POVZETEK

Organizacije za razvoj programske opreme (v nadaljevanju PO) se vedno bolj zavedajo, da je uvajanje sistemov kakovosti predpogoj za dolgoročno preživetje in razvoj. Ključni element za zagotavljanje kakovosti je natančno definiran proces razvoja PO. Model CMM[1] je uveljavljen model za zagotavljanje kakovosti, ki opisuje značilnosti takšnega procesa. Model definira kakovost procesa s pomočjo petih zrelostnih stopenj, ki so določene z množico ciljev na različnih ključnih področjih. Kot tak je smiselno vodilo za organizacije, ki želijo izboljšati svoj proces razvoja PO.

Uporaba modela CMM je primerna tudi v majhnih organizacijah za razvoj informacijskih sistemov (v nadaljevanju IS), ki jih obravnavamo v okviru pričujoče disertacije. Značilnosti teh organizacij sta predvsem delovanje z omejenimi viri in posebnosti pri razvoju specifične vrste PO, kar pa ne zmanjšuje pomembnosti zagotavljanja kakovosti na tem področju. Pri tem je nepogrešljivo merjenje procesa razvoja PO. Cilj disertacije je zato definirati model za kvantitativno spremljanje razvoja PO, ki bo majhni organizaciji za razvoj IS omogočal doseganje četrte stopnje zrelosti po modelu CMM.

Predlagana rešitev temelji na dvojnosti zastavljene naloge. Četrta stopnja modela CMM zelo natanko določa proces in merjenje, kar narekuje podrobno in obsežno izvedbo obojega. V nasprotju s tem so zahteve majhnih organizacij predvsem enostavnost, preprostost in uporabnost. Predlagana rešitev z uporabo izvirnih rešitev doseže oboje: zadosti smiselnemu delu zahtev modela CMM in zagotovi zahtevano uporabnost za majhne organizacije.

Rešitev obsega tri dele:

a) **proces razvoja PO**, ki ustreza zahtevam četrte stopnje modela CMM in odraža način delovanja majhne organizacije za razvoj PO za podporo IS,
b) **merski proces**, ki določa način izvajanja merjenja in predstavlja sestavni procesa razvoja PO ter
c) **merski informacijski model**, ki predstavlja vsebino merjenja in smiselno dopolnjuje oba procesa.

Vsi trije deli skupaj določajo celoto, ki jo s skupno besedo imenujemo model za kvantitativno spremljanje razvoja PO.

a) Prvi korak na poti do rešitve je natančna opredelitev procesa razvoja PO. Za model CMM je značilno, da je zahteven za uporabo in s tega stališča presega zmožnosti majhnih organizacij. V ta namen preverimo dve možnosti: bodisi namesto modela CMM uporabimo enostavnejša modela PSP[2] in TSP[3] ali pa prilagodimo model CMM zahtevam

[1] Model CMM je krajši naziv za Model stopenj zrelosti (ang. Capability Maturity Model) [Pau93].
[2] Model PSP je krajši naziv za Osebni proces razvoja PO (ang. Personal Software Process) [Hum95].
[3] Model TSP je krajši naziv za Skupinski proces razvoja PO (ang. Team Software Process) [Hum00].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

manjših organizacij. Čeprav modela PSP in TSP temeljita na principih modela CMM, se prva možnost izkaže kot približna, nepopolna rešitev. V okviru druge možnosti formalno določimo *prilagoditev modela CMM* za potrebe majhnih organizacij. Slednjič s kombinacijo obeh pristopov določimo proces razvoja PO, ki ga poimenujemo **prilagodljivi proces razvoja PO**.

V proces vgradimo tri izvirne rešitve, ki omogočajo sprotno prilagajanje zahtevam na natanko definiran način med samim izvajanjem razvoja. *Večcikličen proces* razdeli razvoj na obvladljive samostojne dele, kar omogoča natančnejše planiranje in učinkovitejše odzivanje na spremembe. *Prilagodljivo izvajanje korakov* omogoča sprotno prilagajanje razvoja zahtevam in veliko razširljivost procesa, medtem ko *pravila prilagajanja* zagotovijo, da je proces tudi po prilagajanju ustrezno definiran. V disertaciji smo prilagodljivi proces razvoja PO določili in prikazali njegovo izvedbo.

Druga dva dela rešitve predstavljata naš merski pristop, ki je vpet v proces razvoja. Da bi merjenje ustrezno definirali, smo si za izhodišče izbrali preverjen pristop PSM[4], ki ustreza sodobnim standardom in se veliko uporablja. Pristop PSM predstavlja merjenje v dveh delih, ki ju povzamemo in prilagodimo našim potrebam.

b) **Merski proces** določa štiri glavne aktivnosti: *uvajanje, planiranje, izvajanje* in *ovrednotenje merjenja*. Naš pristop deli merjenje na dva dela. *Stalni del* merjenja je tisti, ki ga mora vsaka organizacija nujno izvajati, medtem ko *posebni del* obravnava posebnosti posameznih projektov. Proces določa, da merjenje operativno izvajajo vsi izvajalci posameznih aktivnosti. Merski podatki se zbirajo v *centralnem repozitoriju*[5], tu pa se s postopki združevanja in izpeljav določi še preostale rezultate. S tem izvajanje merskih postopkov potisnemo na nižjo raven, ki omogoča natančnejše in bolj ažurno merjenje, predvsem pa s predvideno avtomatizacijo postopkov in repozitorija razbremenimo merskega analitika. Tak pristop sicer zahteva uvedbo discipliniranega merjenja na ravni posameznika, vendar se merjenje operativno izvaja na enostaven, vedno enak način.

Druga značilnost merskega procesa so analize merskih podatkov, ki jim posvetimo veliko pozornosti. Predvidimo štiri vrste analiz: *ocenjevanje*, *analizo izvedbe*, *analizo performans* in *napovedovanje*, pri čemer sta pomembni predvsem prva in zadnja.

Ocenjevanje je bistvenega pomena za natančno planiranje, ki ima posebno vlogo tudi v prilagodljivem procesu razvoja PO. Zato v tem delu predlagamo izvirne rešitve za izboljšanje natančnosti ocenjevanja, predvsem pa vse ocene ustrezno statistično utemeljimo na podatkih preteklih projektov.

Napovedovanje je naša izvirna vrsta analize, ki jo predlagamo za izboljšanje učinkovitosti dela merskega analitika. Predvidimo popolnoma avtomatizirano izvedbo te analize, ki obsega stalen nabor pokazateljev. Ti se dinamično spreminjajo na podlagi podatkov preteklih projektov in izmerjenih vrednosti. Ta način ponuja hiter progled značilnosti procesa in nadomešča druge analize, ki se izvajajo zgolj po potrebi.

[4] Pristop PSM je krajši naziv za Praktično merjenje PO (ang. Practical Software Measurement) [McG01].
[5] Centralni repozitorij je enotna zbirka raznovrstnih podatkov organizacije in vseh njenih projektov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

c) Zadnji del disertacije obsega **vsebinski opis merjenja**, ki podpira predlagane rešitve. Merjenje predstavimo na štirih ravneh: na ravni *merskega informacijskega modela*[6], *meta modela*[7], več *naborov* meritev in pokazateljev ter *merskega repozitorija*. V okviru tega dela predlagamo tudi izviren način za določanje meritev. *Kombinirana metoda za določanje meritev* omogoča, da izpeljavo popolnega nabora meritev utemeljimo na dejanskih informacijskih potrebah majhne organizacije ter merjenje izvajamo s stalnim naborom neposrednih meritev. Kombinirana metoda vsebuje dobre lastnosti obeh znanih pristopov za določanje meritev: pristopa od zgoraj navzdol[8] in pristopa od spodaj navzgor[9].

Otipljivi rezultati disertacije so predlagani *nabori stalnih merskih izhodišč*, *merskih entitet*, *(osnovnih, združenih in izpeljanih) meritev* ter *pokazateljev*, ki so bili izpeljani s pomočjo kombinirane metode. Izhodišče za določanje meritev predstavlja *nabor stalnih merskih izhodišč*, ki izhaja iz ciljev druge, tretje in četrte stopnje modela CMM ter iz značilnosti majhne organizacije. Nabor *merskih entitet* predstavlja objekte, ki jih je treba meriti. Neposredno merjenje glavnih značilnosti definira *nabor osnovnih meritev.* Osnovne meritve pokrivajo štiri *razrede merjenja*: velikost izdelkov, porabo časa in napredovanje, vire in stroške ter kakovost izdelkov. Določimo tudi *nabor združenih meritev*, kjer z uporabo enega izmed petih načinov združevanja opišemo, kako lahko iz rezultatov meritev sestavljamo nove rezultate. *Nabora izpeljanih meritev in pokazateljev* sta neposreden rezultat uporabe kombinirane metode, pri čemer so pokazatelji povezani z različnimi analizami, ki jih definiramo v okviru opisa merskega procesa. V okviru disertacije smo predstavili nabore, ki predstavljajo merjenje v tipični majhni organizaciji. Posamezni projekti lahko nabore v skladu s potrebami dopolnijo.

Vsi trije deli disertacije predstavljajo celovit *model za kvantitativno spremljanje razvoja PO* v majhni organizaciji za razvoj IS. Zastavljene cilje model dosega z jasnimi koncepti, prilagodljivostjo, relativno majhnostjo in predvsem uporabnostjo.

Model je bil kratko predstavljen v več majhnih organizacijah za razvoj PO, kjer je bila potem izvedena še anketa. Ta je pretežno potrdila pravilnost predlaganih rešitev. To je še zlasti res, če ob odgovorih upoštevamo različno zrelost, cilje in izkušnje anketiranih organizacij na področju merjenja razvoja PO.

[6] Merski informacijski model definira strukturo elementov, s katerimi opišemo vsebino merjenja.
[7] Meta model na več ravneh definira merske podatke in njihova medsebojna razmerja.
[8] Metoda GQM (ang. Goal-Question-Metric) [Bas88] je primer pristopa od zgoraj navzdol.
[9] Primer pristopa od spodaj navzgor je metoda MQG (ang. Metric-Question-Goal) [Het93].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

ABSTRACT

A MODEL FOR THE QUANTITATIVE TRACKING OF THE SOFTWARE PROCESS COMPLIANT WITH THE FOURTH LEVEL OF THE CAPABILITY MATURITY MODEL (CMM)

At this point in time a higher level of quality in software production is necessary for the effective development and long lasting survival of software organizations. A controlled software process is the key issue for quality. The Capability Maturity Model (CMM) is a successful quality model, which defines the characteristics of such a process. This model defines five maturity levels; each containing a number of precisely described key process areas and goals. It can be used as a successful guide for software process improvement in software organizations of different sizes.

The CMM is particularly appropriate for software process improvement in small software organizations that develop a specific kind of software, namely information systems. The main characteristics of these organizations are restricted resources (people, money) and a specific software process. Quality remains an important issue for such organizations and its achievement requires the ability to suitably measure the software process.

The goal of this dissertation is to define a software measurement model for small software organizations that produce information systems. Our model is consistent with the definition of the fourth maturity level of the CMM.

Our solution is built on two principles. The fourth level of the CMM defines the software process and measurement very precisely, making the implementation of a measurement model rather complicated and complex. Yet small organizations expect a simple, understandable, and practical solution to the problem of measurement. Our solution offers a compromise of both principles: it is consistent with all reasonable demands of the CMM in small organizations and provides a practical and usable solution for typical small organizations.

The complete solution consists of three parts:

a) **the software process** is consistent with the goals of the fourth maturity level of CMM and expresses software production in a small software organization that develops information systems,
b) **the measurement process** describes how measurement is done and fits a defined software process, and
c) **the measurement information model** defines the content of measurement and supports the measurement process.

All three parts compose the model for quantitative tracing of software development.

a) The first step is to accurately define the software process. The CMM is a complex quality model and as such it is too complicated for direct use in a small organization. We examine

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

two solutions to this problem. First, it is possible to replace the CMM with one of two well-known process models that implement CMM principles in a simpler way, or second, it is possible to adapt the CMM for use in small organizations. The Personal Software Process (PSP) and the Team Software Process (TSP) are two suitable models for CMM replacement, but they generate an approximate and incomplete process from the point of view of small organizations. In the second solution we define a form of adaptation of the CMM for small organizations. The combination of both solutions is the basis for our definition of **the adaptive software process**, which is the final result of the first part of dissertation.

Three innovations are implemented in the adaptive software process. They enable the process to be defined precisely, and at the same time remain dynamically sensitive to changes during software development. First, the software process is defined as a sequence of seven cycles, each consisting of some phases and many steps. Cycles define manageable parts of software development and enable exact planning and timely reactions. Second, each step can be executed in a more or less complete way. This makes our process extendable. Third, our process defines precisely how to forego certain cycles, phases or steps if they are not needed in the concrete software project. In our dissertation we define and demonstrate the implementation of a completely adaptable software process.

In the second part of the dissertation we define the measurement approach, which is integrated in the adaptive software process already described. The measurement approach is based on an adequately defined and well-known approach named the Practical Software Measurement (PSM), which complies with modern measurement standards. The PSM consists of two models: the Measurement Process Model and the Measurement Information Model. We adapt both for use in small organizations.

b) **The Measurement Process Model** defines four main measurement activities: how to establish and sustain measurement commitment, how to plan measurement, how to perform it, and how to evaluate measurement. In our approach we divide all measurement activities into two parts: the permanent part defines obligatory measurement issues for each project in a small organization, while the special part complies with the special needs of a particular project. The performer of each project development activity has to collect a defined set of measurement data. The measurement results are stored in an automated central repository where some other measurement results are also calculated. In this way we collect low-level measurement data enabling a more accurate measurement and a simplified operational task for the measurement analyst. However, we demand the disciplined operation of each project member in the sense of PSP, even while the measurement activities are executed in a simple and always identical manner.

   The analysis of measurement data defines the remaining core measurement activities. We define four types of analysis: assessment, feasibility analysis, performance analysis, and prediction. The first and last types are the most significant in our measurement approach.

   Assessment is a crucial type of analysis for punctual planning and this is the reason for its special role in the adaptive software process. In this part we implement some innovations that make assessments more exact. All assessments use statistical methods and are based on data regarding similar past projects.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Prediction is our genuine type of analysis enabling the measurement analyst to dynamically follow any changes in a project's characteristics. It defines a constant set of measurement indicators that present the main relations between current measurement data, planned values, and past project data. We define a fully automated mode of prediction in which indicators dynamically change in correspondence with changes in the current measurement data. Automated prediction can replace other types of analysis that are executed only on demand. All the analysis components of prediction in this model are automated and built into the central repository information system.

c) The last part of the dissertation presents **the content of measurement**, which is consistent for all the described solutions. The content description has four levels:
   - *the Measurement Information Model* defines the structure of the content description,
   - *the Meta model* describes the characteristics of all levels of measurement data and relations between them,
   - *collections of measurement entities, attributes, measures, indicators* and *measurement issues* define measurement precisely, and
   - *the measurement repository* implements the described measurement.

   In this part we present an original method for the definition of a suitable collection of measures and indicators from a defined set of measurement issues. Our combined method is based on the information needs of a small software organization, but it always measures only a well-defined set of attributes of existing entities. As such it unites the best characteristics of both top-down and bottom-up methods.

   Collections of measurement entities, attributes, (basic, composed and calculated) measures, indicators, and measurement issues are the tangible results of our dissertation. They were all defined using our combined method and they are consistent with the adaptive software process. The collection of permanent measurement issues is derived from the goals of the fourth level of the CMM, as well as from the characteristics of a small organization. However, measurement entities (objects of measurement) originate in the adaptive software process. A collection of basic measures directly measures the main characteristics of entities: the size of products, time consummation, progress, resources, costs, and product quality. A collection of composed measures defines ways to unite similar measurement results in new results, while calculated measures describe how to calculate new results based on those already known. A collection of indicators is the main result of the use of the combined method. It represents the connection between measures and measurement issues. Alternatively, indicators are results of the form of analysis we defined in the measurement process.

All three parts of the dissertation define a complete model for the quantitative tracing of software development in a small organization that produces information systems. The model we have created is compliant with the fourth level of the CMM. Our goal was the use of clear concepts, an adaptive specialised definition, and above all practical application.

Our model has been presented to several small software organizations in Slovenia. After the presentation the organization representatives were asked to complete a questionnaire. The results of the questionnaire show that our model is in general appropriate and effective for small software organizations. This is especially true if we take into account the different levels of maturity, diverse goals, and various experiences in the field of software measurement of each organization.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

ZAHVALA

Najprej se iskreno zahvaljujem mentorju prof. dr. Viljanu Mahniču za strokovno pomoč, nasvete in vložen trud, ki so pripomogli k uspešnemu zaključku doktorske disertacije. Hvala tudi za spodbudo in sodelovanje, ki mi je omogočilo, da sem v zadnjem letu večji del svojega časa lahko posvetil izdelavi disertacije.

Zahvaljujem se tudi prof. dr. Boštjanu Vilfanu in prof. dr. Borutu Robiču za koristne nasvete ob prvi predstavitvi tematike disertacije. Iskrena hvala tudi dr. Boštjanu Slivniku, ki mi je s spodbudnimi nasveti pomagal v najtežjih obdobjih nastajanja disertacije. Zahvaljujem se tudi Marku Poženelu, Slavku Drnovščku in Mateju Gromu za izkazano pomoč. Hvala tudi ostalim sodelavcem na Fakulteti za računalništvo in informatiko.

Posebno zahvalo za vse izkazano potrpljenje in skrb dolgujem staršem na Kozini in v Novi Gorici. Nenazadnje se od srca zahvaljujem Leonidi, ki mi je v zadnjem letu podarila ves prosti čas in mi dajala upanje, ko sem že hotel odnehati. Brez nje te doktorske disertacije ne bi bilo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

KAZALO

POVZETEK I

ABSTRACT V

ZAHVALA IX

KAZALO XI

KAZALO SLIK XIX

KAZALO TABEL XXIII

1. UVOD 1

1.1 PREDSTAVITEV PROBLEMATIKE 1
1.2 IZHODIŠČA DISERTACIJE 3
1.3 PRIČAKOVANI REZULTATI 3
1.4 OMEJITVE DISERTACIJE 4
1.5 PRISTOP PRI IZDELAVI DISERTACIJE 4
1.6 STRUKTURA DISERTACIJE 5

2. PREDSTAVITEV RAZISKOVALNEGA PODROČJA 9

2.1 MERJENJE PROGRAMSKE OPREME 9
2.1.1 Osnove merjenja 9
2.1.1.1 Načela merjenja 9
2.1.1.2 Merske skale 11
2.1.1.3 Vrste meritev 12
2.1.2 Merjenje v tehnologiji programske opreme 13
2.1.2.1 Cilji merjenja v tehnologiji programske opreme 14
2.1.2.2 Merjenje procesov, izdelkov in virov 15
2.1.2.3 Uporaba meritev programske opreme 16
2.1.3 Proces razvoja programske opreme 16
2.1.3.1 Splošna definicija procesa 16
2.1.3.2 Proces razvoja programske opreme 17
2.1.4 Merski program 18
2.1.4.1 Merski plan 18
2.1.4.2 Načini opisa merskega programa 19
2.1.4.3 Model osnovnega merskega procesa po Hetzlu 19
2.1.4.4 Model merskega procesa po McAndrewsu 20
2.1.4.5 Praktično merjenje PO (pristop PSM) 21
2.1.5 Določanje nabora meritev 23
2.1.5.1 Pristop od zgoraj navzdol: metoda Goal Question Metric (GQM) 23
2.1.5.2 Pristop od spodaj navzgor: metoda Metric Question Goal (MQG) 25
2.1.6 Praktične izkušnje pri uporabi meritev 26

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

2.1.6.1 Načela tehnologije programske opreme in načela merjenja 26
Načela tehnologije programske opreme 26
Načela merjenja programske opreme 27
2.1.6.2 Nasveti za izvajanje meritev 27
2.2 MODELI IN STANDARDI 29
2.2.1 Družina standardov ISO-9000 30
2.2.2 Model BOOTSTRAP 30
2.2.3 Model ISO/SPICE 31
2.2.4 Model PROCESSUS 32
2.2.5 Paradigma za razvoj procesa SEL 33
2.2.6 Model stopenj zrelosti (model CMM) 33
2.2.6.1 Predstavitev modela CMM 34
2.2.6.2 Vsebina modela CMM 35
2.2.6.3 Proces razvoja PO z vidika modela CMM 37
2.2.6.4 Uporaba modela CMM 37
2.2.6.5 Izkušnje z modelom CMM 38
2.2.6.6 Sorodni modeli 39
2.2.7 Osebni proces razvoja PO (model PSP) 40
2.2.8 Skupinski proces razvoja PO (model TSP) 42
2.3 MAJHNE ORGANIZACIJE ZA RAZVOJ INFORMACIJSKIH SISTEMOV 46
2.3.1 Merila za določanje majhnosti organizacije za razvoj PO 46
2.3.2 Značilnosti majhnih organizacij pri uporabi modela CMM 48
2.3.3 Značilnosti organizacij za razvoj informacijskih sistemov 49

3. PROCES RAZVOJA PO V MAJHNI ORGANIZACIJI ZA RAZVOJ IS 51

3.1 PRILAGODITEV MODELA CMM 52
3.1.1 Uporaba sorodnih modelov 52
3.1.2 Prilagoditev modela CMM za majhne organizacije 53
3.1.2.1 Pravila prilagajanja modela CMM 54
3.1.2.2 Ustreznejša delitev skupin in vlog modela CMM 56
Skupine na ravni organizacije in projektov 57
Vloge na ravni projektov 58
3.1.2.3 Prilagoditev procesa ciljem organizacije 58
3.1.2.4 Poenostavljeno dokumentiranje 58
3.1.2.5 Majhen obseg meritev 59
3.1.2.6 Poenostavljeno usposabljanje 59
3.1.2.7 Ustrezno zagotavljanje kakovosti 59
3.1.2.8 Primeri prilagoditev modela CMM 60
3.1.3 Izbrani pristop 60
3.2 PRISTOP PRI OBLIKOVANJU PROCESA RAZVOJA PO 61
3.2.1 Prva delna rešitev: posplošen model TSP 61
3.2.2 Druga delna rešitev: proces za razvoj informacijskih sistemov 62
3.2.3 Končni proces: združitev delnih rešitev z dopolnitvami 63
3.2.3.1 Združitev delnih rešitev 64
3.2.3.2 Vključitev manjkajočega dela prilagojenega modela CMM 64
3.2.3.3 Poenostavitev procesa za uporabo v majhni organizaciji 65
3.3 PRILAGODLJIV PROCES RAZVOJA PO 65
3.3.1 Večcikličen proces 66
3.3.2 Prilagodljivi koraki 68
3.3.3 Pravila prilagajanja procesa problemu 71

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3.3.3.1 Določitev načina uporabe procesa 71
3.3.3.2 Opredelitev nujnih delov vsakega projekta 72
3.3.3.3 Opredelitev nujnih delov vsakega izvajanega cikla 73
3.3.3.4 Opredelitev nujnih delov vsake izvajane faze 73
3.3.3.5 Opredelitev načina izvedbe izvajanih korakov 73
3.3.3.6 Upoštevanje pravil za pravočasno planiranje 73
3.4 OPIS PRILAGODLJIVEGA PROCESA RAZVOJA PO 74
3.4.1 Predstavitev formalnega načina opisa procesa 74
3.4.2 Delovanje majhne organizacije za razvoj informacijskih sistemov 76
3.4.2.1 Organizacijska struktura majhne organizacije 76
3.4.2.2 Repozitorij 76
3.4.2.3 Izvajanje aktivnosti 77
3.4.2.4 Delovanje vodstva 78
3.4.2.5 Delovanje skupin SEPG in SQA 78
3.4.2.6 Izvajanje projektov 78
3.4.2.7 Sestanki v organizaciji 79
3.4.2.8 Podporne aktivnosti 79
3.4.3 Vloge in skupine prilagodljivega procesa 79
3.4.4 Vsebinski opis prilagodljivega procesa za razvoj PO 81
3.4.4.1 Splošne značilnosti procesa 81
3.4.4.2 Proces na najvišji ravni 81
3.4.4.3 Podporne aktivnosti 82
3.4.4.4 Prvi cikel: Začetek projekta 83
3.4.4.5 Drugi cikel: Raziskava zahtev 85
3.4.4.6 Tretji cikel: Načrtovanje 86
3.4.4.7 Četrti cikel: Izgradnja 87
3.4.4.8 Peti cikel: Testiranje 89
3.4.4.9 Šesti cikel:Uporaba 90
3.4.4.10 Sedmi cikel: Zaključek projekta 91
3.4.5 Rezultati projekta 92
3.4.5.1 Opis poteka projekta 93
3.4.5.2 Opis procesa 93
3.4.5.3 Izdelki 93
3.4.5.4 Pomožna dokumentacija 94
3.5 ZAKLJUČEK POGLAVJA 94

4. POSTOPKOVNI DEL MERSKEGA PRISTOPA: MERSKI PROCESNI MODEL 97

4.1 MERSKI PRISTOP 97
4.1.1 Izkušnje organizacij za razvoj PO 97
4.1.2 Značilnosti majhne organizacije za razvoj informacijskih sistemov 99
4.1.3 Predstavitev merskega pristopa 99
4.2 MERSKI PROCESNI MODEL 100
4.2.1 Uvedba merjenja 100
4.2.1.1 Pridobivanje podpore organizacije 101
4.2.1.2 Določanje odgovornosti 101
4.2.1.3 Skrb za potrebne vire 102
4.2.2 Planiranje merjenja 102
4.2.2.1 Definiranje merskih izhodišč in določanje njihove prioritete 103
Definiranje merskih izhodišč *103*

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Razvrščanje merskih izhodišč *104*
Določanje prioritete merskih izhodišč *106*
4.2.2.2 Izbira in opis meritev 107
Izbira razreda meritev *107*
Izbira primernih meritev *109*
Opis podatkovnih zahtev *109*
4.2.2.3 Vključevanje merjenja v proces razvoja PO 110
Opis razvojnega okolja *110*
Določanje priložnosti za merjenje *111*
Opis zahtev za vgraditev meritev *112*
4.2.3 Izvajanje merjenja 112
4.2.3.1 Zbiranje in obdelava podatkov 112
Zbiranje podatkov *113*
Preverjanje podatkov *114*
Normalizacija podatkov *114*
4.2.3.2 Analiza merskih izhodišč 115
4.2.3.3 Odločanje 117
Poročanje o rezultatih *117*
Izbiranje ukrepov *118*
Ukrepanje *118*
4.2.4 Ovrednotenje merjenja 119
4.2.4.1 Ovrednotenje meritev in pokazateljev 119
4.2.4.2 Ovrednotenje merskega procesa 120
4.2.4.3 Dopolnjevanje podatkovne baze izkušenj 120
4.2.4.4 Določanje in implementacija izboljšav 121
4.2.5 Razmerje med merskim procesom in procesom razvoja PO 121
4.3 ANALIZE MERSKIH PODATKOV 122
4.3.1 Ocenjevanje 123
4.3.1.1 Izbira pristopa 124
4.3.1.2 Prilagoditev pristopa 126
4.3.1.3 Izvedba ocenjevanja 126
Ocenjevanje velikosti končnega izdelka *128*
Ocenjevanje obsega dela *133*
Ocenjevanje porabe časa *137*
Ocenjevanje stroškov *140*
Ocenjevanje kakovosti *141*
4.3.1.4 Ovrednotenje ocen 145
4.3.2 Analiza izvedljivosti 145
4.3.2.1 Ovrednotenje planov 147
4.3.2.2 Ocenjevanje vpliva 148
4.3.2.3 Določanje alternativ 148
4.3.3 Analiza performans 148
4.3.3.1 Primerjava planiranih vrednosti z dejanskimi 150
4.3.3.2 Ocenjevanje vpliva 150
4.3.3.3 Napovedovanje rezultatov 151
4.3.3.4 Ovrednotenje alternativ 151
4.3.4 Napovedovanje 151
4.3.4.1 Stalno napovedovanje globalnih vrednosti 153
Velikost končnega izdelka *154*
Velikosti izdelkov ciklov *154*

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Obseg dela celotnega projekta 155
Obseg dela posameznih ciklov in faz 155
Dolžina trajanja in datum zaključka projekta 156
Dolžina trajanja in mejniki posameznih ciklov in faz 156
Stroški projekta ter stroški faz in ciklov 156
Kakovost končnega izdelka 156
Kakovost na štiri načine 157
4.3.4.2 Predstavitev napovedanih vrednosti 158
4.4 ZAKLJUČEK POGLAVJA 162

5. VSEBINSKI DEL MERSKEGA PRISTOPA: MERSKI INFORMACIJSKI MODEL 165

5.1 OPIS VSEBINE MERJENJA 165
5.2 MERSKI INFORMACIJSKI MODEL 166
5.2.1 Informacijske potrebe, merska izhodišča in merski koncepti 166
5.2.2 Merski sestav 167
5.2.2.1 Entitete 167
5.2.2.2 Merljivi atributi 167
5.2.2.3 Osnovne meritve 168
5.2.2.4 Združene meritve 169
5.2.2.5 Izpeljane meritve 169
5.2.2.6 Pokazatelji 170
5.2.2.7 Informacijski izdelki 170
5.2.3 Postopek merjenja in merski plan 170
5.3 META MODEL - MODEL PODATKOV O MERJENJU 171
5.3.1 Zahteve, ki vplivajo na meta model 171
5.3.2 Opis meta modela 173
5.3.2.1 Celovit pogled na meta model 173
5.3.2.2 Meta model na ravni organizacije 174
Organizacijski opis merjenja 174
Opis organizacijskega procesa razvoja PO 176
Opis osebja in skupin organizacije 178
Združeni rezultati vseh projektov 178
5.3.2.3 Meta model na ravni projekta v teku 179
Projektni opis merjenja 180
Opis projektnega procesa PO 181
Opis projektne skupine 182
Opis konfiguracij 182
Izmerjene vrednosti 183
Določene vrednosti (združene ali izpeljane) 184
Ocene, napovedi in rezultati analiz 185
5.3.2.4 Meta model, ki predstavlja dokončane projekte 186
Združeni rezultati projekta 187
5.4 DOLOČANJE VSEBINE MERJENJA 188
5.4.1 Izpeljava nabora meritev z uporabo metode GQM 189
5.4.2 Izpeljava nabora meritev z metodo MQG 190
5.4.3 Izpeljava nabora meritev s kombinirano metodo 190
5.4.3.1 Uporaba metode GQM 193
5.4.3.2 Uporaba metode MQG 193
5.4.3.3 Ugotavljanje razmerja med meritvami in merskimi izhodišči 194

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5.4.3.4 Preverjanje popolnosti 196
5.4.3.5 Preverjanje zadostnosti 196
5.4.3.6 Celoten postopek določanja nabora meritev 197
5.5 VSEBINA MERJENJA 198
5.5.1 Nabor stalnih merskih izhodišč 198
5.5.1.1 Druga stopnja modela CMM: Ponovljiv proces 201
Upravljanje z zahtevami *201*
Planiranje projektov *202*
Zasledovanje in nadzor nad projekti *202*
Urejanje razmerij s poddobavitelji *203*
Zagotavljanje kakovosti PO *203*
Upravljanje s konfiguracijo *203*
5.5.1.2 Tretja stopnja modela CMM: Definiran proces 203
Osredotočenje na proces, ki ga uporablja organizacija *204*
Definicija procesa razvoja PO organizacije *204*
Program izobraževanja *204*
Integrirano vodenje projektov razvoja PO *204*
Tehnologija izdelkov PO *204*
Koordinacija med različnimi skupinami *204*
Pregledi med kolegi *205*
5.5.1.3 Četrta stopnja modela CMM: Upravljan proces 205
Kvantitativno vodenje procesov *205*
Upravljanje za kakovost *206*
5.5.2 Nabor entitet in atributov 206
5.5.2.1 Nabor merskih entitet 206
Aktivnosti *206*
Izdelki *207*
Viri *208*
5.5.2.2 Nabor merskih atributov 208
5.5.3 Nabor osnovnih in dela združenih meritev 209
5.5.3.1 Poraba časa in napredovanje 209
5.5.3.2 Velikost izdelkov 210
Merske metode za merjenje velikosti *210*
Sistematičnost meritev velikosti *211*
Merjenje celotne velikosti in merjenje spremembe velikosti izdelka *212*
Osnovne in združene meritve velikosti izdelka *213*
5.5.3.3 Viri in stroški 214
5.5.3.4 Kakovost izdelkov 216
Odkrite napake *217*
Odpravljene napake *218*
5.5.4 Nabor preostanka združenih meritev, izpeljanih meritev in pokazateljev 219
5.5.4.1 Trije primeri izpeljave pokazateljev in (združenih ali izpeljanih) meritev 220
Obseg dela aktivnosti *220*
Produktivnost aktivnosti *221*
Kakovost na podlagi napak *222*
5.5.4.2 Preostali del nabora pokazateljev in meritev 224
5.5.4.3 Nabor pokazateljev z vidika analiz 224
Ocene *225*
Pokazatelji izvedljivosti *225*
Pokazatelji performans *226*

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Napovedi 228
5.5.4.5 Preverjanje ustreznosti določenih naborov 228
5.6 SKUPNI REPOZITORIJ MERSKIH PODATKOV 229
5.7 ZAKLJUČEK POGLAVJA 231

6. ZAKLJUČEK 233

6.1 PREIZKUS PREDLAGANIH REŠITEV V PRAKSI 233
6.2 REZULTATI DISERTACIJE 237
6.3 PRISPEVKI K ZNANOSTI 238
6.4 PREDLOGI ZA NADALJNJE DELO 239
6.5 SKLEPNA MISEL 239

PRILOGA A: PRILAGODLJIV PROCES RAZVOJA PO 241

A.1 VLOGE IN SKUPINE 241
A.1.1 Vloge in skupine na ravni organizacije 241
A.1.2 Vloge na ravni projekta 242
A.2 OPIS PRILAGODLJIVEGA PROCESA RAZVOJA PO 243
A.2.1 Celoten proces 243
A.2.2 Cikel 0: Podporne Aktivnosti 244
A.2.3 Cikel 1: Začetek projekta 245
A.2.4 Cikel 2: Raziskava zahtev 247
A.2.5 Cikel 3: Načrtovanje 248
A.2.6 Cikel 4: Izgradnja 250
A.2.7 Cikel 5: Testiranje 252
A.2.8 Cikel 6: Uporaba 254
A.2.9 Cikel 7: Zaključek projekta 255
A.2.10 Enostaven korak (posameznik) 256
A.2.11 Zahtevnejši korak (posameznik) 257
A.2.12 Zahteven korak (več članov) 258
A.3 REZULTATI PRILAGODLJIVEGA PROCESA 259
A.3.1 Opis projekta 259
A.3.2 Opis procesa 260
A.3.3 Izdelki 261
A.3.4 Pomožna dokumentacija 262

PRILOGA B: MERSKI PRISTOP 263

B.1: MERSKI PROCESNI MODEL 263
B.1.1 Glavne aktivnosti, podaktivnosti in koraki 263
B.1.2 Umeščenost merskega procesa v proces razvoja PO 264
B.1.3 Analize 265
B.2: MERSKI INFORMACIJSKI MODEL 266
B.2.1 Meta model 266
B.2.2 Razredi meritev 268
B.2.3 Stalen nabor merskih izhodišč 269
B.2.3.1 Majhne organizacije za razvoj informacijskih sistemov 269
B.2.3.2 Druga stopnja modela CMM: Ponovljiv proces 270
B.2.3.3 Tretja stopnja modela CMM: Definiran proces 271
B.2.3.4 Četrta stopnja modela CMM: Upravljan proces 272
B.2.4 Nabor meritev 273
B.2.4.1 Nabor osnovnih meritev 273
Poraba časa in napredovanje 273

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Viri in stroški 273
Velikost izdelkov 274
Kakovost izdelkov 277
B.2.4.2 Nabor združenih meritev 278
B.2.4.3 Nabor izpeljanih meritev 279
B.2.5 Nabor pokazateljev 280
B.2.5.1 Ocenjevanje 280
B.2.5.2 Analiza izvedljivosti 281
B.2.5.3 Analiza performans 281
B.2.5.4 Napovedovanje 282
B.2.6 Povezava med stalnimi merskimi izhodišči in pokazatelji 284
B.2.6.1 Stalna merska izhodišča in pokazatelji 284
B.2.6.2 Pokazatelji in stalna merska izhodišča 286

PRILOGA C: VPRAŠALNIK 287

LITERATURA 293

POMEMBNEJŠE SPLETNE STRANI 311

SLOVAR POMEMBNEJŠIH POJMOV 313

IZJAVA 315

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

KAZALO SLIK

SLIKA 1.1: STRUKTURA DOKTORSKE DISERTACIJE 6
SLIKA 2.1: RAZMERJE MED ISO STANDARDI IN MODELI, OPISUJEJO MERJENJE PO [RIC02] 14
SLIKA 2.2: PRIČAKOVANJA UDELEŽENCEV PRI MERJENJU PO [ROZ00] 15
SLIKA 2.3: UPORABA MERITEV PO [ROZ00] 16
SLIKA 2.4: FAZE MODELIRANJA PROCESA [ROZ00] 17
SLIKA 2.5: MODEL MERSKEGA PROCESA PO HETZLU [HET93] 20
SLIKA 2.6: SPECIFIKACIJA MERITVE ZA ŠTETJE VRSTIC PROGRAMSKE KODE [HET93] 20
SLIKA 2.7: MODEL MERSKEGA PROCESA PO MCANDREWSU [MCA93] 21
SLIKA 2.8: STRUKTURA PRISTOPA PSM [MCG01, DOD02] 22
SLIKA 2.9: METODA GQM (GOAL QUESTION METRIC) [BAS88] 24
SLIKA 2.10: PRIMER OPISA MERSKEGA CILJA PRI METODI GQM [BAS88] 24
SLIKA 2.11: PRIMER IZVEDBE MQG PRISTOPA (METRIC QUESTION GOAL) [HET93] 25
SLIKA 2.12: DRUŽINA STANDARDOV ISO-9000 [ISO92, ISO94] 30
SLIKA 2.13: STRUKTURA PROCESA PO METODOLOGIJI BOOTSTRAP 31
SLIKA 2.14: STRUKTURA MODELA ISO SPICE 32
SLIKA 2.15: KLJUČNA PODROČJA MODELA PROCESSUS [ROZ97] 32
SLIKA 2.16: PARADIGMA ZA RAZVOJ PROCESA SEL [SEL95] 33
SLIKA 2.17: STRUKTURA MODELA STOPENJ ZRELOSTI (SW-CMM V1.1) 35
SLIKA 2.18: PROCES RAZVOJA PO NA 1. STOPNJI MODELA CMM [PAU93A] 36
SLIKA 2.19: PROCES RAZVOJA PO NA 2. STOPNJI MODELA CMM [PAU93A] 36
SLIKA 2.20: PROCES RAZVOJA PO NA 3. STOPNJI MODELA CMM [PAU93A] 36
SLIKA 2.21: PROCES RAZVOJA PO NA 4. STOPNJI MODELA CMM [PAU93A] 37
SLIKA 2.22: PROCES RAZVOJA PO NA 5. STOPNJI MODELA CMM [PAU93A] 37
SLIKA 2.23: PREMOŽENJE PROCESA RAZVOJA PO ORGANIZACIJE [PAU93] 38
SLIKA 2.24: DELEŽI OCENJENIH ORGANIZACIJ PO STOPNJAH CMM [SEI97, SEI02] 38
SLIKA 2.25: DELEŽI ORGANIZACIJ PO STOPNJAH CMM IN PREGLEDIH [SEI97, SEI02] 38
SLIKA 2.26: ČAS NAPREDOVANJA PODJETIJ NA VIŠJO STOPNJO [SEI02] 39
SLIKA 2.27: RAZMERJE MED MODELI CMM, TSP IN PSP [HUM98] 40
SLIKA 2.28: SEDEM VERZIJ MODELA PSP [HUM95] 40
SLIKA 2.29: STRUKTURA PROCESA PSP0 [HUM95] 41
SLIKA 2.30: KLJUČNA PODROČJA MODELA CMM, KI JIH POKRIVA MODEL PSP [HUM95] 42
SLIKA 2.31: STRUKTURA MODELA TSP [HUM00] 43
SLIKA 2.32: DELOVANJE METODE TSP [HUM00] 44
SLIKA 2.33: DELEŽ POKRITOSTI PROJEKTNIH AKTIVNOSTI CMM PRI TSP [DAV02] 45
SLIKA 2.34: DELEŽ POKRITOSTI ORGANIZACIJSKIH AKTIVNOSTI CMM PRI TSP [DAV02] 45
SLIKA 3.1: RAZMERJE MED MODELI CMM, TSP IN PSP 53
SLIKA 3.2: PODROČJA PRILAGODITVE MODELA CMM MAJHNIM ORGANIZACIJAM 54
SLIKA 3.3: DEKLARATIVNI IN INFORMATIVNI DEL MODELA CMM [PAU93A] 55
SLIKA 3.4: IZPELJAVA POSPLOŠENEGA MODELA TSP IZ MODELOV PSP IN TSP 61
SLIKA 3.5: IZPELJAVA TEHNOLOŠKEGA VIDIKA PROCESA NA PODLAGI TREH METODOLOGIJ 63
SLIKA 3.6: CELOTEN POSTOPEK IZPELJAVE PRILAGODLJIVEGA PROCESA RAZVOJA PO 66
SLIKA 3.7: TRINIVOJSKA STRUKTURA PRILAGODLJIVEGA PROCESA RAZVOJA PO 67
SLIKA 3.8: ŠTIRJE NAČINI IZVEDBE KORAKA V PRILAGODLJIVEM PROCESU RAZVOJA PO 69

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Slika 3.9: Dva načina uporabe prilagodljivega procesa 72
Slika 3.10: Prilagodljiv proces razvoja PO 82
Slika 3.11: Cikel 0: Podporne aktivnosti 83
Slika 3.12: Cikel 1: Začetek projekta 84
Slika 3.13: Cikel 2: Raziskava zahtev 85
Slika 3.14: Cikel 3: Načrtovanje 86
Slika 3.15: Cikel 4: Izgradnja 88
Slika 3.16: Cikel 5: Testiranje 89
Slika 3.17: Cikel 6: Uporaba 90
Slika 3.18: Cikel 7: Zaključek projekta 91
Slika 3.19: Tri skupine rezultatov prilagodljivega procesa razvoja PO 92
Slika 4.1: Merski procesni model po pristopu PSM [McG01, DoD02] 100
Slika 4.2: Glavna aktivnost »Uvajanje merjenja« v pristopu PSM 100
Slika 4.3: »Planiraj merjenje« v pristopu PSM 102
Slika 4.4: Prva podaktivnost v okviru »Planiranja merjenja« (pristop PSM) 103
Slika 4.5: Dve vrsti merskih izhodišč pri našem merskem pristopu 103
Slika 4.6: Razredi meritev v našem merskem pristopu 106
Slika 4.7: Druga podaktivnost v okviru »Planiranja merjenja« (pristop PSM) 108
Slika 4.8: Način izbire meritev v pristopu PSM 108
Slika 4.9: Način izbire meritev v našem merskem pristopu 109
Slika 4.10: Združevalne strukture v našem merskem pristopu 110
Slika 4.11: Tretja podaktivnost v okviru »Planiranja merjenja« (pristop PSM) 111
Slika 4.12: Primer merskega plana (pristop PSM) 112
Slika 4.13: »Izvajanje merjenja« v pristopu PSM 113
Slika 4.14: Prva podaktivnost v okviru »Izvajanja merjenja« (pristop PSM) 113
Slika 4.15: Druga podaktivnost v okviru »Izvajanja merjenja« (pristop PSM) 116
Slika 4.16: Vrste analiz v našem merskem pristopu 117
Slika 4.17: Tretja podaktivnost v okviru »Izvajanja merjenja« (pristop PSM) 117
Slika 4.18: »Ovrednotenje merjenja« v okviru pristopa PSM 119
Slika 4.19: Štiri vrste analiz našega merskega pristopa 122
Slika 4.20: Tipična razmerja med merjenimi vrednostmi 123
Slika 4.21: Izvedba ocenjevanja v okviru pristopa PSM 125
Slika 4.22: Dva načina izvedbe ocenjevanja v našem merskem procesu 125
Slika 4.23: Ocenjevanje po našem merskem pristopu 127
Slika 4.24: Grafični prikaz regresijske premice in dveh območij zaupanja 133
Slika 4.25: Ocenjevanje obsega dela po našem merskem pristopu 133
Slika 4.26: Ocenjevanje porabe časa po našem merskem pristopu 138
Slika 4.27: Ocenjevanje stroškov po našem merskem pristopu 140
Slika 4.28: Dve vrsti stroškov: stroški iz obsega dela in dodatki stroški 140
Slika 4.29: Ocenjevanje kakovosti po našem merskem pristopu 143
Slika 4.30: Koraki analize izvedljivosti 146
Slika 4.31: Primer pokazatelja kumulativnih količin pri analizah izvedljivosti 146
Slika 4.32: Primer pokazatelja posamičnih vrednosti pri analizah izvedljivosti 147
Slika 4.33: Primer pokazatelja doseganja omejitev 147
Slika 4.34: Koraki analize performans 149
Slika 4.35: Razmerje med ocenjevanjem, analizo performans in napovedovanjem 151
Slika 4.36: Koraki analize napovedovanja 153
Slika 4.37: Napovedovanje po našem merskem pristopu 153
Slika 4.38: Primer globalne napovedi po našem merskem pristopu 159
Slika 4.39: Primer napovedi za dele procesa po našem merskem pristopu 159

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

SLIKA 5.1: POSTOPNA SESTAVA MERSKEGA PLANA NA PODLAGI INFORMACIJSKIH POTREB 167
SLIKA 5.2: STRUKTURA MERSKEGA SESTAVA 168
SLIKA 5.3: PRIMER: TRIJE DELI OPISA MERSKIH PODATKOV PO [KIT01] 172
SLIKA 5.4: CELOTNA SESTAVA META MODELA PO NAŠEM MERSKEM PRISTOPU 173
SLIKA 5.5: STRUKTURA META MODELA NA RAVNI ORGANIZACIJE 175
SLIKA 5.6: ER DIAGRAM ZA ORGANIZACIJSKI OPIS MERJENJA 175
SLIKA 5.7: ER DIAGRAM ZA OPIS ORGANIZACIJSKEGA PROCESA RAZVOJA PO 177
SLIKA 5.8: ER DIAGRAM ZA OPIS OSEBJA IN SKUPIN ORGANIZACIJE 178
SLIKA 5.9: ER DIAGRAM ZA ZDRUŽENE REZULTATE VSEH PROJEKTOV 179
SLIKA 5.10: STRUKTURA META MODELA NA RAVNI PROJEKTA 179
SLIKA 5.11: ER DIAGRAM ZA PROJEKTNI OPIS MERJENJA 180
SLIKA 5.12: ER DIAGRAM ZA OPIS PROJEKTNEGA PROCESA RAZVOJA PO 181
SLIKA 5.13: ER DIAGRAM ZA OPIS PROJEKTNE SKUPINE 182
SLIKA 5.14: ER DIAGRAM ZA OPIS KONFIGURACIJ 183
SLIKA 5.15: ER DIAGRAM ZA PREDSTAVITEV IZMERJENIH VREDNOSTI 183
SLIKA 5.16: ER DIAGRAM ZA PREDSTAVITEV DOLOČENIH VREDNOSTI 184
SLIKA 5.17: ER DIAGRAM ZA PREDSTAVITEV OCEN, NAPOVEDI IN REZULTATOV ANALIZ 185
SLIKA 5.18: STRUKTURA META MODELA NA RAVNI DOKONČANIH PROJEKTOV 186
SLIKA 5.19: ER DIAGRAM ZA OPIS ZDRUŽENIH REZULTATOV DOKONČANEGA PROJEKTA 187
SLIKA 5.20: POSTOPEK DOLOČANJA PRIMERNEGA NABORA MERITEV 191
SLIKA 5.21: OSNOVNA IDEJA KOMBINIRANEGA PRISTOPA 192
SLIKA 5.22: UGOTAVLJANJE RAZMERJA MED MERITVAMI IN MERSKIMI IZHODIŠČI 194
SLIKA 5.23: CELOTNA STRUKTURA KOMBINIRANE METODE 197
SLIKA 5.24: MERJENJE NA DRUGI STOPNJI MODELA CMM 201
SLIKA 5.25: PRIMER MERSKIH IZHODIŠČ ZA KLJUČNO PODROČJE MODELA CMM 202
SLIKA 5.26: MERJENJE NA TRETJI STOPNJI MODELA CMM 203
SLIKA 5.27: MERJENJE NA ČETRTI STOPNJI MODELA CMM 205
SLIKA 5.28: AKTIVNOSTI V PRILAGODLJIVEM PROCESU RAZVOJA PO 206
SLIKA 5.29: POMEN POJMOV ZA OPIS KAKOVOSTI IZDELKOV 216
SLIKA 5.30: PRIMER POKAZATELJA ZA OBSEG DELA AKTIVNOSTI 221
SLIKA 5.31: PRIMER POKAZATELJA ZA KAKOVOST IZDELKOV 224
SLIKA 5.32: PRIMER POKAZATELJA ZA STROŠKE KAKOVOSTI 228

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

KAZALO TABEL

TABELA 2.1: TIPI MERSKIH SKAL 11
TABELA 2.2: STATISTIČNE OPERACIJE ZA POSAMEZNE TIPE MERSKIH SKAL 12
TABELA 2.3: PREGLEDNICA Z MERJENJEM POVEZANIH ISO STANDARDOV 14
TABELA 2.4: NAČELA TEHNOLOGIJE PO/I [BAS88] 26
TABELA 2.5: NAČELA TEHNOLOGIJE PO/II [BAS88] 27
TABELA 2.6: NAČELA MERJENJA PO/I [BAS88] 27
TABELA 2.7: NAČELA MERJENJA PO/II [BAS88] 27
TABELA 2.8: KLJUČNA PODROČJA PO ZRELOSTNIH STOPNJAH (SW-CMM V1.1) 36
TABELA 2.9: DELEŽ TIPIČNIH PRAKS CMM PO VRSTAH AKTIVNOSTI IN STOPNJAH [DAV02] 45
TABELA 2.10: VELIKOST MAJHNIH ORGANIZACIJ PO [LAR00, ORC00] 47
TABELA 2.11: VELIKOST MAJHNIH PROJEKTOV V SKLADU Z [HUM00, HUM00B] 47
TABELA 2.12: PRIPOMBE MAJHNIH ORGANIZACIJ GLEDE UPORABE MODELA CMM 48
TABELA 3.1: TABELA ZA OPIS NAVODIL PRILAGODLJIVEGA PROCESA 75
TABELA 4.1: PRIMERNOST RAZLIČNIH PRISTOPOV OCENJEVANJA 125
TABELA 4.2: PRIMER NORMALIZACIJSKE TABELE S TIPIČNIMI RAZMERJI VELIKOSTI 130
TABELA 4.3: ZGLED UPORABE NORMALIZACIJSKE TABELE ZA OCENJEVANJE VELIKOSTI 130
TABELA 4.4: PRIMER TIPIČNIH RAZMERIJ AKTIVNOSTI PRILAGODLJIVEGA PROCESA 134
TABELA 4.5: PRIMER UPORABE TABELE RAZMERIJ AKTIVNOSTI 136
TABELA 5.1: ENTITETE ZA ORGANIZACIJSKI OPIS MERJENJA 177
TABELA 5.2: ENTITETE ZA OPIS ORGANIZACIJSKEGA PROCESA RAZVOJA PO 177
TABELA 5.3: ENTITETE ZA OPIS OSEBJA IN SKUPIN ORGANIZACIJE 178
TABELA 5.4: ENTITETE ZA ZDRUŽENE REZULTATE VSEH PROJEKTOV 178
TABELA 5.5: ENTITETE ZA PROJEKTNI OPIS MERJENJA 181
TABELA 5.6: ENTITETE ZA OPIS PROJEKTNEGA PROCESA RAZVOJA PO 182
TABELA 5.7: ENTITETI ZA OPIS PROJEKTNE SKUPINE 182
TABELA 5.8: ENTITETE ZA OPIS REZULTATOV MERJENJA 186
TABELA 5.9: ENTITETE ZA OPIS ZDRUŽENIH REZULTATOV DOKONČANIH PROJEKTOV 188
TABELA 5.10: UPORABA TERMINOLOGIJE V RAZLIČNIH METODAH 192
TABELA 5.11: REZULTATI CIKLOV PRILAGODLJIVEGA PROCESA RAZVOJA PO 207
TABELA 5.12: MERITVE PORABE ČASA 209
TABELA 5.13: MERITVE NAPREDOVANJA 209
TABELA 5.14: OSNOVNE MERITVE VELIKOSTI IZDELKOV 214
TABELA 5.15: ZDRUŽENE MERITVE VELIKOSTI IZDELKOV 214
TABELA 5.16: MERITVE VIROV 215
TABELA 5.17: MERITVE STROŠKOV 216
TABELA 5.18: RAZREDI NAPAK PO STANDARDU IBM [HUM95] 217
TABELA 5.19: RAZREDI NAPAK PO NAŠEM PRISTOPU 218
TABELA 5.20: OSNOVNE MERITVE KAKOVOSTI IZDELKOV 218
TABELA 5.21: ZDRUŽENE MERITVE KAKOVOSTI IZDELKOV 219
TABELA 5.22: PRVI PRIMER DOLOČANJA IZKORISTKA ODPRAVLJANJA NAPAK 223
TABELA 5.23: DRUGI PRIMER DOLOČANJA IZKORISTKA ODPRAVLJANJA NAPAK 223

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

1. UVOD

Področje razvoja programske opreme (v nadaljevanju PO) narekuje potrebo po nenehnem raziskovanju. Številni znanstveniki in strokovnjaki po svetu se trudijo, da bi ponudili nove, boljše pristope za učinkovitejše delo. V skladu s [Fen94] naj bi bili ti utemeljeni **na objektivnih raziskavah** z **jasnimi izhodišči** in **definiranim namenom**. To smo si izbrali za vodilo tudi pri izdelavi pričujoče disertacije.

1.1 Predstavitev problematike

Tehnologija PO pokriva različna področja razvoja PO. Globalno gledano ta zajemajo *specifikacijo PO, načrtovanje le-te, opisujejo programske tehnike in orodja, načine za zagotavljanje kakovosti in varnosti PO ter načine vodenja programskih projektov* [Som92]. Posebno mesto med področji pripada zagotavljanju kakovosti PO, ker je od učinkovite izvedbe aktivnosti tega področja v veliki meri odvisna uspešnost doseganja ciljev razvoja PO.

Zagotavljanje kakovosti včasih napak enačimo s preverjanjem kakovosti. Slednje je namreč enkratna aktivnost pred zaključkom razvojnega cikla, ki služi zgolj za izločanje vgrajenih napak. To za kakovosten izdelek ni dovolj; zagotoviti moramo razvoj PO z manj napakami, za kar potrebujemo učinkovito vgraditev aktivnosti za zagotavljanje kakovosti v celoten proces razvoja PO. Ker je ta proces sestavljen iz *množice aktivnosti, metod, postopkov in transformacij* [Som92, Pau93], je treba zagotoviti kakovostno izvedbo vsakega elementa posebej. Z drugimi besedami to pomeni, da je kakovost PO neposredna posledica kakovostnega procesa razvoja PO.

Znanih je več modelov za zagotavljanje kakovosti. Eden najbolj uveljavljenih je model stopenj zrelosti (v nadaljevanju model CMM) [Pau93]. Odlikuje se po natančni definiciji koncepta *petih stopenj zrelosti* (angl. maturity levels) razvojnega procesa. Vendar model CMM kot tak ne določa načina, kako določeno stopnjo doseči, temveč le, kaj je v okviru izbrane stopnje treba doseči [Pau93a]. To je s stališča uporabnosti najprimerneje, saj si lahko vsaka organizacija prikroji model za svoje potrebe; vendar po drugi strani takšna definicija pušča tudi veliko odprtih vprašanj v zvezi z izvedbo modela.

Model ima dve tipični uporabi: omogoča ugotavljanje kakovosti procesa razvoja PO v organizaciji z določanjem kritičnih pomankljivosti v njem (angl. software process assessment) ter ovrednotenje razvojnih procesov več različnih organizacij (angl. software capability evaluation). V številnih velikih organizacijah so te aktivnosti izvajali sodelavci Software Engineering Institute (v nadaljevanju SEI). Analiza rezultatov v teh organizacijah je pokazala, da je kar 62.5% pregledanih organizacij na prvi ali drugi zrelostni stopnji in samo 14.1% na četrti ali peti [SEI02]. To se odraža tudi v raziskanosti posameznih stopenj zrelosti modela: na prvi, drugi in tretji stopnji je ta že dodobra opisan (tudi s stališča uporabe), medtem ko sta četrta in peta stopnja slabše raziskani. To pomeni, da se na četrti (in peti) stopnji odpira možnost za nadaljnje raziskave in eksperimentiranja.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Razvojni proces na četrti stopnji zrelosti izpolnjuje vse zahteve druge in tretje stopnje, poleg tega pa še zbira kvantitativne podatke o različnih vidikih razvojnega procesa ter le-te uporablja za izboljšavo kakovosti procesa. Pri tem model predpisuje zgolj področja meritev in ne predlaga izvedbe merjenja, enako kakor ne razlaga načina uporabe dobljenih rezultatov. Ravno izbira primernega merskega pristopa je ključnega pomena za proces razvoja PO [Oma97]. Treba je določiti tak merski pristop, ki bo v skladu s potrebami in zahtevami organizacije na četrti stopnji zrelosti omogočal merjenje različnih značilnosti procesa razvoja PO in organizacije nasploh [McA93].

V literaturi je predlaganih veliko merskih pristopov, ki dosegajo različno zastavljene cilje [Kan95, Fen97, Mil88, Oma97]. Vendar izkušnje kažejo, da so številni omejeni na specifične probleme, veliko jih v praksi ni preverjenih, poleg tega pa je z vidika uporabnosti zelo pomembna tudi njihova zahtevnost [Oma97, Hor98]. Najboljše izkušnje na področju merjenja so zbrane v standardu ISO/IEC 15939 [ISO01], ki je usklajen tudi z zadnjimi verzijami modela CMM. Merski pristop Praktično merjenje PO (v nadaljevanju pristop PSM) [McG01] je utemeljen na tem standardu. Sestavljata ga dva modela: merski procesni model in merski informacijski model. Pristop PSM je primerna podlaga za definicijo merskega pristopa, ki se smiselno vklaplja v proces razvoja PO izbrane organizacije.

Značilnosti merjenja so odvisne tudi od vrste PO, ki jo razvijamo [Low96, Oma97]. To se najprej odraža v samem procesu razvoja PO [Kan95], še pomembnejše pa so možnosti (in omejitve) pri določanju merskega procesa in vsebine merjenja (recimo merjenje OO razvoja [Her98], merjenje funkcijskih jezikov [Van95]). Zaradi obsežne uporabe je zelo zanimivo področje razvoja PO za podporo informacijskim sistemom (v nadaljevanju IS), ki temeljijo na uporabi sodobnih sistemov za upravljanje s podatkovnimi bazami (PB). Za tovrstno PO je kakovost izdelkov pomembna predvsem z vidika vzdrževalnosti in krajšega časa razvoja, kar se odraža v specifičnem tehnološkem procesu. Značilnosti tega so izrazitejše aktivnosti zajema zahtev, več sodelovanja z uporabnikom, upoštevanje podatkovnega in procesnega vidika razvijajočih izdelkov in podobno.

Poleg tega je v primeru razvoja PO za podporo IS nujna uporaba različnih programskih orodij, ki lahko podpirajo del ali celoto procesa razvoja PO v organizaciji. V okviru določanja vsebine merjenja je treba definirati splošne vmesne rezultate, ki so tipični za večino orodij, ki so običajno v uporabi. V primeru PO za podporo IS je smiselno definirati recimo uveljavljene diagrame (recimo diagram entiteta-razmerje, diagram podatkovnih tokov, graf hierarhije modulov), s katerimi modeliramo podatkovni in procesni del nastajajoče PO [Mar85].

Pri vgrajevanju merskega pristopa je pomemben še en vidik. Učinkovito izvajanje merjenja in analize merskih rezultatov zahteva veliko virov in je običajno pokazatelj visoke stopnje zrelosti organizacije, zato ga srečujemo predvsem v velikih tehnološko razvitih organizacijah [Oma97]. Temu so prilagojeni tudi učinkoviti merski pristopi, med drugim tudi pristop PSM. Po drugi strani pa je merjenje v manjših organizacijah za razvoj PO (kakršnih je v Sloveniji največ) zelo redko in ne daje zadovoljivih rezultatov, kar je z vidika konkurenčnosti gotovo težava. Majhne organizacije imajo poseben način poslovanja in omejene vire, merjenje pa zahteva prevelik obseg dela [Joh98]. Zato tudi neposredna uporaba obstoječih pristopov ni mogoča. Dejansko še ne poznamo merskega pristopa, ki bi ustrezal četrti stopnji zrelosti v majhnih organizacijah za razvoj PO za podporo IS, ki uporabljajo sodobno upravljanje s PB.

Značilnosti majhnih organizacij za razvoj PO za podporo IS določajo novo raven zahtev. Doslej predstavljene ideje so temeljile na ustreznosti: recimo model CMM ustrezno določa

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

proces razvoja PO in pristop PSM primerno opredeljuje merjenje. Zahteve majhnih organizacij so minimalnost, enostavnost in uporabnost. S tega stališča že uporaba modela CMM in pristopa PSM v majhni organizaciji ni preprosta, temveč zahteva prilagajanje. Iskanje pravega kompromisa med kompleksnostjo modelov in zahtevami majhne organizacije je rdeča nit pričujoče disertacije.

1.2 Izhodišča disertacije

Opisane okoliščine porajajo izhodišča za raziskovalno delo. **Osnovni izhodišči** pričujoče disertacije sta dve:

- določiti želimo merski pristop, ki bo ustrezal **zahtevam 4. stopnje modela CMM** in
- bo primeren za uporabo **v majhnih organizacijah** za **razvoj PO za podporo IS**.

Na podlagi teh izhodišč določimo več ciljev, ki natančneje opredeljujejo vsebino disertacije. Naslednje **cilje** je mogoče razbrati tudi iz strukture disertacije:

- Glede na prvo izhodišče mora proces razvoja PO v majhni organizaciji ustrezati zahtevam četrte stopnje modela CMM. Ker je model CMM primeren predvsem za večje organizacije [Pau99], moramo najprej raziskati, kako ga lahko uporabimo v majhnih organizacijah.
- Majhna organizacija za razvoj IS ima posebnosti, ki se odražajo v njenem procesu razvoja PO. Smiselno je določiti tipične značilnosti majhne organizacije in jih na ustrezen način vgraditi v proces razvoja PO.
- Raziskati je treba znanje in izkušnje drugih raziskovalcev na področju merjenja PO. Na tej podlagi je treba izbrati primeren merski pristop, ki ga bomo prilagodili zahtevam majhnih organizacij za razvoj IS.
- Izbrani merski pristop določimo s postopkovnega in vsebinskega dela. Izvesti je treba prilagoditev obeh delov merskega pristopa posebnostim izbrane vrste organizacije.
- Pomemben del oblikovanja merskega pristopa bo zajemal vgrajevanje izvirnih postopkov in elementov. Ti se bodo nanašali tako na postopkovni kot vsebinski del, pa tudi na sam način določanja pristopa.
- Zadnje dejanje bo natančna definicija vseh pomembnih elementov merskega pristopa in njihova sestava v celovito sliko.

Na podlagi zastavljenih ciljev bomo določili več rezultatov. Ob koncu disertacije bomo vse rezultate združili v celoto, ki jo bomo s skupnim imenom imenovali *model za kvantitativno spremljanje razvoja PO v majhni organizaciji za razvoj PO za podporo IS.*

1.3 Pričakovani rezultati

Ob izdelavi disertacije pričakujemo naslednje izvirne **rezultate**:

- opis prilagoditve modela CMM za majhne organizacije;
- definicija procesa razvoja PO, ki je primeren za majhne organizacije za razvoj IS;
- definicija ustreznega merskega pristopa in njegovih sestavnih delov: merskega procesa in vsebine merjenja;

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- podrobna predstavitev merskega procesa, njegove izvedbe ter njegova umestitev v proces razvoja PO;
- podrobna predstavitev vsebine merjenja na več ravneh: na ravni opisa strukture informacijskega modela, na ravni vsebine podatkov o merjenju ter na ravni naborov merskih izhodišč[10], merskih entitet[11], meritev[12] in pokazateljev[13];
- predstavitev metode za določanje primerne vsebine vseh naborov;
- podrobnejša predstavitev izvirnih rešitev, recimo načina izvajanja merjenja, napovedovanja, združevanja rezultatov in podobno.

Vsi rezultati imajo v okviru raziskave svoje mesto in se medsebojno dopolnjujejo.

1.4 Omejitve disertacije

Naloga je zastavljena zelo široko, zato se moramo smiselno **omejiti** z več vidikov:

- **Obsežnost problema:** nesmiselno je iskati splošno rešitev zastavljenega problema, saj se tudi majhne organizacije med seboj lahko precej razlikujejo. Omejili se bomo na tipično majhno organizacijo za razvoj IS, ki jo bomo tudi definirali. Rešitev, ki jo bomo na tej podlagi določili, bo primerna za širok krog majhnih organizacij.

- **Obsežnost merskega pristopa**: tudi pri določanju vsebine merjenja se bomo omejili na tipične sestavne dele. Otipljiv rezultat disertacije bodo nabori meritev in pokazateljev, ki omogočajo učinkovito in enostavno uporabo. Rezultat bo predstavljal primeren in smiselen nabor na primerni ravni natančnosti, ki ga bo po potrebi mogoče nadgraditi.

- **Obsežnost opisa procesa razvoja PO in merskega pristopa**: popolna definicija procesa razvoja in merskega pristopa presega obseg disertacije. Za oba dela bomo na usklajen način predstavili vse pomembnejše dele, pri čemer bodo opisi do smiselne ravni podrobnosti.

- **Preverjanje modela v praksi**: rezultate disertacije bomo želeli preveriti tudi v praksi. Vendar žal v Sloveniji ni majhne organizacije na ustrezni stopnji zrelosti po modelu CMM. Poleg tega je popolno preverjanje merskega modela dolgotrajno [Ive03] in ga ni mogoče izvesti v zastavljenem časovnem okviru. Zato bomo izvedli enostavnejše preverjanje v štirih majhnih slovenskih organizacijah. Predlagan merski model jim bomo predstavili, se o njem pogovorili in mnenje sogovornikov zajeli z anketo.

1.5 Pristop pri izdelavi disertacije

Pri izdelavi disertacije smo si zastavili nekaj načel, ki se odražajo tudi v njeni vsebini. Namen izbranih načel je zagotoviti ustrezno izvedbo zastavljene naloge oziroma najboljši rezultat.

[10] Mersko izhodišče predstavlja formaliziran opis merskega cilja, ki izhaja iz informacijskih potreb organizacije.
[11] Merska entiteta predstavlja objekt, ki ga merimo.
[12] Meritev je opis postopka, s katerim izmerimo (ali določimo) določeno značilnost merske entitete.
[13] Pokazatelj predstavlja (običajno grafični) prikaz rezultatov ene ali več meritev z določenega vidika.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Prvo načelo: Pomembna značilnost izbrane naloge je dvojnost. Zahteve modela CMM, še zlasti njegove četrte stopnje, določajo podrobno definiran proces razvoja PO. Ta zahteva **ustrezno** izvedbo aktivnosti številnih področij, zaradi česar je model zahteven za izvedbo. Po drugi strani zahteve majhnih organizacij navajajo **minimalnost**, enostavnost in praktičnost, ki so v nasprotju z zahtevami modela CMM.

V disertaciji tega razkoraka ne dojemamo kot težavo, temveč kot možnost za raziskave. Zavedamo se, da vseh zahtev ne bo mogoče doseči, vendar na prilagajanje raje gledamo kot na priložnost za iskanje učinkovitih rešitev. Pri tem posebno pozornost namenjamo ustrezni izvedbi prilagajanja.

Drugo načelo: Praktično gledano model za kvantitativno spremljanje obsega več zaporednih rezultatov: prilagoditev modela CMM majhni organizaciji, proces razvoja PO in merski pristop, ki je sestavljen iz dveh delov. Čeprav je zadnji rezultat najbolj otipljiv, so pomembni vsi rezultati. S tega stališča skušamo principe ustreznosti (model CMM) in minimalnosti (majhne organizacije) vgraditi v vse rezultate.

Tretje načelo: Da bi ustrezno zastavili raziskavo, jo vedno utemeljimo na formalnih zahtevah. Kot primer lahko navedemo recimo zahteve četrte stopnje modela CMM, standard ISO/IEC 15939 in pristop PSM. Tudi proces razvoja PO v majhni organizaciji zasnujemo na preverjenih modelih PSP in TSP, enako kombinirano metodo za izpeljavo nabora meritev.

Četrto načelo: V raziskavah skušamo vedno kombinirati formalne izpeljave s praktičnimi znanji. Tako recimo pri določanju nabora meritev upoštevamo izkušnje drugih organizacij, ki so navedene v literaturi.

Peto načelo: Naša rešitev bo temeljila na nekaj glavnih izvirnih pristopih. Vendar skušamo v proces razvoja in merski pristop uvajati inovacije v vseh delih, kjer je to mogoče. Velikokrat so dobri rezultati kombinacija znanih pristopov z nekaterimi novimi elementi.

Šesto načelo: Razvite rešitve skušamo preveriti in prikazati njihovo uporabo. Preverjanje opravimo z ugotavljanjem povezav med različnimi rešitvami. Tudi predlagana izvedba nekaterih rešitev (recimo procesa razvoja PO) omogoča preverjanje ustreznosti.

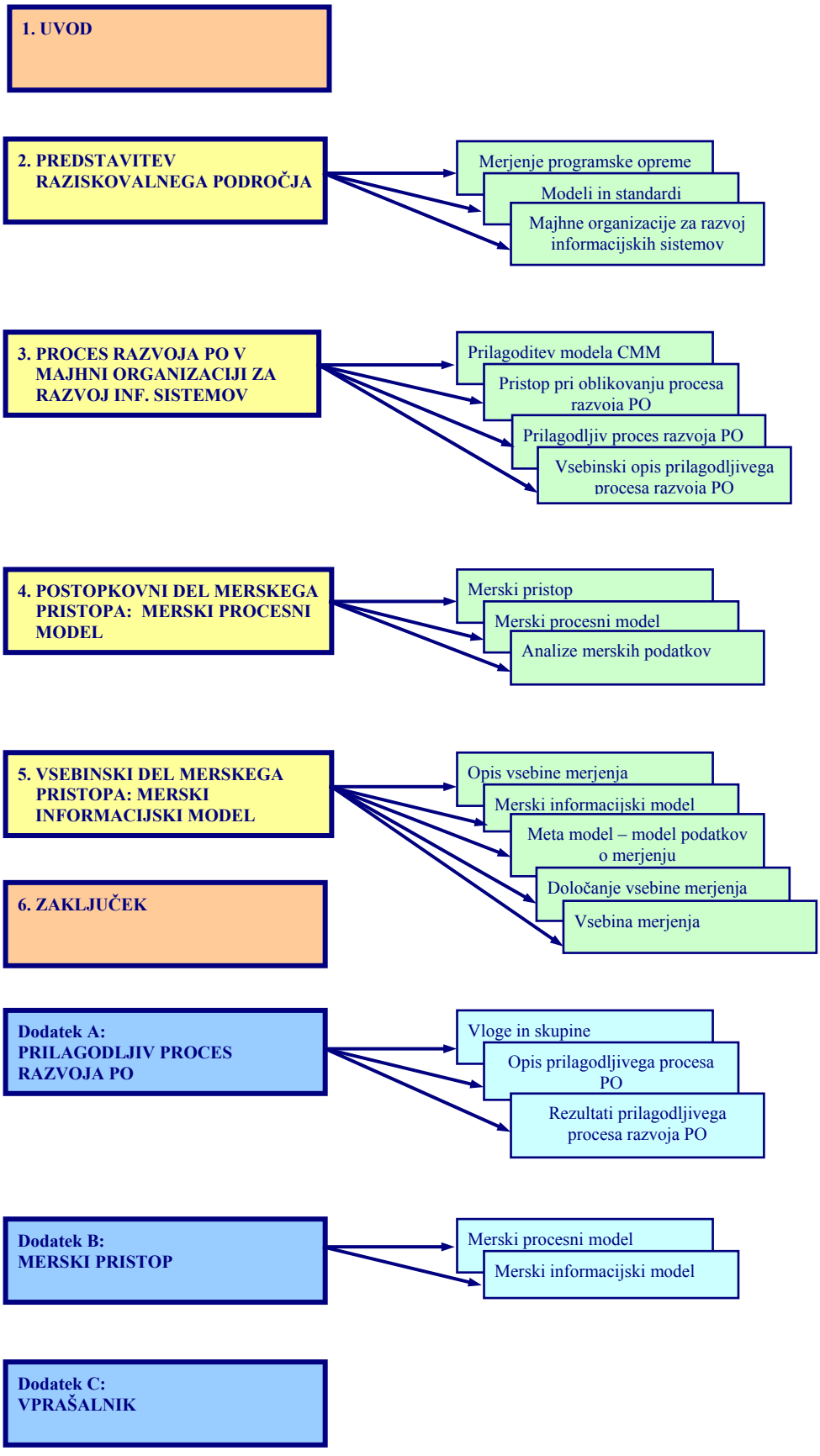
1.6 Struktura disertacije

Disartacija poleg zaključka in uvoda obsega še štiri poglavja in tri dodatke, ki so okvirno predstavljeni v nadaljevanju. Struktura je razvidna tudi iz slike 1.1.

V **drugem poglavju** predstavljamo raziskovalno področje. Predstavitev obsega tri dele:

- Prvi del je namenjen splošnemu pregledu področja merjenja in še zlasti merjenja v tehnologiji PO. Ogledamo si več pomembnih področij (recimo pristop PSM), ki jih moramo spoznati pred raziskavami v disertaciji. Podpoglavje zaključimo z izborom nasvetov in izkušenj na področju merjenja PO.
- Drugi del je namenjen predstavitvi standardov in modelov na področju zagotavljanja kakovosti PO. Osrednja vloga je posvečena modelu CMM, podrobno pa predstavljamo tudi modela PSP in TSP, ki imata vidnejšo vlogo v nadaljevanju.
- Zadnje podpoglavje predstavlja značilnosti majhnih organizacij za razvoj IS.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 1.1: Struktura doktorske disertacije

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Tretje poglavje opisuje prilagodljiv proces razvoja PO, ki predstavlja pomemben rezultat disartacije. Poglavje obsega štiri dele:

- V prvem delu obravnavamo možnosti prilagajanja modela CMM za majhne organizacije. V okviru tega opišemo dve rešitvi: uporabo obstoječih modelov PSP in TSP in določitev prilagoditve modela CMM za majhne organizacije za razvoj IS.
- V drugem delu predstavimo pristop, ki smo ga uporabili za določanje procesa razvoja PO za majhne organizacije. Pri tem opišemo dve delni rešitvi (prva izhaja iz modela TSP, druga pa iz posebnosti razvoja IS) ter njuno združevanje v dokončno definiran proces z upoštevanjem zahtev prilagojenega modela CMM.
- Tretji del prikaže tri osnovne značilnosti nastalega prilagodljivega procesa razvoja PO: večcikličen proces, prilagodljive korake in pravila za prilagajanje procesa problemu.
- V četrtem delu kratko prikažemo delovanje tipične majhne organizacije za razvoj IS ter dejansko izvedbo prilagodljivega procesa razvoja. V okviru tega prikažemo skupine in vloge, opišemo aktivnosti procesa ter rezultate le-teh.

Četrto poglavje opisuje postopkovni del našega merskega pristopa. Obsega tri podpoglavja:

- Prvo podpoglavje predstavlja značilnosti našega merskega pristopa, ki je zasnovan na pristopu PSM.
- Drugi del opisuje aktivnosti v okviru merskega procesa, ki predstavlja postopkovni del merjenja. Prikažemo vse štiri glavne aktivnosti, podrobneje predvsem planiranje in izvajanje merjenja. Prikažemo tudi, kako se opisani merski proces ujema s prilagodljivim procesom razvoja PO.
- Tretji del je v celoti posvečen analizam merskih podatkov. Naš pristop predvideva štiri vrste analiz: ocenjevanje, analize izvedljivosti, analize performans ter napovedovanje. Zaradi pomembnosti sta podrobneje predstavljeni ocenjevanje in napovedovanje.

Peto poglavje opisuje še drugi, vsebinski del našega merskega pristopa. Tega predstavimo na več ravneh, ki jim ustrezajo posamezna podpoglavja:

- V kratkem prvem podpoglavju predstavljamo način predstavitve vsebinskega dela merskega modela.
- Drugo podpoglavje opisuje najvišjo raven predstavitve vsebine. Informacijski merski model definira strukturo opisa vsebine merjenja.
- Tretje podpoglavje predstavlja podroben meta model s podatki o merjenju. Meta model je predstavljen na treh ravneh (organizacijski, projektni in ravni preteklih projektov) s številnimi diagrami entiteta-razmerje.
- V četrtem podpoglavju predstavimo izvirno kombinirano metodo za določanje nabora meritev. Metoda je določena kot kombinacija metod GQM in MQG oziroma pristopov od spodaj navzgor in od zgoraj navzdol.
- V petem podpoglavju s pomočjo kombinirane metode določimo več naborov, ki predstavljajo vsebino merjenja: nabor stalnih merskih izhodišč, nabor merskih entitet, nabor (osnovnih, združenih in izpeljanih) meritev ter nabore pokazateljev.

V **zaključnem poglavju** vse zaokrožimo v celovito sliko: model za kvantitativno spremljanje razvoja PO. Ob tem tudi opišemo preverjanje disartacije v praksi, povzamemo rezultate disertacije ter predlagamo nadaljnje raziskave.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Disertacija vsebuje še tri dodatke, v katerih podrobneje opisujemo glavne rezultate modela za kvantitativno spremljanje procesa razvoja PO.

Dodatek A obsega podrobno predstavitev prilagodljivega procesa razvoja PO. Obsega tri dele:

- Prvi del prikazuje skupine in vloge, ki jih definira prilagodljiv proces.
- Drugi del vsebuje podroben opis aktivnosti procesa na treh ravneh: na ravni celotnega procesa, na ravni posameznih ciklov ter na ravni posameznih korakov.
- Zadnji del predstavlja štiri vrste rezultatov procesa: opis projekta, opis procesa, izdelke in pomožno dokumentacijo.

Dodatek B prikazuje opis merskega pristopa. Sestavljata ga dva dela:

- Prvi del predstavlja merski procesni model. Predstavljene so aktivnosti, analize in umeščenost merskega procesa v prilagodljiv proces razvoja PO.
- Drugi del prikazuje vsebinski del. Obsega predstavitev entitet meta modela, razredov meritev ter naborov stalnih merskih izhodišč, vseh vrst meritev ter pokazateljev. Zadnji del prikazuje povezavo med stalnimi merskimi izhodišči in pokazatelji.

Dodatek C prikazuje vprašalnik, ki smo ga uporabili pri izvedbi ankete o tematiki disertacije v več slovenskih majhnih organizacijah za razvoj IS.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

2. PREDSTAVITEV RAZISKOVALNEGA PODROČJA

Splošno gledano disertacija pokriva tri področja:

- **področje merjenja PO** kot širšega področja, v katerega je umeščen ciljni model;
- **področje modelov in standardov za zagotavljanje kakovosti razvoja PO** (s poudarkom na modelu stopenj zrelosti in sorodnih modelih);
- **področje specifične vrste PO**, torej področje majhnih organizacij za razvoj PO za podporo IS.

V tem poglavju bomo vsako od teh področij kratko predstavili.

2.1 Merjenje programske opreme

S stališča disertacije moramo področje merjenja PO pobliže spoznati z več vidikov:

- Najprej moramo v splošnem razumeti osnove merjenja in posebej merjenja PO, da bi lahko umestili našo nalogo v to področje.
- Pomemben korak pri izvedbi naloge je določitev procesa razvoja PO, ki nam predstavlja okvir, v katerega vgradimo merjenje PO.
- Jedro disertacije je vezano na določitev merskega programa v organizaciji. Ta opredeljuje merski proces in vsebino merjenja. Osnovne značilnosti merskega programa in primerne načine izvedbe tega prikažemo v tretjem delu podpoglavja.
- Pri določanju vsebine merjenja igrajo pomembno vlogo načini za določanje nabora primernih meritev, ki si jih ogledamo v četrtem delu.
- Na področju praktične implementacije merjenja je bilo objavljenih veliko izkušenj, priporočil in nasvetov, ki nam lahko veliko pomagajo pri odločitvah o sestavi našega merskega pristopa. Te kratko predstavimo ob koncu podpoglavja.

Določene podrobnosti bodo predstavljene na ustreznih mestih tudi kasneje.

2.1.1 Osnove merjenja

Merjenje je eno osnovnih načel v znanosti, ki je prisotno še posebej v tehničnih vedah. Čeprav se je merjenje v svojem zgodovinskem razvoju različno uveljavilo na različnih področjih, so načela merjenja jasno matematično definirana.

2.1.1.1 Načela merjenja

Merjenje (ang. measurement) je proces, ki ga pogosto srečujemo v vsakdanjem življenju. Omogoča nam, da stvari ali pojme numerično ali simbolno ovrednotimo in s tem učinkoviteje izvajamo željeno dejavnost. Merjenje je v [Fen97] opredeljeno na naslednji način:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Definicija: *Merjenje je proces, pri katerem so atributom entitet iz realnega sveta pripisani simboli ali številke na tak način, da atribute opisujejo v skladu z jasno določenimi pravili.*

Predpogoj za merjenje je jasna definicija objektov merjenja iz realnega sveta (*entitet)*. Z merjenjem zajemamo informacijo o intuitivno dobro razumljeni značilnosti entitete (*atributu*). To počnemo na predpisan način (s *pravili*), kar nam omogoča konsistentno izvajanje merjenja ter nedvoumno razlago dobljenih rezultatov. Na podlagi teh pravil zasnujemo *mersko teorijo* (ang. theory of measurement), v okviru katere natančno definiramo vse vidike merjenja. To nam daje podlago za vse aktivnosti, ki so povezane z merjenjem.

Obstaja več merskih teorij, ki so določene z vrsto pravil. Znana je recimo t.i. *predstavitvena merska teorija* (ang. representional theory of measurement) [Fen97]. Teorija formalizira intuitivno dojemanje realnega sveta s t.i. empiričnimi razmerji (ang. empirical relations), ki izražajo razmerja med entitetami po nekem atributu.

Primer: Iz znanih dejstev o starosti posameznih oseb recimo znamo za par oseb izpeljati empirično razmerje **biti starejši**:

```
Andrej je star.
Nataša ni stara.      ⇒    Andrej je starejši kot Nataša.
```

Formalno lahko na merjenje gledamo kot na preslikavo iz realnega (izkustvenega) sveta v formalni (matematični) svet. V našem primeru tako intuitivno vsako osebo preslikamo v neko izmerjeno vrednost, empirično razmerje med parom oseb pa ustreza neki binarni relaciji med tako izmerjenima vrednostima. To nas pripelje do formalnejše definicije merjenja [Fen97]:

Definicija: *Merjenje je preslikava iz izkustvenega sveta v formalni, relacijski svet. Izmerjena vrednost ali mera (ang. measure) je številka ali simbol, v katero se preslika določena značilnost entitete.*

Da bi bili preslikavi empiričnega razmerja in entitet ustrezni, morata izpolnjevati t.i *predstavitveni pogoj* (ang. representation condition): če so entitete realnega sveta v nekem empiričnem razmerju, mora preslikana relacija to razmerje med merami ustrezno ohranjati. Predstavitveni pogoj za naš primer je naslednji:

```
A je starejši od B ⇒ Starost(A) > Starost(B)
```

Primer:

```
Starost(Andrej) = 84
Starost(Nataša) = 12   ⇒   Starost(Andrej) > Starost(Nataša)
```

Med dvema entitetama lahko obstaja veliko empiričnih razmerij. Več jih obstaja tudi glede na izbran atribut. V opisanem primeru lahko osebe primerjamo tudi **po višini** ali **teži**, v okviru atributa starosti pa lahko govorimo tudi o empiričnih razmerjih **biti mlajši** ali **biti veliko starejši**. Seveda niso vse empirična razmerja binarna; primer unarnega razmerja je recimo **biti polnoleten** ali **biti najstnik**. V teh primerih so predstavitveni pogoji naslednji:

```
A je višji od B  ⇒  Višina(A) > Višina(B)
A je mlajši od B  ⇒  Starost(A) < Starost (B)
A je veliko starejši od B ⇒ Starost(A) > Starost(B)+10
A je polnoleten  ⇒  Starost(A) > 17
A je najstnik    ⇒  Starost(A) > 9 & Starost(A)< 20
```

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Veljavna je vsaka mera, ki ustreza predstavitvenemu pogoju. Isto mero lahko uporabimo za več empiričnih razmerij, vendar je v tem primeru veljavnih mer manj.

2.1.1.2 Merske skale

Uvedemo pojem *merske skale* (ang. measurement scale), ki nam predstavlja skupek treh elementov: preslikave entitete v izmerjeno vrednost, empiričnega razmerja ter relacije.

Smisel merjenja je v lažji uporabi in sklepanju o opisanih atributih entitet. Vendar obstajajo med merskimi skalami pomembne razlike, saj nekatere omogočajo večjo uporabnost kot druge. Z vidika uporabnosti merjenja nas zanimajo predvsem trije problemi:

a) Katera izmed več ustreznih preslikav atributov v številke ali simbole je uporabnejša?
b) Kako lahko preverimo, če relacija ustrezno predstavlja izbrano empirično razmerje?
c) Kakšne so razlike med več ustreznimi relacijami oziroma katera je uporabnejša?

Prvi problem zahteva čim natančnejšo, konkretnejšo in intuitivno razumljivejšo preslikavo atributov entitet v mere, najbolje v konkretne numerične vrednosti. Drugi problem govori o problemu predstavljivosti, ki ga je mogoče reševati z upoštevanjem določenih relacijskih aksiomov, ki jih vpeljemo. Odgovor na tretji problem so različni *tipi merskih skal*, ki nam predstavljajo različne stopnje uporabnosti. Vsaki merski skali lahko določimo tip, ki mu pripada. Vse merske skale istega tipa nudijo enake možnosti uporabe. Poznamo pet osnovnih tipov merskih skal [Fen97]:

- Nominalna ali imenska skala (ang. nominal scale);
- Ordinalna ali urejenostna skala (ang. ordinal scale);
- Intervalna skala (ang. interval scale);
- Razmernostna skala (ang. ratio scale);
- Absolutna skala (ang. absolute scale).

Če imamo več veljavnih mer za neko empirično razmerje, med njimi obstajajo preslikave, ki jih imenujemo *dopustne transformacije* (ang. admissable transformations). Vsak tip merskih skal ima definirano množico dopustnih transformacij. Na ta način lahko ugotovimo, v kateri tip sodi naša merska skala. Bolj kot je omejujoča množica dopustnih transformacij, bolj natančna je naša merska skala. Iz tabele 2.1 so razvidne značilnosti merskih skal [Fen97].

Tip merske skale:	Značilnosti:	Množica dovoljenih transformacij:	Primeri:
Nominalna skala	Več različnih razredov Oznake razrede razlikujejo, nimajo pa pomena	Preslikave 1:1 M' = f(M) =	Poimenovanja, klasifikacija (vrste testiranj PO, tipi napak, vrste razvojnih ciklov, ipd.)
Ordinalna skala	Več različnih razredov, vrstni red razredov pomemben Oznake razredov predstavljajo vrstni red, z njimi ne da računati	Mon. naraščajoče f. M(x)≥M(y) ⇒ M'(x)≥M'(y) = < > ≤ ≥	Zaporedna klasifikacija (šolske ocene, kakovostni razred mesa, stopnje CMM, prioriteta napake, kakovostni razred kode ipd.)
Intervalna skala	Več različnih razredov, vrstni red in razlike med razredi pomembni Obstaja enota Iz oznak lahko določamo razliko, ne razmerij	Funkcije oblike: M'(x) = aM(x)+b (a>0) = < > ≤ ≥ + -	Merilo območja brez ničle (čas - koledar, temperatura v C ali F, mejni termini v koledarju aktivnosti, ipd.)
Razmernostna skala	Več različnih razredov, vrstni red,razlike in razmerja med razredi pomembni Obstaja enota in ničla Oznake razredov omogočajo aritm. računanje	Funkcije oblike: M'(x) = aM(x) (a>0) = < > ≤ ≥ + - / *	Absolutno merilo (čas trajanja, dolžine, teža, čas razvoja PO, čas testiranja,obseg programske kode, obsega dokumentacije, ipd.)
Absolutna skala	Število vseh elementov, edina preslikava je preštetje elementov Mogoča vsa aritmetika nad štev. elementov	Štetje elementov M' = M = < > ≤ ≥ + - / *	Prešteti elementi (število entitet, število modulov, število razvijalcev, ipd.)

Tabela 2.1: Tipi merskih skal

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Primer: Starost osebe lahko merimo z neposredno meritvijo v različnih časovnih enotah (recimo letih ali mesecih). Preslikava iz ene mere v drugo je potemtakem množenje z ustrezno konstanto, kar jo uvršča med razmernostne merske skale:

```
StarostVMesecih(Miha) = 342        StarostVLetih(Miha) = 28.5
                                ⇓
      StarostVMesecih(Miha) = StarostVLetih(Miha) * 12
    StarostVLetih(Miha) = StarostVMesecih(Miha) * 0.0833
```

S pomočjo tipa merske skale lahko ugotovimo, kakšne analize zbranih izmerjenih vrednosti lahko uporabljamo. V tabeli 2.2 so predstavljene nekatere statistične operacije, ki jih lahko uporabljamo za posamezne tipe merskih skal [Fen97, Par96]. Pri tem velja, da so za vsak tip merske skale primerne tudi operacije predhodnika (recimo operacije nominalnega tipa so primerne tudi za ordinalne skale). Podrobnosti naj si bralec ogleda v [Par96, Fen97].

Tip merske skale:	Srednja vrednost:	Primerne statistične operacije:	Primerni statistični testi:
Nominalna skala	Najpogostejša vrednost	Frekvenca, Korelacija zveznosti	Neparametrični testi
Ordinalna skala	Mediana	Korelacija rangov, Spearman r, Kendall τ ...	Neparametrični testi
Intervalna skala	Aritmetična sredina	Standardna deviacija, Povprečna deviacija, Korelacija momentov	Neparametrični testi
Razmernostna skala	Geometrična sredina Harmonična sredina	Odstotkovna variacija, Korelacijsko razmerje	Parametrični testi

Tabela 2.2: Statistične operacije za posamezne tipe merskih skal

Formalno je merjenje sestavljeno iz naslednjih aktivnosti:

1. Določanja atributov izbranih entitet iz realnega sveta;
2. Določanja smiselnih empiričnih razmerij med atributi;
3. Določanja relacij za vsako empirično razmerje;
4. Določanja preslikave entitet realnega sveta v številke ali simbole;
5. Preverjanja predstavitvenega pogoja (relacija med merami mora ohranjati empirično razmerje).

2.1.1.3 Vrste meritev

Meritve lahko razvrstimo (kategoriziramo) z več vidikov [Fen97, Hum89]:

Merjenje je *neposredno ali direktno* (ang. direct, explicit), če je izvedeno samo na podlagi enega atributa entitete. V kolikor je pri določanju udeleženih več atributov ali entitet, je merjenje *posredno ali indirektno* (ang. indirect, derived). Merjenja prve vrste so običajno osnovna merjenja, medtem ko druga merjenja omogočajo opise kompleksnejših razmerij.

Merjenje je *objektivno* (ang. objective), kadar ni odvisno od okolja in ga je mogoče izvesti samo na en način. Tipičen primer je štetje entitet. *Subjektivno* (ang. subjective) merjenje vključuje človeško presojo oziroma vpliv okolja (recimo merjenje kakovosti izdelka).

Absolutne (ang. absolute) mere so tiste, ki se ne spreminjajo z dodajanjem novih elementov. *Relativne* (ang. relative) mere se z novimi elementi spreminjajo (recimo standardna deviacija). Objektivna merjenja imajo običajno absolutne mere, subjektivna pa relativne.

Pri *dinamičnih* (ang. dynamic) merjenjih je pomembna časovna komponenta (recimo trenutna količina opravljenega dela), medtem ko na *statična* (ang. static) merjenja čas ne vpliva (recimo velikost izdelka).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Merjenja *za napovedovanje* (ang. predictive) so tista, ki so izvedena pred izvedbo opravila, ki na merjenje pomembno vpliva. Nasprotno so merjenja *za spremljanje* (ang. explanatory) izvedena po opravilu.

2.1.2 Merjenje v tehnologiji programske opreme

Čeprav je merjenje pomembna prvina številnih tehničnih disciplin, je igralo na področju tehnologije PO dolgo vlogo obrti [Fen97, Pit99]. Razlogi za to izhajajo iz značilnosti razvoja PO, zaradi katerih je operativna izvedba merjenja zahtevna. Zaradi tega merjenje v tehnologiji PO ni bilo samostojno področje, temveč prej zbirka različnih aktivnosti, modelov in praks [Fen97], ki obsega:

- oceno stroškov in obsega dela;
- modele in meritve učinkovitosti;
- zbiranje podatkov;
- modele in meritve za spremljanje kakovosti;
- modele zanesljivosti;
- modele za ovrednotenje performans;
- merjenje strukturiranosti in zahtevnosti (kompleksnosti);
- ocenjevanje zmožnosti oziroma zrelosti organizacij;
- upravljanje z merjenjem;
- ovrednotenje metod in orodij.

Podobno je bilo z raziskanostjo področja. Raziskave in praktični poskusi so se izvajali v različnih okoljih in z različnimi cilji, kar ima za posledico precejšnjo nepreglednost področja [Mil88]. Določeni pristopi in metode so se uveljavili, včasih tudi zaradi ugleda predlagatelja ali okolja, iz katerega so prišli. Ne glede na to moramo omeniti pomembne prispevke Halsteada (principi merjenja in meritve velikosti) [Hal77], Albrechta (meritev funkcijskih točk) [Alb83], McCabeja (meritve kompleksnosti) [McC76], Putnama (ocenjevanje stroškov) [Put78], Boehma (model COCOMO) [Boe81], Basilija (metoda GQM) [Bas88], Humphreya (model CMM) [Hum89] in drugih.

Vendar se v zadnjem obdobju z izdatnejšim uvajanjem načel kakovosti na področju razvoja PO stvari spreminjajo [Fen97]. V ospredje so stopili standardi in modeli za zagotavljanje kakovosti razvoja PO, ki so bili razviti v zadnjem desetletju. Ti so na podlagi izkušenj najprej določili sistematičen postopek merjenja, v zadnjem času pa natančno umestili merjenje kot samostojno področje [BSi02, BSi]. Standard ISO/IEC 15939 (Informacijska tehnologija – Merski proces za PO) [ISO01], natančno opredeljuje terminologijo in način razumevanja merskega procesa. V zadnjih verzijah modela CMMI [SEI02a, SEI02b] recimo merjenje nastopa kot posebno ključno področje, ki je usklajeno s tem standardom.

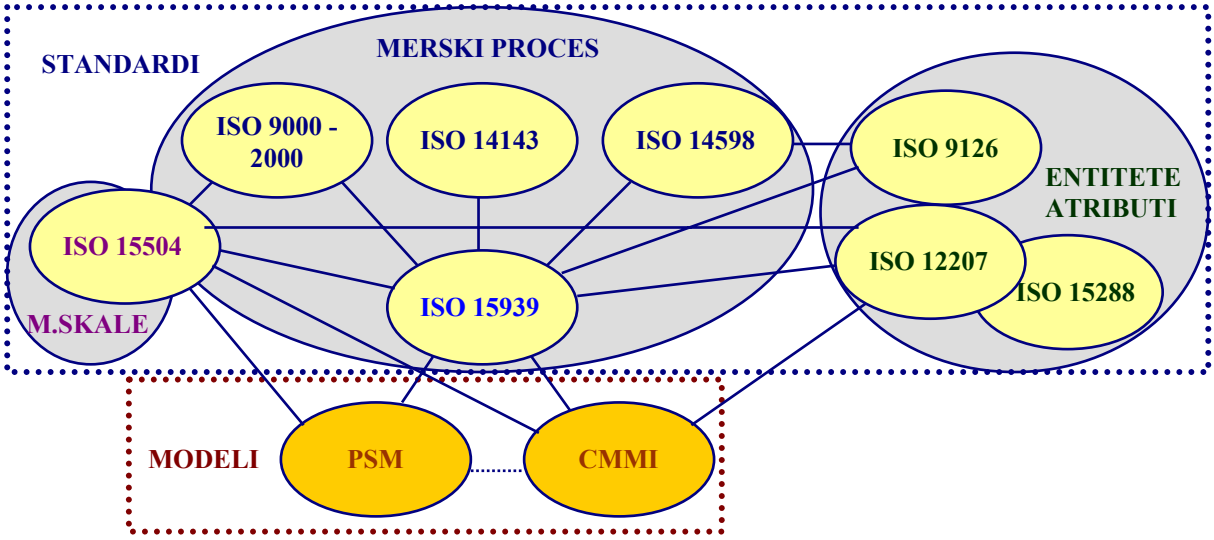
Poleg standarda ISO/IEC 15939 so z merjenjem PO povezani tudi drugi (starejši) ISO standardi, ki bodisi definirajo merjenje ali ga deloma vsebujejo [BSi02, Ric02]. Predstavljeni so v tabeli 2.3, slika 2.1 pa celovito prikazuje razmerja med temi ISO standardi[14] [Ric02]. Razvidno je, da se na standarde navezujeta dva modela (CMMI in PSM), ki bosta igrala ključno vlogo tudi v pričujoči disertaciji. Predstavljena sta v razdelkih 2.2.6 in 2.1.4.5.

[14] Kratka predstavitev standardov se nahaja v [BSi02] in [Ric02]. Standardi so dosegljivi na domači strani International Standard Organisation – ISO [ISO].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Oznaka	Naziv	Slovenski naziv
ISO 9126	Software Product Quality	Kakovost izdelkov PO
ISO 14143	Functional Size Measurement	Merjenje funkcionalnosti
ISO 14598	Evaluation of Software Products	Ovrednotenje izdelkov PO
ISO/IEC 15504	IT – Software Process Assessment	IT – Ocenjevanje procesa razvoja PO
ISO/IEC 15939	IT – Software Measurement Process	IT – Merski proces za PO
ISO/IEC 15288	IT – Life Cycle Management – System Life Cycle Processes	IT – Upravljanje življenjskih ciklov – Procesi za življenjske cikle sistemov
ISO/IEC 12207	IT - Software Life Cycle Processes	IT – Procesi za življenjske cikle PO
ISO 9000-2000	Quality Management Systems	Sistemi za upravljanje kakovosti
Model CMMI	Capability Maturuty Model Integration	Združeni model stopenj zrelosti
Model PSM	Practical Software Measurement	Praktično merjenje PO

Tabela 2.3: Preglednica z merjenjem povezanih ISO standardov



Slika 2.1: Razmerje med ISO standardi in modeli, opisujejo merjenje PO [Ric02]

Poleg standardov ISO so pomembni viri informacij o merjenju tudi angleški standard AMI (ang. Aplications of Metrics in Industry) in dokumenti SEI (ang. Software Engineering Institute) [SEI].

Razvoj področja merjenja PO lepo simbolizira uporaba termina metrika PO. V preteklosti je bila to ključna beseda za merjenje PO, ki se je uporabljala povsod. Danes metrike razumemo kot (pre)splošen termin, ki zajema najrazličnejše aktivnosti povezane z merjenjem PO. Terminu metrike se zato izogibamo [ISO01, Fen97] in raje uporabljamo termin merjenje PO, oziroma termine, ki natančneje opisujejo aktivnost: recimo model, standard, mera in druge.

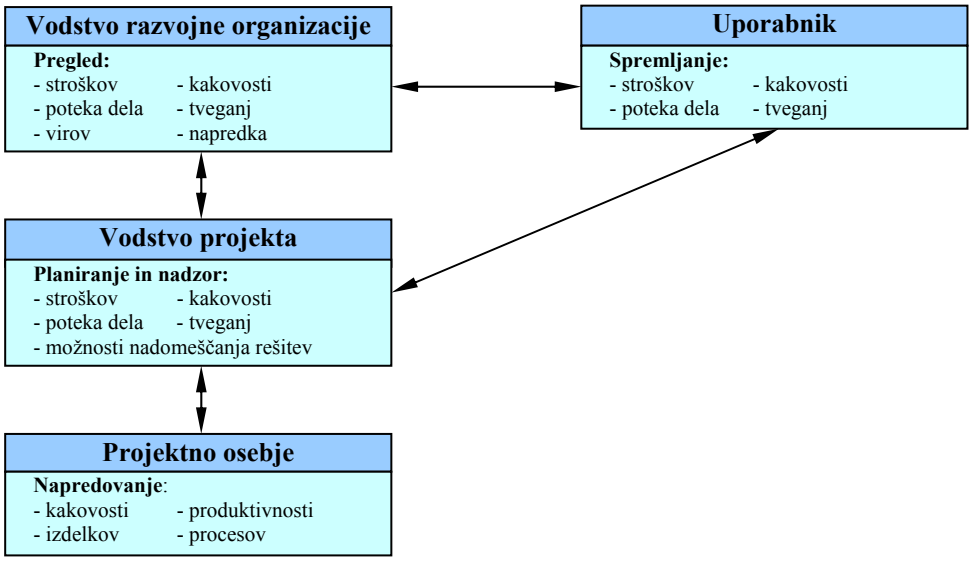
2.1.2.1 Cilji merjenja v tehnologiji programske opreme

Z merjenjem v tehnologiji PO želimo doseči naslednje cilje [SEL95, Fen97]:

1. Razumeti PO;
2. Voditi projekte razvoja PO:
   - planirati in ocenjevati,
   - primerjati dejanske in napovedane vrednosti in
   - izvajati ovrednotenje modelov.
3. S pomočjo kvantitativnega spremljanja nenehno izboljševati proces razvoja PO:
   - razumevanje,
   - ocenjevanje in
   - shranjevanje pridobljenih izkušenj v obliki standardov, pravil, postopkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pri merjenju je udeleženih več strani: razvijalci, uporabniki, vodstvo organizacije za razvoj PO in še kdo. Udeleženci merjenja imajo različne cilje (želje in potrebe) [Roz00] (slika 2.2).



Slika 2.2: Pričakovanja udeležencev pri merjenju PO [Roz00]

2.1.2.2 Merjenje procesov, izdelkov in virov

Pri razvoju PO merimo tri vrste objektov [Fen97, Roz00]:

- procese (ang. processes),
- izdelke (ang. products) in
- vire (ang. resources).

Pri vsaki vrsti lahko merimo *notranje* (ang. internal) ali *zunanje* (ang. external) atribute. Notranji atributi so tisti, ki jih lahko določimo neposredno na objektu, brez da bi na to vplivalo okolje (recimo velikost, struktura). V nasprotju s tem so zunanji atributi lahko določeni samo z upoštevanjem vpliva okolja (recimo uporabnost, kakovost).

Proces razvoja PO nam v splošnem prestavlja množico aktivnosti, ki so povezane z razvojem PO. Pomembna prvina procesa je čas, kar je povezano tudi z zaporedjem izvajanja aktivnosti v procesu. Pri procesih oziroma aktivnostih lahko merimo tri notranje atribute: trajanje, obseg dela in število napak. Med zunanjimi atributi merimo predvsem stroške. Po potrebi lahko merimo še nekatere druge zunanje atribute, pri čemer skušamo te razbrati iz razmerij notranjih atributov.

Izdelki nam predstavljajo vse rezultate izvajanja aktivnosti razvoja PO, recimo programsko kodo, specifikacije, dokumentacijo, poročila in podobno. Kot bomo videli v nadaljevanju, so eden od treh elementov (vsakega) procesa. Obstaja veliko notranjih atributov, ki jih je zaradi določenosti izdelka relativno enostavno meriti; pogosto je mogoče merjenja celo avtomatizirati. Najbolj značilen atribut je velikost, obstaja pa še vrsta drugih, recimo strukturiranost, zahtevnost, funkcionalnost in podobni. Tudi zunanjih atributov je več; običajni so kakovost, razumljivost, uporabnost, primernost za vzdrževanje in drugi.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

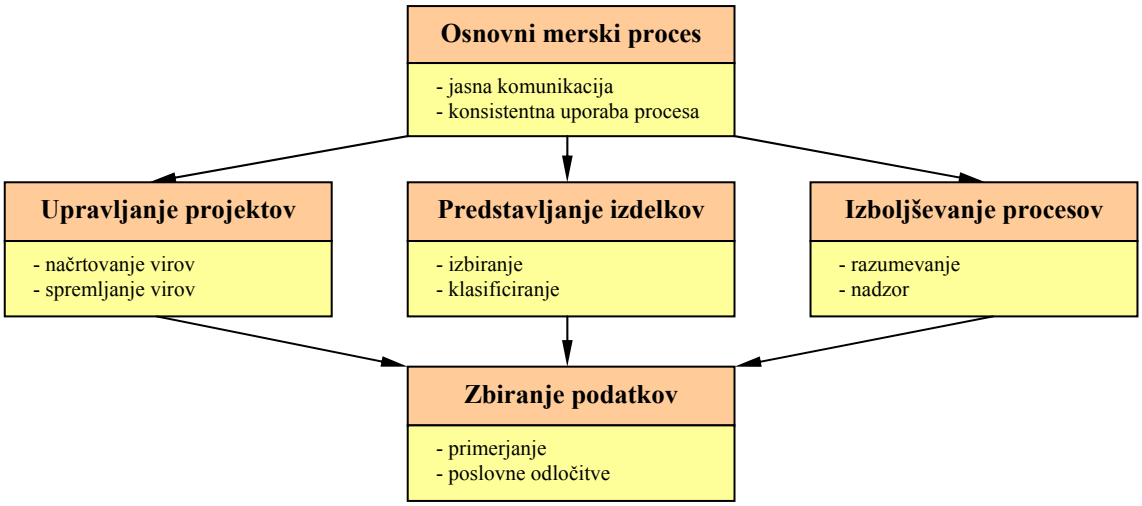
Viri nam predstavljajo osebje (posameznike in skupine), materiale, orodja in metode, ki jih trošimo ob izvajanju aktivnosti ali projekta. Pri virih nas običajno zanima obseg udeleženih virov, stroški in kakovost. Tipični notranji atributi so zopet velikost, cena, strukturiranost (skupine), medtem ko pri zunanjih atributih merimo predvsem kakovost, produktivnost, zanesljivost in uporabnost.

2.1.2.3 Uporaba meritev programske opreme

Meritve so ustrezno operativno definirane, če izpolnjujejo dva pogoja [SEL95, Flo97]:

- *Komunikativnost*: vsakdo natančno razume, kaj je namen meritve ter kakšne so predpostavke in okoliščine merjenja;
- *Ponovljivost*: če meritev ponovimo v enakih okoliščinah, dobimo enake rezultate.

Slika 2.3 predstavlja namen merjenja PO [Roz00]. Iz slike so razvidni trije vidiki merjenja: merjenje procesa, izdelkov in virov projekta.



Slika 2.3: Uporaba meritev PO [Roz00]

2.1.3 Proces razvoja programske opreme

Proces razvoja PO določa način delovanja organizacije za razvoj PO, torej tudi merjenja. V okviru naše naloge bo odigral pomembno vlogo, zato moramo spoznati nekaj njegovih osnovnih značilnosti.

2.1.3.1 Splošna definicija procesa

Proces je v širšem smislu definiran na naslednji način:

Definicija: *Proces je zaporedje korakov z namenom doseganja določenega cilja [IEE92].*
Poleg cilja, ki določa namen procesa, je osnovna značilnost procesa definiran časovni razpored izvajanja določenih aktivnosti. Posamezne aktivnosti morajo biti jasno opredeljene z opisom, vhodnimi in izhodnimi pogoji. Določene so na več ravneh: recimo koraki so najmanjša aktivnost, več korakov sestavlja faze in več faz celoten proces. Pri tem so pomembni tudi izvajalci posameznih aktivnosti in izdelki, ki predstavljajo rezultate (izhode) izvajanja aktivnosti ali vhode za aktivnosti [Fen97, Roz00]. Pri izvajanju aktivnosti si pomagamo z orodji.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Proces je dobro definiran, ko je dokumentiran, zagotavlja usposabljanje ter se redno izvaja. Proces mora biti tudi obvladljiv in merljiv ter primerno opisovati vse tri elemente: izdelke, aktivnosti in izvajalce (posameznike, skupine, orodja). Opis obsega vsaj [Roz00, Flo97]:

- način izvedbe vseh aktivnosti;
- začetek in konec vsake aktivnosti (pogoje za začetek, stanje po zaključku aktivnosti);
- vhode in izhode vsake aktivnosti (izdelke, ki jih te potrebujejo ali proizvedejo);
- izvajalce posameznih aktivnosti;
- opis namena posameznih aktivnosti;
- zaporedje aktivnosti.

Splošna definicija omogoča definicijo procesov na različnih področjih in v različne namene. Procese lahko definiramo tudi na različnih ravneh, kar pomeni, da lahko določeni procesi sestavljajo druge procese. Definicija omogoča tudi prepletanje različnih procesov v celovit proces. Tak primer je proces razvoja PO, ki združuje v sebi tehnološki, upravljalski, merski in druge podporne procese [Flo97].

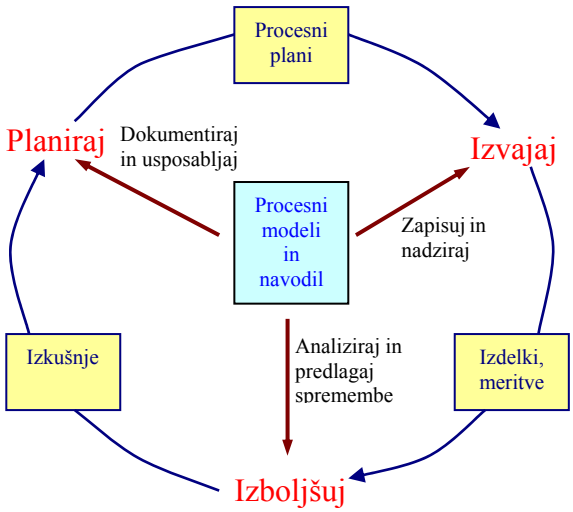
2.1.3.2 Proces razvoja programske opreme

Definicija: *Proces razvoja PO je sistem vseh opravil in potrebnih orodij, standardov, metod, postopkov in praks, ki so udeleženi pri proizvodnji in razvoju izdelkov PO skozi celoten življenjski cikel PO [Roz00].*

Definicija: *Proces razvoja PO je množica aktivnosti, metod, praks in transformacij, ki jih ljudje uporabljajo pri razvoju in vzdrževanju PO in z njo povezanih izdelkov (projektnih planov, načrtov, kode, testnih primerov, uporabniških priročnikov)[Pau93].*

Procese razvoja PO lahko definiramo enako kot vsak izdelek PO: ugotovimo potrebe, določimo cilje in kriterije kakovosti, zasnujemo nov proces na podlagi obstoječega stanja, definiramo nov proces, ga preverimo in postopoma izboljšujemo. Slika 2.4 prikazuje tri faze modeliranja procesa [Roz00]:

- **Planiraj:** zasnova osnove procesa razvoja PO glede na obstoječe stanje;
- **Izvajaj:** spremljanje stanja procesa;
- **Izboljšuj:** izvedba potrebnih sprememb, da dosežemo pričakovano stanje.



Slika 2.4: Faze modeliranja procesa [Roz00]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Opis procesa razvoja PO je potreben za planiranje, izvajanje in izboljšanje procesa oziroma njegovih sestavnih delov. Pri planiranju nam opis procesa na ravni organizacije predstavlja vzorec za planiranje aktivnosti pri projektu razvoja PO. V ta namen prilagodimo elemente in arhitekturo definiranega procesa razvoja PO na ravni organizacije za potrebe projekta. Proces razvoja projekta je vzorec za planiranje delovnih izdelkov, aktivnosti in izvajalcev.

Proces razvoja PO na ravni organizacije operativno definira delovne produkte, aktivnosti in udeležence, ki sestavljajo proces na ravni projekta. Tako omogoča izvajanje projekta s pomočjo [Roz00, SEL95]:

- standardov in vzorcev za delovne izdelke,
- dokumentiranih procesov, metod in postopkov za aktivnosti ter
- odgovornosti posameznih udeležencev.

Definiran proces je podlaga tudi za izboljšave. Uporablja se za predstavitev razvoja in vzdrževanja PO, podlago za izvajanje meritev ter procesne analize, repozitorij za shranjevanje naučenih lekcij in izkušenj, navodil za testiranje ter kot medij za sporočanje sprememb. Definiran proces služi kot [Roz00]:

- osnovni opis trenutne prakse razvoja PO;
- orodje za analizo in primerjavo procesov razvoja PO;
- repozitorij za skupno znanje (know-how) o razvoju PO;
- vzorec za načrtovanje razvoja PO;
- možnost za razširjanje opisa procesa, ki je primeren za vgraditev;
- stična točka za izboljšave razvoja PO.

2.1.4 Merski program

Uspešnost doseganja ciljev vsake organizacije je tesno povezana z možnostjo objektivne ocenitve svojega stanja ter obstojem nadzornih vzvodov, ki ji omogočajo ustrezen odziv [Flo97]. V ta namen moramo v organizacijski proces razvoja PO vpeljati merjenje. To mora biti popolno in dobro definirano ter vpleteno v proces razvoja PO. Poleg tega mora biti uporaba merjenja konsistentna, ponovljiva ter se mora postopoma nadgrajevati [Fen97]. Te cilje lahko dosežemo z definicijo merskega programa. Ta predstavlja integracijo vseh merskih aktivnosti v organizaciji z upravljalskega vidika [Fen97].

2.1.4.1 Merski plan

Merski plan opisuje uvajanje merskega programa v organizaciji. Celovit merski plan daje odgovore na naslednjih šest vprašanj:

Zakaj?
Uvajanje merjenja v organizacijo mora imeti smisel – podpirati mora poslovne cilje organizacije. Ti se določajo v vodstvu organizacije, torej mora pobuda in določitev ciljev merjenja priti od zgoraj.

Kaj?
Najpomembnejše vprašanje je določiti, katere meritve organizacija potrebuje. Ustaljen pristop [Roz00, SEL95, Fen97] je uporabiti metodo GQM [Bas88] (ta bo predstavljena v razdelku 2.1.5.1), ki omogoča na podlagi poslovnih ciljev določiti primerne meritve. Fenton [Fen97] opozarja, da je pri tem potrebno upoštevati zmožnost (zrelost) organizacije, da tako določene

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

meritve sploh izvede. Drugi pristop [Het93] predlaga, da najprej določimo meritve, ki jih lahko učinkovito izvedemo, ter na podlagi teh spremljamo ustrezne cilje. Ta pristop je predstavljen v razdelku 2.1.5.2.

Kje?
Učinkovito merjenje zahteva natančen opis, kje se bodo meritve izvajale oziroma kako bo merjenje integrirano v celoten razvoj PO. V ta namen je smiselno vse dejavnosti merjenja opisati z merskim procesom, ki v skladu z definicijo procesa ustrezno zajame vse podrobnosti opisa meritev. Trije modeli merskih procesov so predstavljeni v razdelkih 2.1.4.3, 2.1.4.4 in 2.1.4.5. Če uporabimo drugačen pristop, mora opis zajemati vse elemente (kot [SEL95]).

Kdaj?
Določena merjenja se izvajajo enkrat v okviru razvoja, druga se izvajajo neprenehoma. Če za opis uporabimo proces, je ta informacija že vsebovana, sicer jo moramo posebej določiti.

Kako?
Zelo pomembno je izbrane merske podatke ustrezno zbirati, obdelovati, preverjati in hraniti. Zato moramo predvideti ustrezne načine, pri čemer moramo paziti na kakovost podatkov (točnost, popolnost). Priporočljivo si je pomagati z orodji, vendar vedno to ni mogoče. Za hranjenje podatkov je treba predvideti ustrezne kapacitete – bazo merskih podatkov. Ker so merski podatki namenjeni uporabi, je treba tudi omogočiti dostop do njih.

Kdo?
Merski plan mora določiti odgovorne za vse aktivnosti nad merskimi podatki: zbiranje, oblikovanje, preverjanje, analiziranje, predstavljanje, uporabo in ovrednotenje. Treba se je zavedati, da so nekatere vloge v organizaciji namenjene tem aktivnostim, druge pa izvajajo vsi člani organizacije. Različni člani imajo tudi različne potrebe in poglede na podatke.

Merski plan je treba dopolnjevati, ko organizacija pridobi več izkušenj ali se merske zahteve spremenijo. Zgled dobrega merskega načrta je recimo [SEL95], planiranje uvajanja meritev pa je celovito predstavljeno tudi v [Flo97].

2.1.4.2 Načini opisa merskega programa

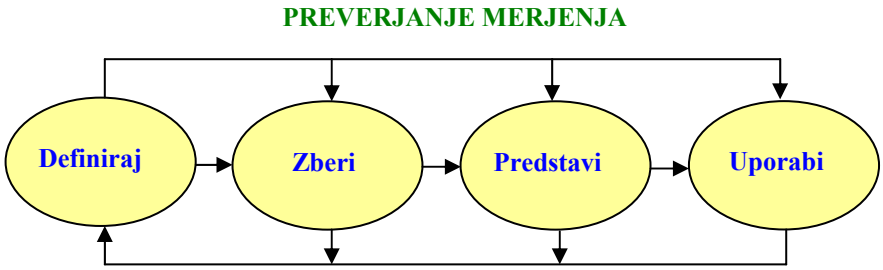
Merske programe je mogoče predstaviti na različne načine, zato se dolgo ni uveljavil enoten pristop in terminologija. Težavo odpravlja standard ISO/IEC 15939 – Informacijska tehnologija: Okvir merskega procesa PO [ISO01]. V njem je natanko določena terminologija in vsebina procesa merjenja. Standard je vgrajen v pristopu z imenom Praktično merjenje PO (PSM), ki se je zelo hitro razširil (tudi po zaslugi naročnika - ameriškega obrambnega ministrstva) [McG01]. Terminologija standarda ISO/IEC 15939 je upoštevana tudi v modelu CMMI (glej razdelek 2.1.2). V nadaljevanju sta predstavljena dva zgodnejša modela merskih procesov ter pristop PSM.

2.1.4.3 Model osnovnega merskega procesa po Hetzlu

Hetzel [Het93] definira model osnovnega merskega procesa (ang. Basic Measurment Process Model), ki temelji na naslednjih štirih aktivnostih (slika 2.5):

- definiciji, kaj želimo meriti,
- zbiranju rezultatov definiranih meritev,
- predstavitvi zbranih informacij in
- uporabi rezultatov meritev za sprožanje ustreznih akcij.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 2.5: Model merskega procesa po Hetzlu [Het93]

Vzporedno s temi aktivnostmi se vrši preverjanje postopka merjenja. Za aktivnosti ni nujno, da se izvajajo zaporedno.

Definicija je namenjena čim natančnejšemu definiranju posameznih meritev. Te so predstavljene s specifikacijami meritev, ki za vsako meritev obsegajo vsaj naslednje dele: naziv, opis, zbiranje, pogostost, enoto, obseg in uporabo. Primer specifikacije je na sliki 2.6.

Naziv:	**KLOC**
Opis:	Število novih ali spremenjenih vrstic izvorne kode pri projektu ali sistemu v tisočih.
Zbiranje:	Avtomatizirano zbiranje z uporabo orodja KLOC-Count vsakič, ko se začne testiranje posameznih modulov. Podrobnejši opis orodja se nahaja v ...
Pogostost:	Vsakič, ko se izvorna koda spremeni ali vnese v knjižnico.
Enota:	Zaokroženo število tisočic števila vrstic, kjer je vrstica definirana kot poljubna neprazna vrstica (torej tudi deklaracije, komentarji, ipd).
Obseg:	Več ali enako 0, navzgor ni omejen.
Izkušnje:	Meritev se že uporablja 10 mesecev. Rezultati so zadovoljivi. Izmerjene vrednosti so bile med 1 in 312.
Uporaba:	Meritev uporabljamo tudi za ocenjevanje potreb projekta. Ocenjena vrednost v KLOC je vhodni podatek orodju za ocenjevanje obsega dela in produktivnosti
Preverjanje:	PB projekta vsebuje obseg dela in KLOC za 105 projektov. Stopnja korelacije je 0.82, pri 12% projektov pa je ocenjen obseg dela odstopal več kot 20%. Enako pričakujemo pri trenutnem projektu. Meritve ne smemo uporabiti za merjenje posameznika, ker velja ta stopnja korelacije samo za projekt kot celoto.
Težave:	- Ne upošteva / šteje vrstice ustvarjene izvorne kode, ki ni namenjena za izročitev uporabniku (recimo generatorja testnih primerov ipd). - Ne obstaja avtomatiziran zajem za module, ki so napisani v prog. jeziku ... na platformi ...

Slika 2.6: Specifikacija meritve za štetje vrstic programske kode [Het93]

Zbiranje rezultatov zajema vse aktivnosti zbiranja in shranjevanja merskih rezultatov v skladu s specifikacijo meritev. Izvedbo zbiranja priključimo ostalim procesnim aktivnostim na čim bolj nevsiljiv način, po možnosti pa jo tudi avtomatiziramo. Zajete vrednosti ovrednotimo že ob zajemu, zbrane podatke pa hranimo za bodočo uporabo.

Predstavitev zbranih informacij obsega analizo zbranih podatkov ter njihovo predstavitev ciljni publiki. Med analizo in pripravo predstavitve se moramo vedno zavedati, kaj želimo pokazati, zato je jasnost predstavitve včasih pomembnejša kot natančnost zbranih rezultatov. Tako pri analizi kot pri predstavitvi si pomagamo z orodji.

Uporaba rezultatov meritve izvede ukrepe za izboljšanje kakovosti razvojnega procesa ali produktov na podlagi zbranih podatkov. Cilj je vsaj razširiti znanja o posameznih vidikih razvojnega procesa.

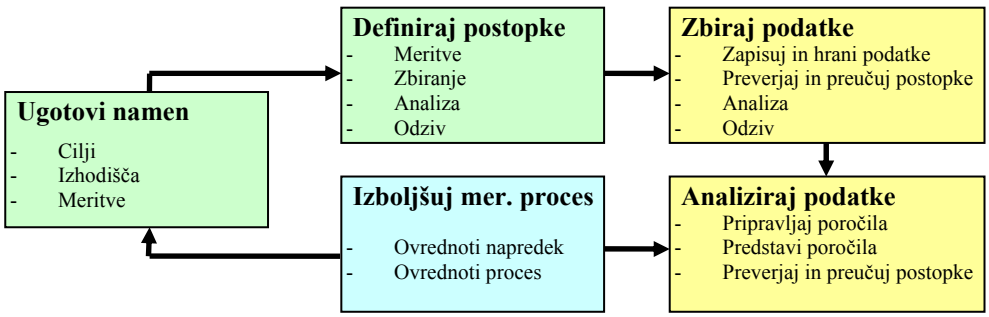
Preverjanje merjenja je aktivnost, ki poteka vzporedno z ostalimi in poskrbi, da so te popolno in pravilno izvedene. Obsega preverjanja pravilnosti, popolnosti in uporabnosti. Tipični pristopi zajemajo naknadno preverjanje in navzkrižne kontrole, koristno pa je v nadzor vključiti tudi udeležene razvijalce.

2.1.4.4 Model merskega procesa po McAndrewsu

McAndrewsov model merskega procesa (ang. measurement process framework) [McA93] obsega tri faze (slika 2.7). Vsako fazo sestavlja ena ali dve aktivnosti, ki sta definirani z več značilnostmi: namenom (ali ciljem), vhodom, opravili, načini preverjanja ter izhodom. Faze (in aktivnosti) merskega modela so naslednje:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- planiranje:
  - ugotovi namen,
  - definiraj postopke.
- implementacija – vgraditev:
  - zbiraj podatke,
  - analiziraj podatke.
- razvoj:
  - izboljšuj merski proces.



Slika 2.7: Model merskega procesa po McAndrewsu [McA93]

Ugotovi namen je aktivnost, ki določi namen in cilje uporabe merjenja. Pri tem v skladu s pristopom GQM določi izhodišča, cilje, vprašanja ter meritve ali pokazatelje. Ob zaključku imamo identificirane uporabnike, nabor merljivih ciljev ter preverjeno množico meritev.

Definiraj postopke je aktivnost, s katero natančno določimo tri stvari: meritve, postopke zbiranja teh in mehanizme shranjevanja. To storimo v več korakih, končni rezultat pa je natančno definiran merski načrt, ki vsebuje ustrezno delitev odgovornosti in metode preliminarne analize.

Zbiraj podatke je aktivnost namenjena zbiranju, zapisovanju in shranjevanju merskih podatkov v skladu z določenimi postopki. Ker se postopki sproti preverjajo, so rezultat te faze poleg preverjenih podatkov tudi predlogi za izboljšavo postopkov zbiranja.

Analiziraj podatke opravi analizo zbranih podatkov in o ugotovitvah poroča. Pri tem je pomemben del izdelava poročil, ki nazorno prikažejo rezultate analize.

Izboljšuj merski proces je zadnja aktivnost, v kateri na podlagi poročil sprejemamo odločitve in določimo področja izboljšav merskega procesa. Pri tem skušamo ovrednotiti dosežen napredek in proces. Sprejete odločitve so navodilo za popravke merskega procesa v naslednji iteraciji njegovega razvoja.

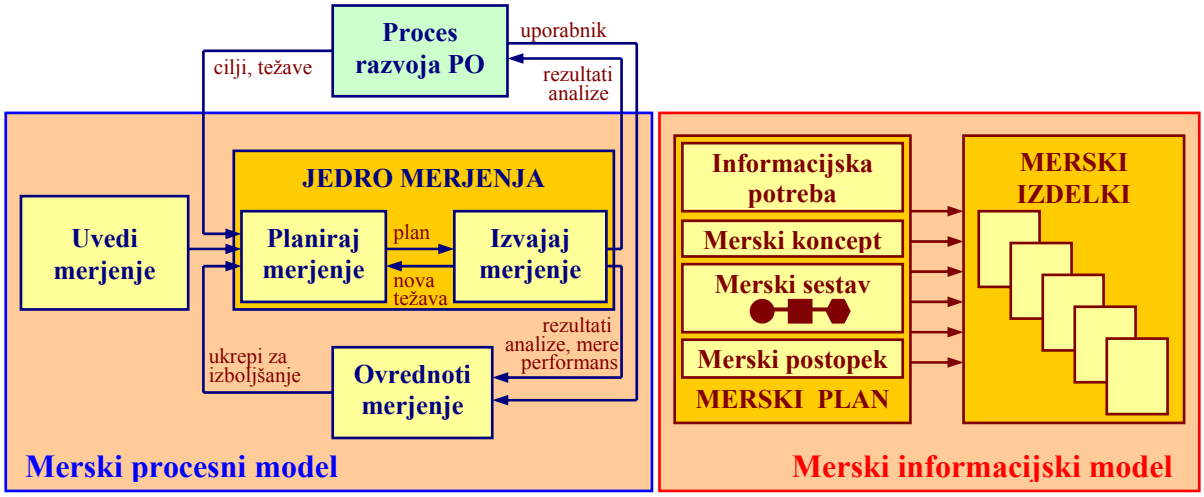
2.1.4.5 Praktično merjenje PO (pristop PSM)

Praktično merjenje PO [McG01, DoD98[15], DoD02[1], PSM] opisuje pristop za definiranje merskega programa v okviru projekta razvoja PO. Pristop je utemeljen na standardu ISO/IEC 15939. Izhodišče za merjenje je zahteva, da izbrane meritve ustrezajo informacijskim zahtevam projekta ter da se merjenje izvaja na strukturiran in ponovljiv način. V ta namen pristop določi dva modela, ki sta podlaga za definicijo merjenja (slika 2.8):

[15] V [DoD98] in [DoD02] je opisana tehnična izvedba pristopa PSM.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- merski informacijski model (ang. Measurement Information Model) in
- merski procesni model (ang. Measurement Process Model)



Slika 2.8: Struktura pristopa PSM [McG01, DoD02]

Merski informacijski model je formalni mehanizem za povezavo informacijskih potreb s procesi in izdelki, ki jih merimo. V ta namen je definirana struktura, iz katere vsak projekt razvije svoj informacijski model, ki določa vsebino merjenja. Ta je osnovni vir za izvajanje merskega procesa, vodi planiranje in izvajanje zbiranja merskih podatkov ter aktivnosti analize. Uporablja terminologijo, ki je usklajena s standardom ISO/IEC 15939.

Izhodišče za določanje modela so *informacijske potrebe* (ang. information needs), ki lahko služijo projektu ali organizaciji. Vezane so na zastavljene cilje in področja potencialnih težav: probleme, tveganja ali pomanjkanje informacij. Na podlagi teh se določi *merski koncept* (ang. measurement concept), ki je predstava o entitetah, ki jih moramo meriti zaradi informacijskih potreb. Formaliziran opis merskega koncepta je *merski sestav* (ang. measuremnt construct).

Ta natanko opisuje merjenje s tremi ravnmi meritev: *osnovnimi meritvami* (ang. base measurement), *izpeljanimi meritvami* (ang. derived measurement) in *pokazatelji* (ang. indicator). Osnovna meritev je neodvisna meritev atributa merske entitete (procesa, izdelka ali vira). Izpeljana meritev je kombinacija dveh ali več (osnovnih ali izpeljanih) meritev. Pokazatelj je kombinacija dveh ali več mer, ki so prikazane v skladu z informacijsko potrebo.

Merski postopek opisuje zbiranje in uporabo podatkov iz merskega sestava. Zbirka vseh informacijskih potreb, merskih sestavov in postopkov je *merski plan*. Izvedba merskega plana proizvede *informacijske izdelke* (ang. information products), ki ustrezajo vsem informacijskim potrebam. Za razvoj merskega programa moramo razumeti informacijske potrebe, definirati množico merskih konceptov in poznati merljive entitete projekta.

Merski procesni model določa izvedbo merjenja v skladu z definiranim informacijskim modelom. Narejen je v skladu s pristopom Planiraj Izvedi Preveri Ukrepaj (ang. Plan Do Check Act) [Dem86] in je direktna implementacija pristopa, ki ga opisuje standard ISO/IEC 15939 [ISO01]. Obsega štiri dele:

- uvajanje merjenja (ang. Establish and Sustain Commitment),
- planiranje merjenja (ang. Plan Measurement),
- izvajanje merjenja (ang. Perform Measurement) in
- ovrednotenje merjenja (ang. Evaluate Measurement).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Uvajanje merjenja poskrbi za ustrezno podporo merjenju tako na ravni projekta kot organizacije. Obsega zagotavljanje virov in podpore za izvajanje merjenja.

Planiranje merjenja obsega določanje informacijskega modela in vgraditev merjenja v obstoječi proces razvoja PO. Rezultat je popolno definiran pristop, ki temelji na informacijskih potrebah.

Izvajanje merjenja obsega zajem merskih podatkov, njihovo obdelavo, uporabo za analizo informacijskih potreb, pripravo merskih izdelkov in predstavitev le-teh. Izvajanje merjenja implementira merski plan in pripravi zahtevane informacije za odločanje.

Ovrednotenje merjenja izvede merjenje in analizo samega merskega procesa. Oceni meritve in zmožnost procesa ter na podlagi tega predlaga akcije za nenehno izboljšavo merjenja.

Posamezni deli merskega procesa se izvajajo iterativno in na ta način poskrbijo za nenehno nadgrajevanje procesa.

Podrobneje je pristop PSM predstavljen v poglavjih 4 in 5. Rezultat disertacije je namreč izviren pristop za merjenje v majhnih organizacijah za razvoj IS, ki je utemeljen na pristopu PSM, zato je primerjalno opisan glede na pristop PSM.

2.1.5 Določanje nabora meritev

Najpomembnejši cilj merskega plana je določiti nabor meritev za učinkovito doseganje ciljev organizacije. Pri tem ločimo dva nasprotujoča si pristopa:

- od zgoraj navzdol (angl. top-down) ali ciljno usmerjen pristop (ang. goal based approach);
- od spodaj navzgor (angl. bottom-up) ali objektni pristop (ang. object based approach).

Pri prvem pristopu najprej opredelimo, kaj želimo doseči (cilje) in na podlagi tega postopoma preko vprašanj določimo ustrezne meritve. Te nam omogočajo spremljati, v kolikšni meri te cilje dosegamo. Najbolj znan primer pristopa je metoda GQM (ang. Goal Question Metric).

V nasprotju s tem pristop od spodaj navzgor temelji na ideji, da lahko učinkovito merimo samo določene značilnosti konkretnih entitet (izdelkov, virov ali procesov) in da samo te nudijo dobro podlago za spremljanje razvoja PO. Zato si ciljev ne določimo vnaprej, temveč si jih razberemo na podlagi izvedenih meritev. Ta pristop ni tako uveljavljen, predstavlja pa ga metoda MQG (ang. Metric Question Goal).

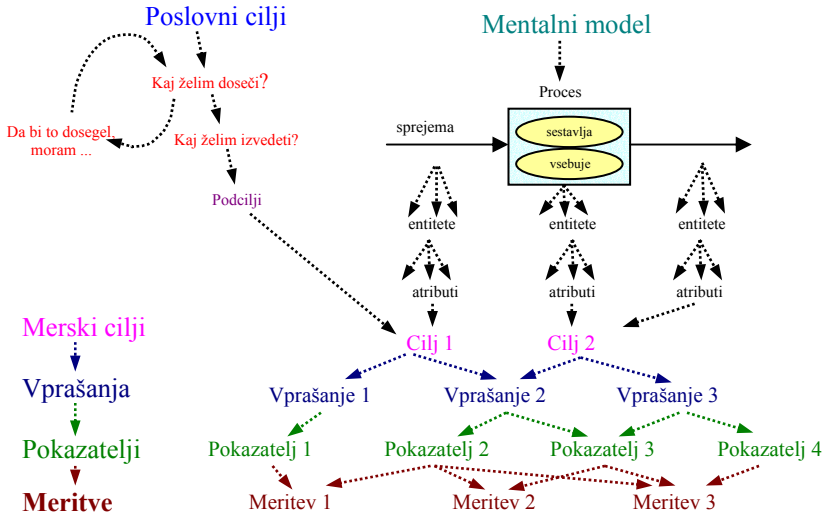
Pristopa si delita več skupnih značilnosti. Oba temeljita na jasnih modelih, kjer je osnovna prvina ponavljanje. Oba tudi predvidevata neprestano izboljševanje merjenja na podlagi preverjanja trenutnih rezultatov.

2.1.5.1 Pristop od zgoraj navzdol: metoda Goal Question Metric (GQM)

Najbolj znan predstavnik pristopa od zgoraj navzdol (top-down) je metoda GQM (ang. Goal Question Metric) [Bas88, Par96]. Idejna očeta pristopa sta raziskovalca Victor Basili in Howard Rubin. Metoda se je zelo uveljavila, bila nadgrajena (recimo [Sym91]) in se veliko uporablja v industriji, kjer daje dobre rezultate (recimo NASA [SEL95], Motorola [Das94], Nokia [Kil01]) [Fen97].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Metoda GQM (slika 2.9) v več korakih prevede neformalne poslovne cilje organizacije v formalno mersko strukturo. Osnovno vodilo je določiti, kaj želimo z merjenjem doseči, zato najprej na podlagi želja in potreb organizacije definiramo, kakšne merske cilje želimo zagotoviti ali spremljati. Vsakemu merskemu cilju ustreza množica vprašanj z merljivimi odgovori, te odgovore pa dobimo z uporabo primernih merjenj. Določanje ustreznih vprašanj in meritev je zahtevno, zato je metoda natančno določena s predlogami in navodili.



Slika 2.9: Metoda GQM (Goal Question Metric) [Bas88]

Podrobnejši opis metode: izhodišče za merjenje so informacijske potrebe organizacije oziroma njeni poslovni cilji. Za izbrano vlogo (udeleženca) določimo seznam entitet, ki ga razdelimo na več skupin (izdelki, viri, aktivnosti). Nato vsaki entiteti pripišemo množico vprašanj, ki se nanašajo na izbrano vlogo. Iz te izberemo tisto podmnožico vprašanj, ki ustreza poslovnemu cilju ter na podlagi teh izpeljemo podcilje in iz podciljev merske cilje.

Merski cilji so lahko aktivni ali pasivni. Aktivni cilji imajo za posledico spremembe (recimo povečati zanesljivost), medtem ko so pasivni cilji namenjeni razumevanju ali spoznavanju (recimo ocenitev uporabnosti izdelka). Merski cilji so predstavljeni s 4 elementi: objektom, namenom, vidikom in opisom okolja in omejitev. Vsak element je natančno določen s stavkom predpisane oblike, ki ga določimo po predlogi (slika 2.10).

Objekt: Proces razvoja PO v podjetju XYZ.
Namen: Proučiti razvojne napake v PO **z namenom** zmanjšanja stroškov razvoja in povečanja kakovosti končnega izdelka.
Vidik: Prouči pojavljanje, odkrivanje in odpravljanje napak **z vidika** skupine za izboljšanje procesa.
Okolje: Podjetje za telekomunikacije. CMM stopnja 2. 2500 zaposlenih, 450 razvija PO. Uporabljajo Oracle, programski jezik C++. Omejitev: prouči se samo dokončane projekte.

Slika 2.10: Primer opisa merskega cilja pri metodi GQM [Bas88]

Na vsak merski cilj vežemo enega ali več (merljivih) vprašanj, ki se lahko nanašajo na izdelke ali proces. Vprašanja o izdelkih se nanašajo na same izdelke (atribute, stroške, spremembe, napake) ali zagotavljanje kakovosti le-teh (modeli, veljavnost, učinkovitost). Vprašanja o procesu so podobno povezana s samim procesom (uporabo) ali kakovostjo tega (veljavnost, modeli). Vsakemu odgovoru ustreza eden ali več pokazateljev. Pokazatelje merimo z (objektivnimi in subjektivnimi) meritvami, pri čemer je lahko ista meritev udeležena pri več pokazateljih. Pri izbiri meritev imajo prednost tiste, ki so vezane na več pokazateljev.

Prednost metode GQM je osredotočenost na smisel zbiranja informacij, zaradi česar je metoda upoštevana in razširjena [Fen97]. Vendar ima metoda tudi nekaj težav [Fen97, Het93]:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Večnivojska drevesna struktura izpeljave ima za rezultat (pre)veliko število meritev.
- Metoda ne omogoča povezovanja meritev, ki so bile izpeljane iz različnih ciljev.
- Težko je določiti ustrezne cilje, če so poslovni cilji zelo splošni.
- Meritve so določene od vrha navzdol, zato dejanski izvajalci merjenj nanje nimajo vpliva, čeprav opazijo očitne slabosti.
- Predlagatelji meritev se ne zavedajo težav pri izvajanju meritev, zato je mogoče, da bodo izvajane neustrezno ali rezultati prilagojeni pričakovanjem.

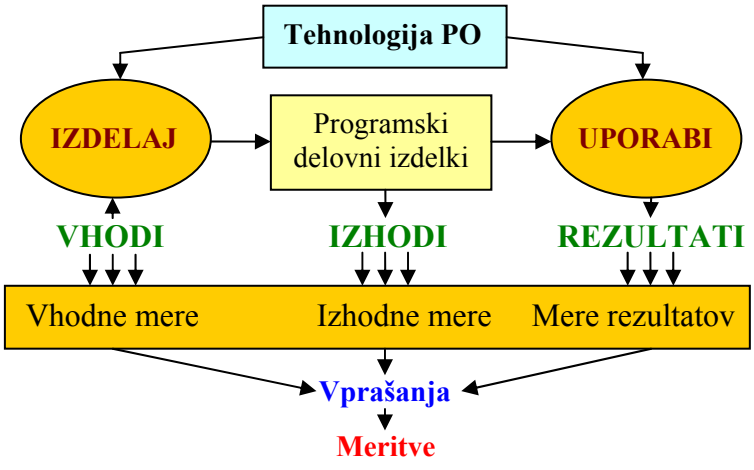
2.1.5.2 Pristop od spodaj navzgor: metoda Metric Question Goal (MQG)

Metodo MQG (ang. Metric Question Goal) [Het93] sta predstavila William Hetzel in William Silver, razvita pa je bila na Univerzi v Marylandu v začetku devetdesetih. Temelji na ideji, da morajo biti meritve zasnovane tam, kjer se dejansko izvajajo – v okviru tehnološkega procesa. Samo tako definirane meritve je mogoče učinkovito izvajati in z rezultati teh meritev lahko pridobimo vse smiselne informacije. Na podlagi rezultatov si lahko postavljamo smiselna vprašanja in utemeljeno sklepamo, katere cilje dosegamo.

Metoda definira, da sta za vsako aktivnost pri razvoju PO značilna dva objekta: delovni izdelek (rezultat aktivnosti) in izvajalci aktivnosti. Med delovne izdelke štejemo vse izdelke, ki nastanejo med razvojem: recimo študije izvedljivosti, opise zahtev, specifikacije, plane, module programske kode, izvedene teste, poročila in drugo. V procesu razvoja nastajajo izdelki, ki jih razvijalci v nadaljnjem razvoju zopet uporabijo za izdelavo novih izdelkov, dokler veriga ne pripelje do končnega izdelka. V takem procesu merimo osnovne značilnosti pri izdelavi ali uporabi vsakega delovnega izdelka. Pri izdelkih merimo tri vrste lastnosti:

- Vhodne merske lastnosti (ang. input measures): predstavljajo vire (udeležence, orodja), aktivnosti in ostale delovne izdelke, ki smo jih porabili med razvojem izdelka.
- Izhodne merske lastnosti (output measures): opisujejo delovne izdelke (recimo njihovo velikost ali kompleksnost) ter vložen obseg dela izvajalcev.
- Merske lastnosti rezultatov (results measures): opisujejo stopnjo zadovoljstva in izkušnje ob uporabi delovnega izdelka.

Meritve tako pokrivajo vse značilnosti razvoja, zato omogočajo odgovor na vsako (smiselno) vprašanje ne glede na cilje. Na podlagi rezultatov meritev lahko višji razvijalci postavijo utemeljena vprašanja ter preverijo, kakšne cilje dosegajo. To je sprožilec za spremembe na višjih ravneh, ki jih je zopet mogoče zaznati samo z merjenjem osnovnih značilnosti delovnih izdelkov. Meritve se spremenijo samo tedaj, ko se spremenijo postopki ali tehnologija.



Slika 2.11: Primer izvedbe MQG pristopa (Metric Question Goal) [Het93]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Primer izvedbe metode MQG je IOR model merjenja. Kot je razvidno iz slike 2.11 pred in med razvojem programskih izdelkov merimo vhodne mere (ang. I-Measures). Ko je aktivnost končana, njo in izdelke izmerimo in dobimo izhodne mere (ang. O-Measures), ob uporabi izdelkov pa dobimo še mere rezultatov (ang. R-Measures). Z ovrednotenjem vseh treh določimo vprašanja in cilje, iz teh pa izpeljemo dopolnitve meritev.

Metoda MQG temelji na merjenju jasno določenih delovnih izdelkov, kar ima za posledico nabor smiselnih meritev. Ker za meritve ni vnaprej znanih ciljev in izvajalec meritev ne razmišlja o posledicah, so bolj objektivne. Vendar ima metoda nekaj očitnih pomanjkljivosti:

- Osnovna težava je zagotavljanje doseganja izbranih (poslovnih) ciljev. Kdo vodi in kdo nadzira merjenje, komu služijo rezultati meritev?
- Metoda meri vse, kar je smiselno meriti, zato je število meritev lahko zelo veliko.
- Problem je tudi učinkovitost, saj merjenje stane, merimo pa veliko stvari brez razloga.

2.1.6 Praktične izkušnje pri uporabi meritev

Področje meritev PO je zelo obsežno in o njem je bilo veliko zapisanega. Veliko je opisov celovitih merskih rešitev za konkretne organizacije (recimo za Contel [Pfl90], Hewlett-Packard [Gra94], Nokio [Kil01], Bull [Wel94], Siemens [Pau94], NASO [SEL95, Sta94] in druge), še več pa predstavitev posameznih meritev in pristopov (recimo [Alb83, Has01, Mun92, Col94, Pig98, Ohl98, Wit97]). Znani so nasveti, kako določiti merski program (recimo [Fen99, Dek01, DeM95, Pud01]). Predlagani so minimalni obsegi meritev, ki naj bi bili primerni za vsako organizacijo (recimo [SEL95, Car92]), pa tudi meritve za specifične vrste PO (recimo objektno usmerjene programske jezike [Ebe97, Her98], funkcijske programske jezike [Van95], hipermedije [Men98] in podobno).

Podrobna predstavitev različnih idej presega namen disertacije. Najuspešnejše izkušnje so se uveljavile v standardih in modelih in nekatere so posredno predstavljene v naslednjih delih disertacije. Vendar so z vidika naše naloge pomembne tudi nekatere druge izkušnje. Zato v strnjeni obliki predstavljamo izbor izkušenj in nasvetov, ki jih predlaga več razvijalcev PO.

2.1.6.1 Načela tehnologije programske opreme in načela merjenja

Basili [Bas88] je že pred dobrim desetletjem strnil izkušnje pri uporabi meritev v deset načel tehnologije PO in štirinajst načel merjenja, ki (pretežno) veljajo še danes:

Načela tehnologije programske opreme

Tabela 2.4 prikazuje pet načel za (a priori) vgrajevanje kakovosti v proces razvoja PO.

P1	*Ločevati je treba med izgradnjo in analizo PO. Le izboljšava izgradnje ima za posledico višjo kakovost, analiza služi zgolj za nadzor. Dobro izvedena analiza zato ne more nadomestiti pomanjkljive izgradnje, lahko pa omogoči učinkovit nadzor le-te.*
P2	*Izgradnja PO mora vsebovati izdelavo formalnega načrta izvedbe faze, s čemer vgradimo kakovost v temelje razvoja.*
P3	*Za zagotovitev učinkovite izvedbe razvojnega procesa PO je nujno formalizirati analizo ter izboljšavo procesa izgradnje PO.*
P4	*Analiza mora jasno določiti, če so bile tehnološke aktivnosti pri izgradnji PO izvedene pravilno in popolno. To je še posebej pomembno, ker niso vse aktivnosti formalizirane.*
P5	*Razvijalci in uporabniki PO potrebujejo sproten odziv na svoje delo, da bi lahko izboljšali izgradnjo PO, ki jo razvijajo v okviru tekočega projekta. Organizacija za razvoj PO potrebuje odziv po zaključku projekta, da bi lahko ustrezno izboljšala proces izgradnje pri bodočih projektih.*

Tabela 2.4: Načela tehnologije PO/I [Bas88]

Drugi del načel uvaja prilagajanje razvoja različnim vrstam procesov in izdelkov (tabela 2.5).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

P6	*Vsa okolja in programski izdelki se razlikujejo med seboj. Razlike morajo biti očitno predstavljene v specifikacijah in upoštevane med razvojem PO.*
P7	*Obstaja veliko metodologij, ki jih lahko uporabljamo pri razvoju PO, vendar moramo vsako prilagoditi organizaciji, projektu in programskemu izdelku.*
P8	*Prilagajanje procesov značilnostim organizacije, projekta in izdelka oziroma prilagajanje za doseganje višje kakovosti in učinkovitosti mora biti tudi formalizirano.*
P9	*Pri prilagajanju procesov moramo koristno uporabiti izkušnje in znanja, ki pa jih moramo prilagoditi spremenjenim okoliščinam.*
P10	*Zaradi stalne potrebe po prilagajanju je ključna vloga prilagodljivega vodstva, ki spremembe nadzira. Njena vloga mora biti v procesu razvoja PO podprta.*

Tabela 2.5: Načela tehnologije PO/II [Bas88]

Načela merjenja programske opreme

Prva štiri načela opisujejo namen merjenja oziroma uvajanja merskega procesa (tabela 2.6).

M1	*Merjenje je idealen mehanizem za opisovanje, ovrednotenje, napovedovanje in motiviranje za več vidikov pri procesu razvoja PO.*
M2	*Meriti je treba proces razvoja PO kot izdelke PO. Izboljšava izdelka zahteva poznavanje značilnosti izdelka kot postopka izdelave.*
M3	*Obstaja vrsta uporab merjenja, ki pa morajo biti jasno definirane.*
M4	*Merjenje mora biti upoštevano s primerne perspektive. Organizacija, razvijalci, projektni vodje, uporabniki vidijo produkt in proces drugače in to morajo meritve upoštevati. Vsakdo od njih bo želel videti različne stvari in različne stopnje podrobnosti.*

Tabela 2.6: Načela merjenja PO/I [Bas88]

Nadaljnja načela se nanašajo na značilnosti meritev in merskega procesa (tabela 2.7):

M5	*Ker se vseh značilnosti PO ne da objektivno meriti, potrebujemo poleg objektivnih tudi subjektivne mere. Te nam omogočajo, da s primerno natančnostjo ovrednotimo merjeno vrednost.*
M6	*Večina vidikov razvojnega procesa in izdelkov je tako kompleksnih, da jih ni mogoče zajeti z eno samo meritvijo. Zato je potrebno določiti množico meritev glede na določene merske potrebe.*
M7	*Za izvajanje merjenja in analize morajo biti razvita primerna okolja za razvoj in vzdrževanje. Proces razvoja PO mora vsebovati tudi načrtovanje teh aktivnosti, ki pa jih je mogoče prilagajati spremembam.*
M8	*Že znanih merskih modelov in meritev ne moremo direktno uporabiti, temveč jih moramo prilagoditi novim zahtevam ter dodatno preveriti v novem okolju.*
M9	*Običajno je od zgoraj navzdol boljši pristop za določanje primernih meritev, ker omogoča primerno interpretacijo in analizo ciljev meritev, pa tudi sledljivost in prilagodljivost.*
M10	*Za vsako okolje obstaja nabor lastnosti in meritev, ki zagotavlja primeren obseg informacij za predstavitev in uporabo.*
M11	*Pri zbiranju podatkov uporabljamo različne mehanizme, odvisno od narave merjenja (ročno / avtomatsko).*
M12	*Zaradi ovrednotenja in primerjanja projektov potrebujemo PB z merskimi podatki. Podatki morajo odražati tudi okolje, v katerem so bili zbrani.*
M13	*Meritve morajo biti določene tudi z opisom, katerega pomen je odvisen od konteksta uporabe.*
M14	*Postopoma je treba PB razviti v bazo znanja, da bo formalizirala ponovno uporabo zbranih izkušenj.*

Tabela 2.7: Načela merjenja PO/II [Bas88]

2.1.6.2 Nasveti za izvajanje meritev

Nasvete navajamo skupaj z glavnimi viri, ki so jih predlagali:

a) Kakšne meritve potrebujemo?

- Enostavne, učinkovite in poceni. [Rif91, Roz00, SEL95, DeM86, Pfl90, Gra94]
- Meritve morajo biti objektivne in ponovljive. [Bas88, Gra94]
- Treba je razumeti način izvajanja merjenja. [Roz00, Pfl90, Pau94]
- Meritve morajo biti natančno in popolno določene. [Rif91, Pfl90, Gra94]
- Uporabljajo se enostavne definicije faz življenjskega cikla razvoja PO. [Roz00]
- Meritve morajo biti del nečesa večjega – recimo truda organizacije za kakovostnejši proces. [Rif91, Pfl90, Kil01, Pau94]

b) Kaj merimo?

- Merimo naravne rezultate razvoja, čim bližje izvoru. [Het93, Gra94, DeM86, DeM95]
- Osredotočiti se je treba na lokalne meritve, ne splošne. [Roz00]
- Merimo procese, vire in izdelke. [SEL95, Pfl90, Gra94, Pau94]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

c) Obseg merjenja

- Začeti z malim naborom meritev, postopoma nabor širiti. [Rif91, Roz00, Pfl90, Car92]
- Tudi nasploh je treba uporabljati kar najmanj meritev. [Roz00, DeM95, Gra94]
- Obseg merjenja in nadzora je treba prilagoditi resnosti cilja (posledicam). [Bas88]
- Izogibati se je treba preobsežnim rezultatom meritev. [Roz00, DeM95]

d) Koga meriti?

- Ne smemo meriti posameznika, temveč skupine. [Rif91, SEL95]
- Meritve ne smejo biti sredstvo za obtoževanje, temveč za spodbudo. [Rif91]
- Analizo meritev je treba ločiti od razvijalcev PO. [Roz00, DeM86]

e) Stroški merjenja

- Predvideti je treba stroške za izvedbo meritev. [Roz00, Pau94]
- Meritve stanejo veliko denarja, obsega dela, zahtevajo uvajanje. [Rif91, DeM86]
- Uvedba dobrih meritev zahteva spremembo navad in načina dela. [Rif91, Pau94]
- Uvajanje meritev mora biti postopno in zahteva dovolj časa. [Rif91, DeM86]

f) Avtomatizacija merjenja

- Zbiranje in poročanje naj bo avtomatizirano, če je le mogoče. [Rif91, Gra94]
- Ne pričakuj, da ti bo uspelo popolnoma avtomatizirati zajem meritev. [Roz00, Pau94]
- Tudi avtomatizacija analize rezultatov ima meje. [SEL95, Pau94]
- Uporabi dostopna, komercialna orodja. [Roz00, Pfl90]

g) Pomanjkljivosti meritev

- Pričakuj površno, nenatančno in nekonsistentno izvedbo merjenja. [Roz00, Pau94]
- Tudi merski podatki so nenatančni, napačni, neusklajeni, nepopolni. [SEL95, Pau94]
- Razlaga dobljenih rezultatov je omejena s sposobnostmi analitika. [SEL95]
- Obsega dela za odpravo napak ni mogoče natančno predvideti ali izmeriti. [Roz00, Wel94]
- Splošen nabor dobro definiranih meritev ne obstaja. [Roz00, Ebe97, Van95]
- Zares uporabne PB že zbranih merskih rezultatov ni. [Roz00, Van95]

h) Smisel in vsebina merjenja

- Namen merjenja mora biti znan in jasen. [Bas88, Roz00, Gra94, Kil01, Ali02]
- Treba je razumeti visokonivojske značilnosti procesa. [Bas88, Roz00, Pfl90]
- Meritve naj izhajajo iz ciljev organizacije. [Bas88, Rif91, Roz00, Pfl90, Gra94, Kil01]
- Merjenje mora upoštevati posebnosti (razvoja, PO, ipd.). [Fen97, Van95, Ebe97]
- Meritve morajo odražati potrebe vseh uporabnikov. [Rif91, Gra94, Ali02]
- Meritve naj imajo pomen v okviru procesa, naj bodo koristne. [Bas88, Pfl90]
- Meritve naj služijo za izboljšavo ali spremembo, ne zgolj primerjavo. [SEL95, Bas88]
- Pozitivni učinki merjenja naj bodo vidni čim prej. [Roz00, DeM86]

i) Meritve so več kot le zbiranje podatkov

- Cilj je uporaba ugotovitev merjenja in ne samo merjenje.[Roz00, Pfl90, Gra94, Fen99]
- Bistvo je načrtovanje, razumevanje in uporaba meritev, ne zbiranje podatkov. [Rif91]
- Načrtuj vsaj trikrat toliko sredstev za analizo kot za zbiranje podatkov. [Roz00]
- Tudi analiza ne sme imeti prevelikega obsega, sicer trpi razvoj. [Het93]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

j) Napovedovanje

- Pri merjenju si je treba zastaviti pričakovanja. [Roz00, DeM86, Pfl90]
- Meritve morajo omogočati napovedovanje. [Roz00, SEL95, DeM86, Pfl90]

k) Praktični nasveti

- Zbiraj podatke o obsegu dela vsaj enkrat mesečno. [Roz00, Gra94, Wel94]
- Zbiraj podatke o številu napak samo za tisto PO, katere razvoj je bil nadziran. [Roz00]
- Za merjenje velikosti PO uporabi število vrstic programske kode. [Roz00, Pfl90]
- Jasno definiraj, kateri del PO je vštet v merjenje in kateri ne. [Roz00, Pau94]

Za celovito spoznavanje področja merjenja PO bralcu priporočamo [Fen97], zanimiva pa so tudi dela [DeM86] in [Kan95]. Praktični pregled merjenja nudi [Oma97], pregled stanja meritev pa tudi [SEL95, Rif91, Kul00]. Za podrobnejše spoznavanje posameznih vidikov naj bralec uporabi literaturo iz zadnjega dela disertacije.

Fenton [Fen00a] ugotavlja, da bo razvoj na področju merjenja potekal v dveh smereh: v smeri empiričnega proučevanja in implementacije merjenja ter v smeri razvoja novih modelov. Pomen empiričnega raziskovanja je na področju merjenja PO očiten [Bas86, Fen94, Fen97, Fen01]. Raziskave uporabe meritev (recimo [Kul00]) pa kažejo tudi pomen razvoja in uporabe uveljavljenih modelov in standardov. Na to kaže tudi vsebina podpoglavja 2.2.

2.2 Modeli in standardi

Kakovost PO je direktna posledica kakovostnega procesa razvoja PO. Na tem izhodišču je bilo razvitih več metod in principov za zagotavljanje kakovosti PO. V [Piv98] jih avtor poimenuje s skupnim imenom nadzorni mehanizmi ter jih razvršča v štiri skupine:

- standardi kakovosti sistemov (recimo družina standardov ISO-9000 [Kah95, ISO92, ISO94, ISO], standardi ESA [Maz94]),
- standardi za PO,
- modeli za ovrednotenje procesa razvoja PO (model BOOTSTRAP [Haa94, Bootstrap], model stopenj zrelosti – model CMM [Pau93, SEI02b, SEI], model ISO/SPICE [Ama98, Rou95, Kuv95, SPICE], model PROCESSUS [Roz97, Her98] ter drugi) in
- pregledi IS.

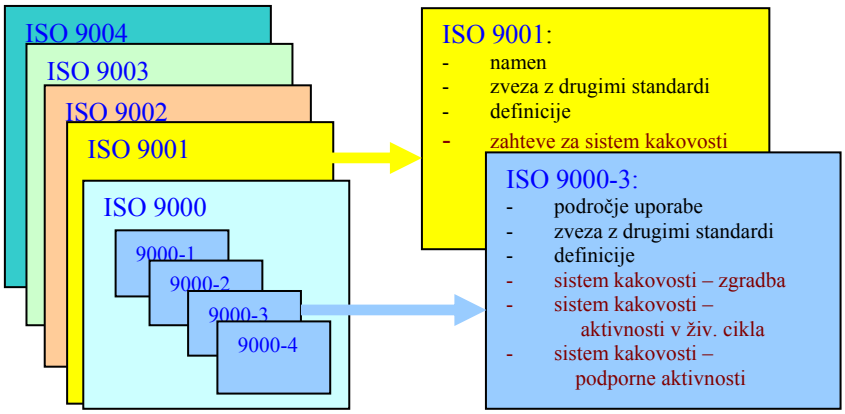
Vsi omenjeni mehanizmi skušajo čim natančneje določiti zakonitosti za zagotovitev kakovosti procesa razvoja PO, pri čemer nekateri temeljijo na dognanjih drugih področij (recimo tehnološkega procesa proizvodnje), drugi pa so že v izhodišču omejeni na posebnosti razvoja PO [Pau95]. V nadaljevanju smo kratko predstavili značilnosti nekaj izbranih pristopov. Med njimi je večina modelov splošnih in skuša z enotnim modelom zajeti problematiko razvoja PO v vsaki organizaciji (recimo standardi ISO 9000, model CMM, model ISO/SPICE). Drugi modeli skušajo naslavljati specifične zahteve organizacije (recimo model SEL [SEL95]).

V okviru disertacije nas posebej zanima model CMM. Model si bomo ogledali podrobneje, enako kakor modela PSP in TSP. Oba modela dopolnjujeta model CMM in predstavljata zanimivo izhodišče za naše nadaljnje delo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

2.2.1 Družina standardov ISO-9000

Družino standardov za vodenje in zagotavljanje kakovosti je razvila mednarodna organizacija za standardizacijo ISO (ang. International Standard Organization) [ISO]. Del družine je standard ISO 9001 [Kah95, ISO92], ki opisuje model zagotavljanja kakovosti v razvoju, proizvodnji, namestitvi in servisiranju. Standard je splošen, zato ožja področja naslavljajo posebni dokumenti. Razvoj PO določa dokument ISO 9000-3 (»Smernice za uporabo ISO 9001 pri razvoju, dobavi in vzdrževanju PO«) [ISO94]. ISO 9001 opisuje zahteve, ki jih mora izpolnjevati kakovostna organizacija pri proizvodnji dobrin. Te zahteve ISO 9000-3 nadgradi z opisom sistema kakovosti, ki natančno opredeli odgovornosti vodstev dobavitelja in naročnika, določi aktivnosti procesa razvoja PO ter ustrezne podporne aktivnosti (slika 2.12). V praksi to pomeni, da mora biti proces razvoja PO ustrezno dokumentiran ter podprt s pregledi izdelkov in presojo samega procesa. Zadnja različica standarda (ISO 9001-2000) pristop ustrezneje definira. Standard ne vsebuje več posebnih dokumentov, vsebinsko pa daje več poudarka merjenju [BSi02].



Slika 2.12: Družina standardov ISO-9000 [ISO92, ISO94]

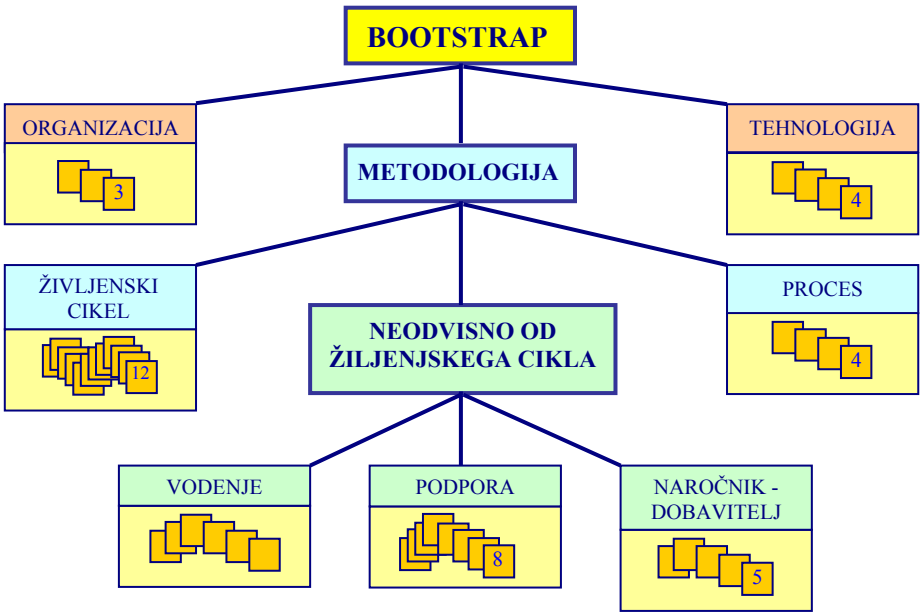
Vir težav pri uporabi standardov je način opisa, ki ga uporabljajo. Opisani so namreč v obliki navadnega besedila, kar onemogoča natančno preverjanje. Nejasnosti so tudi pri razmejitvi med tehnološkimi in procesnimi aktivnostmi, enako niso ustrezno pokrita vsa področja tehnološkega razvoja (recimo analiza zahtev) [Pau94a, Pau95, Mah99]. Čeprav je mersko spremljanje del ISO 9000 standardov, pa to ni določeno tako eksplicitno kot pri modelih CMM in SPICE (predvsem manjka statistično ovrednotenje rezultatov). Opredelitev merjenja je natančnejša od različice ISO 9001:2000 dalje [BSi02].

V razdelku 2.1.2 so bili našteti tudi drugi ISO standardi, ki so povezani z merjenjem PO. Podrobnejša predstavitev teh presega namen disertacije. Bralec si jih lahko ogleda v [BSi02] oziroma na domači strani ISO [ISO]. Standardi so predstavljeni tudi v [BSi, IEEE in TickIT].

2.2.2 Model BOOTSTRAP

BOOTSTRAP [Haa94, Bootstrap] je metodologija, ki je bila razvita na podlagi modela CMM, upošteva pa tudi smernice drugih modelov in standardov (ISO 9001, ISO 9000-3, ISO 12207 ESA). Model je bil razvit v okviru programa ESPRIT Evropske Unije. Cilj je bil razviti natančno in uporabno metodologijo za oceno velikih in majhnih organizacij. BOOTSTRAP ocenjuje stopnjo zrelosti organizacijskega procesa ali izbranega projekta na podoben način kot model CMM, zato so tudi rezultati primerljivi. V ta namen definira referenčni procesni model, ki je podlaga za izvedbo ocenjevanja. Model je visokonivojsko zastavljen, tako da vsak njegov del pokriva več področij. Struktura procesa je razvidna iz slike 2.13.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 2.13: Struktura procesa po metodologiji BOOTSTRAP

Ocena omogoča primerjavo zrelosti z evropskim povprečjem ali natančnejšo primerjavo z ocenjenimi organizacijami v svojem segmentu. Ocenjevalci morajo biti posebej usposobljeni in potrjeni s strani BOOTSTRAP Inštituta. Ocenjevanje se izvaja na 3 načine: kot samoocenitev, ocenjevanje izbranega področja ali popolno ocenjevanje. Slednje traja približno štiri tedne, rezultate pa BOOTSTRAP inštitut hrani tudi v svoji PB [Bootstrap].

2.2.3 Model ISO/SPICE

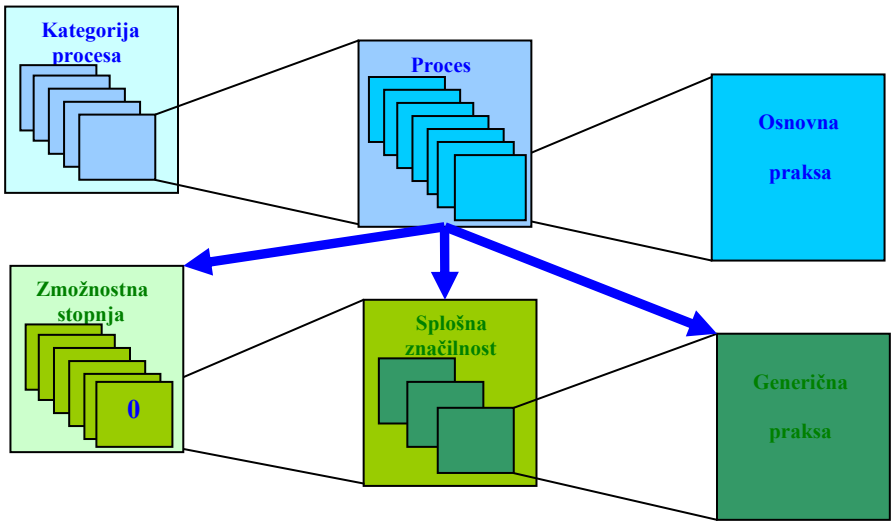
Model SPICE (Software Process Improvement Capability dEtermniation) – ISO/IEC TR 15504 je mednarodni standard za oceno procesa razvoja PO. Namen modela je poenotenje različnih pristopov na tem področju [Ama98, Rou95, Kuv95, SPICE]. V 9 delih opisa SPICE definira referenčni model (slika 2.14), ki določa nabor zaželenih procesov in praks dveh vrst. *Osnovne prakse* so aktivnosti, ki so ključnega pomena za izvajanje 35 procesov. Združene so v pet kategorij:

- odnosi med naročnikom in dobaviteljem;
- tehnologija PO;
- vodenje projektov;
- podpora drugim procesom;
- delovanje na ravni organizacije.

Generične prakse so aktivnosti, ki so prisotne v vseh procesih. Sestavljajo 11 skupnih značilnosti, te pa naprej še šest stopenj zmožnosti. Stopnje zmožnosti obsegajo vrednosti med 0 in 5:

Stopnja 0: osnovna praksa se ne izvaja;
Stopnja 1: osnovna praksa se izvaja, vendar se ne planira in mersko spremlja;
Stopnja 2: osnovna praksa se planira, izvaja in spremlja ter ustreza standardom;
Stopnja 3: osnovna praksa se izvaja v skladu s standardnim, dobro definiranim procesom;
Stopnja 4: rezultati procesa se natančno mersko spremljajo, možno zanesljivo napovedovanje;
Stopnja 5: obstajajo kvantitativni cilji za izboljšanje procesa, uporaba novih tehnologij in idej.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 2.14: Struktura modela ISO SPICE

Pri ocenjevanju procesa za osnovne in generične prakse določimo ustreznost in prisotnost, nakar rezultate zberemo in predstavimo v vektorski obliki za vsako stopnjo posebej. Model SPICE omogoča objektivno ocenjevanje procesa, pri čemer za razliko od modela CMM rezultat predstavlja zmožnost samega procesa razvoja PO in ne zrelosti organizacije.

2.2.4 Model PROCESSUS

Model PROCESSUS [Roz97, Her98] je bil razvit na Univerzi v Mariboru in predstavlja integracijo modela CMM in standarda ISO 9001 oziroma ISO 9000-3. Tako model CMM kot standard ISO 9001 poudarjata pomen dokumentiranega procesa razvoja PO, vendar imata različno strukturo. Model CMM je bolj natančen in določen, standardi ISO pa pokrivajo določena področja, ki jih model CMM ne [Pau94a, Pau95, Mah99].

PROCESSUS razširja prve tri stopnje modela CMM z nekaj novimi ključnimi področji in aktivnostmi (v obstoječa ključna področja), ki so določene na podlagi standardov ISO (slika 2.15). Tako se uporabnost modela poveča, saj ga lahko uporabljamo tudi za načrtovanje, usmerjanje in izvajanje aktivnosti za dvig zrelostne stopnje [Roz97].

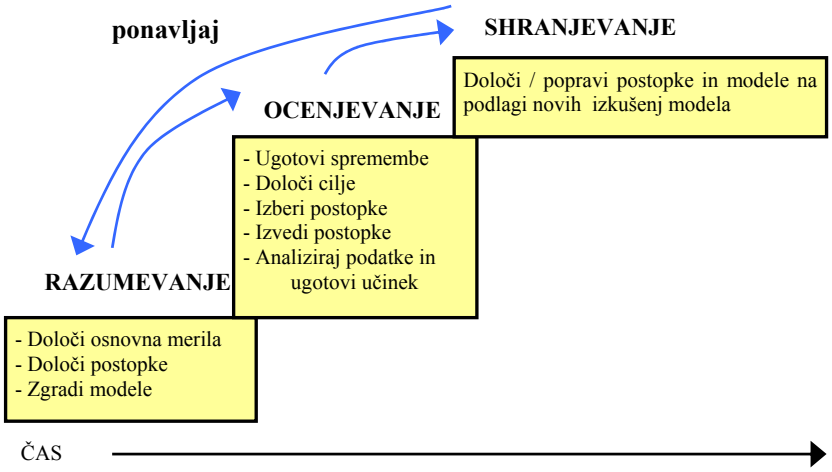
Stopnja	Ključna področja	Ključna področja (angleško)
Stopnja definiranega procesa	Osredotočenje na proces organizacije	Organization Process Focus
	Definicija procesa razvoja PO	Organization Process Definition
	Nadzor dokumentov	**Document Control**
	Program izobraževanja	Training Program
	Integrirano vodenje projektov PO	Integrated Software Management
	Tehnologija izdelkov PO	Software Product Engineering
	Koordinacija različnih skupin	Intergroup Coordination
	Pregledi med kolegi	Peer Reviews
	Upravljanje z vgrajenim proizvodom	**Included Product Management**
Stopnja ponovljivega procesa	**Upravljanje sistema kakovosti**	**Quality System Management**
	Upravljanje organizacijske strukture	**Organizational Structure Management**
	Upravljanje dobave proizvoda	**Product Delivery Management**
	Upravljanje pogodb	**Contract Management**
	Upravljanje z zahtevami	Requirements Management
	Planiranje projektov	Software Project Planning
	Zasledovanje in nadzor nad projekti	Software Project Tracking and Oversight
	Urejanje razmerij s poddobavitelji	Software Subcontract Management
	Zagotavljanje kakovosti PO	Software Quality Assurance
	Upravljanje s konfiguracijo	Software Configuration Management
Začetna stopnja		

Slika 2.15: Ključna področja modela PROCESSUS [Roz97]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

2.2.5 Paradigma za razvoj procesa SEL

Splošni modeli za zagotavljanje kakovosti se pogosto uporabljajo, vendar se je treba zavedati, da imajo organizacije različne cilje. Posamezna področja so lahko zanje različno pomembna, kar je motiv za izgradnjo svojega procesa razvoja PO doseganje specifičnih ciljev. Na tem pomisleku temelji Paradigma za razvoj procesa [SEL95], ki so jo razvili v oddelku z imenom SEL (Software Engineering Laboratory), ki je del NASE [SEL]. Paradigma je sestavljena iz treh pomenskih delov: razumevanja, ocenjevanja in shranjevanja (glej sliko 2.16). Koraki posameznih delov se izvedejo eden za drugim, nakar se ponavljajo in tako omogočajo, da se vse najnovejše spremembe ciljev organizacije odražajo v naslednjih iteracijah.



Slika 2.16: Paradigma za razvoj procesa SEL [SEL95]

Cilji in izkušnje organizacij se spreminjajo, zato je prvi del paradigme **razumevanje** teh. V njem si postavimo merila, določimo procese in določimo modele, ki jih bomo v nadaljevanju uporabljali. Sledi **ocenjevanje**, kjer se izvede večina aktivnosti. V skladu s postavljenimi merili ugotovimo področja, na katerih dejansko stanje odstopa od pričakovanj. Zastavimo si cilje in izvedemo spremembe. Na koncu izmerimo spremembe in ugotovimo dejanski učinek naših akcij. Tretji del je **shranjevanje** pridobljenih izkušenj v obliki pravil, standardov, postopkov in podobnega. Te lahko določimo na novo ali samo popravimo že obstoječe.

Paradigma SEL ne preverja zrelosti procesa ali popolnosti nabora pokritih področij, saj je vse prepuščeno potrebam organizacije. S tega stališča je paradigma neprimerljiva z drugimi opisanimi modeli in jo navajamo samo kot primer drugačnega pristopa.

2.2.6 Model stopenj zrelosti (model CMM)

Model stopenj zrelosti (ang. Capability Maturity Model - CMM) je nastal na Software Engineering Institute (v nadaljevanju SEI) pri univerzi Carnegie Mellon v ZDA. Razvoj se je začel leta 1986 na pobudo ameriškega obrambnega ministrstva. Cilj je bil razviti metodo za ocenjevanje sposobnosti razvijalcev PO, ki sodelujejo z vladnimi organizacijami. Leta 1988 je bil zasnovan model za zagotavljanje kakovostnega razvoja PO [Hum89], ki je z verzijama 1.0 iz leta 1991 in 1.1 [Pau93, Pau93a] iz leta 1993 doživel široko uporabo. Na podlagi teh je nastalo veliko dokumentov, ki podrobno opisujejo posamezne dele modela. Model so dograjevali še naprej do leta 1998, vendar do nove dokončane verzije ni prišlo – nazadnje so razvili osnutek verzije 2.0 (SW-CMM v2 draft C)[CMM, SEI].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Sledil je razvoj modelov še za druga sorodna področja uporabe (recimo SE-CMM za sistemski inženiring, SA-CMM za nakup PO ipd.) in združitev vseh modelov v t.i. sestavljeni CMM (ang. CMM Integrated ali CMMI) [CMMI, SEI]. Ta je predstavljen na dva načina:

- **stopenjsko** (ang. staged) [SEI02b]: Osnovna izvedba modela CMM, ki definira pet stopenj zrelosti procesa organizacije za razvoj PO. Na vsaki stopnji določi ključna procesna področja in cilje, ki jih mora organizacija doseči.
- **zaporedno** (ang. continous) [SEI02a]: Izvedba, ki se zgleduje po modelu SPICE. Definira zrelost posameznih ključnih procesnih področij, ki se določa glede na dosežene cilje oziroma uporabljene tipične prakse. Pred modelom CMMI se ta način opisa ni uporabljal.

V zadnjih letih se model CMMI z dodajanjem novih področij spreminja še naprej. Novejša verzija (CMMI v1.1) je izšla marca 2002 in pokriva štiri področja uporabe [SEI02a, SEI02b]. Avgusta 2002 so posebej predstavili tudi del verzije CMMI v1.1, ki je namenjen razvoju PO [SEI02c, SEI02d]. Ne glede na novosti modela CMMI pa ostaja del modela, ki je namenjen razvoju PO, že od leta 1993 pretežno nespremenjen, kar dokazuje tudi enostavna preslikava modela CMM v CMMI [SEI00]. Tudi praksa potrjuje, da je osnovni model SW-CMM zelo razširjen in se še uporablja [Pau01].

V našem primeru želimo uporabiti model v primerni, a čim enostavnejši obliki. Zato bomo uporabili SW-CMM iz leta 1993, čeprav bomo na določenih mestih omenjali tudi novejši model CMMI. V naši rešitvi bomo recimo upoštevali posebno ključno področje modela CMMI, ki je namenjeno merjenju in analizi. Zaradi enostavnosti namesto popolnega naziva »model SW-CMM v1.1« uporabljamo krajši naziv »model CMM«.

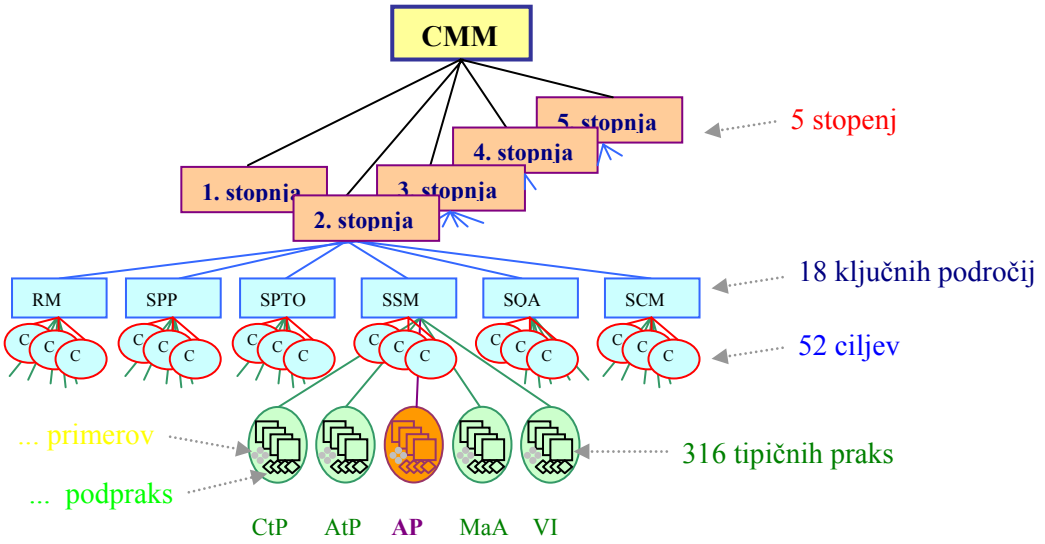
2.2.6.1 Predstavitev modela CMM

Ena izmed načel pristopa Total Quality Management (TQM) pravi, da se je treba osredotočiti na kakovost, da bi lahko dosegli krajši razvojni cikel, večjo produktivnost, večje zadovoljstvo strank in boljši poslovni uspeh [Pau99]. Model CMM je dejansko uspešna implementacija tega načela na področju razvoja PO. Temelji na ideji, da za izboljšanje učinkovitosti razvoja PO ni dovolj uporaba novih tehnologij in orodij, temveč je treba poskrbeti predvsem za kakovosten, dobro definiran in nadziran proces razvoja PO.

SEI je izvedel obsežno raziskavo učinkovitih pristopov v uspešnih organizacijah za razvoj PO v ZDA in natančno določil, kaj to pomeni [Hum89]. Nastal je hierarhičen model, v okviru katerega je definiranih 5 stopenj zrelosti (ang. maturity levels). Za vsako stopnjo so določena ključna področja (ang. key process area ali KPA), cilji (ang. goal), tipične prakse (ang. key practices), podprakse (ang. subpractices) in primeri (ang. examples). V okviru teh elementov so definirane še druge sestavine modela, kot so organizacijska struktura, skupine, vloge, aktivnosti, dokumenti in podobno [Pau93, Pau93a].

Vsaka stopnja določa ustrezno definiran proces razvoja PO in predstavlja določeno stopnjo sposobnosti organizacije za razvoj PO, kar poenostavljeno imenujemo tudi zrelost (ali zmožnost) organizacije. Ključna področja predstavljajo množico medsebojno povezanih aktivnosti, ki jih izvajamo na področju, ki je pomembno za razvoj PO (recimo izobraževanje razvijalcev). Cilje dosežemo, če na primeren način izvajamo aktivnosti ključnih področij. Praktično je to mogoče (ne pa nujno) doseči z uporabo tipičnih praks, ki so v določenih primerih podrobneje opredeljene s podpraksami oziroma primeri. Slika 2.17 predstavlja razmerja med posameznimi deli strukture v modelu CMM.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 2.17: Struktura modela stopenj zrelosti (SW-CMM v1.1)

Organizacija doseže določeno stopnjo zrelosti, ko izpolni cilje vseh ključnih področij te in nižjih stopenj zrelosti. Model je deskriptiven (in ne preskriptiven), saj ključna področja in cilji določajo KAJ je treba v okviru vsake stopnje doseči, ne pa KAKO. Tipične prakse, podprakse in primeri, ki pojasnjujejo KAKO izvajati aktivnosti, so namreč samo običajni pristopi, ki niso obvezni. Organizacija lahko zadosti ciljem tudi na drugačen način.

Pomembna značilnost modela CMM je zahteva po vgraditvi aktivnosti z vsemi potrebnimi podpornimi aktivnostmi v proces razvoja PO. S tega stališča lahko aktivnosti oziroma tipične prakse razdelimo na pet skupin aktivnosti:

- zavezanost za izvajanje (ang. Commitment to Perform - CtP);
- sposobnost za izvajanje (ang. Ability to Perform - AtP);
- izvajane aktivnosti (ang. Activities Performed - AP);
- merjenje in analiza (ang. Measurment and Analysis - MaA);
- preverjanje izvedbe (ang. Verifying Implementation - VI).

Na vsakem ključnem področju so definirane tipične prakse, ki pripadajo vsaki od skupin aktivnosti. Najpomembnejše prakse so iz skupine »Izvedene aktivnosti«, ostale pa poskrbijo za učinkovito, ponovljivo in trajno izvedbo. Tu lahko opazimo podobnost z modelom SPICE: skupina »Izvajane aktivnosti« bi z vidika modela SPICE predstavljala osnovno prakso, medtem ko bi ostale skupine sodile med generične prakse. Model CMMI delitev tipičnih praks po zgledu modela SPICE že povzema pri zaporedni predstavitvi modela CMM.

2.2.6.2 Vsebina modela CMM

Ključna področja in zrelostne stopnje so razvidne iz tabele 2.8. Ponovljiv proces je urejen predvsem s projektnega vidika. Definiran proces uvaja dobre tehnološke pristope na ravni celotne organizacije, kar je podlaga za proces na 4. stopnji, ki se ukvarja s kakovostjo procesa in izdelkov. Proces na 5. stopnji zmore obvladati spremembe in se nenehno izboljševati.

Načelo modela CMM dejansko temelji na sledljivosti procesa razvoja PO: zrelejša kot je organizacija, večja je sledljivost njenega procesa. V nadaljevanju so s tega stališča kratko predstavljene značilnosti posameznih zrelostnih stopenj (slike od 2.18 do 2.22). Podrobnejši opis ostalih elementov modela se nahaja v [Pau93, Pau93a].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Stopnja	Ključna področja	Ključna področja (angleško)	
5. stopnja Optimizirajoč proces	Preprečevanje napak	Defect Prevention	DP
	Uvajanje tehnoloških sprememb	Technology Change Management	TCM
	Uvajanje sprememb v procesu	Process Change Management	PCM
4. stopnja Upravljan proces	Kvantitativno vodenje procesov	Quantitative Process Management	QPM
	Upravljanje za kakovost PO	Software Quality Management	SQM
3. stopnja Definiran proces	Osredotočenje na proces organizacije	Organization Process Focus	OPF
	Definicija procesa razvoja PO	Organization Process Definition	OPD
	Program izobraževanja	Training Program	TP
	Integrirano vodenje projektov PO	Integrated Software Management	IPM
	Tehnologija izdelkov PO	Software Product Engineering	SPE
	Koordinacija različnih skupin	Intergroup Coordination	IC
	Pregledi med kolegi	Peer Reviews	PR
2. stopnja Ponovljiv proces	Upravljanje z zahtevami	Requirements Management	RM
	Planiranje projektov	Software Project Planning	SPP
	Zasledovanje in nadzor nad projekti	Software Project Tracking & Oversight	SPTO
	Urejanje razmerij s poddobavitelji	Software Subcontract Management	SSM
	Zagotavljanje kakovosti PO	Software Quality Assurance	SQA
	Upravljanje s konfiguracijo	Software Configuration Management	SCM
1. stopnja Kaotičen proces			

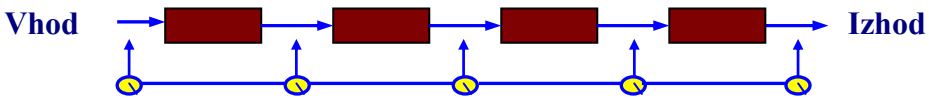
Tabela 2.8: Ključna področja po zrelostnih stopnjah (SW-CMM v1.1)

1. Kaotičen proces: Ta stopnja ustreza organizaciji, v kateri ne uporabljajo (skoraj) nobenega pristopa za obvladovanje razvojnega procesa. Zato je znan le vhod in izhod, proces razvoja PO pa je naključen, včasih celo kaotičen. Definiranih je malo procesov in uspešen rezultat dela je odvisen zgolj od sposobnih posameznikov.



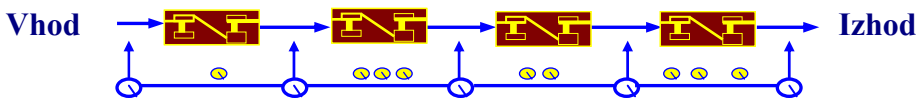
Slika 2.18: Proces razvoja PO na 1. stopnji modela CMM [Pau93a]

2. Ponovljiv proces: Organizacija na drugi stopnji ima vzpostavljene osnovne procese vodenja projektov, ki omogočajo spremljanje stroškov, trajanja razvoja in učinkovitosti dela. Proces je definiran do take mere, da omogoča ponovitev enakega rezultata razvoja, kot je bil v preteklosti na podobnem projektu že dosežen.



Slika 2.19: Proces razvoja PO na 2. stopnji modela CMM [Pau93a]

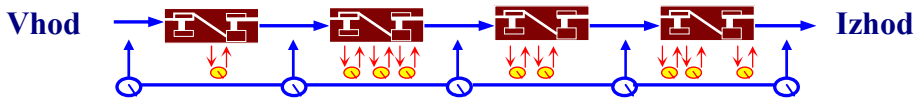
3. Definiran proces: Organizacija poleg upravljalskih obvladuje tudi vse tehnične aktivnosti, ki so dokumentirane in standardizirane v enoten proces razvoja PO organizacije. Projekti uporabljajo prilagojen standardni proces razvoja PO, ki je potrjen s strani organizacije.



Slika 2.20: Proces razvoja PO na 3. stopnji modela CMM [Pau93a]

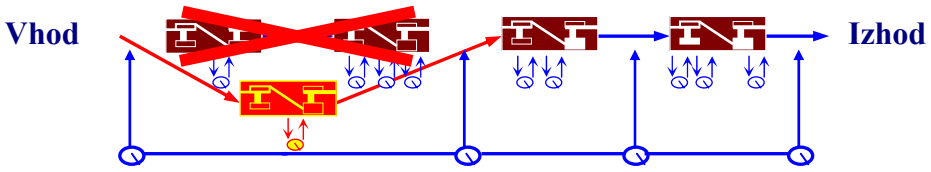
4. Upravljan proces: Četrta stopnja predstavlja organizacijo, ki svoj proces razvoja PO in kakovost izdelkov spremlja z natančnimi meritvami. Organizacija na ta način kvantitativno razume in nadzira proces razvoja PO in kakovost izdelkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 2.21: Proces razvoja PO na 4. stopnji modela CMM [Pau93a]

5. Optimizirajoč proces: Organizacija na tej stopnji zmore svoj proces razvoja PO nenehno izboljševati z uvajanjem inovativnih idej in tehnologij ter spremljanjem njihovega vpliva na proces. Organizacija je zmožna ovrednotiti vpliv sprememb.



Slika 2.22: Proces razvoja PO na 5. stopnji modela CMM [Pau93a]

Posledica domišljenega modela je natančna in dokaj obsežna definicija. SW-CMM verzije 1.1 iz leta 1993 tako sestavlja 5 stopenj, na katerih je opisanih 18 KPA, 52 ciljev in kar 316 tipičnih praks. Osnovna dokumentacija [Pau93, Pau93a] obsega približno 500 strani. Model CMMI je še obsežneje opisan, saj tako stopenjski kot zaporedni način predstavitve področja razvoja PO obsegata približno 650 strani dokumentacije [SEI02c, SEI02d].

2.2.6.3 Proces razvoja PO z vidika modela CMM

Z vidika modela CMM je pomemben standarden proces razvoja PO na ravni organizacije. Ta predstavlja operativno definicijo postopka, ki se uporablja v organizaciji in je predloga za določanje vseh projektnih procesov razvoja PO v organizaciji. Opisuje ključne procesne elemente, ki jih mora vsebovati vsak proces, in njihova medsebojna razmerja. Elementi predstavljajo natančno določeno in zaokroženo skupino povezanih aktivnosti, vire, njihova razmerja in odvisnosti. Pri tem velja posebej izpostaviti množico skupin in vlog, ki jih model CMM določa tako na ravni organizacije kot projekta [Pau93, SEI02c, SEI02d].

Vrstni red izvajanja aktivnosti je razviden iz arhitekture procesa. V skladu z zahtevami modela CMM je treba proces razvoja PO določiti, vzdrževati, preverjati in izboljševati s formaliziranimi pristopi na ravni organizacije (dokumentiranje, preverjanje izvajanja, ipd.). Ker določa predpisan način izvajanja vseh aktivnosti v organizaciji, je ključnega pomena za dolgoročno stabilnost in napredovanje organizacije [Pau93].

Na projektni ravni je poudarek na uporabnosti procesa. Proces v okviru projekta je določen iz organizacijskega procesa in predstavljen s standardi, postopki, orodji in metodami. Da bi proces nedvoumno tolmačili, so podana tudi navodila in kriteriji za prilagajanje. Model CMM dopušča v okviru procesa uporabiti poljubne življenjske cikle, ki jih je organizacija pred tem preverila in potrdila.

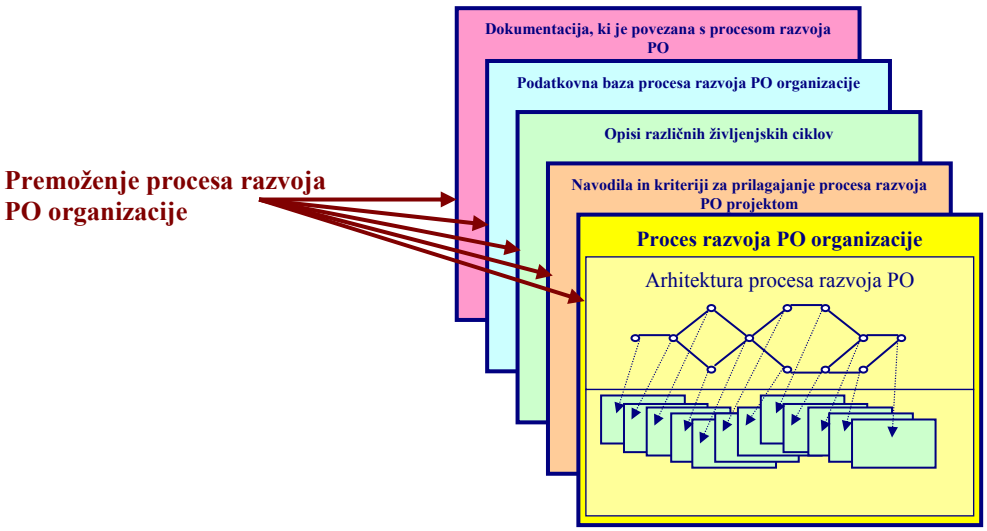
Poleg opisa procesa razvoja PO v organizaciji hranimo z njim povezane dele, ki jih skupaj imenujemo *premoženje procesa razvoja PO organizacije* (slika 2.23).

2.2.6.4 Uporaba modela CMM

Model CMM se uporablja za ocenjevanje organizacij za razvoj PO na dva načina [Mah99]:

- z namenom izboljšanja procesa razvoja (s strani njih samih) ter
- z namenom ugotovitve zmožnosti za izvedbo naročila (s strani naročnikov).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

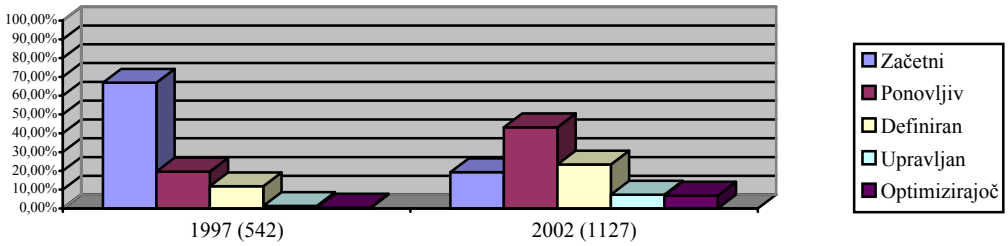


Slika 2.23: Premoženje procesa razvoja PO organizacije [Pau93]

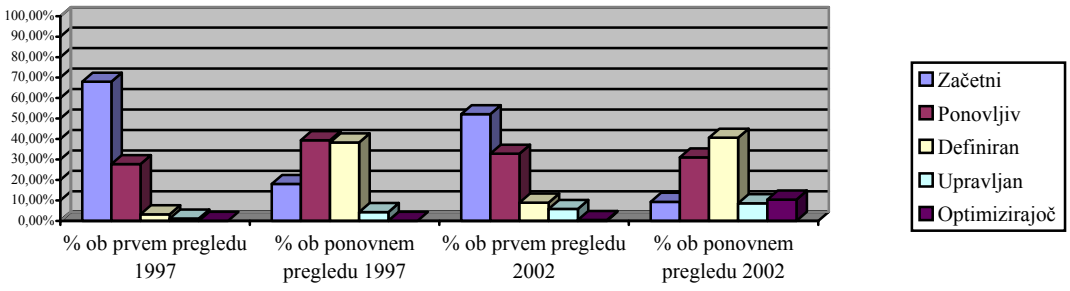
Za oba načina obstajajo natanko definirani postopki (vprašalniki). Izvajajo jih ocenjevalci, ki so preverjeni s strani SEI. Tak pristop zagotavlja objektivnost in primerljivost rezultatov. Ocenjevalci preverijo letno več kot 100 organizacij, pri čemer se povečuje delež ponovnih ocenitev. Predlagan čas med eno in drugo ocenitvijo je od 18 do 30 mesecev, pri čemer krajšanje tega časa običajno ne pripomore k uspešnosti ocenitve [Mah99, SEMA]. Ko se organizacije lotijo aktivno izboljševati proces v skladu z zahtevami modela CMM, potrebujejo v povprečju za napredovanje ene stopnje med 17 in 28 meseci. Najtežji je prehod med tretjo in četrto stopnjo [SEI02].

2.2.6.5 Izkušnje z modelom CMM

Slike 2.24, 2.25 in 2.26 kažejo rezultate ocenitev v računalniški industriji, ki so bili zbrani na SEI do leta 1997 [SEI97] in do avgusta letošnjega leta [SEI02]. Prikazani so deleži ocenjenih organizacij po stopnjah modela CMM, razmerja med prvo in ponovno oceno ter časi napredovanja po zrelostni lestvici.

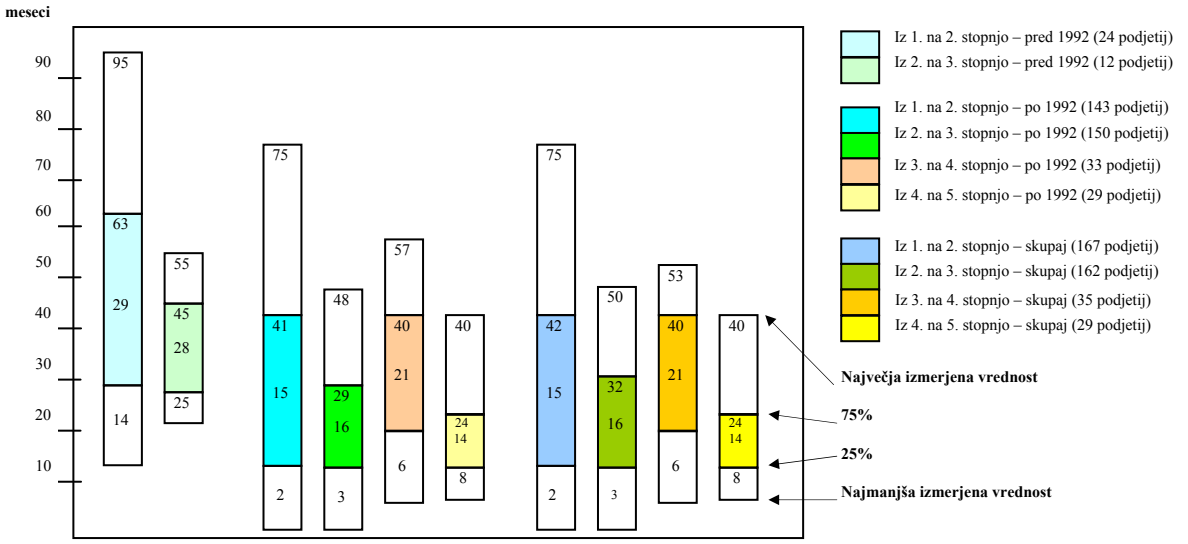


Slika 2.24: Deleži ocenjenih organizacij po stopnjah CMM [SEI97, SEI02]



Slika 2.25: Deleži organizacij po stopnjah CMM in pregledih [SEI97, SEI02]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 2.26: Čas napredovanja podjetij na višjo stopnjo [SEI02]

Poleg dokumentov SEI [SEI] je v literaturi mogoče zaslediti tudi številne opise izkušenj pri uporabi modela CMM, recimo za organizacije Raytheon [Dio93], Boeing [Wig97], Motorola [Das94], Hewlett-Packard [Low96], vojaško letalsko bazo Hughes [Hum91] in druge. Pogosti so opisi uporabe modela CMM kot vodila za izboljšanje kakovosti procesa razvoja PO (ang. Software Proces Improvement ali SPI) (recimo [Ray99, Her94, Dio93, Wig97]).

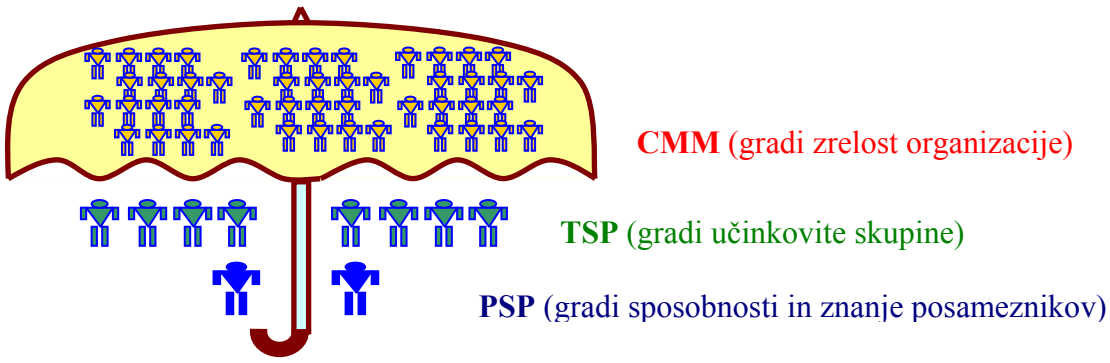
Na najvišjih stopnjah je majhen odstotek organizacij, čeprav se stanje izboljšuje. Tipično gre za velike, tehnološko zelo razvite organizacije, ki za to namenijo dovolj virov in temeljito implementirajo vse zahteve CMM [SEI97, SEI02, Fen97, Oma97]. Veliko je zapisanega tudi o uporabi modela CMM v specifičnih okoliščinah (recimo [Pau01]) in še zlasti v majhnih organizacijah (glej razdelek 2.3.2). Pri teh je površno poznavanje modela CMM pogosto [Bro94, Sch00]. Tipični spodrsljaji pri razumevanju modela CMM so opisani tudi v [Wie96].

2.2.6.6 Sorodni modeli

Model CMM primerno definira nadziran proces razvoja PO, zato pogosto služi kot vodilo za izboljšanje kakovosti procesa organizacij za razvoj PO. Vendar je model CMM deskriptiven model in čeprav je natančno določen, uporabniki modela potrebujejo bolj specifična navodila, kako aktivnosti učinkovito implementirati v določenih okoliščinah, recimo v majhnih organizacijah z omejenimi viri [Pau99, Hum99]. Težav z uporabo modela CMM se je zavedel tudi SEI in predstavil dva preskriptivna modela, ki naj bi to vrzel zapolnila [Hum98].

Najprej je bil predstavljen osebni proces razvoja PO (ang. Personal Software Process – PSP), ki je definiral model procesa za posamezne inženirje v skladu z zahtevami modela CMM. Na podlagi tega je bil razvit še skupinski proces razvoja PO (ang. Team Software Process - TSP), ki določa proces za učinkovito delovanje skupine razvijalcev PO. Oba modela sta predvsem dopolnitev modela CMM, ki naj bi pripomogla k usposabljanju razvijalcev in skupin za učinkovitejše delo na ravni organizacije [Hum98, Hum98a, Hum98b]. Ne glede na to pa lahko modela uporabljamo tudi samostojno. Slika 2.27 predstavlja uporabnost vseh treh modelov v organizaciji za razvoj PO.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

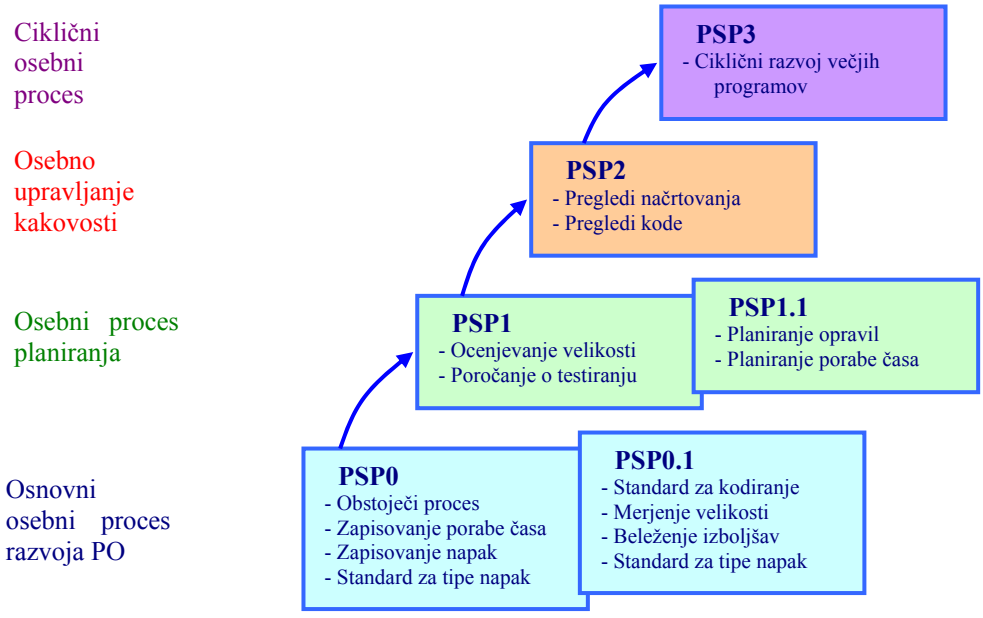


Slika 2.27: Razmerje med modeli CMM, TSP in PSP [Hum98]

2.2.7 Osebni proces razvoja PO (model PSP)

Osebni proces razvoja PO [Hum95, Hum97, Hum00a, PSP] je preskriptivni model procesa za inženirje v organizaciji za razvoj PO, ki temelji na načelih modela CMM. Sestavlja ga množica metod in praks za načrtovanje, merjenje in vodenje poteka dela inženirja pri razvoju PO. Uporaba osebnega procesa ni omejena na opis opravil v okviru razvoja PO, temveč ga lahko učinkovito uporabimo tudi samostojno pri razvoju osebnih projektov.
Splošni cilj uporabe modela PSP je pravočasen in poceni razvoj PO brez napak, ki temelji na načelih učinkovitega, nadzorovanega in odgovornega dela inženirja [Hum97a]. V model so tako sestavili tiste metode in prakse modela CMM, ki jih lahko uporablja posameznik pri svojem delu. V skladu s tem inženir svoj proces postopoma dopolnjuje, pri čemer začne z osnovnimi postopki in konča s tistimi, ki mu omogočajo popolno obvladovanje svojega dela.

Slika 2.28 prikazuje sedem verzij modela PSP, ki predstavljajo osebne procese razvoja PO od najbolj enostavnega (PSP0) do najbolj popolnega (PSP3). Vsaka verzija obsega več področij, pri čemer višja verzija vedno vsebuje vsa področja nižje verzije. Inženir si bo na začetku za cilj izbral proces PSP0. Ko bo operativno obvladal vsa področja te verzije, bo njegov cilj PSP0.1, ki ga bo dosegel z izpolnitvijo dodatnih zahtev. Po zgledu modela CMM bi lahko verzije razvrstili v 4 stopnje, ki so tudi razvidne iz slike 2.28.



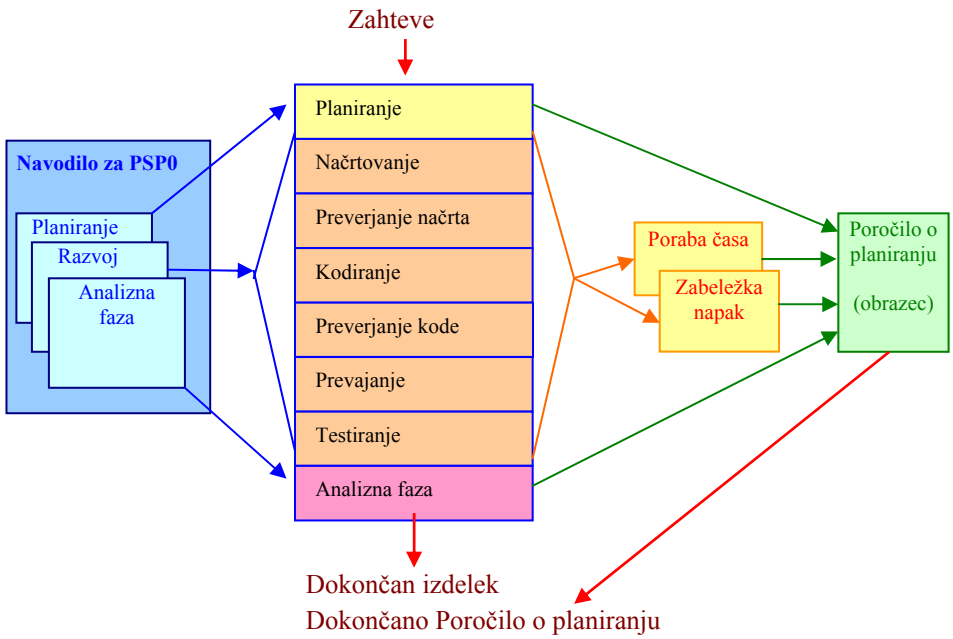
Slika 2.28: Sedem verzij modela PSP [Hum95]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

1. stopnja
Obsega verziji PSP0 in PSP0.1. PSP0 vzpostavlja nadziran osebni proces z osnovnimi merjenji in beleženjem rezultatov. PSP0.1 dopolnjuje PSP0 z uvajanjem standardne izvedbe kodiranja, merjenjem velikosti in uporabo obrazca za beleženje možnosti za izboljšavo.
2. stopnja
Govori o planiranju. PSP1 tako uvaja poročilo o testiranju ter planiranje velikosti in porabe virov, PSP1.1 pa planiranje opravil in porabe časa. Veliko je govora o metodah za napovedovanje (recimo PROBE [Hum95]).
3. stopnja
Uvaja upravljanje s kakovostjo. PSP2 uvaja tehnike za pregledovanje planiranja in kode, ki omogočajo učinkovito odkrivanje napak in ocenjevanje kakovosti izdelkov inženirja. PSP2.1 dodaja tehnike za preverjanje postopkov planiranja.
4. stopnja
Obsega verzijo PSP3, ki dodaja tehnike za obvladovanje razvoja večjih programov. Razvoj poteka s cikličnim ponavljanjem področij verzije PSP2 na delih programov.

Vsaka verzija modela PSP je definirana z množico dokumentov, ki pripadajo štirim tipom:

a) **Navodilo** (ang. script) je prikaz vseh korakov, ki sestavljajo proces (recimo PSP0) ali del procesa (recimo planiranje v PSP1). Izvajanje korakov v podanem zaporedju pove, kako proces poteka. Koraki so podani kot kratek opis aktivnosti (v enem stavku). Namen navodila, vhodni in izhodni kriteriji so pri vsakem navodilu posebej navedeni.
b) **Obrazec** (ang. form) je vnaprej določen vzorec, ki pomaga pri zbiranju in zapisovanju rezultatov v natanko določeni obliki. Ker natanko določa število in vrsto podatkov, nam služi kot predloga in opomnik, kaj je treba opraviti.
c) **Zabeležka** (ang. log) je obrazcem podobna predloga, le da jo uporabimo samo po potrebi. Običajno ima manj podatkov in služi za sprotno shranjevanje rezultatov.
d) **Standard** (ang. standard) je opis pravilne izvedbe določene aktivnosti (recimo štetja vrstic kode pri merjenju velikosti) ali izgleda določenega dokumenta. Vsebuje jasne opise elementov aktivnosti: opis, tipične vrednosti, primere, izjeme in drugo.

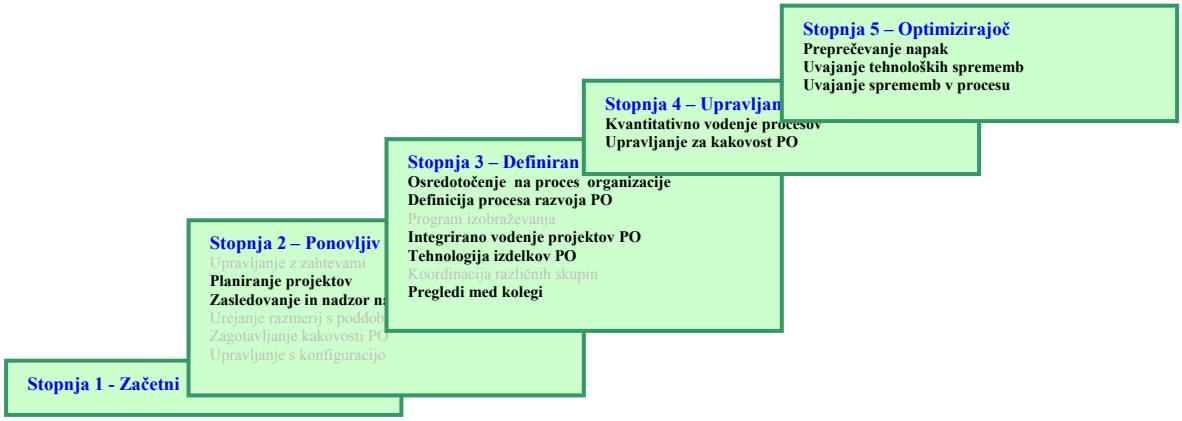


Slika 2.29: Struktura procesa PSP0 [Hum95]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Slika 2.29 prikazuje, kako se dejansko izvaja osebni proces v skladu z modelom PSP0. Inženirjev prvi korak pri razvoju PO je planiranje (ang. planning) na podlagi definiranih zahtev (ang. requirements). To se opravi po navodilu za izvajanje PSP0 (ang. PSP0 script) oziroma navodilu za izvajanje planiranja (ang. planning script). Rezultate planiranja (predvideno velikost končnega izdelka, porabo časa, nabor potrebnih opravil) se shrani v obrazec »Poročilo o planiranju« (ang. plan summary). Nato inženir razvija izdelek v skladu s fazami izbranega življenjskega cikla, ki ga opisujejo navodila za razvoj (ang. development script). V PSP0 ta vsebuje faze načrtovanja (ang. design), preverjanja načrtovanja (ang. design review), kodiranja (ang. coding), preverjanja kode (ang. coding review), prevajanja (ang. compile) in testiranja (ang. test). Med razvojem inženir beleži podatke o porabljenem času in odkritih napakah v zabeležke porabljenega časa (ang. time recording log) in napak (ang. defect recording log). Ko je razvoj končan, se v skladu z navodilom izvede analizna faza (ang. postmortem script), v kateri se izmeri končno velikost izdelka, ki se jo skupaj z izvlečkom podatkov o napakah in porabljenem času zapiše v »Poročilo o planiranju«. To poročilo in dokončani izdelek sta končna rezultata.

Osebni proces višjih verzij bi lahko predstavili z dopolnitvijo slike 2.29 z novimi dokumenti in razmerji. Humphrey v [Hum95] PSP0 definira z 11-imi dokumenti (navodili, obrazci, zabeležkami in standardi), PSP1 z 19-imi, PSP2 z 25-imi in PSP3 s kar 39 dokumenti. Količina dokumentov ustrezno odraža zahtevnost posameznih verzij modela PSP.



Slika 2.30: Ključna področja modela CMM, ki jih pokriva model PSP [Hum95]

S stališča naše raziskave je pomembno, v kakšni meri lahko model PSP nadomesti model CMM. Z vidika posameznega razvijalca osebni proces ustreza 5. stopnji zrelosti, vendar pokriva le tista ključna področja, ki so za posameznika smiselna. Na sliki 2.30 so krepko izpisana ključna področja modela CMM, ki jih osebni proces (vsaj deloma) pokriva. Zajeta so vsa ključna področja 4. in 5. stopnje, manjkajo pa nekatera področja iz 2. in 3. stopnje, ki so pomembna za vodenje projektov in uvajanje postopkov na ravni organizacije. Ta za osebni proces niso smiselna.

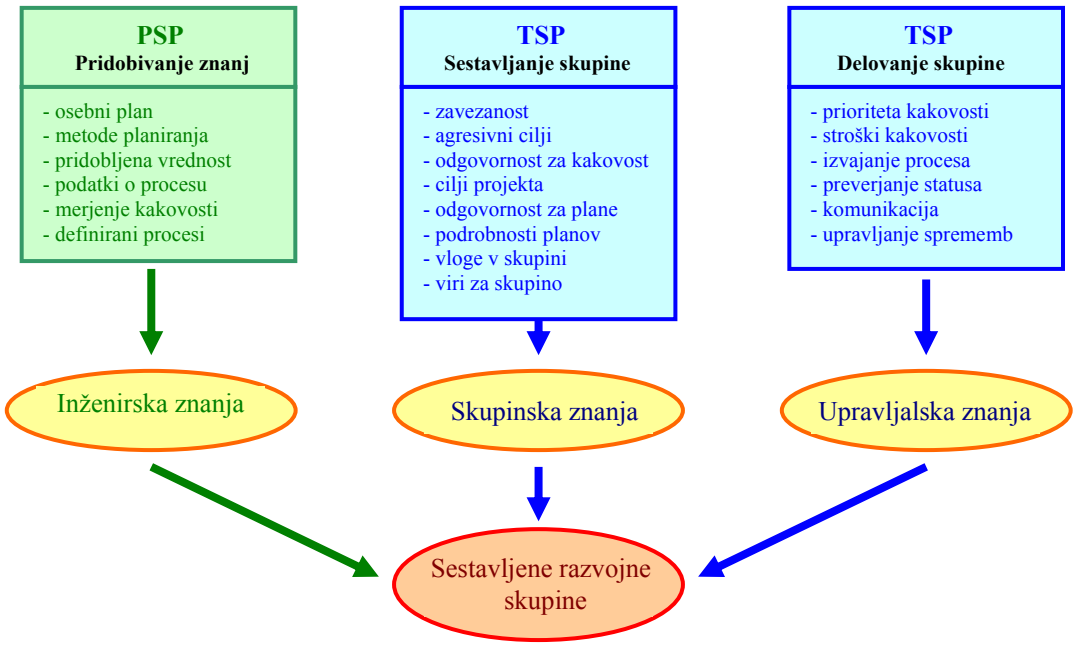
2.2.8 Skupinski proces razvoja PO (model TSP)

Skupinski proces razvoja [Hum00, Hum00b, TSP] je nadgradnja modela PSP, ki opisuje proces za učinkovito delovanje skupine razvijalcev PO v organizaciji. Začetek razvoja sega v leto 1996, do danes pa je bilo razvitih več verzij. Zadnje verzije so bile preizkušene v praksi [McA00, Web99] in optimizirane. Model TSP definira pristope in metode, ki so primerne za skupine z manj kot 20 člani (optimalno število članov je med 5 in 12), vendar je v razvoju že model mTSP, ki bo primeren za skupine do 150 razvijalcev. Za potrebe učenja je bil razvit poenostavljen model iTSP (ang. Introduction to TSP) [Hum00].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Skupinski proces razvoja PO je namenjen *učinkovitim skupinam* (ang. effective groups) [Hum00]. Učinkovita skupina je množica razvijalcev s smiselnim, jasnim in dosegljivim skupnim ciljem. Vsebuje vsaj dva člana in vsak član ima natanko določeno vlogo. Člani so motivirani in usposobljeni za doseganje cilja, sodelovanje in podpora pa sta ključna dejavnika uspeha. Skupina ima zagotovljene vse potrebne vire za svoje delovanje.

Predpogoj za uporabo modela TSP je, da posamezni člani uporabljajo model PSP. Tam je osebni proces natančno določen, model TSP pa ta znanja nadgradi z opisom sestavljanja in delovanja učinkovite skupine (slika 2.31).



Slika 2.31: Struktura modela TSP [Hum00]

Sestavljanje skupine (ang. launch) je dejansko določanje konkretnega procesa razvoja PO, ki temelji na ciljih skupine. Predvidoma traja 4 delovne dneve in obsega 9 sestankov, na katerih skupina izvede naslednje aktivnosti:

- vodstvo organizacije določi poslovne cilje skupine;
- skupina določi vloge in cilje delovanja posameznim članom;
- skupina določi strategijo razvoja;
- skupina planira opravila po fazah (top-down);
- skupina razvije plan zagotavljanja kakovosti;
- skupina preveri plane članov in jih uravnoteži (bottom-up);
- skupina izvede oceno tveganj;
- skupina pripravi poročila za vodstvo organizacije in poroča;
- vodstvo organizacije preveri sestavo skupine.

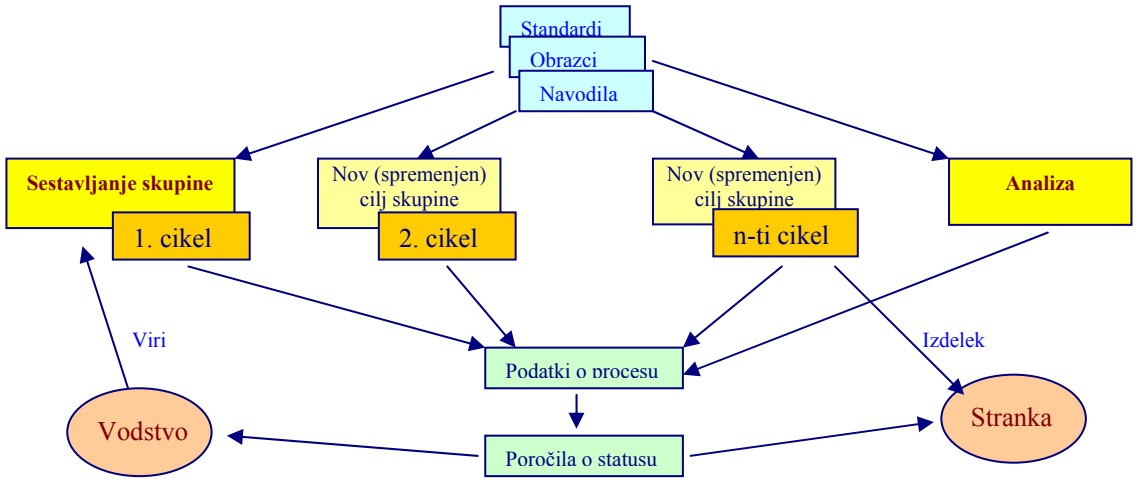
Ko je skupina sestavljena, začne delovati v skladu z izbranim procesom. Drugi del modela TSP predstavlja metode in postopke za učinkovito delo skupine in določa naslednja področja:

- odgovorno vodenje skupine;
- dosledno izvajanje procesa skupine;
- spremljanje izhodišč;
- komunikacijo v skupini;
- poročanje vodstvu organizacije;

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- planiranje;
- ocenjevanje poteka dela;
- prerazporeditev dela med člani skupine;
- pripravo skupine za nove cilje;
- upravljanje kakovosti.

Skupina običajno deluje ciklično, pri čemer vsak cikel zajema obvladljiv del predvidenih aktivnosti razvoja[16] (slika 2.32). Pri tem lahko cikel zajema določen del življenjskega cikla ali časovno obdobje (recimo 3 mesece). Na začetku vsakega cikla je standardizirana aktivnost (ponovne) priprave skupine (ang. re-launch), v kateri skupina ponovno preveri cilje, jih po potrebi prilagodi ter podrobno planira izvedbo cikla. Na koncu vseh ciklov je analiza (ang. postmortem), v kateri skupina analizira opravljeno delo in poroča vodstvu organizacije.



Slika 2.32: Delovanje metode TSP [Hum00]

Ko skupina doseže prvotno določene cilje in izvede analizo, se njena življenjska doba konča ali pa tudi ne. Vodstvo lahko skupini določi nove cilje in ta izvede pripravo na nove cilje. V tem primeru ni potrebno izvesti vseh korakov sestavljanja skupine, temveč le preveriti, če lahko proces skupine bolje prilagodimo novi nalogi. Spremembe nastanejo tudi zaradi izkušenj, ki so bile pridobljene z merjenjem različnih vidikov predhodnega razvoja PO.

Posebno pozornost model TSP posveča zagotavljanju kakovosti. Podobno kot pri modelu PSP je osnovno načelo preprečevanje napak oziroma odpravljanje le-teh čim bolj zgodaj v razvoju. V skladu s tem je v okviru modela TSP določen plan zagotavljanja kakovosti, ki ga skupina sprejme ob sestavljanju. Z vidika zagotavljanja kakovosti so pomembni ponovni pregledi in merjenja, ki omogočajo spremljanje in napovedovanje kakovosti.

Formalno je skupinski proces definiran na enak način kot osebni proces: z množico navodil, obrazcev, zabeležk in standardov. Skupno število dokumentov pri zadnji verziji modela TSP presega 70, kar ustrezno odraža zahtevnost modela [Web99].

Pomembna značilnost modela TSP je opredelitev vlog in njihovih nalog v skupini. V različici modela iTSP [Hum00] Humphrey navaja pet vlog, ki bi jih morala imeti vsaka skupine. To so: vodja skupine (ang. team leader), vodja razvoja (ang. development manager), vodja

[16] *Cikličnost* razumemo v smislu ponavljanja določenih aktivnosti ob začetku vsakega cikla. Vsak cikel se začne s (ponovno) pripravo skupine,. tako da razvoj poteka od priprave do priprave skupine.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

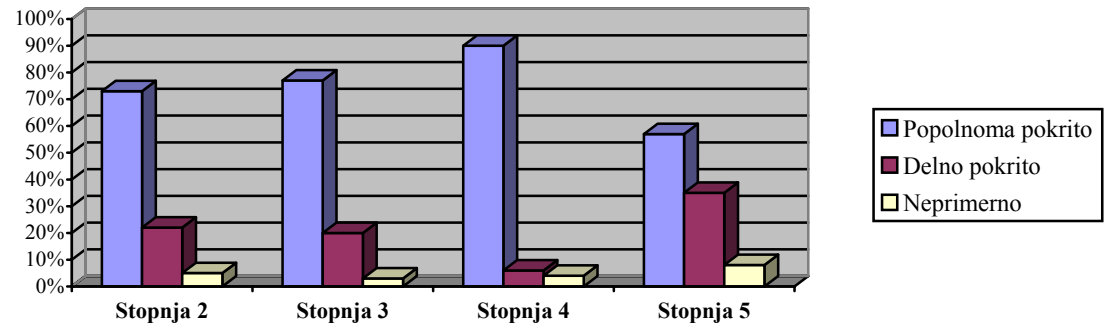
planiranja (ang. planning manager), vodja kakovosti procesa (ang. quality/process manager) in vodja podpore (ang. support manager). Model TSP priporoča še pet vlog več: vodjo razvoja nadomestijo štiri vloge (vodja analize, vodja načrtovanja, vodja izdelave in vodja testiranja), dodatno pa skupinski proces določa še vodjo procesa in vodjo stikov z naročnikom. Model TSP tudi omogoča, da lahko isti član (z izjemo vodje skupine) opravlja več vlog naenkrat.

V [Dav02] je opisana primerjava med modeloma TSP in CMM. Preverili so tipične prakse vseh ključnih področij z izjemo področja »Upravljanje razmerij s poddobavitelji«. Definirali so dve vrsti aktivnosti: **projektne aktivnosti** so namenjene izvajanju projekta, medtem ko **organizacijske aktivnosti** naslavljajo samo potrebe na organizacijski ravni. Izmed 294 preverjenih tipičnih praks je 139 projektnih, 70 organizacijskih in 85 takih, ki sodijo v obe skupini istočasno. S tega stališča so nato ocenili, v kolikšni meri model TSP pokriva aktivnosti modela CMM.

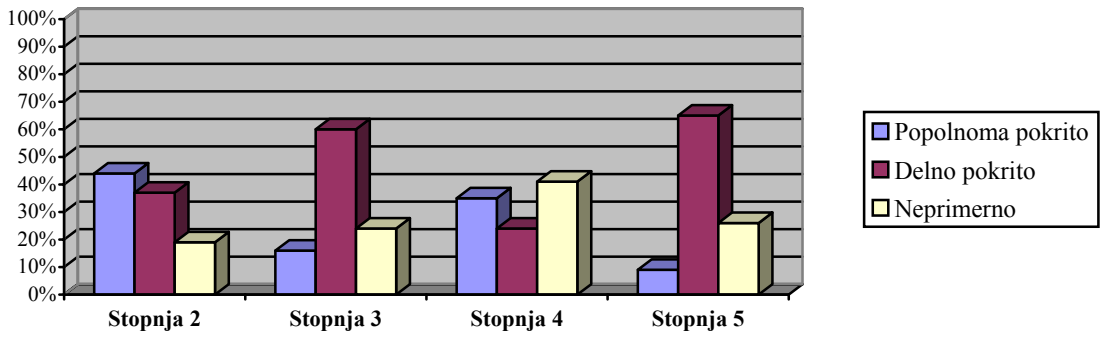
Rezultati so razvidni iz tabele 2.9 ter slik 2.33 in 2.34. Model TSP zelo dobro pokriva projektne aktivnosti modela CMM, medtem ko je pokrivanje organizacijskih aktivnosti slabše. To se kaže še posebej pri treh ključnih področjih 3. stopnje (»Osredotočenju na proces organizacije«, »Definiciji procesa razvoja PO« in »Programu izobraževanja«) ter pri »Uvajanju tehnoloških sprememb«, ki sicer sodi na 5. stopnjo modela CMM.

	Stopnja 2	Stopnja 3	Stopnja 4	Stopnja 5	Skupaj
Projekt	73%	42%	45%	14%	**47%**
Organizacija	0%	37%	6%	50%	**24%**
Oboje	27%	21%	48%	36%	**29%**

Tabela 2.9: Delež tipičnih praks CMM po vrstah aktivnosti in stopnjah [Dav02]



Slika 2.33: Delež pokritosti projektnih aktivnosti CMM pri TSP [Dav02]



Slika 2.34: Delež pokritosti organizacijskih aktivnosti CMM pri TSP [Dav02]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

2.3 Majhne organizacije za razvoj informacijskih sistemov

Namen pričujoče disertacije je predstaviti merski model, ki bo ustrezal zahtevam četrte stopnje modela CMM in bo primeren za majhne organizacije za razvoj IS, ki temeljijo na uporabi sodobnih sistemov za upravljanje s PB. Vrsta PO pomembno določa značilnosti merskega modela: najprej s specifičnim procesom razvoja PO, še bolj pa z možnostmi (in omejitvami) pri izbiri ustreznih parametrov merjenja [Van95, Ebe97, Her98].V našem primeru ima organizacija dve značilnosti, ki si ju bomo podrobneje ogledali:

- organizacija je majhna in
- organizacija razvija IS, ki temeljijo na sodobnih sistemih za upravljanje PB.

2.3.1 Merila za določanje majhnosti organizacije za razvoj PO

V splošnem je majhnost organizacije dokaj jasna kategorija: gre za organizacijo za razvoj PO, ki zmore zaradi svojih omejenih virov (ponavadi kadrov) izvajati svojo dejavnost (razvoj PO) v omejenem obsegu. Vendar moramo biti natančnejši in odgovoriti na dve vprašanji:

- kako formalno določamo majhnost in
- kdaj v skladu s tem pravimo, da je organizacija majhna.

Za določanje majhnosti se uporabljajo različni pristopi. Tako govorimo o majhnih organizacijah oziroma podjetjih (ang. small organizations, small companies) [Pau99, Joh98, Lar00, Hum95, Olo99, Ant01] in majhnih projektih (ang. small projects) [Pau99, Joh98, Hum95, Sch00]. Nekateri kot ločeno kategorijo omenjajo še majhno poslovanje (ang. small business) [Bro94, Joh98]. Vsi pristopi opisujejo organizacije, ki imajo z vidika razvoja PO podobne značilnosti. Za določanje majhnosti uporabljata predvsem dve merili:

- število vseh zaposlenih oziroma razvijalcev PO v organizaciji in
- velikost projektov, ki jih je organizacija sposobna izvajati.

Majhno število zaposlenih v organizaciji je najpomembnejša značilnost majhne organizacije. Pri tem lahko govorimo o številu vseh zaposlenih ali samo o številu zaposlenih, ki sodelujejo pri razvoju PO. Pri organizacijah, ki se ukvarjajo izključno z razvojem PO (v [Joh98] so te označene kot »majhne organizacije«), je pravzaprav vseeno, saj se običajno le majhen delež kadrov ne ukvarja z razvojem PO. Več težav je pri organizacijah, kjer razvoj PO predstavlja samo del dejavnosti. Tam je smiselno šteti samo razvijalce PO ali ustrezno upoštevati nekatere druge kriterije (recimo organizacijsko strukturo). Ker to običajno ni več enostavno objektivno meriti, nekateri avtorji [Joh98] raje govorijo o »majhnem poslovanju«, pri katerem je velikost vedno določena s številom vseh zaposlenih.

Število kadrov in njihove značilnosti določajo sposobnost za izvajanje različno velikih projektov razvoja PO. Majhne organizacije so sposobne sodelovati samo pri »majhnih« projektih, pri katerih naenkrat sodeluje malo razvijalcev in trajajo relativno malo časa. Majhno je tudi število projektov, ki jih zmorejo izvajati istočasno ali jih lahko zaključijo v določenem obdobju [Pau99].

V [Joh98] lahko preberemo, da ima majhna organizacija manj kot 50 razvijalcev, majhno poslovanje pa do 500 zaposlenih. Isti avtorici v analizi malih organizacij v ZDA [Bro94]

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

navajata povprečno število 40 zaposlenih, organizacije z manj kot 10 zaposlenimi pa naj ne bi bile sposobne uporabljati modela, ki bi bil tako zahteven kot je model CMM. Tudi v [Sch00] ima majhna organizacija 40 razvijalcev.

To se ujema z uradnim merilom za majhne organizacije v EU, ki ga navajata [Lar00, Orc00]. To podobno kot [Joh98] navaja 50 zaposlenih kot mejo za majhne organizacije. Vendar avtorji ugotavljajo, da ne moremo enačiti organizacij z 1, 5 in 50 zaposlenimi, zato definirajo tri (pod)vrste majhnih organizacij:

- izredno majhne (ang. eXtra eXtra Small ali XXS): od 1 do 2 zaposlena;
- zelo majhne (ang. eXtra Small ali XS): od 3 do 15 zaposlenih;
- majhne (ang. Small ali S): od 16 do 50 zaposlenih.

Kot je razvidno iz tabele 2.10 je velikost organizacije odvisna še od tega, koliko izdelkov izdeluje oziroma koliko projektov istočasno razvija.

	1 ali 2 zaposlena	**3 do 15 zaposlenih**	**16 do 50 zaposlenih**
1 izdelek / projekt	XXS	XS	S
Od 2 do 5 izdelkov / projektov	-	XS	S
Več kot 5 izdelkov / projektov	-	S	S

Tabela 2.10: Velikost majhnih organizacij po [Lar00, Orc00]

Velikost projektov, ki jih organizacija razvija je posreden način ugotavljanja velikosti organizacije. Majhni projekti zahtevajo malo sodelujočih, potekajo malo časa ali imajo omejene druge vire, zato se organizacije, ki jih izvajajo, obnašajo zelo podobno kot majhne organizacije. Pogosto so viri organizacije zares omejeni ali pa je razvoj PO samo ena izmed številnih dejavnosti organizacije. Čeprav je merilo posredno, ga je včasih enostavneje preveriti kot na podlagi poslovnih ciljev in organizacijske strukture oceniti njeno majhnost.

Majhni projekti zajemajo manj kot 20 razvijalcev [Joh98]. Po empiričnih podatkih je razmerje med številom razvijalcev projekta in številom zaposlenih v majhni organizaciji približno 1:2. V [Pau99] je majhen projekt tisti, ki traja 3 do 4 mesece in zaposluje 5 ali manj razvijalcev. V okviru predstavitve modelov PSP in TSP so predlagane velikosti projektov, ki so primerne za uporabo prvega ali drugega procesa [Hum00, Hum00b] (tabela 2.11).

Velikost projekta	**Število razvijalcev**	**Čas razvoja**	**Primeren model**
Majhen (ang. small)	3 do 5	6 mesecev	TSP
Zelo majhen (ang. very small)	2 do 3	4 meseci	TSP
Izredno majhen (ang. tiny)	1 do 2	2 meseca	TSP
Oseben (ang. personal)	1	1 teden	PSP
Smešen (ang. ridicolous)	1	1 uro	-

Tabela 2.11: Velikost majhnih projektov v skladu z [Hum00, Hum00b]

Obe merili sta primerni za klasifikacijo organizacij po velikosti. Vendar ima majhnost organizacije za posledico tak način delovanja, ki poleg omejenih virov povzroča še druge težave pri uporabi modela CMM.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

2.3.2 Značilnosti majhnih organizacij pri uporabi modela CMM

Majhne organizacije so zaradi omejenih virov bolj občutljive in njihov najpomembnejši cilj je preživetje. Zato je skrbno upravljanje vseh dejavnosti s ciljem najugodnejšega izplena za njih življenjskega pomena. Porabljeni viri se morajo oplemenititi v kar najkrajšem času. Poslujejo na enostaven način, vendar je poslovanje turbulentno, spremembe pogoste in nenapovedane. Zaposlujejo malo posameznikov, ki običajno nastopajo v več vlogah, tako tehničnih kot vodstvenih. Težave rešujejo s prilagodljivostjo, inovativnostjo in trdim delom. Ker so stroški razvoja odločilen dejavnik, jih nepotrebne naložbe lahko pokopljejo.

V skladu s tem na uporabo modela CMM pogosto gledajo kot na velik začetni vložek (strošek) z negotovim izidom. Tabela 2.12 prikazuje dvanajst najpogostejših pripomb v zvezi z uporabo modela CMM, ki so bile zbrane v 190 anketiranih majhnih ameriških organizacijah [Joh98].

	Opis težave z modelom CMM	% pripomb
1.	Premalo virov (časa, osebja, denarja)	16.3 %
2.	Preveliki stroški izboljšanja procesa razvoja PO (splošno)	11.1 %
3.	Preveliko število vlog, ki jih predpisuje model CMM	8.9 %
4.	Model CMM je primeren samo za velike organizacije	8.4 %
5.	Model CMM je preveč zahteven, težko razumljiv model	6.3 %
6.	Pridobitve z uporabo modela CMM ne odtehtajo velikih stroškov	6.3 %
7.	Stranke ne podpirajo izboljšanja procesa	6.3 %
8.	Pomanjkanje informacij o modelu CMM in prilagoditvah	5.3 %
9.	Neprimerno delovno okolje za uporabo modela CMM	5.3 %
10.	Model CMM predvideva prevelik obseg usposabljanja	4.7 %
11.	Težave z določanjem neodvisnih skupin v skladu z modelom CMM	4.7 %
12.	Model CMM predvideva preveč skupin - birokratski proces	4.7 %

Tabela 2.12: Pripombe majhnih organizacij glede uporabe modela CMM

Podobne ugotovitve navaja tudi [Sch00], ki je mnenje majhnih organizacij o modelu CMM strnil v šest točk. Model CMM je:

- preveč zahteven in težko prilagodljiv,
- že na začetku zahteva prevelik vložek virov (osebja, denarja, časa),
- zahteva preveč dokumentacije,
- preveč strog in strukturiran in
- nima takojšnega pozitivnega učinka.

To je povezano z najpogostejšimi napakami, ki jih naredijo majhne organizacije pri implementaciji modela CMM [Sch00]. Predvsem je značilen neprimeren pristop pri uporabi modela. Poleg uporabe modela brez prilagoditev je pogosta naivna predstava, da je mogoče v majhni organizaciji implementirati nespremenjen model CMM z majhno porabo virov. To izhaja tudi iz pomanjkljivega poznavanja in usposabljanja o modelu CMM, pogosto pa se implementacija začne tudi brez primernega načrta.

Vendar ne smemo pozabiti, da določene lastnosti majhnih organizacij tudi prispevajo k lažji implementaciji modela CMM [Pau99, Joh98, Sch00, Ant01]. Majhne organizacije so veliko bolj prilagodljive kot velike, kar pomeni, da je lažje uvajati novosti. Ker se hitreje odzivajo na spremembe, je učinek sprememb tudi lažje opazovati in mogoče hitreje ukrepati. Ker je organizacija majhna, je vodenje enostavnejše in bolj učinkovito, kar tudi skrajša postopke. Posamezniki ponavadi opravljajo več različnih delovnih nalog, kar jim daje večji pregled nad

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

dogajanjem in večji občutek odgovornosti. Zato se osebje počuti bolj povezano z organizacijo in bolje sodeluje v vseh aktivnostih organizacije. Tudi komunikacija in nadzor kakovosti (brez formaliziranih postopkov) je v majhnih organizacijah boljša, kar tudi izhaja iz tesnejše vpletenosti posameznikov v poslovanje organizacije. Vse to se običajno kaže v solidni produktivnosti že brez uporabe formalnih metod za zagotavljanje kakovosti.

2.3.3 Značilnosti organizacij za razvoj informacijskih sistemov

Značilnosti razvoja IS bi lahko strnili v tri točke:

a) specifičen tehnološki proces,
b) vmesni rezultati razvoja in
c) uporaba programskih orodij.

a) Razvoj IS zahteva tehnološki proces razvoja PO, ki je prilagojen posebnostim te vrste PO (recimo [Dor97]). Proces ima naslednje pomembne značilnosti:

- Pomemben del razvoja je posvečen raziskovanju problema. Ker je od tega odvisen končni rezultat, je temu posvečenih več faz (strateško načrtovanje, zajem in analiza zahtev), ki lahko časovno obsegajo največji del razvoja.
- Zelo pomembna je vloga naročnika, saj on določa zahteve in uspešnost končnih rezultatov. Proces razvoja mora vsebovati veliko komunikacije z naročnikom.
- Končni izdelek je sestavljen iz procesnega in podatkovnega dela, enako kakor tudi vmesni rezultati. Oba dela razvijamo skupaj.
- Razvoj od opisa zahtev do izdelka je jasno definiran in določa več specifičnih vmesnih rezultatov. Ti opisujejo končni rezultat z različnih vidikov.
- Pomemben je tudi izgled in funkcionalnost izdelka, zato je potrebno oblikovanju uporabniškega vmesnika med razvojem posvetiti veliko pozornosti.
- Posebno pozornost je treba nameniti dokumentaciji. Poleg opisa izdelka mora ta vsebovati kakovostna navodila za uporabo ter sistem pomoči.
- Poleg razvoja izdelka je pomembno tudi uvajanje tega v uporabo. Ob koncu življenjskega cikla je treba dodati fazi namestitve in vzdrževanja ter dopolniti fazo testiranja. Vse te aktivnosti je treba ustrezno planirati, izvajati in jih nadzirati.
- Pri uvajanju izdelka v uporabo je potrebno poleg pravilnega procesnega in podatkovnega izdelka poskrbeti tudi za uporabnost obeh: pri procesnem delu poskrbeti za ustrezne nefunkcionalne zahteve (zmožnost, varnost, nastavitve), pri podatkovnem pa za polnjenje PB, arhiviranje in podobno.
- Pri nameščanju je treba poskrbeti za pravilno strategijo namestitve, ustrezno polnjenje PB, izobraževanje uporabnikov in podobno.
- Vzdrževanje traja tudi potem, ko je projekt razvoja izdelka končan. Treba je predvideti postopke spremljanja stanja sistema in odpravljanje težav med delovanjem.
- Tehnološki postopek zahteva posebne vloge v skupini razvijalcev: recimo analitika, razvijalca in skrbnika PB, osebje za izvedbo namestitve in vzdrževalce.

b) Pri razvoju se pojavi več vmesnih rezultatov. Ti so posledica postopka opisa zahtev z več vidikov (procesnega, podatkovnega) in postopnega razvoja načrta in izdelka [Goo95]. Vmesni rezultati se dopolnjujejo in rastejo v končen izdelek, kar omogoča uvajanje metod za preverjanje pravilnosti razvoja, kot so recimo navzkrižna preverjanja. Vmesni izdelki so zaradi večje jasnosti pogosto opisani z diagrami in grafi [Mar85]. Tipični izdelki so:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- seznam zahtev za opis vseh zajetih zahtev;
- diagrami entitete-razmerja za opis podatkovnega vidika;
- diagrami podatkovnih tokov za opis procesnega vidika;
- funkcijska dekompozicija za predstavitev strukture izdelka;
- razne matrike za predstavitev razmerij med različnimi elementi;
- konceptualni diagram za prestavitev sestave procesnega vidika izdelka;
- prototipi za poenostavljeno predstavitev posameznega dela procesiranja.

c) Razvoj IS je primeren za avtomatizacijo, saj ta omogoča lažji razvoj, navzkrižna preverjanja vmesnih rezultatov in avtomatsko izvajanje določenih faz. Orodja običajno podpirajo določene metodologije in tehnike razvoja PO (recimo Oracle [Dor97]).

S tem zaključujemo pregled raziskovalnega področja.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3. PROCES RAZVOJA PO V MAJHNI ORGANIZACIJI ZA RAZVOJ IS

Prvi korak na poti do modela za kvantitativno spremljanje procesa razvoja PO je določitev ustreznega procesa razvoja PO. Za uvajanje meritev je potreben natančno definiran proces razvoja PO z vsemi elementi: aktivnostmi, viri in izdelki. Če tega ni, nimamo kaj meriti oziroma so rezultati meritev neprimerljivi. Glede na zastavljeno nalogo mora biti to upravljan proces v smislu modela CMM, ki bo ustrezal majhni organizaciji za razvoj IS.

Doslej smo ugotovili, da je model CMM uspešen model za zagotavljanje kakovosti, ki na primeren način obvladuje vsa pomembna področja in po mnenju mnogih opisuje referenčni primer procesa, kakršnega naj bi uporabljala vsaka organizacija [Pau99, Her94]. Vendar je model CMM zaradi svoje kompleksnosti primeren predvsem za velike organizacije za razvoj PO. Za majhne organizacije je model preveč zahteven in obsega več, kot potrebujejo oziroma zmorejo [Pau99, Bro94, Sch00]. Ker želimo model CMM vseeno uporabiti pri določanju našega procesa, moramo **v prvem podpoglavju** najprej raziskati, na kakšen način bi to najprimerneje storili.

Odločitev o uporabi prilagojenega modela CMM bo vplivala na način določanja procesa, ki ga bomo predstavili **v drugem podpoglavju**. Običajen pristop predvideva ustrezno definiranje izhodiščnega procesa, nato pa postopno nadgradnjo in preverjanje doseženih ciljev [Pau95a]. Tak pristop bomo uporabili tudi mi, le da bomo za izhodišče uporabili dva procesa. Prvi bo okvirno ustrezal zahtevam modela CMM, drugi pa bo odražal tehnološke posebnosti razvoja IS. Oba bomo združili in nadgradili v končni proces razvoja PO.

Ena najpomembnejših zahtev pri definiciji procesa je prilagojenost majhnim organizacijam za razvoj IS, ki posredno pomeni zahtevo po enostavnosti in uporabnosti. Uporaba modela CMM nas navaja k drugačnemu cilju: k natanko določenemu procesu razvoja, ki je primerno zapleten za razumevanje in uporabo. Majhne organizacije so različno velike in v praksi razvijajo različno velike projekte, zato potrebujejo oboje: včasih zelo podrobno definiran proces, drugič spet proces, ki ustrezno izvede samo nekaj najnujnejših faz. V vsakem primeru pa potrebujejo proces, ki izvede vse, kar bi glede na obseg naloge moral v skladu z zahtevami modela CMM.

Za rešitev omenjene težave predlagamo dinamično prilagodljiv proces. Ta je dovolj natančno določen in je gledano kot celota dokaj obsežen, zato ga lahko majhna organizacija uporablja za zahtevne projekte. Vendar proces omogoča sprotno poenostavljanje v skladu z natanko določenimi pravili. Med izvajanjem projekta se lahko postopoma spremeni v zelo enostaven proces, ki pa mu spričo natančnih pravil nič ne manjka z vidika zahtev modela CMM. Načela dinamično prilagodljivega procesa bomo predstavili **v tretjem podpoglavju**.

V četrtem podpoglavju bomo kratko vsebinsko predstavili proces razvoja PO, ki smo ga na tak način določili. Začeli bomo s kratko predstavitvijo formalnega opisa procesa, ki ga uporabljamo za predstavitev procesa v prilogi. Ker moramo proces umestiti v ustrezno okolje,

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

bomo okvirno predstavili še delovanje tipično majhne organizacije ter skupine in vloge v njej. Glavni del bo okvirna predstavitev organizacijskega procesa razvoja PO do ustrezne stopnje podrobnosti. V zadnjem delu podpoglavja bomo pokazali še vmesne in končne rezultate, ki nastanejo ob izvajanju opisanega procesa razvoja PO.

Ob zaključku poglavja bomo kratko povzeli najpomembnejše rezultate.

3.1 Prilagoditev modela CMM

Tako v velikih kot majhnih organizacijah obstaja vrsta področij, ki jih organizacija mora obvladovati. Vsaka organizacija za razvoj PO recimo potrebuje dokumentirane uporabnikove zahteve, učinkovito vodenje projektov, ustrezno planiranje, primerno organizacijo dela in podobno. Model CMM na primeren način obvladuje vsa pomembna področja in opisuje referenčni primer procesa, kakršnega naj bi uporabljala vsaka organizacija [Pau99, Sch00]. Rezultati raziskave [Bro94] to potrjujejo: večina izmed 190 anketiranih majhnih organizacij v ZDA ocenjuje model CMM kot najprimernejše vodilo za izboljšanje kakovosti procesa razvoja PO v organizaciji. To je tudi razlog, zakaj smo v našem primeru izbrali model CMM za izhodišče.

Vendar je model CMM narejen na podlagi izkušenj velikih organizacij in zato primeren predvsem za večje, zahtevnejše organizacije [Pau99, Joh98, Sch00]. To se kaže predvsem v opisu tipičnih praks, ki so zahtevne za realizacijo in s tega stališča neprimerne za manjše organizacije. Zahtevnost modela za majhne organizacije kažejo tudi drugi elementi. Model CMM določa več kot 25 različnih vlog v organizaciji [Orc00, Ant01], čeprav v majhni organizaciji pogosto sploh ni toliko zaposlenih. Model predvideva veliko razvojne dokumentacije in obsežne spremljajoče dejavnosti (izobraževanje, nadzor, merjenje), kar je za majhno organizacijo lahko prehudo breme. Te ugotovitve potrjujejo tudi rezultati raziskav [Bro94, Sch00].

Poleg dejstva, da je model CMM zahteven za uporabo v majhnih organizacijah, je pomembno tudi to, da model obsega več, kot majhne organizacije zares potrebujejo. Enake cilje bi lahko v majhni organizaciji dosegli tudi na drugačen, ustreznejši način.

Neposredna uporaba modela CMM zato očitno za majhne organizacije ni primerna [Ant01, Bro94, Orc00, Pau99] in model CMM moramo uporabiti drugače. Možnosti sta dve:

- uporabiti sorodne modele, ki vsebujejo načela modela CMM in so primernejši za majhne organizacije ali
- prilagoditi CMM, da bo primeren za majhne organizacije.

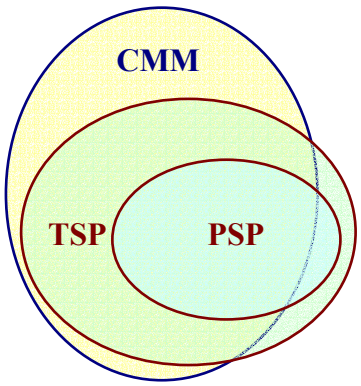
3.1.1 Uporaba sorodnih modelov

Model CMM lahko nadomestimo z uporabo osebnega ali skupinskega procesa razvoja PO (glej razdelka 2.2.7 in 2.2.8) [Pau99, Rož02, Rož02a]. Oba modela sta preskriptivna, torej natanko določata, kako realizirati ustrezen proces. S tega stališča je njuna uporaba še enostavnejša. Modela obsegata vse tiste aktivnosti modela CMM, ki jih je z njunega vidika smiselno uporabiti. V okviru svojega namena oba določata proces na 5. zrelostni stopnji [Hum99].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Vendar z vidika majhne organizacije temu ni tako. Model PSP je namenjen posamezniku in ne obsega aktivnosti, ki so namenjene skupinskemu delu. Zato tudi ne obsega določenih projektnih ali organizacijskih aktivnosti, ki jih tudi majhna organizacija nujno potrebuje. Zaradi tega je model PSP primeren samo v izjemnih primerih, če ima organizacija samo nekaj zaposlenih in ti delujejo predvsem individualno. Druga možnost je uporaba osebnega procesa za realizacijo podprocesov v organizaciji [Rož02, Rož02a].

Uporaba modela TSP temelji na predpostavki, da je proces razvoja PO v majhni organizaciji dejansko enak procesu v skupini, če je število razvijalcev v obeh približno enako. To je žal samo deloma res. Kot je razvidno iz zadnjega dela razdelka 2.2.8, tak pristop zelo ustrezno pokriva projektne aktivnosti, ne pa tudi vseh organizacijskih aktivnosti modela CMM [Dav02]. Model TSP ne obvladuje nekaterih pomembnih organizacijskih področij, kot je recimo upravljanje procesa razvoja PO v organizaciji. Slika 3.1 prikazuje razmerje med modeli CMM, TSP in PSP, pri čemer površine ne ustrezajo deležu pokrivanja modelov.



Slika 3.1: Razmerje med modeli CMM, TSP in PSP

Če se odločimo za uporabo modela TSP, bomo v majhni organizaciji na primeren način dosegli glavnino ciljev CMM. S tem bomo okvirno zadostili zahtevam druge stopnje modela CMM, vendar višje samo z modelom TSP ne bomo mogli. Skupinski proces je torej primeren za začetek, vendar moramo postopoma v majhno organizacijo uvesti tudi manjkajoče aktivnosti na organizacijski ravni [Rož02].

3.1.2 Prilagoditev modela CMM za majhne organizacije

Ker je model CMM za majhne organizacije prezahteven, ga lahko poskušamo prilagoditi, da bo ustrezal zmožnostim majhne organizacije [Ant01, Pau99, Rož02]. Prilagajanje modela CMM majhnim organizacijam je znano raziskovalno področje [Pau99, Hum99, Hum98, Ant01, Joh98, Orc00, Red00, Sch00]. Avtorji so se reševanja problema lotili z različnih izhodišč, vendar so pogosto prišli do podobnih zaključkov.

Tudi naša prilagoditev vsebuje nekatere znane ideje, vendar je nadgrajena z izvirnimi predlogi. Prilagojena je posebnostim majhnih organizacij za razvoj IS in konkretno implementirana v okviru prilagodljivega procesa za razvoj PO.

Slika 3.2 prikazuje skico predlagane prilagoditve. Najprej opredelimo pravila prilagajanja, s katerimi določimo, katere dele modela CMM lahko spreminjamo in na kakšen način. Nato predstavimo konkretne predloge na več izbranih področjih.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 3.2: Področja prilagoditve modela CMM majhnim organizacijam

3.1.2.1 Pravila prilagajanja modela CMM

Osnova za prilagajanje je v samem modelu CMM, saj model nikjer ne predpisuje, kako morajo biti določene aktivnosti izvedene, temveč samo zahteva, da morajo biti cilji doseženi na ustrezen način. Čeprav so tipične prakse natanko opredeljene, te niso obvezne. Cilje lahko dosegamo na drugačen način, ki je za majhno organizacijo primernejši [Hum98]. Pri tem upoštevamo naslednja pravila:

a) Pri prilagajanju ohranjamo natančnost definicije procesa

Pomembna prednost modela CMM je natančna definicija procesa. Če pri prilagajanju to izgubimo, bo prilagojeni model neustrezen. Zato je pomemben cilj prilagajanja ohranjanje ustrezno nadzorovanega procesa. S tega stališča lahko definiramo dve vrsti prilagajanja:

- enakovredno prilagajanje in
- neenakovredno prilagajanje.

Enakovredno prilagajanje: Zaradi deskriptivnega modela je prilagajanje relativno enostavno, če zamenjujemo enakovredne dele modela. Taka primera sta recimo izvedba aktivnosti na drugačen (enakovreden ali zahtevnejši) način ali zamenjava ene vloge z drugo, ki ima enake zadolžitve. Pri enakovrednem prilagajanju moramo zgolj paziti, da nadomestek ustreza vsem zahtevam izvirnika.

Neenakovredno prilagajanje: Veliko težje je v (pogostejšem) primeru, ko skušamo del modela nadomestiti z neenakovrednim, enostavnejšim ali slabšim delom. Tedaj prilagajanje dejansko pomeni poenostavitev, ki zahteva tudi opuščanje nekaterih zahtev modela CMM. Nekatere zahteve bodo opuščene kot nujna posledica poenostavitve, druge pa bomo opustili ali spremenili zaradi smiselnosti.

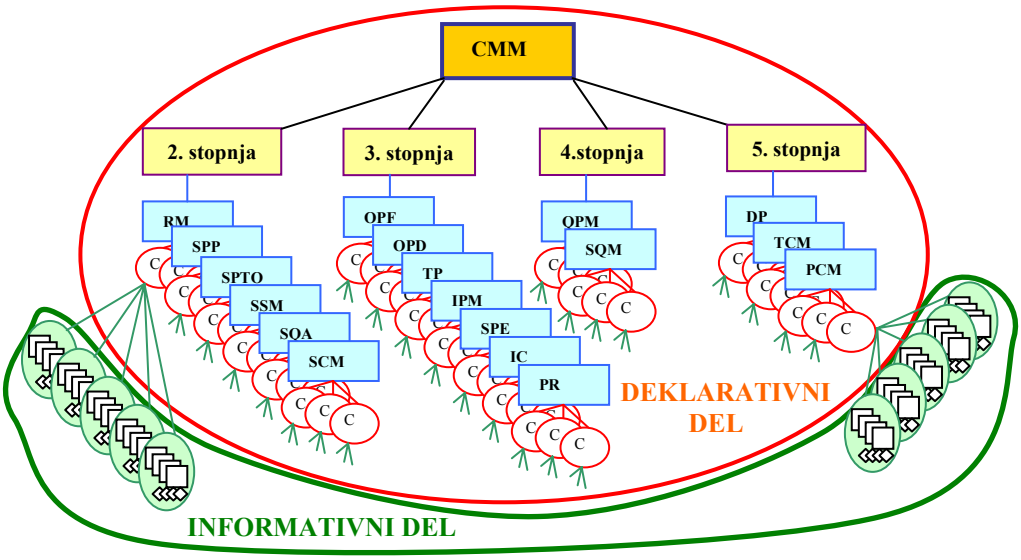
- V prvem primeru je običajno prilagoditev nujna, zato moramo spremenjeno funkcionalnost modela sprejeti. Recimo: če skupino nadomestimo z eno samo vlogo, bo posledica tega opuščanje sestankov skupine.
- V drugem primeru bomo zahteve spremenili zaradi smiselnosti. Sprememba bo vplivala tudi na druge dele modela, zato potrebujemo celovit pogled na model. Tipičen primer je opuščanje vloge, ki zahteva premestitev njenih glavnih zadolžitev med druge vloge.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

b) Prilagajamo samo nujni (deklarativni) del modela CMM

V našem primeru je potrebno prilagajanje modela CMM zaradi omejenih zmožnosti majhnih organizacij, zato želimo model kar najbolj poenostaviti. Osnovno vodilo pri prilagajanju je implementirati vse nujne dele modela CMM na primeren način. Opustimo lahko tiste dele, ki za majhne organizacije niso potrebni. Veliko delov modela bo treba drugače, enostavneje implementirati. Presoja, kaj je nujno, kaj smiselno in kaj odveč je jedro prilagoditve.

Prvi korak je zato določiti tiste elemente opisa modela CMM, ki določajo nujni del modela. Te predstavlja deklarativni del modela: pet stopenj, 18 ključnih področij in 52 ciljev [Pau93a]. Ti elementi so dovolj splošno definirani, da so primerni tudi za majhne organizacije. Ostali deli opisa modela CMM (316 tipičnih praks, še več podpraks in primerov) predstavljajo informativni del modela. Ti deli so predstavljeni na način, ki pogosto ni primeren za uporabo v majhnih organizacijah. To je tisti del modela, ki ga je treba smiselno prilagoditi za majhne organizacije. Iz slike 3.3 sta razvidna oba dela modela.



Slika 3.3: Deklarativni in informativni del modela CMM [Pau93a]

c) Izpustimo dele modela, ki niso primerni za majhne organizacije

Če si ogledamo model CMM z vsebinske plati lahko presodimo, da je mogoče brez škode izpustiti tudi določena ključna področja ali cilje. Primer je ključno področje »Upravljanje razmerij s poddobavitelji«. Za majhne organizacije ni smiselno, da bi imele še poddobavitelje. Glavnino ciljev vseh ostalih ključnih področij je treba doseči tudi v majhnih organizacijah [Pau99, Ant01], vendar moramo prilagoditi izvedbo zmožnostim majhne organizacije.

d) Poenostavimo dele modela, ki služijo institucionalizaciji procesa razvoja PO

Vse aktivnosti modela CMM so razdeljene v pet skupin aktivnosti (glej razdelek 2.2.6.2). Štiri skupine obsegajo podporne aktivnosti, ki institucionalizirajo aktivnosti skupine »Izvajane aktivnosti« in te so v vseh ključnih področjih zelo podobne. Splošno gledano so te aktivnosti preveč zahtevne za majhne organizacije, zato jih poenostavimo na naslednji način:

»Zavezanost za izvajanje« zahteva obstoj dokumentiranega postopka na organizacijski ravni, ki zavezuje udeležence za izvedbo aktivnosti ključnega področja. Poleg tega na številnih ključnih področjih zahteva jasno določitev odgovornega vodje. V našem primeru bomo

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

predvideli, da obstajajo standardizirani postopki za vse pomembne dejavnosti organizacije. To niso splošni dokumenti, temveč kratko in razumljivo (po korakih) opisana navodila za izvedbo raznih aktivnosti. Odgovorni izvajalci bodo določeni za vsako aktivnost v organizaciji, vendar bo število vlog (vodij) manjše.

»Sposobnost za izvajanje« obsega tri področja: dostop do ustreznih znanj (usposabljanje), ustrezno financiranje in dostop do orodij. V primeru majhne organizacije tudi predvidevamo ustrezno usposabljanje za izvajanje vseh aktivnosti ključnega področja, vendar tega izvajamo na manj formalen in cenejši način. Predvidevamo tudi, da s financiranjem in dostopom do orodij ni nobenih težav. Za slednje imamo zadolženega posebnega člana projektne skupine (vodjo podpore).

»Merjenje in analiza« običajno določa tipično prakso za mersko spremljanje statusa izvedenih aktivnosti ključnega področja. V našem primeru bomo predvideli mersko spremljanje vseh aktivnosti v organizaciji, torej poenostavitev ni potrebna.

»Preverjanje izvedbe« je formalizacija preverjanja izvajanja aktivnosti treh vrst:

- občasno s strani vodstva organizacije,
- občasno ali po potrebi s strani vodje projekta in
- preverjanje postopkov s strani skupine SQA.

V majhni organizaciji je običajno organizacijska struktura enostavnejša in komunikacija med člani boljša, zato te postopke poenostavimo. Vsi merski podatki in rezultati se nahajajo (ali so dostopni) v skupnem repozitoriju in ustrezni člani imajo dostop in možnost, da jih kadarkoli preverijo. Ker so na ravni organizacije predvideni redni tedenski sestanki, na katerih se poroča tudi o poteku vseh projektov, pregledi za vodstvo organizacije in projekta niso posebej predpisani. Skupina SQA določi enega izmed svojih članov za vodjo kakovosti pri projektu. Ta ima preko svojih projektnih obveznosti neposreden vpogled v vse aktivnosti projektne skupine, o čemur tudi poroča vodji projektne skupine in (občasno) vodji skupine SQA.

Opuščanje oziroma drugačna izvedba tipičnih praks, ki sodijo v skupino *»Izvajane aktivnosti«*, bo razvidna iz predstavitve procesa v nadaljevanju.

e) Model CMM prilagajamo tipični majhni organizaciji za razvoj IS

Veliko lažje je, če imamo pred očmi konkretno organizacijo in skušamo model CMM prilagoditi njej. To je z vidika majhne organizacije tudi najbolj prav, saj ne iščemo splošnih navodil (ta predstavlja model CMM), temveč konkretne rešitve. V našem primeru o konkretni organizaciji ne moremo govoriti, lahko pa govorimo o tipični majhni organizaciji za razvoj IS, kar je podobno.

S tem smo si ogledali vsa pravila prilagajanja. Sledi opis prilagoditev po področjih.

3.1.2.2 Ustreznejša delitev skupin in vlog modela CMM

Število zaposlenih in njihova organizacijska struktura je za uspešno izvajanje dejavnosti organizacije ključnega pomena, zato je model CMM s tega stališča podrobno razdelan in določa veliko število različnih skupin in vlog. Tako model CMM predvideva vsaj 7 skupin in več kot 25 različnih vlog [Pau93]. To za majhne organizacije veliko preveč, zato je treba število skupin, vlog in njihovih nalog prilagoditi razpoložljivim virom v organizaciji.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Za majhne organizacije z malo člani in enostavno organizacijsko strukturo ni primerna specializacija posameznih članov, temveč težnja, da bi usposobile vse člane za opravljanje različnih nalog. To je sicer težko doseči, vendar globalne aktivnosti organizacije težijo k temu cilju. Tako recimo organizacija vsem omogoča različna usposabljanja, posveča veliko pozornost prenosu izkušenj in predvideva zamenljivost pri izvajanju različnih vlog. To sicer kratkoročno povzroča težave pri izvajanju projektov, vendar dolgoročno koristi organizaciji.

V skladu s tem tudi vsi člani aktivno sodelujejo pri izvajanju projekta. Težimo k temu, da so člani čim bolj enakomerno obremenjeni tako po porabi časa kot po zadolžitvah. Model CMM predvideva hierarhijo petih ravni vodstvenih vlog [Pau93a, SEI02b], kar zmanjšamo na tri ravni: vodja (oziroma vodstvo) organizacije, vodja projekta in vodje posameznih delov razvoja, ki so v resnici tudi izvajalci. Na ravni organizacije obstajata še vodji dveh skupin.

Skupine na ravni organizacije in projektov

S stališča majhne organizacije model CMM določa preveč skupin, zato smo na organizacijski ravni predvideli samo dve. Druge skupine smo opustili in njihove naloge pripisali ostalim skupinam. Vsaka skupina vsebuje vsaj enega člana (vodjo). Na ravni projekta smo vse skupine nadomestili z vlogami vodij, ki so zadolženi za izbrano področje. Pri tem smo predvideli, da vodji po potrebi pomagajo drugi člani, kar se podrobno določi ob planiranju projekta. Skupine smo prilagodili na naslednji način:

- **Skupina za tehnologijo procesa za razvoj PO (SEPG):** skupina skrbi za vzdrževanje in razvoj organizacijskega procesa in repozitorija. Obstaja tudi v našem primeru, njene naloge so razširjene z nalogami skupin, ki so ukinjene. Skupina razporeja svoje člane posameznim projektom, da tam opravljajo vlogo vodje procesa in merjenja.
- **Skupina za zagotavljanje kakovosti PO (SQA)**: skupina skrbi za ustrezno kakovost procesa in je s tega stališča potrebna tudi v našem primeru. Člani skupine SQA sodelujejo pri posameznih projektih v vlogi vodij kakovosti.
- **Skupina za tehnologijo PO**: po prilagoditvi ta razvojna skupina ne obstaja. Nadomeščajo jo vsi člani projektne skupine (z izjemo vodje kakovosti in procesa), ki sodelujejo pri razvoju. V skupino sodijo tudi vodje razvoja (analize, načrtovanja, izgradnje in testiranja).
- **Skupina za zajem sistemskih zahtev**: skupina ni potrebna, na ravni procesa jo nadomešča vodja analize.
- **Skupina za sistemsko testiranje**: tudi ta skupina ni potrebna. Njeno vlogo prevzame skupina za zagotavljanje kakovosti PO, na projektni ravni pa jo nadomešča vodja testiranja in vsi člani projektne skupine.
- **Skupina za upravljanje konfiguracije PO**: tudi te skupine ne predvidimo, njeno vlogo pa v okviru projekta prevzame vodja podpore.
- **Skupina za usposabljanje**: te skupine ne potrebujemo. V okviru projekta prevzame nekatere njene naloge vodja projekta, za nadzor pa poskrbijo člani skupine SQA.

Model CMM določa, da morajo biti nekatere vloge in skupine neodvisne, kar pomeni, da ne smejo biti zaupane isti osebi (skupini) oziroma nekomu, ki je podrejen odgovornemu izvajalcu. Neodvisnost zagotavljamo s samostojno skupino SQA na ravni organizacije. S tem, ko določene vloge na ravni projekta izvajajo njeni člani, te vloge niso pod nadzorom vodje projekta, temveč lahko član skupine SQA preko vodje skupine SQA neovirano dostopa do vodstva organizacije. Skupina SQA ima pristojnosti tudi pri preverjanju usposabljanja in sistemskega testiranja, za kar se tudi zahteva neodvisnost. Če za opravljanje nalog v okviru določenega projekta ni dovolj članov skupine SQA, se za pomoč določi člane iz drugih projektnih skupin.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Vloge na ravni projektov

Pri vlogah smo zaradi učinkovitejše izvedbe (specifičnih) razvojnih aktivnosti procesa določili nekaj več vlog, kot jih predvideva model TSP (vodja analize, dve vodji načrtovanja, dve vodji izvedbe, vodja testiranja), dodali pa smo tudi odgovorno osebo za sodelovanje z naročnikom. S tem postane število vlog kar veliko (mogoče celo večje od števila članov projektne skupine), vendar naš pristop predvideva, da lahko več vlog opravlja isti član projektne skupine. Zaželjeno je, da si člani čim bolj enakomerno razporedijo vloge. Pri poimenovanju vlog smo se nekoliko zgledovali po modelu TSP [Hum00, Web99], nekaj vlog pa smo tudi preimenovali, da bi bolje odražale značilnosti našega procesa. Vloge so naslednje:

- vodja projekta,
- vodja planiranja,
- vodja analize,
- vodja razvoja PO, ki ga lahko nadomešča več vlog:
    - vodja načrtovanja IS,
    - vodja načrtovanja PB,
    - vodja izgradnje IS,
    - vodja izgradnje PB,
    - vodja testiranja,
- vodja sodelovanja z naročnikom,
- vodja kakovosti,
- vodja podpore in
- vodja procesa in merjenja.

Glavne značilnosti vlog in skupin so predstavljene v dodatku A.1.

3.1.2.3 Prilagoditev procesa ciljem organizacije

Pri prilagajanju modela CMM je treba upoštevati cilje organizacije in pozornost posvetiti tistim področjem, ki so zanjo najbolj zanimiva. To bo v našem primeru zahtevalo predvsem spremembe pri tehnološkem vidiku modela CMM oziroma natančneje pri izvedbi ključnega področja »Tehnologija izdelkov PO« in nekaterih z njim povezanih področij. Tehnološki proces bo zasnovan tako, da bo ustrezal razvoju IS. Določanje ustreznega tehnološkega procesa bo predstavljeno v podpoglavju 3.2.

Doseganje ciljev v majhnih organizacijah je povezano z uporabo ustreznega življenjskega cikla. Model CMM ne zahteva uporabe predpisanega življenjskega cikla, vendar zaradi enostavnosti za majhne organizacije nekateri avtorji [Pau99, Joh98, Orc00] predlagajo kar običajni kaskadni življenjski cikel [Cha86, Kan95]. Tudi mi smo za izhodišče uporabili ta življenjski cikel, vendar smo ga nadgradili (glej podpoglavje 3.2).

Ob prilagajanju je smiselno zastaviti tako strukturo procesa, da bo le-ta odražal posebnosti organizacije in dolgoročno zagotavljal uspešen razvoj PO. V našem primeru želimo zaradi enostavnosti uporabe omogočiti enostavno prilagajanje uporabnikovim zahtevam. V ta namen smo razvili koncept dinamičnega prilagajanja, ki bo predstavljen pri opisu procesa razvoja PO v podpoglavjih 3.3 in 3.4.

3.1.2.4 Poenostavljeno dokumentiranje

Model CMM zahteva vpeljavo nadziranega izvajanja ustreznih postopkov. Formalno to pomeni, da morajo biti proces, projekt in rezultati dela ustrezno dokumentirani, zato določa

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

več vrst dokumentacije (plane, specifikacije, poročila in podobno). Vendar to ne pomeni, da mora biti dokumentacija številna, obsežna in podrobna, temveč mora biti pregledna, razumljiva, uporabna in tudi primerna obsegu delovanja organizacije [Pau99, Joh98, Red00]. To še posebej velja za majhne organizacije, ki si ne morejo privoščiti velikega obsega dela za izdelavo le-te. Veliko lahko nadomestijo z dobro komunikacijo, bližino razvijalcev, ustaljenimi pristopi in predvsem avtomatizacijo izdelave dokumentacije.

Ne glede na to pa sodimo, da mora tudi v majhni organizaciji obstajati določena raven formalne komunikacije, zato pri izločanju nepotrebne dokumentacije nismo preveč strogi. Pri tem poenostavimo pisanje dokumentacije z uporabo že pripravljenih obrazcev, predlog, orodij in večnamenskih dokumentov, ki dokumentacijo tvorijo na podlagi podatkov iz repozitorija.

Postopek dokumentiranja tudi avtomatiziramo. Dokumenti in podatki se zbirajo na enem mestu in v elektronski obliki. Zajemajo se samo nujni podatki, ki so med vnosom ustrezno preverjeni. Preverjajo se tudi celotni dokumenti, če tako predvidevajo standardni postopki.

3.1.2.5 Majhen obseg meritev

Pomembna značilnost zrelih organizacij je nadzor razvojnega procesa. K temu sodi merjenje procesa, projektov in izdelkov. V razdelku 2.1.6.2 predstavljene izkušnje kažejo, da se je pri tem smiselno omejiti na mersko spremljanje majhnega števila dobro definiranih meritev [Pau99, Roz00, Rif91]. Kot bo predstavljeno v drugem delu disertacije so v našem primeru merjenja velikosti, napredovanja, stroškov in kakovosti uvedena kot nujen del vsake aktivnosti, ki jo izvaja posameznik. Merski podatki se v skladu z zahtevami CMM zbirajo na enem mestu. Prilagoditev modela CMM za majhne organizacije se skriva v jasno definiranem in enostavnem procesu merjenja, predvsem pa v skrbno izbranih meritvah in pokazateljih. To področje bomo podrobno predstavili v četrtem in petem poglavju.

3.1.2.6 Poenostavljeno usposabljanje

Del modela CMM je tudi ključno področje, ki predvideva uvedbo usposabljanja na organizacijski ravni z vsemi potrebnimi podpornimi aktivnostmi.V našem primeru formalna usposabljanja poenostavimo in nadomestimo z neformalnimi, recimo selektivnim učenjem po potrebi, samoučenjem, posredovanjem znanja med kolegi, učenjem na daljavo in obiskovanjem zunanjih tečajev. Pozornost posvetimo prepoznavanju že obstoječih znanj in njihovi učinkoviti uporabi. V skladu z zahtevami CMM posvečamo pozornost usposobljenosti za izvajanje posameznih vlog, vendar se pri tem opiramo na neformalne pristope (recimo posredovanje znanja med kolegi). Usposabljanje planiramo po potrebi skupaj z ostalimi opravili raje na ravni projektov in ne na organizacijski ravni, na tej pa ostajajo zadolžitve skupini SQA za preverjanje znanja.

3.1.2.7 Ustrezno zagotavljanje kakovosti

Zelo pomemben del modela CMM je zagotavljanje kakovosti. Temu je deloma ali v celoti posvečenih več ključnih področij, še zlasti pa je zagotavljanje kakovosti pomembno za četrto stopnjo zrelosti procesa. Ne glede na to je za potrebe majhne organizacije smiselno ta področja nekoliko prilagoditi.

Osnovno načelo prilagoditve je prenos odgovornosti za zagotavljanje kakovosti na nižjo raven, torej iz ustreznih skupin v okviru projekta (ki jih po novem več ni) na posameznega razvijalca. S tem sicer dosežemo bolj razdrobljene postopke zagotavljanja kakovosti, vendar so ti enostavnejši in bolje izvedeni. Zajemajo namreč manjše aktivnosti in izdelke, ki so jih

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

posamezniki sami razvili in jih poznajo. Pri tem uporabimo tudi enostavnejše prijeme, vendar poskrbimo, da so ti natanko določeni. Predvidimo tudi, da vsi razvijalci ustrezno obvladajo postopke tega področja [Sch01].

Tudi na višji ravni si lahko privoščimo enostavnejše preverjanje kakovosti, saj izkoriščamo prednosti majhne organizacije. Kot primer bi navedli ključno področje »Pregledi med kolegi«. V našem primeru organizacija nima uvedenih neposrednih pregledov med kolegi, vendar to nadomešča z drugimi praksami, ki dosežejo enak učinek:

- z vpeljavo ponovnega pregleda za vsako aktivnost ali izdelek v skladu s standardnim pristopom [Pau99, Joh98].
- z vpeljavo neodvisnih zunanjih preverjanj, ki lahko vključujejo (poleg članov SQA) tudi člane iz drugih projektov.

Postopki zagotavljanja kakovosti so definirani in opisani v repozitoriju. Poseben poudarek je namenjen objektivnosti in neodvisnosti izvajalcev preverjanja, zato jih izvajajo člani, ki niso neposredno vpleteni v razvoj. Za projekte so to običajno člani skupin SQA ali SEPG, možna pa je tudi praksa, da pri preverjanjih sodelujejo člani drugih projektov. Pomembna je tudi kontrola ustreznosti preverjanja kakovosti (glej recimo [Gra94a, Oma93]).

Predstavili smo glavna načela prilagoditve modela CMM majhni organizaciji. Več vsebinskih podrobnosti prilagoditve bomo predstavili v nadaljevanju, v katerem si bomo ogledali naš pristop pri razvoju procesa in sam prilagodljiv proces razvoja PO.

3.1.2.8 Primeri prilagoditev modela CMM

Naj ob koncu tega razdelka omenimo še nekaj zanimivih primerov prilagajanja modela CMM majhnim organizacijam. Na North Carolina State University so v okviru razvoja procesa EPRAM predstavili formalno prilagoditev vseh tipičnih praks 2. stopnje modela CMM [Ant01]. LOGOS Tailored CMM je tržno uspešna prilagoditev, ki temelji na podobnem pristopu, kot smo ga ubrali mi. Raziskava švedske univerze Umeå predstavlja dinamični model CMM, ki ga je mogoče prilagajati rasti organizacije [Orc00]. Še več primerov je mogoče spoznati na podlagi citirane literature.

3.1.3 Izbrani pristop

Ogledali smo si obe možnosti za uporabo modela CMM v majhnih organizacijah. Prednost uporabe sorodnih modelov je predvsem v tem, da so natanko določeni in preverjeni v praksi, medtem ko nam druga pot omogoča prilagoditi model našim dejanskim potrebam. Vendar noben soroden model ne pokriva vseh zahtev majhne organizacije, v primeru prilagajanja modela CMM pa je natančna izvedba ustrezne prilagoditve zelo zahtevna.

Zaradi tega smo se odločili za kombiniran pristop. Pri določanju procesa smo tako za enega izmed temeljev uporabili model TSP (in posredno PSP), s čemer smo pokrili pomemben del zahtev modela CMM. Na ta način smo uporabili tudi veliko elegantnih rešitev modela TSP, ki so primerne za majhne organizacije. Za manjkajoči del zahtev modela CMM smo poskrbeli ob oblikovanju našega procesa, ko smo uvedli še opisane prilagoditve iz razdelka 3.1.2.

V naslednjem podpoglavju podrobneje predstavljamo postopek določanja procesa razvoja PO, iz katerega je razvidna izvedba našega pristopa. Podrobne formalne realizacije prilagoditve nismo opisovali, temveč smo se omejili na idejno predstavitev. Podrobnosti lahko razberemo iz opisa prilagodljivega procesa v razdelku 3.4.4 in dodatku A.2.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3.2 Pristop pri oblikovanju procesa razvoja PO

Pri definiranju procesa za razvoj PO moramo izhajati iz namena procesa. Oblikovati želimo proces za razvoj PO, ki organizaciji služi kot izhodišče za določanje procesa vsakega projekta. Ker je organizacija majhna, je smiselno imeti samo en organizacijski proces, vendar mora biti ta zastavljen tako, da ga je mogoče enostavno prilagoditi vsakemu projektu. Naš proces za razvoj PO je določen na dveh izhodiščih:

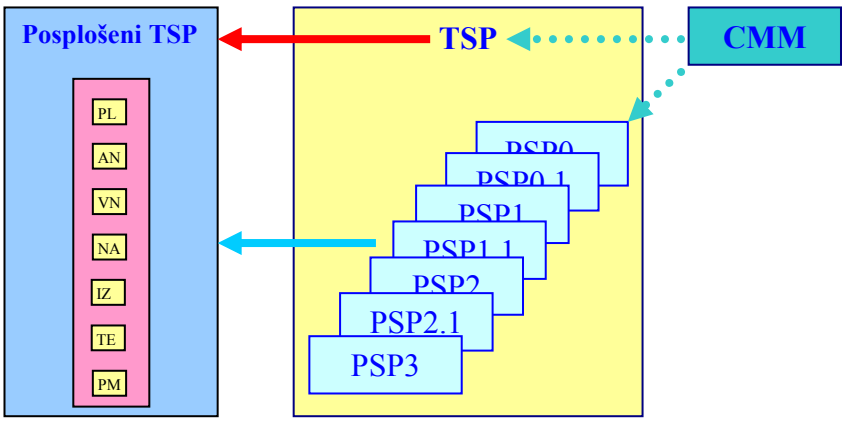
- **na ustrezno definiranem procesu**: ustrezati mora strogim kriterijem za tretjo stopnjo modela CMM in
- **na ustreznem tehnološkem procesu**: prilagojen mora biti majhni organizaciji za razvoj IS, ki temeljijo na uporabi sodobnih sistemov za upravljanje PB.

Na podlagi izhodišč smo najprej zasnovali dve delni rešitvi. Prva delna rešitev temelji na prvem izhodišču, medtem ko druga ustreza zahtevam drugega. Definiranje procesa smo nadaljevali z združevanjem obeh delnih rešitev, nakar smo tako dobljeni proces nadgradili še s tistimi zahtevami prilagojenega modela CMM, ki jih prva delna rešitev ni upoštevala. Zadnji korak na poti do končnega procesa je poenostavitev procesa, ki naj bi tako postal primernejši za majhne organizacije. Pri tem se je pokazala potreba, da v proces vgradimo prilagodljivost, kar zahteva nekaj izvirnih rešitev.

3.2.1 Prva delna rešitev: posplošen model TSP

Osebni proces razvoja PO (glej razdelek 2.2.7) je primerno izhodišče za definicijo osebnega procesa članov projektne skupine. Po zgledu osebnega procesa bo ta na najnižji ravni uvedel nadzorovan proces z merjenjem porabe časa in kakovosti za vse aktivnosti člana. Model PSP bo treba le posplošiti in poenostaviti, saj je trenutno prilagojen drugačni vrsti PO. Pri tem za izhodišče vzamemo model PSP višje verzije (recimo PSP2, PSP2.1 ali PSP3) [Hum95].

Skupinski proces razvoja PO (glej razdelek 2.2.8) je primeren temelj za razvoj našega procesa za majhne organizacije. Je perskriptiven, opisan na primeren način in vsebuje več rešitev, ki so primerne tudi za majhne organizacije. Glede na naše zahteve skupinskemu procesu manjkajo predvsem tisti deli, ki so namenjeni delovanju na organizacijski ravni [Dav02]. Poleg tega TSP ni prilagojen za tako vrsto PO, kot jo mi želimo razvijati. V določenih delih je model tudi preveč zapleten, zato ga je smiselno poenostaviti.



Slika 3.4: Izpeljava posplošenega modela TSP iz modelov PSP in TSP

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Na podlagi TSP zasnujemo posplošeni model TSP, ki ne bo vseboval motečih posebnosti modela TSP in bo primerno izhodišče za nadgradnjo (glej sliko 3.4). To je prva delna rešitev za naš proces majhne organizacije za razvoj IS, ki pa ji manjka še kar nekaj potrebne funkcionalnosti. To bomo dopolnili z uporabo prilagojenega modela CMM. Vendar moramo najprej poskrbeti, da bo naš proces odražal tudi ustrezen tehnološki postopek.

3.2.2 Druga delna rešitev: proces za razvoj informacijskih sistemov

Naša organizacija je namenjena za razvoj IS, ki temeljijo na uporabi sodobnih sistemov za upravljanje s PB in kot taka zahteva ustrezen tehnološki proces. Zato oblikujemo drugo delno rešitev na podlagi teh zahtev.

Prvi korak v tej smeri je zasnova procesa. Ta vsebuje vse potrebne faze razvoja IS razmeščene v kaskadni življenjski cikel [Cha86, Kan95]. Kaskadni življenjski cikel je izbran zaradi svoje enostavnosti, kar je eden od pogojev za definicijo procesa v majhnih organizacijah [Pau99, Joh98]. Vendar lahko že sedaj predvidimo, da bo treba ta življenjski cikel med določanjem procesa nadgraditi. Zasnova procesa obsega naslednje faze:

- strateško načrtovanje,
- zajem zahtev,
- analizo zahtev,
- načrtovanje procesnega dela,
- načrtovanje podatkovnega dela,
- izgradnjo procesnega dela,
- izgradnjo podatkovnega dela,
- testiranje,
- namestitev in
- vzdrževanje.

Ob zasnovi proces še ne določa aktivnosti na nižjih ravneh. Pri oblikovanju teh upoštevamo dobre pristope in rešitve uveljavljenih metodologij, ki so namenjene razvoju IS. V naš proces smo vgradili izbrana načela treh metodologij:

- informacijsko inženirstvo (Information Engineering - IE) [Mar81];
- strukturirano metodologijo za sistemsko analizo in načrtovanje (Structured System Analysis and Design Method - SSADM) [Goo95];
- CASE metodo za razvoj aplikacij (CASE Application Development Method - CADM) [Dor97, Dor97a].

Obstaja še vrsta drugih ustreznih metodologij, vendar smo na začetku iskali predvsem enostavne, učinkovite prijeme, zaradi česar smo izbrali ravno te tri.

Informacijsko inženirstvo (IE) Jamesa Martina je v praksi preizkušena metodolologija, kar je bil razlog za njeno uporabo pri določanju našega procesa. V našem procesu smo uporabili predvsem osnovno idejo opisovanja rezultatov posameznih faz razvoja (analize in načrtovanja) z ustreznimi diagrami [Mar85]. Tako predvidevamo tri različne vrste diagramov: diagrame entiteta razmerje (ERD), diagrame podatkovnih tokov (DFD) in diagrame za opis strukture, med katere sodita recimo diagram funkcijske dekompozicije in kontekstni diagram.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

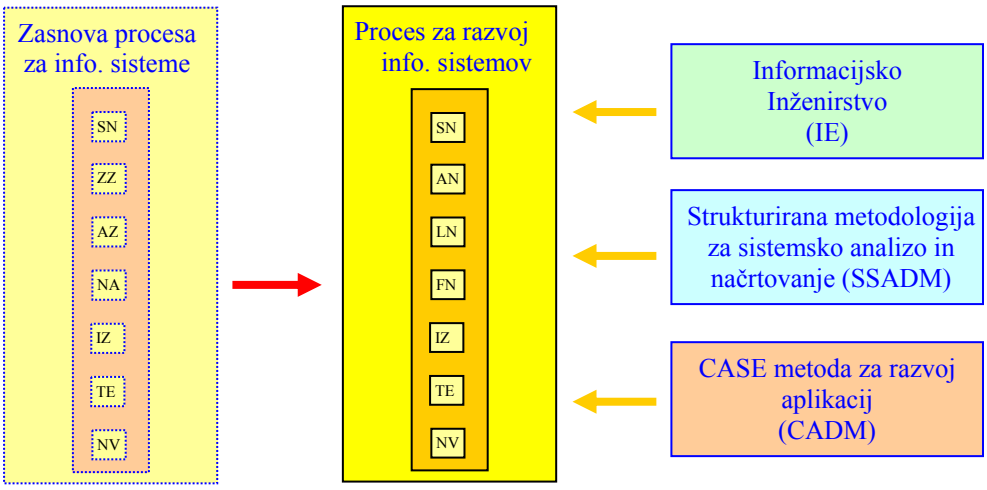
Strukturirana metodogija za sistemsko analizo in načrtovanje (SSADM) je nastala z združitvijo učinkovitih prijemov in rešitev različnih metodologij in je kot taka primeren vir dobrih rešitev za naš proces. Čeprav je namenjena samo delu razvoja (strateškemu načrtovanju, analizi zahtev in načrtovanju), smo uporabili naslednje ideje:

- veliko sodelovanja z naročnikom, predvsem z vidika nudenja možnosti izbire med več alternativami v določenih korakih razvoja;
- navzkrižno preverjanje procesnega in podatkovnega opisa v vseh ključnih fazah;
- delitev načrtovanja na logični in fizični del;
- zbiranje in formalen pregled rezultatov ob koncu vsakega dela razvoja.

CASE metodo za razvoj aplikacij (CADM) smo upoštevali predvsem zaradi njene primernosti za razvoj IS in njenega praktičnega vidika. CADM je namreč podprta z orodjem Oracle[17]. Iz te metodologije smo upoštevali recimo:

- idejo o planiranju vsakega dela razvoja posebej;
- idejo o pripravi standardov za razvoj;
- oblikovanje posameznih rezultatov faz;
- načine izvedbe faz zajema podatkov, testiranja, namestitve in vzdrževanja.

Z upoštevanjem zasnove in dobrih rešitev vseh treh metodologij je nastala druga delna rešitev, ki ustrezno odraža tehnološke značilnosti razvoja IS (slika 3.5). Pri oblikovanju te smo pazili na splošnost in nismo procesa omejili na noben način.



Slika 3.5: Izpeljava tehnološkega vidika procesa na podlagi treh metodologij

3.2.3 Končni proces: združitev delnih rešitev z dopolnitvami

Proces razvoja PO do konca določimo v treh korakih:

- najprej združimo obe delni rešitvi v skupen proces,
- proces nadgradimo, da bo odražal še manjkajoče zahteve CMM in
- proces poenostavimo, da bo primernejši za uporabo v majhni organizaciji.

[17] Oracle je metodo CADM nadgradil z z metodologijo CDM [Dor98], ki omogoča vključevanje različnih življenjskih ciklov. Za naše potrebe zadošča metoda CADM.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3.2.3.1 Združitev delnih rešitev

Obe delni rešitvi najprej združimo v skupni proces, ki bo že dokaj blizu končnemu rezultatu. Pri določanju skupnega procesa upoštevamo vse značilnosti obeh delnih rešitev. Pri tem se pokaže več neskladij, ki jih ni mogoče enostavno rešiti, temveč ta zahtevajo drugačne pristope, recimo spremembo strukture procesa, uvedbo dodatnih aktivnosti ali kaj podobnega. V nadaljevanju predstavljamo tri najpomembnejša neskladja.

- Zahteva nadziranega procesa je natančno planiranje in spremljanje poteka razvoja. Prva delna rešitev to vsebuje, ker predvideva, da je problem ob začetku projekta že znan in je planiranje na podlagi rezultatov preteklih projektov mogoče. Pri drugi delni rešitvi je velik del razvoja posvečen raziskavi zahtev, torej natančno vnaprejšnje planiranje ni mogoče. Lahko sicer planiramo potreben čas za izvedbo analize, ne moremo pa predvideti, kakšne bodo natanko zahteve. Zahteve pa bistveno določajo obseg nadaljnjega dela.

- Posplošeni TSP uvaja natančno planiranje in mersko spremljanje poteka razvoja in kakovosti. Osnova za to je ocena velikosti končnega izdelka s pomočjo podatkov o preteklih projektih. Pri razvoju IS se med razvojem pojavlja tudi več vmesnih izdelkov, ki opisujejo končni izdelek. Za oceno velikosti končnega izdelka je smiselno uporabljati tudi njihovo velikost, kar zahteva spremembo načina ocenjevanja velikosti. Poleg tega merimo sedaj velikost z dveh vidikov: procesnega in podatkovnega.

- Tretje neskladje so aktivnosti uporabe končanih izdelkov. Za realizacijo tega dela procesa bo potrebno uvesti poleg dodatnih faz, ki jih vsebuje druga delna rešitev, tudi dodatne aktivnosti. Ena zahtevnejših je realizacija vzdrževanja, ki zahteva izvajanje aktivnosti tudi tedaj, ko je projekt formalno že zaključen.

3.2.3.2 Vključitev manjkajočega dela prilagojenega modela CMM

Prva delna rešitev že vsebuje veliko elementov prilagojenega modela CMM, ki so se ob združevanju prenesli tudi v skupni proces. Vendar v tem procesu še vedno manjka organizacijski vidik razvoja PO. Ta zahteva vgraditev dodatnih aktivnosti, ki pa večinoma ne bodo vgrajene neposredno v organizacijski proces, temveč bodo zajete v načinu delovanja organizacije in njeni organizacijski strukturi. Zadovoljili se bomo z opisom tistih vidikov organizacije, ki se jih naša problematika neposredno dotika.

Ker je naš proces namenjen majhni organizaciji, smo uporabili zahteve prilagojenega modela CMM iz razdelka 3.1.2. Proces smo dopolnili na naslednji način:

- določili smo način delovanja organizacije v tistem delu, ki ga ne določa prilagojeni model CMM (recimo strategija delovanja);
- definirali smo skupni repozitorij za vse podatke organizacije in projektov;
- definirali smo centralizirano zbiranje merskih podatkov vseh aktivnosti v repozitoriju;
- aktivnosti projekta smo prilagodili aktivnostim organizacije, pri čemer smo določene aktivnosti prenesli na raven organizacije;
- vloge pri izvajanju projekta smo uskladili z vlogami in skupinami organizacije.

Dopolnjeni proces ustreza zastavljenim zahtevam, vendar je dokaj zapleten in obsežen. Smiselno je proces poenostaviti ali posplošiti, kar smo storili v zadnjem koraku.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3.2.3.3 Poenostavitev procesa za uporabo v majhni organizaciji

Ideja o poenostavljanju procesa je utemeljena z omejenimi viri organizacije, vendar je treba najprej presoditi, kateri so tisti deli procesa, ki bi jih lahko poenostavili.

Zaradi majhnosti organizacije smo se odločili, da bo v organizaciji samo en organizacijski proces razvoja. Ta mora vsebovati vse, kar organizacija za razvoj IS potrebuje za izvajanje kakršnegakoli projekta. To pomeni, da moramo predvideti tudi izvajanje zahtevnih projektov, kar narekuje natančen in tehnološko zapleten proces. Pri takem projektu lahko sodeluje (s stališča majhne organizacije) velika projektna skupina, ki zahteva vse predvidene vloge. Predstavljen proces to omogoča in s tega stališča nima odvečnosti.

Poleg tega smo želeli razviti proces, ki bo na ravni upravljanega procesa CMM. Upoštevali smo dobre rešitve skupinskega procesa razvoja in ob dograjevanju procesa smo zahteve CMM smiselno prilagajali majhni organizaciji. Aktivnosti, ki so bile vgrajene v proces zaradi tega vzroka očitno tudi ne smemo opuščati.

Vendar večina resničnih projektov v majhni organizaciji ne bo tako zahtevnih. Prilagajanje organizacijskega procesa konkretnemu projektu je torej tista aktivnost, s katero lahko naš zapleteni proces poenostavimo. Organizacijski proces naj bo v izhodišču zastavljen širše, vendar naj vsebuje elemente, ki omogočajo učinkovito prilagajanje konkretnemu problemu. Ob prilagajanju se lahko poenostavi tako tehnološki kot projektni vidik razvoja, potreba po izvajalcih in pričakovani rezultati. Bistveno je, da postopek prilagajanja vedno ohranja smiselne zahteve upravljanega procesa (v skladu z zahtevami prilagojenega modela CMM) v tistem delu razvoja, ki se zares izvaja.

Čeprav smo s tem proces razvoja PO za majhne organizacije vsebinsko dorekli, potrebujemo spremembo doslej predstavljene strukture procesa. Razlog za to so že neskladja, ki smo jih odkrili ob združevanju obeh delnih rešitev v razdelku 3.2.3.1. Tudi vpeljava učinkovitega prilagajanja zahteva spremembe v strukturi procesa. V zadnjem koraku definicije našega procesa razvoja zato uporabimo sveže ideje in definiramo spremenjeno strukturo procesa, ki oboje upošteva.

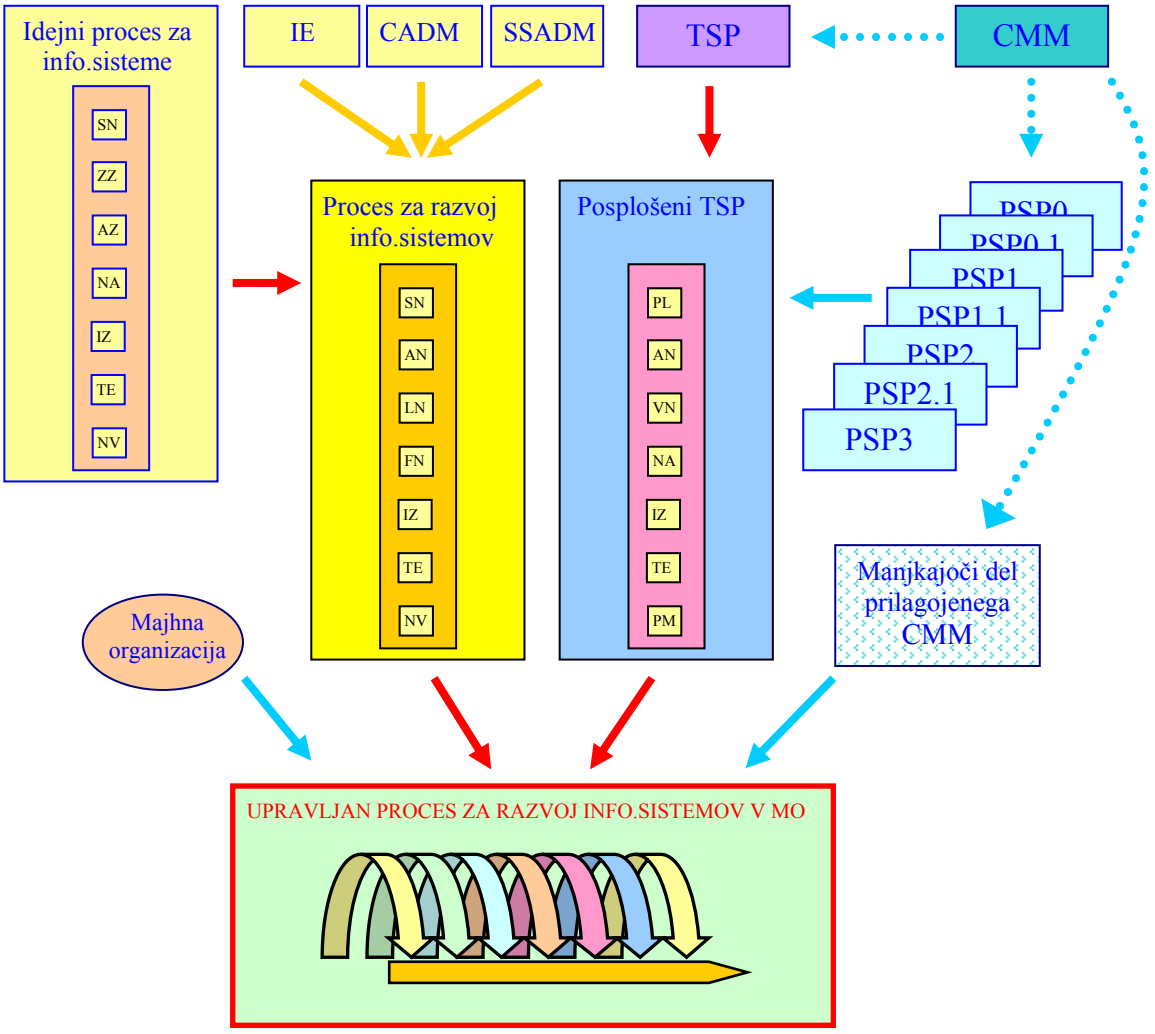
Nastali proces imenujemo **prilagodljiv proces razvoja PO**. To je preskriptiven proces, ki je namenjen razvoju IS v majhni organizaciji in z vidika majhnih organizacij dosega zahteve četrte stopnje modela stopenj zrelosti. Proces bo podrobneje predstavljen v naslednjem podpoglavju. Slika 3.6 predstavlja celoten postopek izpeljave prilagodljivega procesa razvoja PO: izpeljavo obeh delnih rešitev, združevanje teh v skupni proces, vključitev manjkajočega dela in poenostavitev.

3.3 Prilagodljiv proces razvoja PO

Prilagodljivost procesa razvoja PO je izvedena s tremi pristopi:

- uporabo večcikličnega procesa,
- uporabo prilagodljivih korakov in
- natančno definiranimi pravili prilagajanja procesa.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 3.6: Celoten postopek izpeljave prilagodljivega procesa razvoja PO

3.3.1 Večcikličen proces

Procese običajno definiramo tako, da celoten procesa razbijemo na več faz, vsako fazo pa naprej na posamezne korake (glej razdelek 2.1.3). Pri tem nam faza predstavlja zaključeno množico med seboj povezanih aktivnosti, ki dajejo določen rezultat. Korak je pomensko jasno definiran del izvedbe faze. Z vidika vsebinskega opisa procesa predstavlja najmanjšo aktivnost in ima natanko določenega izvajalca, ki je za izvedbo koraka odgovoren.

V našem primeru tako strukturo procesa nadgradimo še s tretjo ravnjo: cikli[18]. Celoten proces je tako sestavljen iz:

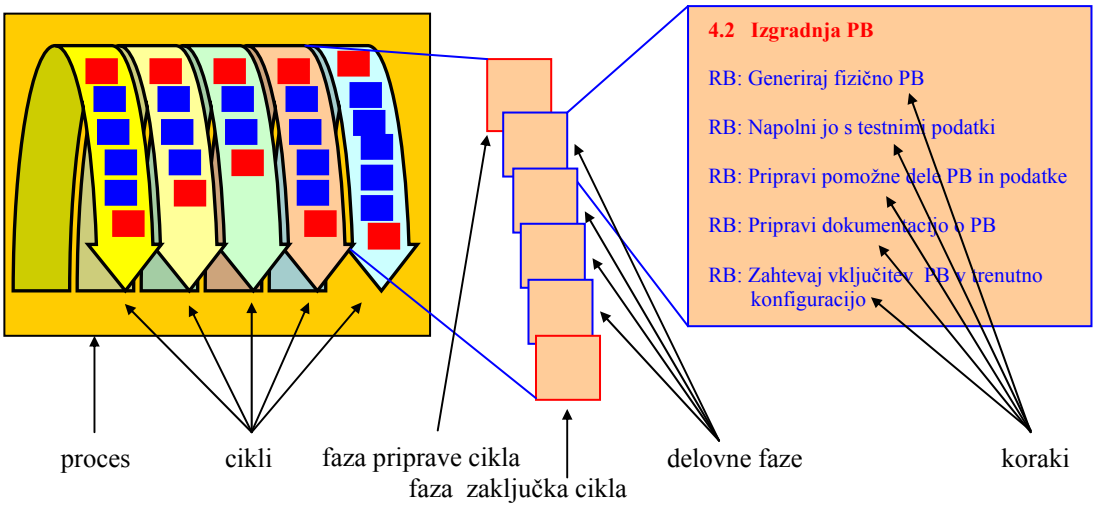
- več (sedmih) ciklov,
- vsak cikel iz več faz in
- vsaka faza iz več korakov.

[18] Cikle smo spoznali že pri modelu TSP (razdelek 2.2.8), vendar so v našem procesu drugače izvedeni. Poleg tega imajo v našem procesu drugačno vlogo; v modelu TSP recimo nadomeščajo faze iz običajne terminologije opisa procesa, v našem primeru pa predstavljajo višjo raven opisa procesa.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Cikel združuje več faz v pomensko zaokrožen del procesa s pomembnimi rezultati (recimo testiranje). Glavna značilnost ciklov sta *fazi priprave in zaključka cikla.* Faza *priprave cikla* se izvede ob vsakem začetku cikla, medtem ko se faza *zaključka cikla* izvede ob vsakem zaključku. To omogoča ciklično izvajanje od faze priprave prvega do faze priprave naslednjega cikla oziroma od faze zaključka prvega do faze zaključka naslednjega cikla. Zato ju imenujemo *ponovljivi fazi*.

Ostale faze imenujemo *delovne faze* in določajo tehnološki vidik procesa. Delovne faze se izvajajo zaporedno ena za drugo, vsaka pa pripada samo enemu ciklu. Cikli obsegajo različno število delovnih faz (med 4 in 8). Slika 3.7 prikazuje trinivojsko strukturo prilagodljivega procesa.



Slika 3.7: Trinivojska struktura prilagodljivega procesa razvoja PO

Trinivojska struktura nam poleg cikličnega izvajanja daje tudi dodatno funkcionalnost:

- proces je natančneje in bolj sistematično določen;
- faze so krajše in obsegajo manj korakov, kar ima pozitiven učinek na prilagodljivost;
- ker ima vsaka faza določen rezultat, so rezultati natančneje določeni.

Čeprav se celoten razvoj izvaja v ciklih, pa se delovne faze izvajajo v točno določenem zaporedju, s čemer ohranjamo določenost procesa in ustrezen vrstni red razvoja. Ponovljive faze samo razdelijo celoten razvoj na obvladljive dele, uvedejo dodatno funkcionalnost v proces in omogočajo večjo prilagodljivost.

Slaba stran uvedbe ciklov je kompleksnejša sestava procesa. To lahko vpliva na uporabnost procesa, vendar je v našem primeru cikličen proces potreben zaradi raznolikosti uporabe. V organizaciji namreč definiramo samo en proces, ki ga uporabljamo pri vseh projektih in mora vsebovati vse potrebne dele za različne načine uporabe. Načini prilagajanja omogočajo, da ga zelo enostavno poenostavimo, pri tem pa igra pomembno vlogo ravno trinivojska struktura.

Z vidika naše problematike je uvedba večcikličnega procesa potrebna tudi zaradi značilnosti razvoja IS in zahtev upravljanega procesa. Pomemben del razvoja IS namreč predstavlja raziskava zahtev, ki lahko traja celo večji del celotnega razvoja. Dokler popolnoma ne poznamo zahtev, ne moremo natančno predvideti obsega vseh aktivnosti pri razvoju končnega izdelka. Če želimo ob tem planirati obseg dela in pričakovane rezultate, nam preostane dvoje:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- lahko planiramo takoj na začetku razvoja na podlagi površnega poznavanja zahtev in izkušenj izvedenih projektov ali
- lahko prvo izvedemo raziskavo (zajem in analizo) zahtev in nato natančno planiramo.

Rezultat prvega pristopa bodo verjetno nenatančni plani, medtem ko bomo po drugi poti celotno raziskavo zahtev izvedli brez planov. Oboje je v nasprotju z zahtevami modela CMM, ki predvidevajo natančno planiranje vseh aktivnosti razvoja.

Večciklični pristop to težavo rešuje. Celoten razvoj je ustrezno razporejen po ciklih, od katerih ima vsak cikel na začetku ponovljivo fazo »Priprava cikla« in na koncu fazo »Zaključek cikla«. »Priprava cikla« med drugim vedno izvede tudi natančno planiranje aktivnosti cikla. Tako lahko na začetku prvega cikla določimo plan za izvedbo raziskave zahtev, celoten plan pa zgolj približno, ker še ni znanih zahtev za celotno planiranje. Ker raziskava zahtev zajema vse aktivnosti drugega cikla, sledi pred nadaljevanjem razvoja faza priprave naslednjega cikla. Ob izvedbi te faze že natanko poznamo zahteve in lahko pravilno planiramo potek dela za naprej. Večciklični pristop ima še nekaj pozitivnih učinkov:

- Pristop omogoča, da poleg raziskave zahtev v razvoj vključimo še druge dele razvoja, ki jih ne moremo predvideti vnaprej. To sta recimo odločanje naročnika o obsegu rešitve ali izbira najprimernejše tehnične rešitve med razvojem.
- Cikli obsegajo tak del razvoja, ki ga lahko natančno ocenjujemo in zanj planiramo delo. Tudi če bi bile zahteve že na začetku popolnoma znane, bi brez ciklov težko tako natančno planirali potek zelo dolgega projekta.
- Pristop omogoča učinkovito prilagajanje novim zahtevam (s strani naročnika ali zaradi novih ugotovitev) ob začetku ciklov. Tako lahko spremenimo plane, poskrbimo za dodatne vire ali se ustrezno drugače sproti prilagodimo novim dejstvom.
- Cikli omogočajo tudi natančnejšo analizo opravljenega dela, poročanje vodstvu organizacije in bolj ažurno ukrepanje ob ugotovljenih nepravilnostih.
- Ob koncu ciklov je primeren čas, da dodatno in dokončno (po potrebi z naročnikom) preverimo vse rezultate cikla. Če preverjanje uspe, ta del razvoja zaključimo in ga več ne ponavljamo. V nasprotnem primeru pred nadaljevanjem odpravimo težave.

Cikli imajo še eno prednost: v primeru, da določen del razvoja ni potreben, lahko nekatere (ne vsakega) cikle brez škode v celoti izpustimo. Na ta način lahko zelo hitro prilagodimo proces enostavnejšemu problemu. Tak primer je recimo izpuščanje cikla »Raziskava zahtev«, če so znane vse zahteve že na začetku projekta.

V bistvu lahko na enak način izpustimo tudi nekatere faze v posameznih ciklih, če jih reševanje konkretnega problema ne potrebuje. Za izpuščanje faz zadostuje, da vse takšne postopke natanko planiramo ob začetku cikla.

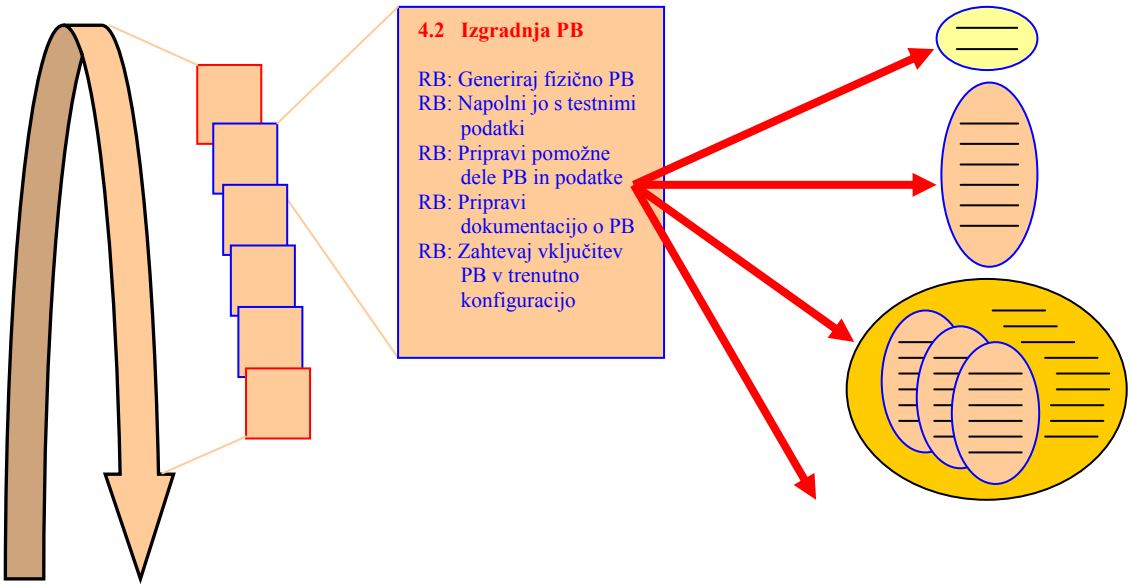
3.3.2 Prilagodljivi koraki

V organizaciji za razvoj IS v splošnem ne moremo vnaprej (preden poznamo problem) napovedati, kako zahtevna bo izvedba korakov posameznih faz. Za nekatere sicer lahko to predvidimo (recimo za korak »Poročanje vodstvu organizacije«, ki se izvaja vedno na enak način), vendar je zahtevnost večine odvisna od dejanskih zahtev. Lahko se zgodi, da bo izvedba določenega koraka zahtevala zelo velik obseg dela ali pa bomo korak kar izpustili. Ta razmislek je podlaga za drugi pristop: prilagodljive korake.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Za vsak korak predvidimo, da se lahko izvede na enega izmed štirih načinov (slika 3.8):

a) kot enostaven korak (posameznika),
b) kot zahtevnejši korak (posameznika),
c) kot zahteven korak (več članov) ali
d) koraka ni treba izvesti.



Slika 3.8: Štirje načini izvedbe koraka v prilagodljivem procesu razvoja PO

Enostaven korak (posameznika) je enostavna, enkratna aktivnost enega posameznika, ki ne potrebuje planiranja in načrtovanja rešitve, temveč zgolj izvedbo ustreznega zaporedja elementarnih aktivnosti[19]. Tipičen primer enostavnega koraka je poročanje vodstvu organizacije. Izvedba enostavnega koraka obsega šest delov:

- ocenitev velikosti rešitve, časa izvedbe, stroškov in števila napak (če je to smiselno);
- izvedbo aktivnosti v skladu s standardnim postopkom in sprotno merjenje časa;
- ponovni pregled v glede na običajne težave in z merjenjem časa in števila napak (če je to smiselno);
- odpravo najdenih napak z merjenjem časa ter števila napak (če je to potrebno);
- ugotovitev velikosti rešitve, porabe časa, stroškov in števila napak (če je to smiselno);
- oddajo rešitve.

Enostaven korak je predstavljen v dodatku A.2.10.

Zahtevnejši korak (posameznika) je obsežnejša aktivnost posameznika, ki obsega raziskavo problema, delitev na podprobleme (po potrebi), planiranje izvedbe, načrtovanje rešitve, samo izvedbo in na koncu še testiranje. Primer zahtevnejšega koraka je recimo podrobno planiranje opravil cikla ali izvedba testiranja dela IS v skladu s standardnim postopkom. V primerjavi z enostavnim korakom je treba dodatno izvesti še:

[19] Termin aktivnost uporabljamo splošno: na različnih ravneh nam aktivnosti predstavljajo lahko cikle, faze, korake ali dele korakov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- analizo problema, da se ga popolnoma razjasni;
- določiti natančnejše ocene velikosti, porabe časa, stroškov in napak, ki obsegajo še območje zaupanja;
- če rešitev naloge zahteva izvedbo več opravil, moramo ta natanko planirati po delih;
- rešitev je treba načrtovati, načrt pa ponovno pregledati;
- sledi izgradnja ali izvedba naloge po standardnem postopku, pri čemer rešitev ponovno pregledamo in odpravimo najdene napake;
- rešitev dodatno še testiramo in odpravimo odkrite napake, če je to smiselno;
- med izvajanjem vseh aktivnosti merimo porabljen čas in število napak;
- ugotovimo skupno velikost rešitve, porabo časa po fazah in število najdenih napak;
- rešitvi priložimo še predloge za izboljšavo procesa ali postopka.

Zahtevnejši korak je predstavljen tudi v dodatku A.2.11.

Zahteven korak (več članov) je aktivnost, ki zahteva sodelovanje več razvijalcev. Pri tem je treba poskrbeti predvsem za delitev problema na podprobleme in pazljivo sestavljanje delnih rešitev. To zahteva naslednje aktivnosti:

- temeljito raziskavo problema;
- dogovor o strategiji reševanja in pravilno razbitje problema na več podproblemov;
- oceno skupne velikosti in porabe časa po fazah skupaj z območjem zaupanja;
- zasnovo končne rešitve;
- **izvedbo reševanja po delih, pri čemer vsak član rešuje svoj podproblem na enak način kot bi izvajal zahtevnejši ali enostaven korak posameznika;**
- sestavo delnih rešitev s temeljitim preverjanjem;
- po potrebi dodatno testiranje celovite rešitve z odpravljanjem napak;
- med izvajanjem sprotno merjenje porabe časa tudi za skupne aktivnosti;
- zbiranje poročil o porabi časa in napakah vseh sodelujočih članov;
- natančno ugotavljanje skupne velikosti izdelka, porabe časa, stroškov in napak;
- oddajo rešitve in predlogov za izboljšavo.

Tak način izvedbe je primeren za različne korake, vendar moramo smiselno uporabiti opisani postopek. Če v skladu z zahtevnim korakom izvajamo recimo »Test arhiviranja in obnove PB«, si več članov razdeli aktivnosti pri testiranju, skupen končni rezultat pa je poročilo o testiranju. Zahtevni koraki so pogosti predvsem pri fazah načrtovanja in izgradnje, drugod pa jih uporabimo po potrebi. Opis izvedbe zahtevnega koraka za več članov je v dodatku A.2.12.

Z uporabo prilagodljivih korakov dosežemo veliko raztegljivost procesa. Najenostavnejši proces tako lahko obsega samo nekaj nujnih aktivnosti – recimo začetek projekta (planiranje), izvedbo in zaključek projekta (analizo in končno poročilo), pa še te izvaja enočlanska projektna skupina kot enostavne korake. V nasprotju s tem pa obsežen projekt zahteva veliko projektno skupino in traja relativno dolgo. Tak projekt bi izvajal večino korakov opisanega procesa, med katerimi bi številne korake kot nekakšne podprocese izvajalo več članov projektne skupine skupaj. Lahko bi se celo zgodilo, da bi bili deli podprocesa tako zahtevni, da bi zahtevali sodelovanje več članov.

Pristop s prilagodljivimi koraki je uporaben tudi za definiranje postopkov merskega spremljanja vseh aktivnosti pri razvoju projekta in tudi v organizaciji nasploh. Ker v vse tri načine izvedbe korakov vključimo spremljanje porabe časa, stroškov in števila napak ter

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

zahtevamo, da se vse aktivnosti v organizaciji izvajajo na enega izmed teh načinov, smo s tem dejansko uvedli učinkovit nadzor nad izvajanjem vseh aktivnosti. Tak pristop spominja na zahtevo modela TSP, da vsi razvijalci uporabljajo osebni proces razvoja, vendar je v našem primeru nadgrajen s prilagodljivimi koraki.

S prilagodljivimi koraki uspešno uvajamo tudi druge dobre prakse na nižje ravni izvajanja aktivnosti (glej razdelek 3.1.2.9). Tak primer so recimo aktivnosti za sprotno vgrajevanje kakovosti v izdelke. Definicija korakov zahteva, da tudi najenostavnejša izvedba koraka zahteva reševanje problema v skladu s standardnim postopkom. Poleg tega se vsak rezultat ponovno pregleda v skladu s primernimi standardi za pregled, ki usmerjajo pozornost na tipične težave rešitve. Oba pristopa zelo povečata kakovost rešitev [Hum95, Hum00a].

3.3.3 Pravila prilagajanja procesa problemu

Zadnji pristop prilagodljivega procesa je definicija pravil za učinkovito uporabo večcikličnega procesa razvoja in prilagodljivih korakov. Izhodišči za določanje pravil sta naslednji:

- **Nepredvidljivost**: Narava razvoja IS narekuje proces, v katerem veliko časa porabimo za ugotavljanje, kaj sploh moramo razviti. To pomeni, da na začetku ne moremo natanko vedeti, koliko dela nas pravzaprav čaka.
- **Določenost**: Za zagotavljanje kakovostnega razvoja izdelka mora biti proces natanko določen. Brez tega je nemogoče planirati in nadzirati izvajanje posameznih ciklov, faz in korakov.

Izhodišči sta si nasprotujoči: po eni strani ugotavljamo, da elementov procesa ne moremo določiti vnaprej, po drugi pa želimo imeti čim natančneje definiran proces. Pravila morajo ubrati tako pot, da zadostijo obema.

Nepredvidljivost zmanjšujemo s postopnim zajemom zahtev, ki omogoča, da po vsaki izvedeni aktivnosti zajema natančneje opredelimo problem. Poleg tega aktivno vključimo naročnika v postopek raziskovanja zahtev, kjer je to mogoče. Izvedba z več cikli pomaga pri prilagajanju razvoja rešitev trenutnemu poznavanju zahtev.

Določenost lahko ohranimo tako, da natanko opredelimo postopek prilagajanja procesa konkretnemu problemu. Ta postopek obsega naslednje korake:

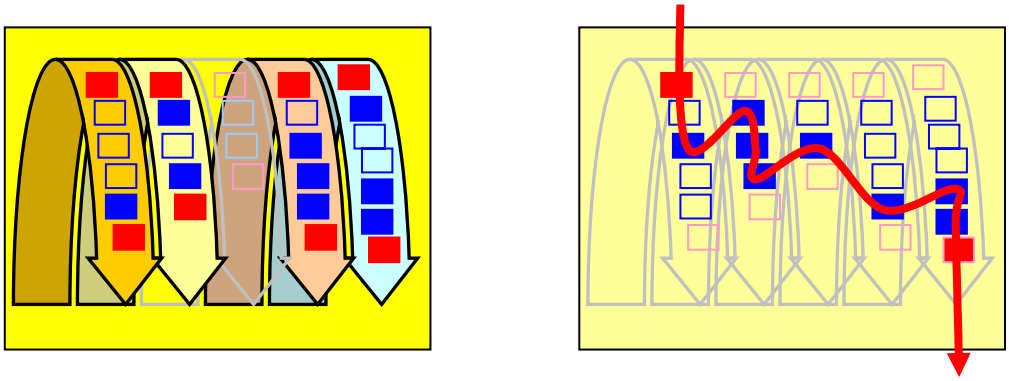
- določitev načina uporabe procesa;
- opredelitev nujnih (nepogrešljivih) aktivnosti vsakega razvoja;
- v primeru izvedbe v več ciklih, opredelitev nujnih delov vsakega izvajajočega cikla;
- opredelitev nujnih delov vsake izvajajoče faze;
- opredelitev načina izvedbe izvajajočih korakov;
- upoštevanje pravil za pravočasno planiranje.

3.3.3.1 Določitev načina uporabe procesa

Prvi korak je odločitev o načinu uporabe procesa. Možnosti sta dve (slika 3.9):

- izvedba razvoja v več ciklih in
- izvedba razvoja v enem ciklu.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Izvedba razvoja v več ciklih Izvedba razvoja v enem ciklu

Slika 3.9: Dva načina uporabe prilagodljivega procesa

Izvedba razvoja v več ciklih je običajen način uporabe procesa. Zanj se bomo odločili, če na začetku poznamo zgolj okvirne zahteve ali pa gre za obsežnejši projekt. To pomeni, da bodo pri planiranju ob začetku projekta določeni globalni plani, vsak cikel pa bo imel tudi pripravo (z natančnim planiranjem) in zaključek cikla (z zbiranjem rezultatov, analizo in poročanjem). Pri tem ni nujno, da se planira izvedba vseh ciklov, temveč vsaj treh: ciklov »Začetek projekta«, »Izgradnja« in »Zaključek projekta«.

Izvedba v enem ciklu je druga možnost. V tem primeru je že na začetku mogoče razbrati, da gre bodisi za enostaven, kratek projekt ali (običajno kratek) projekt, pri katerem je problem natanko opredeljen. To pomeni, da izvedbe ne bo treba deliti na cikle, temveč bomo vse razvojne aktivnosti izvedli z enim samim natančnim planiranjem na začetku in eno celovito analizo poteka na koncu. Pri tem se lahko zgodi, da izvedemo vse delovne faze opisanega procesa, ne smemo pa izvajati ponovljivih faz z izjemo prve »Priprave cikla« in zadnjega »Zaključka cikla«. Lahko se celo zgodi, da gre za dokaj zahtevno in dolgotrajno izvedbo, pri kateri pa so bile vse zahteve in odločitve o obsegu rešitve, tehničnih rešitvah in implementaciji natanko znane že na začetku. Veliko pogostejši bodo drugačni primeri, ko bo razvoj obsegal le nekaj faz.

3.3.3.2 Opredelitev nujnih delov vsakega projekta

Pri obeh načinih izvedbe procesa je nujno, da razvoj obsega vsaj tri dele:

- »Začetek projekta« z vzpostavitvijo projekta (cilji, vloge), globalnim planiranjem opravil in porabe časa ter poročanjem vodstvu;
- izvedbo vsaj ene aktivnosti na predpisan način (sicer projekt nima smisla);
- »Zaključek projekta« z zbiranjem rezultatov, analizo poteka in končnim poročilom.

Obvezno je tudi razviti izdelek, vse aktivnosti mersko spremljati ter pripraviti nujne dele dokumentacije: plane, opis projekta, opis rezultata in končno poročilo.

Nujno je, da smo pri določanju globalnih planov ob začetku projekta natančni, kolikor smo le lahko. Pri izvedbi v enem ciklu je to očitna zahteva, saj je to sploh edino planiranje. Pri izvedbah v več ciklih pa se ta natančnost nanaša predvsem na skupen obseg opravil in izdelkov posameznih ciklov. Ni nujno, da natanko opredelimo faze in korake vsakega cikla, predvideti pa moramo cikle in (vsaj približno) skupen obseg opravil in porabe časa zanje. Pri planih je smiselno upoštevati tudi ustrezne rezerve zaradi sprememb zahtev ali novih ugotovitev ob začetkih ciklov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3.3.3.3 Opredelitev nujnih delov vsakega izvajanega cikla

Če smo se odločili za izvedbo v več ciklih, moramo poleg predstavljenih delov v vsakem izvajajočem ciklu izvesti še vsaj naslednje:

- »Pripravo cikla« z določitvijo natančnih planov cikla in poročanjem;
- glavno oziroma vsaj eno fazo cikla na predpisan način;
- »Zaključek cikla« z zbiranjem rezultatov, analizo in poročanjem.

Ob koncu cikla moramo obvezno pridobiti rezultat izvedene faze ter vsaj naslednje dele dokumentacije: natančne plane, poročilo o izvedbi cikla ter opis rešitve za izvedeno fazo.

3.3.3.4 Opredelitev nujnih delov vsake izvajane faze

Pri izvedbi vsake planirane faze je nujno izvesti vsaj en korak, ki daje otipljiv rezultat. Pri tem je treba paziti, da je izbrani korak smiseln (ne moremo recimo preveriti določenega rezultata, če ga pred tem nismo izdelali). Lahko pa se korak izvede na bolj enostaven način: če je rezultat enostavnejši in ne vsebuje vseh sestavnih delov, ga preverimo takšnega.

3.3.3.5 Opredelitev načina izvedbe izvajanih korakov

Izvedba planiranih korakov je mogoča na enega od treh že predstavljenih načinov:

- kot enostaven korak enega člana,
- kot bolj zahteven korak enega člana ali
- kot zahteven korak, ki ga izvaja več članov.

3.3.3.6 Upoštevanje pravil za pravočasno planiranje

Ključnega pomena za uporabo opisanih pravil je čas njihove uporabe. Pri izvedbi z enim ciklom ni nobenih dvomov. Faza planiranja je samo ena ob začetku projekta in globalni plani obsegajo tudi podrobne plane za vse člane. Tedaj se moramo odločiti o vsem:

- katere delovne faze bomo izvedli (brez ponovljivih faz),
- katere korake posameznih faz bomo izvedli,
- na kakšen način bomo izvedli posamezne korake in
- katere rezultate pričakujemo.

Pri izvedbi z več cikli poteka planiranje večkrat. Najprej ob začetku projekta določimo globalni plan, pozneje pa še natančne plane ob vsakem začetku cikla. Na začetku razvoja se moramo na podlagi okvirnih zahtev in izkušenj čim bolj natančno odločiti:

- katere cikle bomo izvajali pri razvoju,
- kolikšen bo skupen obseg opravil in poraba časa ciklov (skupaj z rezervo) in
- katere končne rezultate pričakujemo.

Ob začetku vsakega cikla se odločamo naprej:

- katere faze v ciklu bomo izvedli (nujno moramo izvesti pripravo in zaključek cikla),
- katere korake posameznih faz bomo izvedli,
- na kakšen način bomo izvedli posamezne korake in
- katere rezultate faz in cikla pričakujemo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Seveda moramo cikle, faze ter korake izbirati tako, da njihovo zaporedje tvori smiselno celoto. Običajno izpuščamo celotne dele razvoja ali uporabe, ki bodisi obsegajo celotne cikle ali faze. Ker je možnosti zelo veliko, tukaj ni smiselno postavljati formalnih omejitev. Pravilnost izbire lahko preizkusimo z določitvijo sosledja rezultatov izbranih aktivnosti. Določene faze ali koraka ne moremo izvesti, če nismo prej izdelali vseh rezultatov, ki jih ta potrebuje za svojo izvedbo.

Iz tega izhaja pomembna prednost prilagodljivega procesa: **dinamično določanje procesa**. Klasično določanje procesa namreč predvideva oblikovanje procesa projekta iz organizacijskega procesa razvoja PO na podlagi navodil za prilagoditev. Prilagoditev se opravi pred začetkom projekta, nato pa se prilagojeni proces uporablja nespremenjen. Tudi pri nas proces projekta določamo na podlagi organizacijskega procesa, vendar ga na začetku prilagodimo toliko, kolikor informacij o problemu imamo. Ko ob izvajanju ciklov problem spoznavamo bolje, ga postopoma prilagajamo naprej. To omogoča bistveno boljšo prilagojenost in s tem natančnost napovedi vseh meril: velikosti opravil in izdelkov, porabe časa, stroškov in kakovosti.

3.4 Opis prilagodljivega procesa razvoja PO

V podpoglavju 3.3 smo predstavili inovativne rešitve prilagodljivega procesa razvoja PO. V tem podpoglavju bomo proces predstavili tudi vsebinsko do določene ravni podrobnosti. Naš namen je predstaviti primer izvedbe procesa, ki bo pomagal pojasniti morebitne nejasnosti.

V prvem delu podpoglavja je predstavljen način formalnega opisa procesa, ki ga uporabljamo v dodatku A.2. Sledi opis delovanja tipične majhne organizacije za razvoj IS. S tem smo določili okvir, v katerega bo umeščen prilagodljiv proces razvoja PO. V zadnjem delu podpoglavja je vsebinski opis procesa, pri katerem smo se omejili na okviren opis glavnih delov procesa: vlog, aktivnosti in izdelkov.

V dodatku A.2 so glavni deli procesa natančneje predstavljeni še s formalno metodo. Procesa tudi tu nismo predstavili v celoti, saj bi bil opis modela preobsežen[20]. Glede na to, da je proces zasnovan v duhu modelov PSP in TSP, si lahko bralec primere izvedbe nekaterih manjkajočih delov opisa ogleda v [Hum95, Hum00, Hum00a, Hum00b].

3.4.1 Predstavitev formalnega načina opisa procesa

Kot smo ugotovili v razdelku 2.1.3.2, moramo pri opisu procesa predstaviti naslednje dele:

1. Izvedene aktivnosti (cikli, faze in koraki):
   - zaporedje izvajanja aktivnosti,
   - opis namena posameznih aktivnosti in
   - način izvedbe aktivnosti.
2. Delovne izdelke, ki nastanejo ob uporabi procesa v okviru projekta:
   - izdelki, ki so potrebni, da se aktivnosti lahko izvedejo in
   - izdelki, ki so rezultat izvedbe aktivnosti.
3. Udeležence oziroma izvajalce posameznih aktivnosti.

[20] Model TSP [Web99, Hum00b] je predstavljen z več kot 70 navodili, obrazci, zabeležkami in standardi. Podroben opis našega procesa bi zahteval še več dokumentov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Obstaja več načinov za opisovanje procesov. Med znane pristope sodijo recimo tehnika strukturiranega opisa analize in načrtovanja (ang. Structured Analysis and Design Technique - SADT) [Roz00], pristop ETVX (ang. Entity-Task-Validation-eXit) [Roz00], diagrami podatkovnih tokov (DFD) [Mar85], odločitvena drevesa, mreže aktivnosti in podobno. Pristopi običajno uporabljajo različne diagramske tehnike za ponazoritev sestave in zaporedja izvajanja posameznih aktivnosti procesa. Za predstavitev ostalih elementov je običajen kratek in jedrnat tekstovni opis.

Opisi procesov so namenjeni predvsem razvijalcem. S tega stališča je pomembna tudi enostavnost in uporabnost predstavitve. Pristop Information mapping [Hor90] svetuje, da za predstavitev procesa uporabimo čim enostavnejšo tehniko, da ne bi že z uporabo zahtevne tehnike prispevali k slabšemu razumevanju opisa. S tega stališča lahko za manjše procese uporabljamo enostavne tehnike, za zapletene pa so primerne bolj izpopolnjene, ki omogočajo več. Enostaven opis procesa uporablja tudi Humphrey pri opisu modelov PSP in TSP [Hum95, Hum00]. Procese opisuje s štirimi dokumenti (navodili, obrazci, zabeležkami in standardi), pri čemer je ključni del opisa navodilo, ostali pa opis dopolnjujejo (glej razdelek 2.2.7). Aktivnosti opiše v tabelah, ki so lahko na različnih ravneh. Tak način je berljiv in lahko prenosljiv.

Za podoben pristop smo se odločili tudi mi. Prilagodljiv proces je v dodatku A.2 formalno opisan z navodili, ki omogočajo opis v skladu z zahtevami iz razdelka 2.1.3.2. Iz navodil je razvidna sestava in zaporedje aktivnosti, njihovi izvajalci in rezultati. Pomen posameznih aktivnosti je razviden iz samega poimenovanja.

Navodila so v našem primeru določena na treh ravneh, saj za naše zahteve takšna stopnja podrobnosti zadošča. V prvem navodilu, ki je edino na najvišji ravni, so predstavljeni cikli in faze, ki sestavljajo celoten proces. Sledi osem navodil, ki predstavljajo vsak cikel posebej ter dodatne aktivnosti. Iz teh navodil so razvidne faze, koraki ter rezultati vseh faz in ciklov. Ob koncu so navedena še splošna navodila z opisom enostavnega, zahtevnejšega in zahtevnega koraka (glej razdelek 3.3.2). Primer tabele z opisom navodila je predstavljen v tabeli 3.1.

Proces:		
Namen:		
Faza:	**Ime faze:**	**Korak:**
	Rezultati:	
	Rezultati:	
Izhodni pogoji:		

Tabela 3.1: Tabela za opis navodil prilagodljivega procesa

Navodila aktivnosti ustrezno definirajo proces. Za popolno predstavitev bi morali določiti navodila še za posamezne vloge, s čemer bi aktivnosti prikazali še z drugega vidika. Mi smo se zadovoljili s preprostejšim opisom vlog in njihovih glavnih zadolžitev v dodatku A.1. Zaradi obsežnosti smo se izognili tudi predstavitvi različnih vrst obrazcev, zabeležk in standardov, smo pa dodatku A.3 prikazali vse rezultate procesa.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3.4.2 Delovanje majhne organizacije za razvoj informacijskih sistemov

Doslej smo na dveh mestih (v podpoglavju 2.3 in razdelku 3.1.2) opisali tipične značilnosti majhne organizacije za razvoj IS, vendar nikjer nismo celovito določili, kako si delovanje tipične majhne organizacije predstavljamo. To predstavo potrebujemo, če želimo vsebinsko predstaviti proces razvoja PO. V organizaciji namreč potekajo še druge aktivnosti, ki neposredno ne sodijo v proces razvoja PO, vendar nanj pomembno vplivajo. Zato bomo v nadaljevanju dopolnili našo predstavo o tipični majhni organizaciji za razvoj IS s področji, ki doslej niso bila pojasnjena.

Tipična organizacija je določena tako, da ustreza našim izhodiščnim zahtevam. Njeno delovanje je tako v veliki meri odraz zahtev 4. stopnje modela CMM, ki smo ga na več področjih (pri formalizaciji postopkov, dokumentiranju, usposabljanju, vlogah) prilagodili majhnim organizacijam. V tem delu bo predstavitev povzemala zahteve prilagojenega modela CMM iz razdelka 3.1.2 in jih ustrezno implementirala. Poleg tega bo v opis delovanja vpletenih tudi več idejnih rešitev, ki služijo za boljšo predstavo.

3.4.2.1 Organizacijska struktura majhne organizacije

V skladu z ugotovitvami razdelka 2.3.1 ima tipična majhna organizacija nekaj deset zaposlenih. Ker gre za organizacijo za razvoj PO predvidevamo, da imajo vsi zaposleni (z izjemo administracije) znanja s področja razvoja IS. To velja tudi za vodstvo organizacije, čeprav se to z razvojem PO neposredno ne ukvarja.

Vsi člani (razen administracije) sodelujejo pri projektih razvoja PO. Število zaposlenih v organizaciji je premajhno, da bi jih lahko delili po specifičnih znanjih. Zato v splošnem velja, da lahko vsi opravljajo vsako delo. Seveda vsi člani niso enako izkušeni, vendar pri razvoju sodelujejo in skrbijo za prenos znanj.

Organizacija izvaja svojo dejavnost s projektnim delom, pri čemer uporablja prilagodljiv proces razvoja PO. Pri projektih vedno določimo vloge, ki smo jih predstavili v okviru prilagoditev modela CMM. Kot rečeno posebnih skupin na ravni projektov ne definiramo, vendar naloge na določenem področju (običajno) izvaja več članov pod vodstvom ustreznega vodje področja. Šele med pripravo ciklov planiramo, kateri člani bodo pri določeni aktivnosti sodelovali. Tako se lahko sproti prilagajamo zahtevnosti opravil in bolj uravnoteženo obremenimo člane projektne skupine. Običajno za izvajanje opravil določimo nekaj bolj in nekaj manj izkušenih članov, da tako poskrbimo tudi za prenos znanja med člani. Člani projektne skupine so pogosto zadolženi za več smiselno izbranih vlog. Projektni vodja običajno ne sme opravljati več vlog.

3.4.2.2 Repozitorij

Predvidevamo, da v organizaciji obstaja centralni repozitorij, ki vsebuje različne vrste podatkov o delovanju organizacije. Repozitorij je podprt z IS, ki omogoča dodajanje, pregled vzdrževanje vseh podatkov ter generiranje različnih vrst poročil. V repozitoriju obstaja sistem uporabniških imen in gesel. Pomembnejši deli repozitorija so:

- Opis standardnih postopkov;
- Plan aktivnosti;
- Podatki o članih;

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Proces razvoja PO na ravni organizacije:
  - organizacijski proces razvoja PO,
  - pravila prilagajanja organizacijskega procesa projektom.
- Podatki o merjenju na ravni organizacije:
  - organizacijski merski proces,
  - organizacijski meta model (podatki z opisom merjenja):
    - merska izhodišča,
    - meritve,
    - pokazatelji.
- Podatki o projektih:
  - proces razvoja PO na ravni projekta,
  - plani (globalni in plani ciklov),
  - podatki o konfiguracijah rezultatov in opisa razvoja (lokacije, opis),
  - opis razvoja (specifikacije),
  - podatki o merjenju na ravni projekta:
    - projektni merski proces,
    - projektni meta model,
    - merski podatki (izmerjeni in določeni, rezultati analiz).
  - vsa poročila.
- Dokumenti (opis uporabe, tipični primeri, opis orodij...);
- Predloge obrazcev;
- Učna gradiva.

Repozitorij se uporablja tudi za pregled statusa aktivnosti ter za preverjanje kakovosti, aktivnosti ali rezultatov. Ta način uporablja vodstvo in skupina SQA za občasne preglede.

Repozitorij predpisuje oddajo rezultatov aktivnosti v elektronski obliki. Pri tem je za določene člane možna samo oddaja, ne pa spreminjanje ali dopolnjevanje.

3.4.2.3 Izvajanje aktivnosti

Delovanje organizacije je natanko predpisano **s standardnimi postopki**. Ti so opisani s strnjenim zaporedjem elementarnih korakov v obliki navodil in se nahajajo v posebnem delu repozitorija. Vsi člani morajo delovati v skladu s temi postopki, pri čemer velja, da so postopki poenostavljeni, prilagojeni potrebam organizacije in (predvsem) učinkoviti. Za vse aktivnosti, ki so definirane s postopki, obstajajo tudi potrebna sredstva in viri za izvajanje. V praksi obstaja ustaljen način dela, ki ustreza postopku, standarden postopek pa to samo zelo natančno definira in omogoča preverjanje kakovosti izvajanja.

Način delovanja organizacije določa, da je za izvedbo vsake aktivnosti (na ravni organizacije ali projekta) natanko določena odgovorna oseba. Ker se aktivnosti vnaprej planira, so znani tudi pričakovani rezultati, ki jih je mogoče preveriti. Ob izvedbi se meri porabljen čas in dosežen rezultat. Rezultati se shranijo v merski del repozitorija. Rezultate po potrebi preveri skupina SQA ali vodstvo.

Nujna zahteva za kakovostno organizacijo dela je natančno spremljanje izvajanja aktivnosti in rezultatov vseh članov. V ta namen v organizaciji obstaja plan aktivnosti za vse člane. Plan se določa samo za aktivnosti na ravni organizacije, medtem ko je projektno delo planirano v okviru projekta. V planu aktivnosti je podan samo skupen čas za projektno delo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Planiranje aktivnosti se določa na podlagi rednih obveznosti posameznikov, rednih skupnih aktivnosti ali trenutnih zadolžitev. Potreben čas za izvedbo aktivnosti se oceni na podlagi preteklih merskih rezultatov. Za vsako izvedeno aktivnost mora član vnesti potreben čas izvedbe in dosežen rezultat. Planiranje je tako kot merjenje avtomatizirano v okviru IS repozitorija.

3.4.2.4 Delovanje vodstva

V majhni organizaciji uporabljamo horizontalno strukturo vodenja, zato je vodstvo sestavljeno samo iz nekaj članov. Glavna funkcija vodstva je vodenje organizacije v skladu z izbrano strategijo razvoja. Poleg tega vodstvo odloča tudi o vseh poslovnih odločitvah pri izbiri in izvedbi projektov. Kot formalni podpisnik dokumentov je odgovorno tudi za razvoj PO. Naloga vodstva odloča tudi o vseh poslovnih odločitvah pri izbiri in izvedbi projektov. Ker je lahko projektov več, si člani vodstva delijo delo in odgovornosti.

Ker je organizacija majhna in člani med seboj dobro komunicirajo, lahko namesto rednih pregledov projektov predvidimo preglede na zahtevo. Formalna potrjevanja s strani vodstva tako ohranimo samo za nujne primere. Nadzor vodstvu omogoča tudi neomejen dostop do podatkov repozitorija ter redni tedenski sestanki, ki jih člani vodstva tudi vodijo.

3.4.2.5 Delovanje skupin SEPG in SQA

Na ravni organizacije sta določeni skupini SEPG in SQA, ki imata (obe) vsaj enega člana.

Glavna naloga skupine SEPG je razvoj, izgradnja, vzdrževanje in nadgradnja organizacijskega procesa razvoja PO in vsega, kar je s tem povezano. Pomembna naloga je tudi skrb za repozitorij in z njim povezan IS. Med obveznosti skupine SEPG spada tudi zagotavljanje nemotenega zbiranja vseh merskih podatkov, njihovo hranjenje in globalna analiza (recimo analiza vseh projektov v določenem letu). Na ravni projekta za to skrbi vodja procesa in merjenja, ki je običajno tudi član skupine SEPG. Skupina prejema predloge za izboljšavo ali spremembo procesa. Kakovost procesa ocenjuje po merskih rezultatih projektov.

Naloga skupine SQA je skrb za ustrezno kakovost aktivnosti, projektov in izdelkov v organizaciji. To zajema delovanje na več področjih:

- pri preverjanju ustreznosti raznih aktivnosti, ki se izvajajo v organizaciji;
- pri preverjanju aktivnosti razvoja PO v projektih in izdelkih, ki pri tem nastanejo;
- pri določanju standardov kakovosti in drugih standardov, ki služijo kot merilo za preverjanje ustreznosti postopkov ali izdelkov;
- pri vzdrževanju in izboljševanju teh standardov.

Večino pregledov izvaja skupina SQA preko ustreznih podatkov in poročil repozitorija. Skupina SQA skrbi tako za aktivnosti, ki se v organizaciji ponavljajo v določenih časovnih obdobjih, kot za tiste, ki se sprožijo samo enkrat kot posledica določenih dogodkov.

3.4.2.6 Izvajanje projektov

Izvajanje projektov je glavna dejavnost organizacije. Ko naročnik odda zahtevo, se z njim sestane vodstvo in podpiše Inicializacijski dokument[21]. Ta je podlaga za začetek projekta.

[21] Inicializacijski dokument se podpiše ob vzpostavitvi projekta. V njem se okvirno določi zahteve projekta.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Vodstvo tedaj (ali že prej) izbere vodjo projekta in člane, ki bodo sodelovali pri projektu. Vodja projekta se sestane z vodstvom organizacije in se dogovori za osnovne značilnosti dela. Sledi prvi sestanek, na katerem se zberejo člani in določijo operativne značilnosti projekta. Projekt poteka v skladu z organizacijskim procesom razvoja, ki ga projektna skupina prilagodi svojim potrebam. Vsebina procesa je predstavljena v razdelku 3.4.4. Med izvajanjem projektni vodja večkrat poroča vodstvu organizacije. Ob koncu projekta skupina pripravi poročilo, ki ga vodja projekta predstavi vodstvu organizacije. S tem je projekt zaključen.

3.4.2.7 Sestanki v organizaciji

V organizaciji so organizirani redni tedenski sestanki med vodstvom, člani skupin SEPG in SQA, vsemi vodji projektov ter po potrebi še s kom. Vsebina sestanka je določena vnaprej. Poleg rednih pozna organizacija tudi dogovorjene sestanke. Ti so sklicani zaradi posebnih okoliščin, od katerih so odvisni tudi udeleženci (recimo vsi člani projektne skupine, naročnik projekta ali kdo drugi). Za razliko od rednih sestankov tu ni predpisane vsebine sestanka. Namen dogovorjenih sestankov je lahko tudi reševanje odprtih vprašanj in sporov med člani.

3.4.2.8 Podporne aktivnosti

Zagotavljanje kakovosti, dokumentiranje, merjenje in usposabljanje so uvedene v skladu z zahtevami prilagojenjega modela CMM.

3.4.3 Vloge in skupine prilagodljivega procesa

Pred opisom aktivnosti moramo spoznati še vloge in skupine, ki jih določa prilagodljiv proces razvoja PO. Na ravni organizacije proces določa samo skupini SEPG in SQA, ki sta bili ustrezno predstavljeni že v razdelku 3.1.2.4, pa tudi v okviru opisa tipične organizacije (glej razdelka 3.4.2.4 in 3.4.2.5). Kot posebno vlogo na ravni organizacije opredelimo le vodjo oziroma vodstvo organizacije. Tudi delovanje vodstva je bilo predstavljeno v razdelku 3.4.2.3, predstavljeno pa je tudi v dodatku A.1.1.

Na ravni projekta ni skupin, zato pa so zanimivejše vloge, ki smo jih doslej le omenjali (glej razdelek 3.1.2.4) in si jih še nismo podrobneje ogledali. Proces zaradi značilnosti tehnološkega procesa določa kar 12 različnih vlog. Čeprav je to za majhno organizacijo veliko, pa v resnici ne povzroča težav. Vsak član projektne skupine namreč lahko izvaja več vlog naenkrat, dopuščamo pa tudi opuščanje nepotrebnih vlog ob prilagoditvah.

Izvedba tega je odvisna od konkretnih zahtev projekta. Pri tem upoštevamo tri načela:

Neodvisnost določa, da član ne more sočasno izvajati dveh vlog v okviru izbranega projekta, če med naloge prve vloge sodi nadzor druge vloge. To je tudi razlog, zakaj vodja projekta ne more izvajati nobene druge vloge.

Združljivost opisuje, katere vloge vsebinsko sodijo skupaj in so zato primernejše, da jih izvaja isti član. Tipičen primer je združljivost vlog s področja razvoja PO.

Časovna razmestitev predpisuje način razdelitve vlog članom projekta, da bodo ti med celotnim potekom projekta enakomerno obremenjeni. Tako ni primerno, če enemu članu zaupamo več vlog z začetka projekta, drugemu pa samo vloge s konca projekta.

Prilagodljiv proces razvoja PO določa, da v okviru projekta določimo naslednje vloge:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- **Vodja projekta** je zadolžen za vodenje in izvedbo projekta. Kot zastopnik projekta se tudi dogovarja z naročnikom in poroča vodstvu organizacije. Poleg tega sproti nadzira potek projekta, odpravlja težave in nasploh skrbi za doseganje zastavljenih ciljev.
- **Vodja kakovosti** je pomemben član, saj s svojim delovanjem (zunanjimi pregledi kakovosti, analizami aktivnosti) zagotavlja primerno kakovost razvoja. Od njega se pričakuje natančno planiranje in objektivna presoja rezultatov pregledov, kar je podlaga za pravočasno ukrepanje. Praksa je, da to vlogo opravlja član skupine SQA.
- **Vodja procesa in merjenja** skrbi za dosledno uporabo projektnega procesa. Sodeluje pri prilagajanju procesa, zbira predloge za spremembe le-tega in po potrebi ukrepa. Zadolžen je tudi za merjenje na ravni projekta, kar pomeni, da le-tega planira in skrbi za njegovo izvajanje. Ključni nalogi merjenja sta ustrezno zbiranje merskih podatkov in izvajanje merskih analiz. Vodja procesa je običajno član skupine SEPG.
- **Vodja podpore** v okviru projekta skrbi za upravljanje konfiguracij, upravlja projektno dokumentacijo ter skrbi za druge podporne dejavnosti, recimo zagotavljanje ustreznih orodij, priprava namestitve rezultatov razvoja, vzdrževanje, arhiviranje in podobno.
- **Vodja sodelovanja z naročnikom** ima tri naloge: vzdrževanje stalne komunikacije z naročnikom, usposabljanje uporabnikov ter oblikovanje uporabniškega vmesnika IS.
- **Vodja planiranja** odgovarja za planiranje porabe časa in opravil, kar zahteva predvsem natančno ocenjevanje ter ustrezno sodelovanje z razvijalci ob določanju planov. Oboje zahteva usposobljenost (recimo poznavanje statističnih metod), seznanjen pa mora biti tudi z načini uporabe rezultatov preteklih projektov. Pomemben del njegovega dela je dinamično prilagajanje spremembam pri planiranju ciklov.
- **Vodja testiranja** z različnimi testiranji preverja kakovost končnih izdelkov. Čeprav naš proces aktivno skrbi za kakovost že prej, je celovito testiranje zelo pomembno. V njegove zadolžitve sodi tudi skrb za ustrezno odpravljanje napak.

Naslednjih pet vlog skrbi za ustrezen tehnološki razvoj izdelka. S tega stališča nadomeščajo skupino za tehnologijo PO (SEG) [Pau93a, SEI02b], ki jo naš pristop ne predvideva. Vse vloge bi lahko nadomestili samo z eno samo vlogo (vodja razvoja), vendar želimo v prilagodljivem procesu poudariti posebnosti razvoja IS (recimo procesni in podatkovni vidik). Vloge v tem delu so tipično zadolžene za del razvoja, ki se ponavadi izvaja v določenem ciklu. Ob prilagoditvi lahko določene vloge tudi opustimo ali združimo. Vloge so naslednje:

- **Vodja analize** je zadolžen za izvedbo raziskave zahtev in logično načrtovanje. Njegova naloga je tudi spremljanje sprememb zahtev.
- **Vodja načrtovanja IS** je zadolžen za ustrezno fizično načrtovanje procesnega dela IS.
- **Vodja načrtovanja PB** je zadolžen za ustrezno fizično načrtovanje PB.
- **Vodja izgradnje IS** je zadolžen za izgradnjo procesnega vidika končnega izdelka, torej programske kode. Njegovo delo obsega tudi odpravljanje napak, pripravo ustrezne dokumentacije in namestitev dokončanega sistema.
- Podobno je **vodja izgradnje PB** zadolžen za izgradnjo PB. Njegova skrb je tudi preverjanje usklajenosti PB z IS, polnjenje podatkov ob namestitvi in dokumentacija PB.

Za izvedbo vsakega koraka prilagodljivega procesa je zadolžena določena vloga (glej dodatke med A.2.2 in A.2.9). Pri tem je pomemben princip prilagodljivega pristopa, ki omogoča izvajanje korakov na več načinov (glej razdelek 3.3.2). Če se korak izvaja na enostaven ali zahtevnejši način enega člana, potem zadolžitev pomeni, da bo korak izvedel sam član, ki izvaja to vlogo. Če korak izvaja več članov, bo zadolžena vloga vodila oziroma usmerjala druge člane. To pomeni, da se mora vsak član vsaj nekoliko spoznati na vsako aktivnost.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Skupine in vse vloge so predstavljene tudi v dodatku A.1.

3.4.4 Vsebinski opis prilagodljivega procesa za razvoj PO

Ogledali smo si delovanje tipične majhne organizacije za razvoj IS in vloge, ki jih določa prilagodljiv proces. S tem smo spoznali vse potrebne okoliščine in lahko razumljivo predstavimo vsebino prilagodljivega procesa razvoja, ki bo predstavljene značilnosti povezala v smiselno celoto. Formalni prikaz procesa (z navodili) je v dodatku A.2.

3.4.4.1 Splošne značilnosti procesa

Proces razvoja PO na ravni organizacije je implementacija prilagodljivega procesa in ima vse značilnosti, ki so bile predstavljene v podpoglavju 3.3[22]. Celoten proces obsega 36 faz, ki so razporejene v 7 ciklov. Tehnološke in projektne aktivnosti sestavljajo korake in faze v različnih ciklih. Delovne faze so večinoma namenjene tehnološkemu izvajanju procesa, medtem ko so projektne aktivnosti zbrane predvsem v prvem in zadnjem ciklu ter v ponovljivih fazah cikla. Prvi in zadnji cikel (»Začetek projekta« in »Zaključek projekta«) sta sploh posebna, saj sta namenjena izključno zagonu in zaključku projekta.

Vsak cikel vsebuje ponovljivi fazi »Priprava cikla« in »Zaključek cikla«, s katerima se cikel začne in zaključi. Na ravni cikla opravljata enako funkcijo, kot jo opravljata cikla »Začetek projekta« in »Zaključek projekta« na ravni celotnega razvoja.

Priprave vseh ciklov (z izjemo prvega) obsegajo pretežno enake naloge. Te so naslednje:

- preverjajo, ali je potrebno spremeniti cilje projekta, globalne plane, način dela ali sestavo projektne skupine zaradi spremenjenih zahtev ali drugih vzrokov;
- natanko planirajo izvedbo opravil, porabo časa, stroške in kakovost tako za cikel kot posamezne člane;
- predstavijo plane in spremembe vodstvu organizacije in jih z njim uskladijo;
- določijo ustrezne standarde, ki so potrebni za izvedbo cikla.

Tudi zaključki ciklov (z izjemo zadnjega) obsegajo običajno enak nabor nalog:

- zberejo vse rezultate cikla in jih ponovno formalno pregledajo z vidika pravilnosti, usklajenosti in popolnosti;
- preverijo uvrščenost rezultatov v trenutno konfiguracijo in po potrebi konfiguracijo dopolnijo;
- opravijo analizo cikla z vidika porabe časa, stroškov, doseženih rezultatov in kakovosti rezultatov;
- na podlagi ugotovitev analize ustrezno ukrepajo (po potrebi lahko ponovijo del razvoja, spremenijo zahteve ali podobno);
- pripravijo poročilo o ciklu in poročajo vodstvu organizacije.

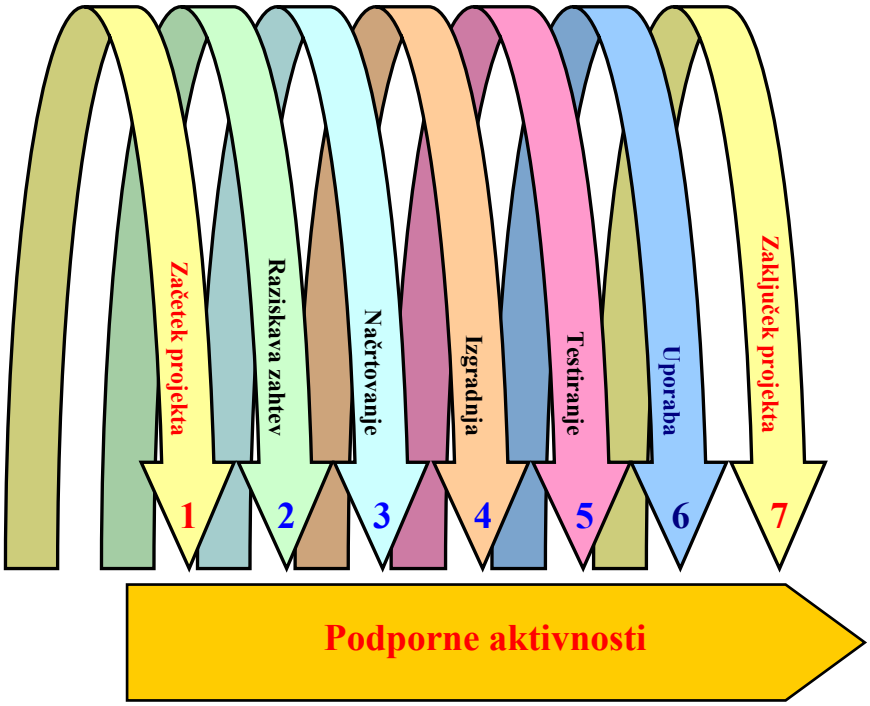
3.4.4.2 Proces na najvišji ravni

Na sliki 3.10 je predstavljen prilagodljiv proces razvoja PO, ki obsega 7 ciklov aktivnosti. Cikli se izvajajo zaporedno od prvega do zadnjega. Poleg tega obstaja še več podpornih aktivnosti, ki se izvajajo ves čas razvoja. Cikli so naslednji:

[22] V podpoglavju 3.3 smo predstavili strukturo in izvirne rešitve v okviru prilagodljivega procesa razvoja PO.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

1. **»Začetek projekta«:** V »Začetku projekta« pripravimo vse potrebno za reševanje problema: sestavimo projektno skupino, si določimo cilje in način dela, izvedemo strateško načrtovanje, pripravimo globalne plane in se o vsem dogovorimo z vodstvom.
2. **»Raziskava zahtev«:** Namen cikla je ugotoviti, kaj naročnik zares hoče, in to jasno zapisati. Obsega zajem vseh zahtev, njihovo analizo in logično modeliranje.
3. **»Načrtovanje«:** V tem ciklu definiramo, kako bomo rešili problem. Pri tem določimo sestavo rešitve, izberemo tehnično rešitev ter definiramo fizičen načrt rešitve.



Slika 3.10: Prilagodljiv proces razvoja PO

4. **»Izgradnja«:** V ciklu »Izgradnja« zares zgradimo rešitev. Najprej podrobno načrtujemo vse potrebne dele, nato pa zgradimo izdelke (programsko kodo in PB) in jih po delih testiramo.
5. **»Testiranje«:** Kot pove že ime je cikel namenjen vsestranskemu testiranju zgrajenih izdelkov. Pri tem opravimo sistemsko testiranje in uporabniško testiranje.
6. **»Uporaba«:** Značilnost razvoja IS je v tem, da moramo poskrbeti tudi za uvedbo izdelka v uporabo. Cikel »Uporaba« izvede namestitev in vzdrževanje.
7. **»Zaključek projekta«:** Ob »Zaključku projekta« zberemo vse rezultate, jih analiziramo, shranimo, pripravimo končno poročilo in zaključimo projekt.

Formalna predstavitev celotnega prilagodljivega procesa je v dodatku A.2.1.

3.4.4.3 Podporne aktivnosti

V ciklih se dogaja glavnina razvojnih aktivnosti. Poleg teh med izvajanjem projekta potekajo tudi podporne aktivnosti, ki jih ne bi mogli uvrstiti v nobeno od faz. Izvajajo se bodisi v rednih časovnih presledkih neodvisno od poteka projekta (recimo »Tedenski sestanki«) ali pa ves čas v odvisnosti od trenutnih potreb razvoja (recimo »Upravljanje konfiguracije«). Aktivnosti so združene v naslednjih 6 faz (slika 3.11):

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 3.11: Cikel 0: Podporne aktivnosti

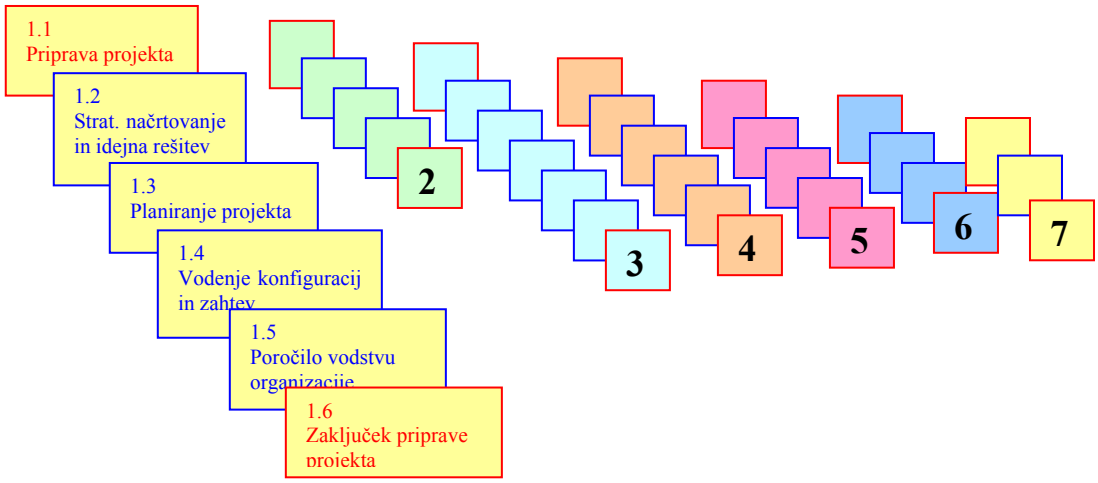
1. **»Tedenski sestanki«** se izvajajo enkrat tedensko na ravni projekta in so namenjeni sprotnemu spremljanju izvajanja projekta. V skladu z ustaljenim dnevnim redom jih vodi projektni vodja. Vsi člani projektne skupine poročajo o aktivnostih, za katere so zadolženi. Posebna pozornost je namenjena rezultatom analize poteka razvoja in kakovosti, ki jo vsak teden pred sestankom opravi vodja procesa in merjenja. Na sestanku se pogovorijo tudi o vseh drugih odprtih vprašanjih, recimo o nujnih spremembah zahtev in podobnem. Sestanki morajo biti kratki, konstruktivni in dajati jasne odgovore oziroma odločitve.
2. **»Zbiranje merskih podatkov«** poteka med izvajanjem vseh aktivnosti. Člani so dolžni po vsaki opravljeni aktivnosti oddati meritve o porabi časa, velikosti rezultata, stroških in kakovosti (številu napak). Zbiranje je avtomatizirano in usklajeno s plani opravil posameznih članov. Vodja procesa in merjenja skrbi za nemoten potek merjenja v skladu z definiranim merskim procesom. Opravlja tudi tedenske analize in analize ob zaključkih ciklov, pri katerih na podlagi merskih podatkov določa pokazatelje poteka in kakovosti.
3. **»Upravljanje konfiguracije«** avtomatsko zbira zahtevke za dodajanje novih izdelkov med trenutno konfiguracijo ves čas izvajanja projekta. Zahtevke preverja vodja projekta, ki tudi odredi dodajanje izdelka v trenutno konfiguracijo, če je to smiselno. Za operativno upravljanje konfiguracije skrbi vodja podpore.
4. **»Upravljanje sprememb zahtev«** skrbi za ustrezno obravnavanje spremenjenih zahtev. Ves čas izvajanja projekta se avtomatsko sprejema zahtevke od članov ali naročnika. Vodja projekta za vsak zahtevek določi njegovo pomembnost in nujnost. Na podlagi tega se sproži postopek za spremembo zahteve bodisi na prvem tedenskem sestanku ali ob zaključku cikla. Za upravljanje sprememb zahtev skrbi vodja analize, ki preverja tudi usklajenost dokumentacije s spremenjeno zahtevo.
5. **»Skrb za izboljšavo procesa«** je naloga vseh članov, zato so dolžni vsi pomagati s predlogi za izboljšave, rešitvami tipičnih težav in podobnim. Predlogi se zbirajo avtomatsko v repozitoriju, vodja procesa in merjenja pa je zadolžen za njihov nadzor, analizo in posredovanje utemeljenih predlogov skupini SEPG.
6. **»Spremljanje tveganj«** je sklop aktivnosti, ki nadzirajo potencialne težave in ukrepajo, v kolikor se težava dejansko pojavi. Tveganja se najprej identificira, za kar poskrbi vodja procesa in merjenja. Za vsako tveganje se zadolži posameznega člana, da ga spremlja in o njem poroča. Ukrepi ob pojavitvi tveganj se planirajo v okviru faze »Planiranje projekta«.

Bralec si lahko ogleda formalno predstavitev podpornih aktivnosti v dodatku A.2.2.

3.4.4.4 Prvi cikel: Začetek projekta

V prvem ciklu se pripravi in zažene projekt. Glavna dela cikla sta strateško načrtovanje ter planiranje opravil in porabe časa za celoten projekt. Kot kaže slika 3.12 cikel obsega šest faz:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 3.12: Cikel 1: Začetek projekta

1. **»Priprava projekta«** je prva faza cikla. Vodstvo organizacije sproži projekt na podlagi Inicializacijskega dokumenta, ki vsebuje okvirne zahteve. Sledi prvi sestanek, na katerem se določijo cilji projekta, vloge posameznim članom, razvojna strategija in način dela. V okviru tega se na podlagi organizacijskega merskega modela določi tudi projektno merjenje. Ugotovi se tudi potrebe po izobraževanju članov in orodjih za razvoj. Rezultata faze sta Inicializacijski dokument in osnutek Projektne knjige. V tej bodo na koncu zbrani vsi dokumenti o poteku projekta (plani, poročila, analize in drugo).
2. **»Strateško načrtovanje in idejna rešitev«** zajema prvo delovno srečanje projektne skupine z naročnikom. Okvirne zahteve Inicializacijskega dokumenta ne zadoščajo, zato je treba pred planiranjem temeljiteje raziskati poslovno okolje in obstoječi sistem naročnika. Ob tem se oblikujejo prve idejne rešitve, ki se jih na prvem sestanku med projektno skupino in naročnikom predstavi naročniku. Ta izbere najprimernejšo idejo in s tem usmeri nadaljevanje razvoja. Vse ugotovitve se shranijo v Strateškem dokumentu, ki predstavlja rezultat te faze.
3. **»Planiranje projekta«** je najpomembnejša faza tega cikla. V njej skušamo zastaviti čim natančnejše globalne plane za izvedbo vseh aktivnosti projekta, čeprav imamo po opravljenem strateškem planiranju zgolj okvirno opredeljene zahteve. Najprej definiramo vse končne izdelke in njihove sestavne dele, ocenimo njihovo velikost in potreben čas za razvoj po fazah. Na podlagi tega naredimo globalni plan opravil, porabe časa in stroškov. Če je projekt manjši oziroma že dovolj natančno poznamo zahteve, lahko izdelamo tudi osebne plane članov. Pri tem izvedemo postopek planiranja v treh delih: najprej vsak član oceni svoj delež, te deleže uskladimo z globalnim planom in nazadnje prerazporedimo obremenitve, da so vsi člani približno enako obremenjeni. Sledi postopek planiranja kakovosti, v katerem na podlagi podatkov preteklih projektov ocenimo število napak po fazah. Identificiramo še tveganja in za vsako tveganje zadolžimo člana skupine. Zadnja koraka sta zunanji pregled ustreznosti vseh planov s strani skupine SQA in dokumentiranje le-teh v Projektni knjigi.
4. **»Vodenje konfiguracij in zahtev«** sproži to dejavnost za potrebe projekta. V ta namen se identificirajo vsi izdelki in vmesni opisi razvoja, ki bodo pod nadzorom konfiguracije. Vodja projekta v sodelovanju z vodjo planiranja in vodjo razvoja odloča o zahtevkih za dodajanje izdelkov v konfiguracijo. Podobno se ravna z zahtevami. Tudi tu je zadolžen vodja projekta v sodelovanju z vodjo planiranja in vodjo razvoja za spremembe in usklajenost zahtev v vseh razvojnih dokumentih. Ob koncih faz in ob začetku ciklov se zahteve posebej preveri. Vse člane se seznani s postopki konfiguriranja in spreminjanja zahtev ter se pripravi plan za vodenje konfiguracij. Ko ustreznost tega potrdi neodvisni pregledovalec (iz skupine SQA), se ga doda Projektni knjigi.

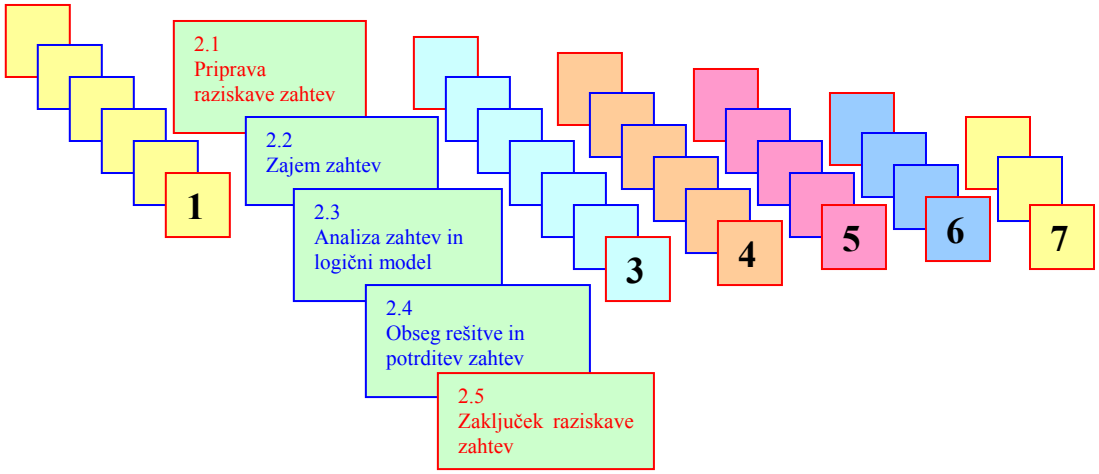
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5. **»Poročilo vodstvu organizacije«** opravi vodja projekta s pomočjo vseh članov. S predstavitvijo projektna skupina argumentirano pojasni svojo vizijo izvajanja projekta. To je podlaga za diskusijo, po kateri vodstvo organizacije potrdi ali zavrže njihovo rešitev. V drugem primeru mora projektna skupina ponovno opraviti potrebne faze in predstavitev ponoviti. V tem delu se nad podatki repozitorija izvajajo tudi analize izvedljivosti[23].
6. **»Zaključek priprave projekta«** je zadnji del »Začetka projekta« in obsega enake korake kot vsi zaključki ciklov. V tem delu se zbere vse rezultate ciklov in se jih formalno še enkrat preveri z vidika pravilnosti, popolnosti in usklajenosti. Nato se opravi analiza izvedbe cikla na podlagi podatkov repozitorija, ki jih člani oddajajo ob izvedbi vsake aktivnosti v skupni repozitorij. V kolikor so ugotovljene pomanjkljivosti ali težave, se predlaga ukrepe, ki običajno obsegajo ponovitev določenega dela razvoja. Z upoštevanjem zbranih predlogov za izboljšave se oblikuje Poročilo o ciklu, v katerem so zajeti vsi pomembni podatki o izvedenem ciklu. To poročilo vodja projekta predstavi na naslednjem tedenskem sestanku organizacije. V kolikor ima vodstvo organizacije pripombe, se te upoštevajo in formalno skupaj s poročilom shranijo v Projektni knjigi.

Podrobnejša predstavitev prvega cikla je v dodatku A.2.3. Tam so razvidni tudi rezultati posameznih faz in celotnega cikla.

3.4.4.5 Drugi cikel: Raziskava zahtev

Drugi cikel je namenjen zajemu uporabnikovih zahtev, analizi le-teh ter oblikovanju Specifikacije zahtev. Ta poleg seznama zahtev obsega še logična modela IS in PB. Na koncu cikla bomo natančno vedeli, kaj naročnik od nas zahteva. Raziskovanje zahtev obsega pet faz, ki so razvidne tudi iz slike 3.13.



Slika 3.13: Cikel 2: Raziskava zahtev

1. **»Priprava raziskave zahtev«** je faza, ki se izvaja na podoben način ob začetku vsakega cikla (z izjemo prvega in zadnjega). Obsega natančno planiranje vseh opravil in porabe časa cikla, kar se določi na podlagi globalnih planov. V tej fazi tudi natanko določimo plane opravil in porabe časa za vse člane na podlagi treh korakov: ocene potrebnega časa za opravila s strani članov, usklajevanja z vsemi plani cikla in prerazporeditve bremen. Zadnji del faze je določanje standardov, ki bodo veljali med izvajanjem zajema in analize zahtev (obrazci, načini opisa ipd.).

[23] Analize izvedljivosti so ena izmed vrst analiz, ki jih naš pristop definira. Vse analize so podrobno predstavljene v podpoglavju 4.3.

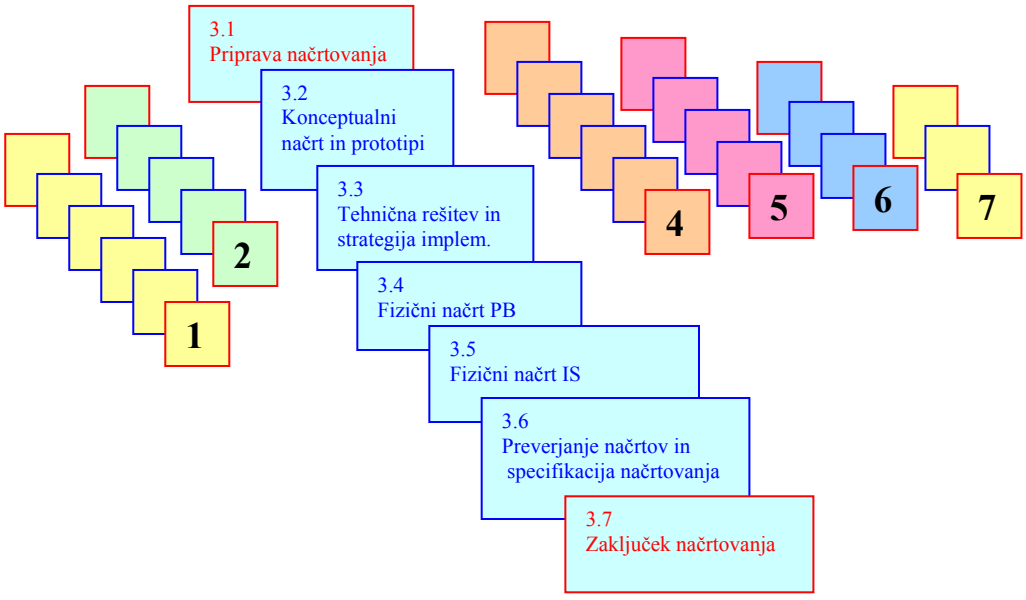
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

2. **»Zajem zahtev«** je lahko zelo obsežna faza, čeprav formalno obsega samo tri korake. Zahteve zbiramo pri naročniku in uporabnikih z intervjuji, vprašalniki in sestanki na različnih ravneh. Preverimo tudi obstoječ sistem z vidika tehnične dokumentacije, uporabe, načina dela uporabnika in njegovih izkušenj. Vse rezultate beležimo v skladu s predpisanim načinom v seznam zahtev, ER diagrame in diagrame podatkovnih tokov.
3. **»Analiza zahtev in logični model«** obsega prepis zajetih informacij v bolj natančen opis nastajajočega sistema. Pri tem modeliramo tako procesni (funkcijska dekompozicija, diagrami podatkovnih tokov) kot podatkovni (ER diagrami) del. S postopkom logičnega modeliranja odpravimo pomanjkljivosti in oba modela uskladimo. Povezave med obema deloma in seznamom zahtev opišemo z več matrikami. Na koncu faze vse rezultate z vidika pravilnosti preveri še član skupine SQA.
4. **»Obseg rešitve in potrditev zahtev«** je naslednja faza. Obseg rešitve je lahko različen in je odvisen od izbire naročnika. Na podlagi logičnih modelov zato pripravimo predstavitev za naročnika, na kateri mu predstavimo več možnih obsegov bodočega sistema. Naročnik izbere enega, obenem pa preveri tudi vse zbrane rezultate iz cikla »Raziskave zahtev«. Na ta način potrdi pravilnost zbranih rezultatov, ki jih po tem pregledu shranimo v dokumentu Specifikacija zahtev.
5. **Zaključek raziskave zahtev** je zadnji korak, ki poleg (v prejšnjem ciklu) omenjenih korakov (zbiranja in preverjanja rezultatov, analize izvedbe, predloga ukrepov in poročanja vodstvu organizacije) izvede še nekaj korakov. Ker imamo sedaj pomemben rezultat razvoja (Specifikacijo zahtev), moramo preveriti njegovo uvrščenost v trenutno konfiguracijo. Med izvedbo cikla smo mogoče tudi odkrili napake pri zasnovi rešitve v prvem ciklu, kar nam daje podlago za dodatne analize. Tudi o teh poročamo vodstvu organizacije.

Rezultati cikla so zbrani v enotnem dokumentu Specifikacija zahtev, opis poteka projekta pa v dopolnjeni Projektni knjigi. Formalen opis cikla raziskave zahtev se nahaja v dodatku A.2.4.

3.4.4.6 Tretji cikel: Načrtovanje

Cilj načrtovanja je opisati, kako bomo izpolnili naročnikove zahteve. Glavne faze obsegajo izdelavo konceptualnega načrta IS, razvoj fizičnih modelov ter odločitev o tehničnih vidikih uporabe s strani naročnika. Cikel na sliki 3.14 obsega sedem faz:



Slika 3.14: Cikel 3: Načrtovanje

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

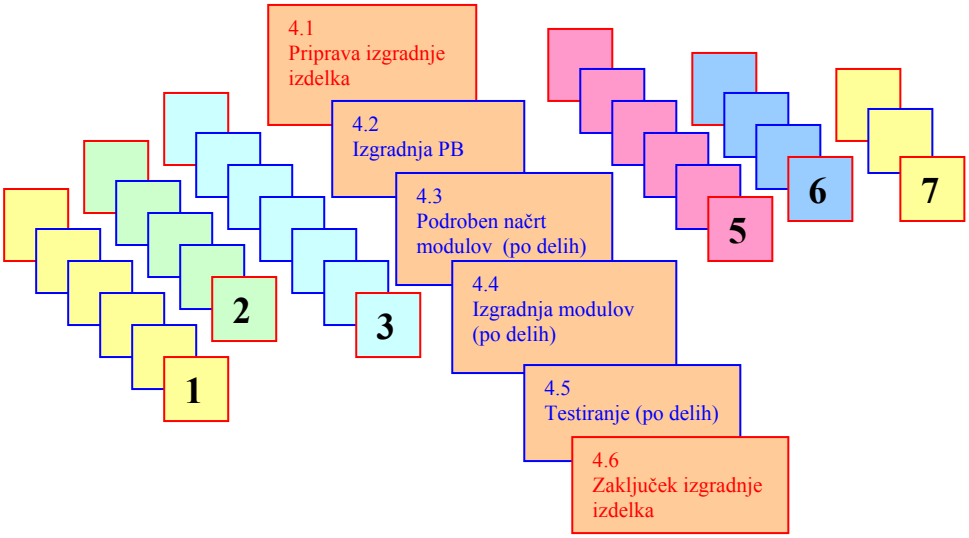
1. **»Priprava načrtovanja«** je začetna faza, ki je podobna začetnim fazam ostalih ciklov. Med že omenjene korake (natančno planiranje cikla in članov, poročanje vodstvu organizacije) ta faza dodaja še ponovno presojo projekta. Preverjanje ciljev, globalnih planov, vlog in ocen tveganja je potrebno zaradi tega, ker so se lahko te stvari spremenile zaradi ugotovitev raziskave zahtev. To je pomemben vidik prilagodljivosti sistema, ki se zna po vsakem ciklu ustrezno odzivati na spremembe. Zadnji del faze določi standarde za izvedbo načrtovanja (recimo obliko grafičnega vmesnika, sistem pomoči, zahteve za dokumentacijo in podobno).
2. **»Konceptualni načrt in prototipi«** je faza, v kateri določimo načrt sestave nastajajoče rešitve. Je pomemben rezultat načrtovanja, ki omogoča delitev dela med več razvijalcev v nadaljevanju. Ker so po koncu prejšnjega cikla popolnoma znane zahteve naročnika in obseg rešitve, lahko pripravimo več predlogov tehničnih rešitev. Za vsako (ali vsaj pomembnejše) pripravimo tudi ustrezne prototipe, da bi uporabniku lažje predstavili funkcionalnost rešitev. Obenem za naslednjo fazo lahko pripravimo tudi več predlogov strategij testiranja (predvsem uporabniškega), namestitve in vzdrževanja.
3. **»Tehnična rešitev in strategija implementacije«** sta rezultata predstavitve predlogov prejšnje faze naročniku. Gre za obsežnejšo predstavitev, ki je zadnja pred izgradnjo rešitve. Naročnik na podlagi predstavljenih možnosti in prototipov določi tehnično rešitev in ustrezne strategije. Na tej podlagi pripravimo plane testiranja, namestitve in vzdrževanja. V zadnjem koraku faze vse plane preveri član skupine SQA.
4. **»Fizični načrt PB«** je izdelan na podlagi logičnega modela, zajema pa opise tabel, razmerij, atributov in drugega. V tem delu razvijemo fizične načrte tudi za tiste dele PB, ki smo jih določili ob izboru strategije za potrebe testiranja, nameščanja ali vzdrževanja. Tipično so to posebne strukture za prenos podatkov starega sistema na novega ali zajem parametrov delovanja IS med vzdrževanjem.
5. **»Fizični načrt IS«** obsega opise vseh modulov, poročil, navigacije med njimi in podobnega. Poseben poudarek je namenjen skupnim delom oziroma ponovni uporabljivosti. Obenem določimo tudi fizične načrte za pomožne programe. Ti služijo za pomoč pri izvedbi testiranja (recimo največje obremenitve), namestitve (recimo prenos podatkov) in vzdrževanja (recimo izpis parametrov delovanja IS).
6. **»Preverjanje načrtov in specifikacija načrtovanja«** je faza, v kateri preverimo usklajenost načrtov procesnega in podatkovnega dela. Ker je pravilnost načrtov za nadaljevanje zelo pomembna, jih preveri še neodvisna stran (skupina SQA). Vse rezultate cikla zberemo v celovit rezultat (Specifikacijo načrtovanja). Enako kot druge pomembne rezultate razvoja tudi Specifikacijo načrtovanja vključimo v trenutno konfiguracijo.
7. **»Zaključek načrtovanja«** je zadnja faza, ki poskrbi za zbiranje vseh rezultatov, analizo izvedbe in kakovosti cikla, ustrezno ukrepanje (po potrebi ponovitev določenih korakov faze ali uskladitev z naročnikom) ter poročilo vodstvu organizacije.

Podrobnosti tretjega cikla z rezultati so predstavljene v dodatku A.2.5.

3.4.4.7 Četrti cikel: Izgradnja

V tem ciklu dejansko zgradimo rešitev in jo delno preverimo. Če je proces razvoja PO podprt z orodji, je pogosto ta del procesa izveden z uporabo raznih generatorjev kode, sicer pa obsega učinkovito izvedbo kodiranja. Več faz je zasnovanih tako, da se lahko izvajajo ločeno po delih (modulih), kar omogoča delitev dela med člane projektne skupine. Cikel je predstavljen na sliki 3.15. Obsega šest faz:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 3.15: Cikel 4: Izgradnja

1. **»Priprava izgradnje«** obsega enake korake kot priprava načrtovanja: ponovno oceno ciljev projekta, po potrebi spremembo globalnih planov in določitev podrobnih planov opravil in porabe časa za cikel ter vse člane. Tudi v tej fazi priprave določimo standarde, ki so tokrat namenjeni izgradnji (recimo standard kodiranja).
2. **»Izgradnja PB«** je faza, v kateri iz fizičnega opisa PB dejansko zgradimo PB. Pri tem zgradimo tudi tiste tabele, ki služijo za testiranje, namestitev in vzdrževanje. Pomemben del faze je tudi polnjenje PB s testnimi podatki – torej podatki prave oblike, ki so namenjeni preizkušanju pravilnosti delovanja PB. PB tudi podrobneje kot doslej opišemo v sistemski dokumentaciji o PB.
3. **»Podroben načrt modulov (po delih)«** je potreben pred kodiranjem posameznih modulov. Na podlagi konceptualnega načrta se delo razdeli med člane, ki prejmejo opise modulov (poročil, funkcij) in njihove funkcionalnosti. V tej fazi člani lahko izdelajo še podrobne načrte, ki bodo podlaga za kodiranje. Vsak načrt najprej sami ponovno pregledajo v skladu s seznamom tipičnih težav, nato pa usklajenost in pravilnost vseh načrtov preveri še član skupine SQA. Načrti se shranijo tudi v trenutno konfiguracijo, da bi se na ta način zagotovilo usklajen razvoj in ponovno uporabo skupnih delov IS (recimo poročanje o napakah).
4. **»Izgradnja modulov (po delih)«** je faza, v katerem člani dejansko napišejo kodo za posamezne dele. Ob kodiranju običajno vnesemo veliko napak, zato je posebna pozornost namenjena zagotavljanju kakovosti. Kodiranje mora potekati v skladu s standardi kodiranja. Po dokončanem pisanju programa se kodo ponovno pregleda s pomočjo seznama glavnih težav. Ta je določen na podlagi izkušenj preteklih projektov in omogoča lažje odkrivanje tipičnih problemov. Sledi prevajanje, pri katerem napake javlja prevajalnik. Vse napake so sproti odpravljene, a tudi zabeležene. V tej fazi se izdela koda tudi za vse pomožne programe. Ob kodiranju se napiše tudi sistemsko dokumentacijo z opisom delov IS.
5. **»Testiranje (po delih)«** je prvo v nizu testiranj. Njegov namen je odkriti čim več napak posameznih delov. Preverja se module in pomožne dele, pri čemer se dostopa tudi do PB in se jo na tak način tudi posredno preverja. Preveri se tudi sistemsko dokumentacijo, ki je bila pripravljena v prejšnjih fazah. Vse napake se sproti odpravi, pri čemer se beleži porabljen čas in število napak. O testiranju se pripravi tudi posebno poročilo, ki zajema opise najdenih napak. Ko je testiranje končano, preverjene dele preveri neodvisna skupina SQA, vsi deli pa se vključijo v trenutno konfiguracijo.

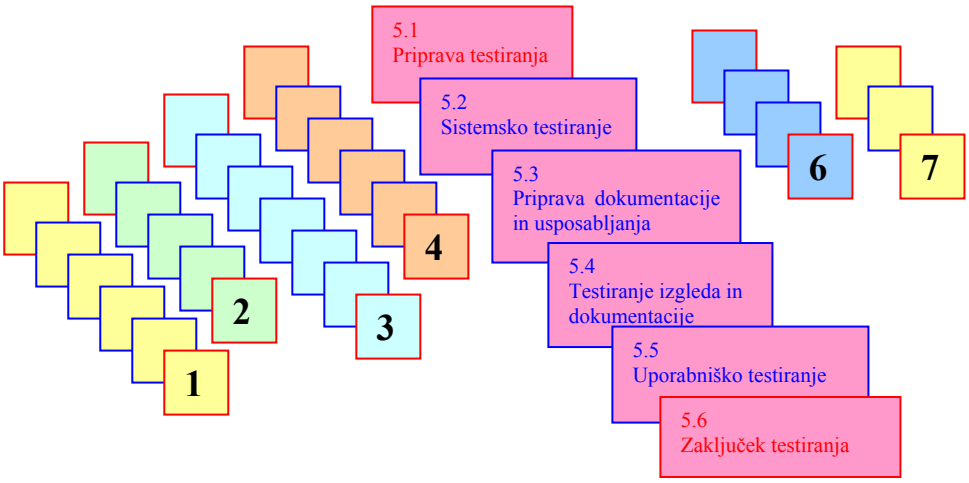
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

6. **»Zaključek izgradnje«** je tipična zadnja faza cikla, v kateri še enkrat preverimo vse rezultate, analiziramo izvedbo in kakovost ter po potrebi ukrepamo oziroma ponovimo dele izgradnje. O značilnostih cikla poroča vodja na naslednjem tedenskem sestanku organizacije. Vsi opisi poteka projekta se shranijo v Projektno knjigo.

Formalna predstavitev cikla »Izgradnja« se nahaja v dodatku A.2.6. Iz opisa so razvidni tudi rezultati faz in celotnega cikla.

3.4.4.8 Peti cikel: Testiranje

Namen cikla testiranja (slika 3.16) je vsestransko preveriti razvite izdelke in jih pripraviti za namestitev. Samo testiranje je temeljito in obsega več faz, pri čemer je dejanska izvedba odvisna od potreb oziroma izbrane testne strategije. Podrobno testiranje je gotovo nujno pri zagotavljanju visoke kakovosti izdelkov, vendar je glavni moto našega procesa predvsem napake preprečevati (z natančno izvedbo raziskave zahtev, vključevanjem naročnika) ali vsaj zgodaj odkrivati (s ponovnimi pregledi in številnimi preverjanji SQA). Cikel obsega šest faz.



Slika 3.16: Cikel 5: Testiranje

1. **»Priprava testiranja«** opravi potrebne prilagoditve projekta ob začetku cikla (sedaj jih ne bi smelo biti veliko) in natanko planira potek vseh faz testiranja. Obenem določi vse potrebne standarde za testiranje, ki opisujejo pristop in izvedbo testov.
2. **»Sistemsko testiranje«** obsega več testov, od katerih je najpomembnejši združitveni test. Ta preveri ujemanje vseh razvitih modulov s konceptualnim načrtom. Sledi test preverjanja performans ter test delovanja ob kritični obremenitvi. Testiramo lahko tudi uspešnost postopka arhiviranja in obnove PB ter prenos podatkov v PB (za polnjenje PB). Pomemben je tudi test varnosti, ki oceni različne vidike varnosti razvite rešitve. Če odkrijemo napake, poskrbimo za njihovo odpravo, pri tem pa vse njihove značilnosti vnesemo v Poročilo o testiranju. Ko je sistemsko testiranje končano, zahtevamo uvrstitev testiranega IS in PB v trenutno konfiguracijo.
3. **»Priprava dokumentacije in usposabljanja«** je namenjena pripravi uporabniške dokumentacije in učne dokumentacije. Sistemska dokumentacija bi doslej morala biti že razvita in preverjena. Po pripravi dokumentacije to preverijo člani skupine SQA.
4. **»Testiranje izgleda in dokumentacije«** preveri uporabniški vmesnik, sistem pomoči, vso razvojno dokumentacijo (specifikacije) in vse vrste dokumentacije, ki opisujejo končne izdelke. Podobno kot v prejšnjih fazah se tudi tu beleži napake v Poročilo o napakah, poskrbi za odpravo pomanjkljivosti ter nazadnje uvrsti IS in dokumentacijo v trenutno konfiguracijo.

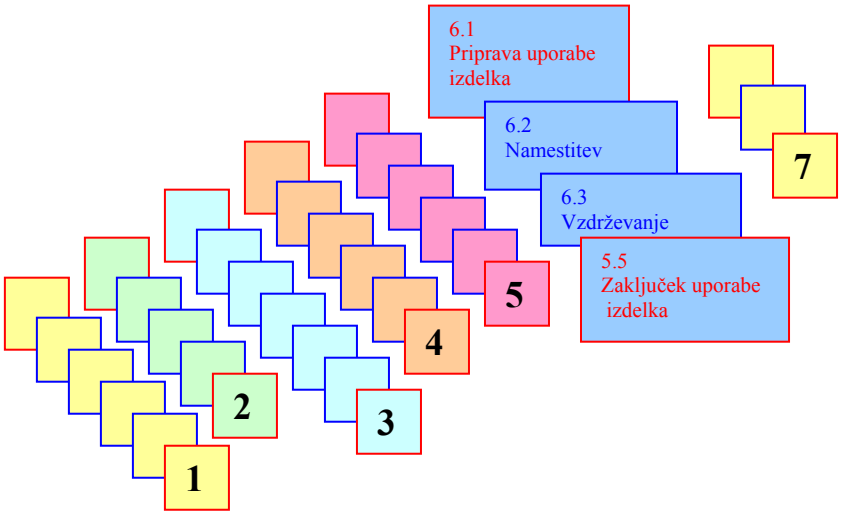
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5. **»Uporabniško testiranje«** obsega osnovno izobraževanje uporabnikov in laboratorijski test sprejemljivosti. Usposabljanje za uporabo dograjenega IS in PB za naročnika (in uporabnike) pomembno vpliva na vtis o celotnem razvoju. V tej fazi naročnik tudi dokončno odloči, ali smo ustrezno rešili zastavljen problem. V okviru predstavitve se preveri tudi prijaznost IS, sistema pomoči in uporabniške dokumentacije. Tudi to testiranje zaključimo s Poročilom o napakah, odpravo napak ter uvrstitvijo v trenutno konfiguracijo.
6. **»Zaključek testiranja«** zbere rezultate, analizira potek cikla in po potrebi ukrepa. Pripravi se poročilo za vodstvo podjetja in vse dokumente o poteku projekta shrani v Knjigo projekta.

Formalna predstavitev cikla testiranja in opredelitev njegovih rezultatov se nahaja v dodatku A.2.7.

3.4.4.9 Šesti cikel:Uporaba

Pomembna značilnost razvoja IS je zadnji del procesa, ki se ne ukvarja s samim razvojem izdelkov, temveč z vpeljavo rešitev v praktično uporabo. Obsega dva dela: namestitev in vzdrževanje. Vsak del sodi v svojo fazo, oba skupaj pa sestavljata jedro cikla »Uporaba« (slika 3.17). Skupaj z začetno in zaključno fazo ima cikel štiri faze.



Slika 3.17: Cikel 6: Uporaba

1. **»Priprava uporabe«** obsega enake aktivnosti kot prve faze drugih ciklov, čeprav sedaj že res ne bi smelo več prihajati do velikih sprememb ciljev projekta ali globalnemu planu izvedbe. Možne so zgolj spremembe strategije aktivnosti nameščanja ali vzdrževanja. Najpomembnejši rezultat faze je natančen plan izvedbe cikla, ki ga enako kot vse doslej preverimo pri vodstvu organizacije. Za izvedbo nameščanja in vzdrževanja predvidimo tudi primerne standarde.
2. **»Namestitev«** moramo temeljito pripraviti, izvedemo pa jo v skladu z izbrano strategijo. Tipična pristopa sta zamenjava vse naenkrat – BigBang (cenejša, a tvegana) in postopna zamenjava (dražja in varnejša) [Goo95]. Postopek lahko zahteva veliko obsega dela in pripravo vrste pomožnih programov in podatkov, za kar moramo poskrbeti že ob planiranju. Po namestitvi IS se uporabnikova PB napolni z realnimi podatki in preveri z uporabniškim testom v realnem okolju. Po potrebi se opravijo še zadnje nastavitve in IS je pripravljen za delo. To naročnik potrdi s prevzemom in podpisom Dokumenta o namestitvi.

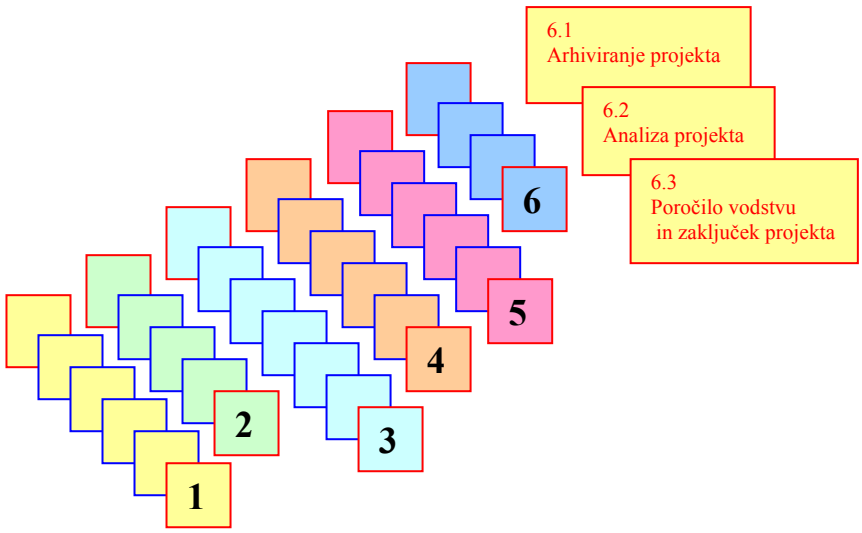
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3. **»Vzdrževanje«** se izvaja v skladu z dogovorjeno strategijo, ki jo naročnik in razvijalec potrdita s podpisom Dokumenta o vzdrževanju. Običajno razvijalec določeno obdobje nudi vsestransko pomoč pri uporabi IS, potem pa se preide na režim občasnih pregledov in interventnega odpravljanja napak. Z vidika razvijalca je koristno zastaviti tudi sistem sprotnega spremljanja delovanja IS. Ta lahko poleg ugotovljenih napak daje še veliko drugih koristnih podatkov. Projekt je praktično zaključen, vendar bo IS in PB treba vzdrževati tudi po njegovem koncu. Predvidevamo, da je za to poskrbljeno z ustreznimi aktivnostmi na ravni organizacije.
4. **»Zaključek uporabe«** je zadnja faza, ki se izvede na enak način kot drugi zaključki ciklov: zbiranje rezultatov, analiza, ukrepi (predvsem odziv na reklamacije naročnika) in poročilo vodstvu organizacije.

V dodatku A.2.8 se nahaja podrobnejša predstavitev šestega cikla.

3.4.4.10 Sedmi cikel: Zaključek projekta

Zadnji cikel poskrbi urejanje vse dokumentacije, arhiviranje, končno poročilo in formalen zaključek projekta. Predstavljen je na sliki 3.18, obsega pa samo tri faze:



Slika 3.18: Cikel 7: Zaključek projekta

1. **»Arhiviranje projekta«** poskrbi za shranjevanje rezultatov razvoja (vseh razvojnih verzij IS in PB) in opisov razvoja (vseh specifikacij in dokumentov). Slednje spravimo v enoten dokument (Knjigo razvoja) projekta ter jih arhiviramo.
2. **»Analiza projekta«** podrobno preveri vse merske podatke projekta in jih obdela s treh vidikov: z vidika poteka projekta, z vidika kakovosti in z vidika ocene dela posameznih članov projektne skupine. Posebej zapiše izvedene rezultate celotnega projekta ter popravi tabelo rezultatov vseh preteklih projektov. Posebej upošteva tudi zbrane predloge za izboljšavo procesa. S pomočjo Projektne knjige in vseh opravljenih analiz se v tej fazi pripravi Končno poročilo o projektu.
3. **»Poročilo vodstvu in zaključek projekta«** je zadnja faza procesa. V njej se na posebnem sestanku Končno poročilo predstavi vodstvu podjetja. Na tem lahko sodelujejo poleg članov projektne skupine tudi člani skupin SQA in SEPG ali naročnik. O poročilu se razpravlja in izpostavi njegove dobre ali slabe strani. Te se upošteva kot napotek za izboljšavo procesa ali vodenje projektov v prihodnje. Pripombe se zapiše tudi v Končno poročilo, ki se ga vstavi v Projektno knjigo in vse skupaj arhivira. S tem je projekt tudi formalno končan.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

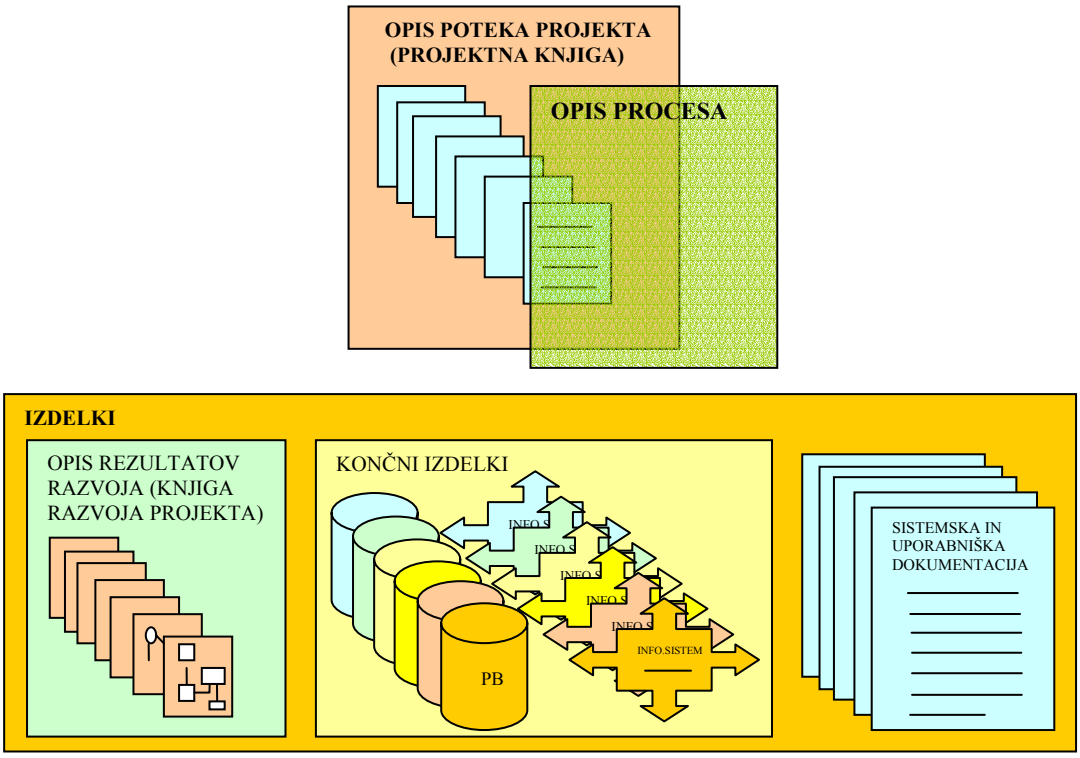
Formalni opis sedmega cikla se nahaja v dodatku A.2.9. V dodatkih A.2.10, A.2.11 in A.2.12 so podani tudi opisi treh načinov izvedbe splošnega koraka (glej razdelek 3.3.2), ki se uporabljajo za izvajanje vseh korakov prilagodljivega procesa.

3.4.5 Rezultati projekta

Predstavitev prilagodljivega procesa bomo zaključili s predstavitvijo rezultatov posameznih faz in ciklov, ki smo jih spoznali ob predstavitvi aktivnosti procesa, razvidni pa so tudi iz navodil v dodatku A.2. Med izvajanjem projekta jih nastane veliko, zato jih je treba sistematično opredeliti. Rezultati prilagodljivega procesa so treh vrst (slika 2.19):

- prva vrsta opisuje izvajanje projekta (recimo vloge, cilji, poročila),
- druga vrsta prikazuje značilnosti procesa (recimo ocene in merski podatki) in
- tretja predstavlja izdelke, ki nastanejo med razvojem. Med temi ločimo tri vrste:

  - opise rezultatov razvoja, ki predstavljajo tudi končni izdelek,
  - končne izdelke dveh vrst: IS in PB ter
  - različne vrste dokumentacije.



Slika 3.19: Tri skupine rezultatov prilagodljivega procesa razvoja PO

Jasno določeni rezultati so nujen sestavni del definiranega procesa in predstavljajo merljive mejnike pri izvajanju procesa. Kot je razvidno iz opisa procesa glavnina rezultatov predstavlja dokumentacijo različnih vrst. Eno izmed področij prilagoditve modela CMM za majhno organizacijo je ravno dokumentiranje [Pau99, Joh98]. Za tem se skriva razmislek, da lahko dobra komunikacija v majhni organizaciji nadomesti formalne dokumente. To je sicer do neke mere res, vendar je neformalno komunikacijo težko definirati, meriti in na njeni podlagi še težje karkoli napovedovati. Zato je bolje določiti tak nabor dokumentacije, ki bo zajemal dovolj, a ne preveč.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pri definiciji našega procesa smo določali rezultate na naslednji način:

- Izločili smo zares nepotrebne dokumente, vendar pri tem nismo bili pretirano strogi. Naše vodilo je bilo, da mora imeti vsaka faza jasno določen rezultat.
- Dokumente smo združevali v knjige, da bi bilo manj končnih rezultatov. Za knjige je iz opisa procesa razvidno, kaj mora v vsakem trenutku vsebovati. Popolnost in pravilnost se zato lažje formalno preverja, še posebej, ker se to počne nad združenimi rezultati ob zaključku ciklov.
- Predvideli smo učinkovito uporabo avtomatizacije. Dokumenti (poročila) bodo avtomatsko generirani na podlagi podatkov repozitorija. Predvidevamo tudi pregled dokumentov v repozitoriju na avtomatiziran način.
- Ker so vsa merjenja in oddaja rezultatov vezana na repozitorij, že pravilna izvedba vseh aktivnosti zagotavlja popolne podatke.

V nadaljevanju kratko predstavljamo nabor rezultatov vseh treh vrst, ki je nastal na tak način.

3.4.5.1 Opis poteka projekta

Projektna knjiga združuje vse dokumente, ki opisujejo potek projekta. Knjiga se zasnuje v prvem ciklu in postopoma dopolnjuje do zadnje faze zadnjega cikla, ko jo zaključimo. Najpomembnejši dokumenti v njej so opis projekta, vse vrste globalnih in podrobnih planov, vsa poročila o ciklih z rezultati analiz in končno poročilo o projektu. Podrobneje je Projektna knjiga predstavljena v dodatku A.3.1.

3.4.5.2 Opis procesa

Opis procesa obsega podatke, ki predstavljajo izvajanje procesa – torej predvsem merske podatke o izvajanju posameznih aktivnosti, plane in podobno. Vsi ti podatki so analizirani in shranjeni tudi v raznih poročilih o projektu, zato lahko večji del opisa procesa najdemo tudi v Projektni knjigi. V njej so recimo shranjene ocene v raznih planih, poleg teh pa še merski podatki, rezultati analiz ter poročila o testiranju.

Ostali del opisa procesa se nahaja v repozitoriju. Tam so predvsem vsi merski podatki posameznih članov, izpeljane in združene meritve ter pokazatelji. Poleg tega so za opis procesa zanimivi tudi drugi podatki, recimo podatki o delovanju sistema v času vzdrževanja. Nabor rezultatov, ki sodijo v opis procesa, je razviden iz dodatka A.3.2.

3.4.5.3 Izdelki

Ugotovili smo, da med izdelke štejemo:

a) vse opise rezultatov razvoja, ki predstavljajo končne izdelke,
b) vse končne izdelke in
c) vso (sistemsko in uporabniško) dokumentacijo.

a) Vsi **opisi rezultatov razvoja** se zberejo v Knjigi razvoja projekta. Med razvojem skušamo te opise najprej oblikovati v več ločenih samostojnih dokumentov - specifikacij, ki predstavljajo preverjene opise rešitve na določeni stopnji razvoja. V Knjigo razvoja projekta jih spravimo šele ob koncu projekta. Glavni deli Knjige razvoja projekta so Inicializacijski dokument, Strateški dokument, Specifikacija zahtev, Specifikacija načrtovanja, podrobni načrti za posamezne dele ter dokumenta o namestitvi in vzdrževanju.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

b) **Končni izdelki** so v našem primeru ločeni na dva dela: na programsko kodo IS in PB. Ker so vsi izdelki pod nadzorom upravljanja konfiguracij, se med razvojem pojavijo isti končni izdelki večkrat v različnih konfiguracijah. Med rezultate štejemo vse različice, vendar novejša različica vedno nadomesti staro, tako da ta pozneje ni več veljavna.

Informacijski sistem oziroma programska koda v njegovem okviru predstavlja procesni del rešitve, ki izvaja operacije nad PB. Lahko je v enem ali več delih, v njegov sestav pa spada uporabniški vmesnik in sistem pomoči. IS predstavljajo recimo prototipi IS in različne različice IS, ki se pojavljajo od zaključka cikla izgradnje naprej. Za preverjanje, nameščanje in vzdrževanje je včasih treba razviti tudi pomožne programe, ki sicer niso neposreden del končnega rezultata.

Drugi del končne rešitve je **podatkovna baza**, ki je nameščena na zunanjem pomnilniku strežnika in napolnjena s primernimi (testnimi ali pravimi) podatki. Enako kot pri IS je treba tudi pri PB predvideti pomožne dele in podatke za testiranje, namestitev in vzdrževanje.

c) Med izdelke štejemo tudi **dokumentacijo**, ki je namenjena uporabniku. Ta obsega dokumentacijo za opis končnih izdelkov (za procesni in podatkovni vidik), navodila za uporabo in učno dokumentacijo, ki služi za izvajanje osnovnega izobraževanja uporabnikov. Tudi dokumentacija je pod nadzorom konfiguracij.

Natančnejši pregled vseh treh vrst izdelkov se nahaja v dodatku A.3.3.

3.4.5.4 Pomožna dokumentacija

Repozitorij obsega tudi druge dokumente, ki so pomembni za izvajanje procesa. Med njimi zavzemajo posebno mesto različni standardi, ki nam definirajo postopke in izdelke. Druga pomembna skupina dokumentov so predloge, recimo za obrazce ali zabeležke. Ta del rezultatov je predstavljen v dodatku A.3.4.

3.5 Zaključek poglavja

Pri določanju procesa razvoja PO smo si zastavili dva cilja:

- proces mora formalno ustrezati, torej mora biti usklajen z zahtevami 4. stopnje modela CMM in posebnostmi specifičnega tehnološkega razvoja IS ter
- proces mora biti primeren za majhno organizacijo: enostaven, praktičen in nezahteven.

Da bi to dosegli, smo pri definiranju procesa uporabili samosvoj pristop: proces smo izpeljali z združitvijo dveh delnih rešitev. Prva je temeljila na preverjenem modelu TSP (in PSP), ki je namenjen skupinam razvijalcev in je usklajen z zahtevami modela CMM. Druga delna rešitev je predstavljala posebnosti enostavnega tehnološkega procesa za razvoj IS. Združeni proces smo nadgradili še z manjkajočimi zahtevami modela CMM, ki smo jih pred tem prilagodili zahtevam majhne organizacije. Te so obsegale predvsem aktivnosti, ki so pomembne za obvladovanje delovanja organizacije. S tem smo zagotovili ustrezno definiran proces, ki izpolnjuje oba cilja.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Prilagoditev modela CMM pomembno vpliva na zmanjševanje zahtevnosti procesa, vendar mora biti proces definiran še vedno popolno, torej z vsemi aktivnostmi. Kot tak proces ne more biti preprost. Vendar razvoj projektov v majhni organizaciji le redko poteka z izvajanjem čisto vseh predpisanih aktivnosti, temveč mnoge aktivnosti opušča. Tu sta se porajali dve vprašanji:

- kako zagotoviti, da izpuščanje aktivnosti pri razvoju ne bo vplivalo na ustreznost izvajanja procesa in predvsem
- kako mehanizem opuščanja učinkovito vgraditi v proces.

Odgovor na to daje inovativna izvedba našega procesa. Poimenovali smo jo prilagodljivi proces razvoja PO, saj omogoča prilagajanje sestave in izvedbe aktivnosti dejanskim zahtevam projekta. Še več, proces lahko dinamično prilagajamo medtem, ko ga izvajamo. Izvedba procesa temelji na treh izvirnih idejah:

- Večciklična struktura procesa omogoča, da celoten razvoj razdelimo na sedem delov - ciklov. Vsak izmed teh se začne s planiranjem in konča z analizo izvedbe, kar omogoča, da šestkrat med razvojem preverjamo doseganje ciljev in po potrebi ukrepamo. Poleg tega so cikli kratki in obvladljivi, omogočajo pa tudi postopno raziskavo zahtev, kar je pri razvoju IS zelo pomembno.
- Vsak korak, ki je najmanjša definirana aktivnost v procesu, se lahko izvede na tri načine: na enostaven ali zahtevnejši način enega izvajalca ter na zahteven način, kjer pri izvedbi sodeluje več razvijalcev. To omogoča, da podrobne izvedbe posameznih korakov ne planiramo natančno, temveč se sproti prilagajamo zahtevam.
- Pravila za prilagajanje procesa problemu določajo, kako prilagajati proces, da ta ostane še vedno ustrezen. Pravila določajo obvezne dele in načine, kako pravilno izvajati opuščanja.

Poleg tega smo v proces vgradili še druge učinkovite principe, recimo zahtevam prilagojene skupine in vloge, sistematičnost rezultatov aktivnosti, načine za zagotavljanje kakovosti (ponovni pregledi, zunanja preverjanja) in mersko spremljanje aktivnosti vsakega posameznika. Ravno slednje bo eden izmed temeljev za uvedbo merskega pristopa.

Proces smo v zadnjem delu poglavja tudi vsebinsko predstavili. Do primerne stopnje podrobnosti smo ga tudi formalno definirali v dodatku A, tako da predstavlja primerno podlago za vgradnjo merskega pristopa.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

4. POSTOPKOVNI DEL MERSKEGA PRISTOPA: MERSKI PROCESNI MODEL

4.1 Merski pristop

V majhni organizaciji za razvoj IS utemeljimo učinkovito merjenje na naslednjih izhodiščih:

- na najboljših izkušnjah in praksah organizacij za razvoj PO, ki so predstavljene v strokovni literaturi, ter
- na specifičnih zahtevah organizacije, torej na zahtevi po doseganju 4. stopnje po modelu CMM, posebnostih razvoja IS in omejitvah majhnih organizacij.

4.1.1 Izkušnje organizacij za razvoj PO

V strokovni literaturi obstaja veliko število opisov izkušenj in praks organizacij z uvedbo merjenja v proces razvoja PO (glej razdelek 2.1.6.2). Na podlagi najboljših praks je bil oblikovan standard ISO/IEC 15939: Informacijska tehnologija: Merski proces PO [ISO01] (glej razdelek 2.1.2). Standard upoštevata oziroma implementirata tudi:

- model CMMI [SEI02c, SEI02d], predvsem ključno področje Merjenje in analiza [Zub01] in še zlasti
- pristop PSM [Ric02] (glej razdelek 2.1.4.5), ki je primerno ogrodje za opis našega merskega pristopa.

Model CMMI na drugi stopnji definira ključno področje merjenje in analiza, ki predpisuje uvedbo merskega procesa in njegovo povezavo s procesom razvoja PO [SEI02c, SEI02d]. Merski proces je tako uveden tudi formalno, kar v prejšnjih različicah ni bilo res. Ključno področje določa naslednje cilje in aktivnosti:

Cilj 1: Uskladi aktivnosti merjenja in analize:
- določi merske cilje;
- določi meritve;
- določi zbiranje podatkov in postopke hranjenja;
- določi postopke analize.

Cilj 2: Poskrbi za rezultate merjenja:
- zbiraj merske podatke;
- analiziraj merske podatke;
- shranjuj podatke in rezultate;
- posreduj rezultate.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Cilj 3: Institucionaliziraj merski proces:
- vzpostavi pristop na ravni organizacije;
- planiraj merski proces;
- priskrbi vire;
- določi odgovornosti;
- usposobi izvajalce;
- upravljaj konfiguracije;
- identificiraj in upoštevaj pomembne lastnike;
- spremljaj in nadzoruj proces;
- objektivno ocenjuj postopke;
- preverjaj status z vodstvom.

Čeprav v našem primeru procesa razvoja PO nismo zasnovali na modelu CMMI, temveč na nekoliko enostavnejšem modelu CMM (glej razdelek 2.2.6), smo vseeno merjenje definirali kot poseben podporni proces, torej v skladu z zahtevami tega ključnega področja modela CMMI. Tudi v tem primeru smo zahteve (enako kot smo to storili pri ključnih področjih modela CMM) smiselno prilagodili potrebam in zmožnostim majhne organizacije.

Merski pristop smo utemeljili na **pristopu PSM** [McG01], ki na primeren način definira postopke in opis merjenja. Ustrezajo nam naslednje značilnosti pristopa:

- Pristop ustreza zahtevam standarda ISO/IEC 15939 in modela CMMI, kar zagotavlja popolnost in pravilnost.
- Pristop je opisan z merskim procesnim modelom in merskim informacijskim modelom, kar omogoča jedrnat in primerno podroben opis.
- Pristop PSM je zastavljen dovolj široko, da omogoča prilagajanje zahtevam majhnih organizacij [Mac01].
- Pristop definira merjenje predvsem kot aktivnost na ravni projekta, kar sicer ne ustreza zahtevam 4. stopnje CMM, vendar je primerno z vidika majhnosti organizacije.

Nekatere značilnosti pristopa PSM niso primerne za našo implementacijo merjenja. Mednje štejemo naslednje:

- Pristop je namenjen predvsem merjenju v okviru projekta. V našem primeru definiramo merjenje na organizacijski ravni, zaradi česar bo treba določene dele dopolniti.
- Pristop vsebuje načelo določanja meritev, po katerem vse meritve izbiramo na podlagi ciljev in težav projekta. V našem primeru so določene meritve obvezne, druge pa odvisne od ciljev in težav projekta ali organizacije.
- Določene dele pristopa je zaradi omejitev majhne organizacije smiselno izvajati drugače, bolj enostavno. Primerno je tudi natančneje določiti posamezne korake, čeprav to vnaša več omejitev.

Pristop je smiselno dograditi z jasno strukturo podatkov o meritvah in drugih z merjenjem povezanih podatkov, kar omogoča jasnejšo definicijo in nadzor. Taka struktura tudi ustrezno podpira predvideno avtomatizacijo.

Poleg tega je smiselno vgraditi dodatno funkcionalnost, ki omogoča enostavnejše spremljanje merskih rezultatov in analiz. Primer je recimo uvedba analize napovedovanja.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

4.1.2 Značilnosti majhne organizacije za razvoj informacijskih sistemov

V našem primeru moramo pri uvajanju merjenja upoštevati tri značilnosti:

- zahteve 4. stopnje modela CMM,
- zahteve specifične vrste PO in
- prilagoditev majhnosti organizacije.

Zahteve četrte stopnje modela CMM natanko opredeljujejo merjenje v organizaciji. Merjenje bo zahtevalo ustrezno strukturo merskega procesa in informacijskega modela. Cilji in tipične prakse posameznih ključnih področij tudi določajo vsebino merjenj, zato bodo izhodišče za izbiro nabora meritev, ki jih je treba (oziroma jih je smiselno) meriti. Kot rečeno je z zahtevami ključnega področja Merjenje in analiza modela CMMI določena tudi izvedba merjenja z merskim procesom.

Organizacija razvija **specifično vrsto PO** (IS), zato ima posebnosti (glej razdelek 2.3.3). V našem pristopu se bodo te odražale predvsem pri naboru meritev, ki bodo upoštevale procesni in podatkovni vidik. Tudi analiza rezultatov temelji na razmerjih med posameznimi rezultati, ki so specifični za tovrstne organizacije.

Majhnost organizacije zahteva predvsem enostavnost, določenost in uporabnost postopkov in rezultatov. Kaže se kot težnja po poenostavljanju in krajšanju tistih postopkov, ki obremenjujejo uporabnika. Uvedba učinkovitega merjenja je sama po sebi zahtevna, zato jo lahko poenostavimo samo do določene stopnje, če ne želimo okrniti osnovnega namena. Ker neposredno vključuje uporabnika (izvajalca aktivnosti), bo pomemben del aktivnosti vedno odvisen od njegovega truda. Vendar lahko z ustrezno organizacijo merjenja velik del aktivnosti tudi avtomatiziramo. V našem primeru predvidevamo, da z avtomatizacijo poenostavimo predvsem zajem merskih podatkov in izvedbo analiz.

4.1.3 Predstavitev merskega pristopa

V skladu z zapisanim je naš merski pristop prilagojen in nadgrajen pristop PSM. Ta je predstavljen z dvema modeloma: merskim procesnim modelom in merskim informacijskim modelom. Vsak model je podrobneje razčlenjen na več ravneh: procesni model na glavne merske aktivnosti, podaktivnosti in korake, vsebina informacijskega modela pa na merske entitete, atribute, (osnovne, izpeljane in združene) meritve ter pokazatelje.

V tem poglavju je podrobneje predstavljen prilagojen merski procesni model, ki ustreza našim zahtevam. Merski informacijski model je predstavljen v petem poglavju. Merski proces je predstavljen z dveh vidikov:

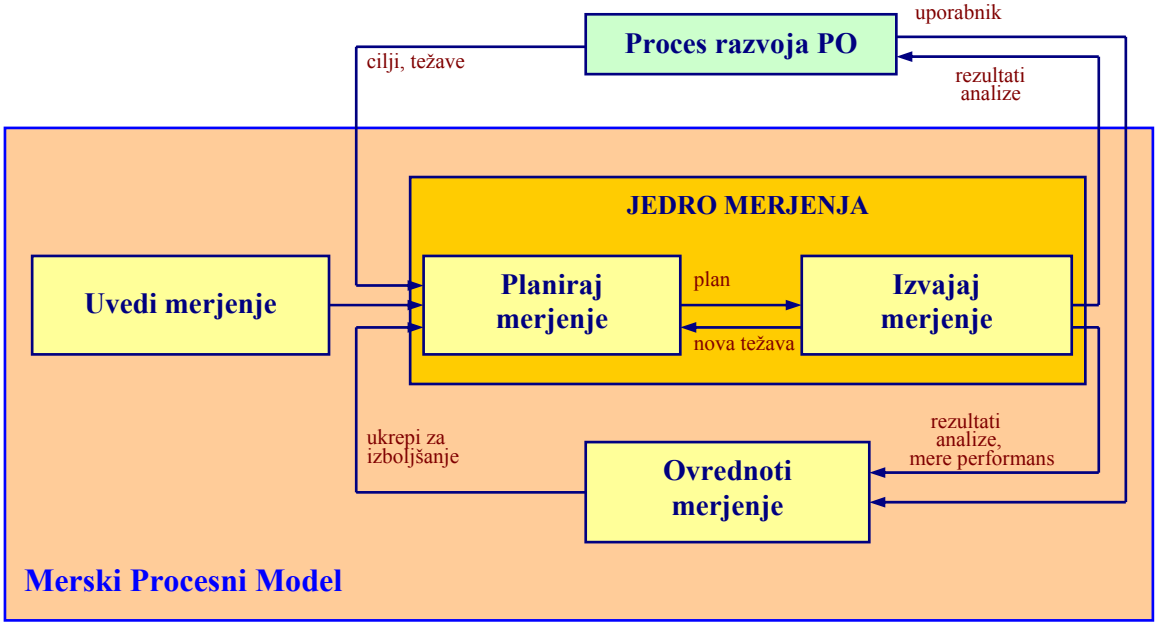
- Opis merskega procesnega modela je predstavljen primerjalno ob opisu posameznih delov pristopa PSM. Globalno gledano naš pristop obsega enake dele, vendar ima več pomembnih razlik na nižjih ravneh. Čeprav smo pristop PSM kratko že predstavili v podpoglavju 2.1.4.5, ga moramo zaradi primerjave z našim pristopom prikazati podrobneje. Ob predstavitvi vsake aktivnosti ali koraka je tako najprej kratka predstavitev pristopa PSM, nato pa opis našega pristopa pri tem delu. V opisu že sproti in posebej na koncu razdelka navajamo tudi umeščenost posameznih delov merskega procesa v prilagodljiv proces razvoja PO.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Posebnosti našega merskega procesa so posebej pozorno predstavljene. Pri tem se na pomembnih področjih spustimo precej v podrobnosti. Tak primer so recimo nekateri deli analize merskih podatkov (ocenjevanje in napovedovanje). Posebnosti vsebinske plati merskega modela (meritve in pokazatelji) so predstavljene v okviru naslednjega poglavja.

4.2 Merski procesni model

Slika 4.1 predstavlja merski procesni model pristopa PSM [McG01, DoD98, DoD02].

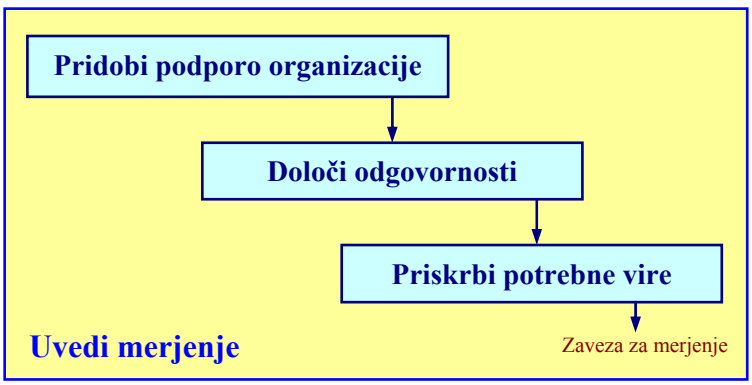


Slika 4.1: Merski procesni model po pristopu PSM [McG01, DoD02]

Enako strukturo obsega tudi naš pristop. Vsaka glavna aktivnost je podrobneje predstavljena v nadaljevanju.

4.2.1 Uvedba merjenja

Namen prve glavne aktivnosti pristopa PSM je zagotovitev pogojev za uspešno vgradnjo merjenja. Ta aktivnost se izvede ob začetku izvajanja merjenja. V okviru merskega procesa je na kratko in splošno določena. Pristop tako ne določa podrobnosti, temveč le naslavlja glavna področja, za katera je treba poskrbeti. Slika 4.2 predstavlja ta področja.



Slika 4.2: Glavna aktivnost »Uvajanje merjenja« v pristopu PSM

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

4.2.1.1 Pridobivanje podpore organizacije

Čeprav je pristop PSM namenjen izvajanju merjenja na ravni projekta, je podpora s strani celotne organizacije ključnega pomena za uspeh. Predvsem gre pri tem za podporo vodstva, pa tudi tehničnega kadra, ki pomaga pri izvedbi zahtevnejših postopkov. Podpora zajema naslednje aktivnosti:

- definicija merskih postopkov,
- zagotovitev virov,
- izbira jasnih in splošno sprejetih ciljev merjenja,
- preverjanje izvajanja merjenja,
- izvajanje analiz in
- uporaba rezultatov analiz.

Naš pristop izhaja iz zahtev modela CMM in kot tak uvaja merjenje na podoben način kot druga pomembna področja: celovito, z definiranimi pristopi, viri, usposabljanjem in preverjanjem kakovosti izvedbe.

Za razliko od pristopa PSM mi merjenje uvajamo kot aktivnost na organizacijski ravni. Zato je merski proces vgrajen v organizacijski proces razvoja PO in ob definiranju projektnega procesa razvoja PO privzamemo tudi ustaljen način merjenja. Ta predvideva ustaljene postopke, zagotavlja vire in načine preverjanja ustreznosti izvajanja. Na ravni organizacije sicer nimamo posebne merske skupine, vendar to nalogo zaupamo skupini SEPG. Ključnega pomena je zbiranje usklajenega nabora merskih podatkov, ki se za vse projekte zapisujejo v centralizirani repozitorij organizacije.

V okviru petega poglavja je predstavljena vsebina merjenja za naš primer ter pristop, s katerim smo ga določili. Ko smo nabor enkrat definirali in zapisali v organizacijski merski proces, ga lahko procesi samo povzemajo. S tem se poenostavi in standardizira uvajanje merjenja na ravni projektov.

4.2.1.2 Določanje odgovornosti

Merjenje je v različnih organizacijah lahko uvedeno na različne načine, vendar mora primerno definirati odgovornosti za več področij v okviru merjenja. Pristop PSM določa naslednje odgovornosti:

- vodstvo organizacije in projektov odgovarja za postopke merjenja, plane in definicije;
- člani projektne skupine in projektni vodja zbirajo podatke;
- odgovorna oseba, ki skrbi za bazo merskih podatkov je določena v projektnih planih;
- merski analitik ali projektni vodja skrbita za analize podatkov in poročanje.

V okviru našega merskega procesa so zadolžitve za vsa ta področja natanko določene. Za definiranje standardiziranih postopkov in merskih definicij na ravni organizacije je poskrbljeno s strani organizacijske skupine SEPG. Njihovo primernost potrdi vodstvo organizacije. Na ravni projekta prevzame vlogo skupine SEPG vodja procesa in merjenja, vlogo vodstva pa vodja projekta. Vsi člani lahko predlagajo izboljšanje merjenja preko predlogov za izboljšavo procesa.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Naš merski proces zahteva, da merske podatke zbirajo čisto vsi izvajalci aktivnosti projekta. Zbirajo jih na ravni posameznih korakov ali delov korakov ter jih zapisujejo v skupni repozitorij. Za repozitorij skrbijo posebni člani organizacijske skupine SEPG, na ravni projekta pa vodja procesa in merjenja. Ta je na ravni projekta zadolžen tudi za opravljanje analiz, pri čemer tesno sodeluje z vodjem kakovosti in vodjem projekta. Za poročanje vodstvu organizacije je zadolžen vodja projekta.

4.2.1.3 Skrb za potrebne vire

V okviru te podaktivnosti pristop PSM opisuje predvsem zagotavljanje potrebnih orodij in usposabljanja za izvajanje merjenja. Orodja lahko zelo olajšajo delo izvajalcem merjenja. Primerna je uporaba PB in tabelaričnih orodij, grafičnih orodij, predstavitvenih orodij, statističnih paketov in podobno. Orodja naj bodo enostavna in učinkovita.

Vsi člani organizacije potrebujejo vsaj osnovna znanja o merjenju. Razvijalci v okviru projekta potrebujejo poleg tega tudi podrobno poznavanje procesa zbiranja podatkov in upravljanja z njimi. Merski analitik mora poznati še izvajanje analize merskih podatkov.

V okviru izvedbe centralnega repozitorija merskih (in drugih) podatkov predvidevamo, da obstaja IS. Ta avtomatizira zbiranje, uporabo, preoblikovanje in poročanje o vseh merskih podatkih. Predvidevamo, da IS vsebuje različna orodja iz predstavljenih skupin.

Usposabljanje za izvajanje meritev se v okviru našega pristopa izvaja na enak način kot za ostala področja. Osnovna načela merjenja so definirana v okviru standardnih postopkov, ki so dosegljivi vsakemu članu projektne skupine. Čeprav je zbiranje zelo poenostavljeno zaradi avtomatizacije, je ustreznemu usposabljanju za zbiranje podatkov posvečeno veliko pozornosti. Vsi člani skupine SEPG so posebej seznanjeni z izvajanjem analiz in drugo uporabo merskih podatkov.

4.2.2 Planiranje merjenja

Planiranje merjenja je v okviru pristopa PSM zelo pomembno, zato je tudi zelo natančno definirano. Namenjeno je sistematičnemu zajemu merskih izhodišč, izbiri in opisu meritev ter vključitvi teh v proces razvoja PO (glej sliko 4.3). Planiranje se temeljito izvede prvič, potem pa se še večkrat ponovi v okviru izboljševanja procesa merjenja. Ob ponovitvah se običajno samo dopolni ali nadomesti izbrana merska izhodišča in meritve.



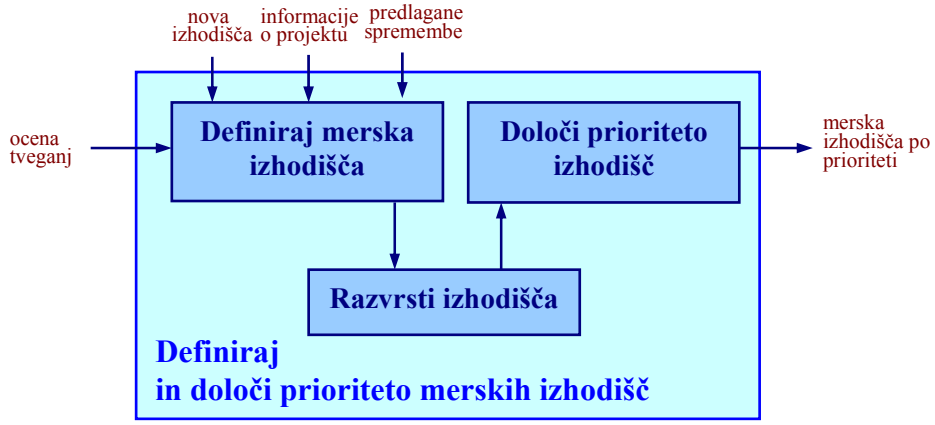
Slika 4.3: »Planiraj merjenje« v pristopu PSM

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

4.2.2.1 Definiranje merskih izhodišč in določanje njihove prioritete

Na sliki 4.4 so vidni posamezni koraki v okviru definiranja in določitve prioritete merskih izhodišč. Pristop PSM definira merska izhodišča **na podlagi ciljev projekta** ali **na podlagi ovir**, ki (lahko) motijo projektno skupino pri doseganju zastavljenih ciljev. Ovire so treh vrst:

- težave, s katerimi se sooča projekt (recimo pomanjkanje osebja),
- tveganja, ki lahko prerastejo v težave (recimo velika spremenljivost zahtev) in
- pomanjkanje informacij, zaradi česar ni mogoče objektivno presojati.



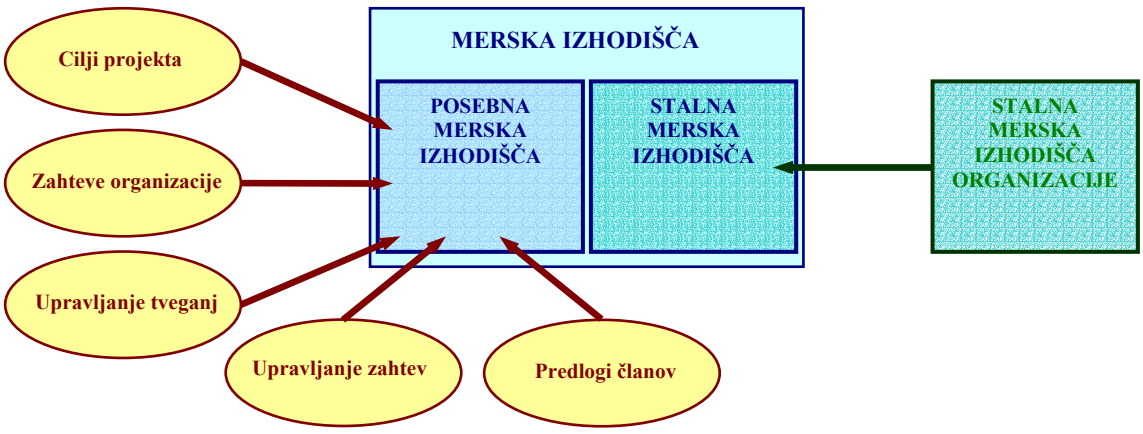
Slika 4.4: Prva podaktivnost v okviru »Planiranja merjenja« (pristop PSM)

Definiranje merskih izhodišč

Pristop PSM določa, da je treba merska izhodišča določiti v tem koraku na podlagi ciljev projekta ali na podlagi ovir. Najpogosteje je glavni vir merskih izhodišč analiza tveganj [Boe91], merjenje pa je za vsak projekt različno - prilagojeno dejanskim potrebam.

Naš pristop obravnava merska izhodišča drugače. Merska izhodišča so dveh vrst (slika 4.5):

- stalna merska izhodišča in
- posebna merska izhodišča.



Slika 4.5: Dve vrsti merskih izhodišč pri našem merskem pristopu

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Zahteve 4. stopnje modela CMM določajo, da je treba uvesti stalno mersko spremljanje določenih atributov procesa, projekta, virov in izdelkov. To nas privede do ideje, da je smiselno nekatera merska izhodišča določiti kot stalna, ki jih mora upoštevati vsak projekt. Med *stalna merska izhodišča* sodi recimo spremljanje obsega dela za izvajanje posameznih aktivnosti procesa ali merjenje velikosti izdelkov. Nabor stalnih merskih izhodišč mora biti majhen, a pokrivati vse potrebne vidike. Definiran je na ravni organizacije.

Poleg tega je nujno omogočiti tudi sprotno določanje merskih izhodišč na podoben način, kot to določa pristop PSM. V našem pristopu poznamo pet virov za *posebna merska izhodišča:*

- določeni cilji projekta, ki zahtevajo posebno mersko spremljanje;
- zahteve vodstva organizacije, ki so dane na tedenskih sestankih organizacije;
- upravljanje tveganj (aktivnost, ki se izvaja neprenehoma);
- upravljanje sprememb zahtev (aktivnost, ki se izvaja neprenehoma);
- predlogi, oddani s strani članov projektne skupine (recimo predlogi za izboljšavo procesa).

Glavnina posebnih merskih izhodišč je določena na podlagi ciljev projekta in analize tveganj. Pet virov omogoča, da imajo tako člani vodstva organizacije kot člani projektne skupine v vsakem trenutku možnost, da predlagajo mersko spremljanje določenih atributov. V naslednjih korakih planiranja merjenja se določi, če bo do tega zares prišlo. Kot bomo videli kasneje, naš pristop omogoča, da se merjenje uvede bodisi takoj ali pa ob začetku naslednjega cikla. To si lahko dovolimo, ker je na podlagi že zbranih merskih podatkov mogoče izvajati (nekatere) analize tudi za preteklost.

Razvrščanje merskih izhodišč

Postopno določanje meritev na podlagi izbranih merskih izhodišč je ena pomembnih prednosti pristopa PSM. Prvi korak na tej poti je razvrščanje merskih izhodišč v sedem skupin *splošnih merskih izhodišč*. Splošna merska izhodišča naj bi zajemala vsa smiselna področja meritev. Ker je splošnih merskih izhodišč sedem, se je relativno enostavno odločiti, kam sodi določeno mersko izhodišče. S to izbiro zelo omejimo določanje meritev v nadaljnjih korakih. Splošna merska izhodišča so naslednja:

- **poraba časa in napredovanje:** merjenje doseganja mejnikov in zaključevanja posameznih aktivnosti procesa razvoja;
- **viri in stroški**: merjenje porabe virov in stroškov glede na opravljeno delo;
- **velikost in stabilnost:** merjenje velikosti izdelkov PO in spremljanje stabilnosti z vidika funkcionalnosti ali zmožnosti;
- **kakovost izdelkov:** spremljanje kakovosti izdelkov z vidika odkrivanja in odstranjevanja napak, zanesljivosti ali drugih lastnosti kakovosti;
- **učinkovitost razvoja:** spremljanje učinkovitosti razvijalcev in zrelosti procesa v skladu z zahtevami modela CMM;
- **tehnična primernost:** merjenje tehničnih rešitev z vidika uporabe tehnologij, metod in orodij;
- **zadovoljstvo naročnika:** spremljanje parametrov, ki odražajo zadovoljstvo naročnika.

Preden opišemo naš pristop glede razvrščanja merskih izhodišč, se nekoliko pomudimo pri načinu izbiranja meritev na podlagi znanih izhodišč. Možne so tri rešitve:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

a) Izbor meritev na podlagi ciljev organizacije
Meritve lahko določimo na podlagi svojih potreb sami. Metoda GQM (glej razdelek 2.1.5.1) recimo določa sistematičen in preverjen pristop, kako na podlagi poslovnih ciljev (oziroma merskih izhodišč) določimo meritve. Žal pristop zahteva izkušnje in precej truda, zato ni najbolj primeren za sprotno določanje meritev.

b) Določitev splošne množice meritev na podlagi izkušenj
Različne organizacije imajo bogate izkušnje na področju merjenja in več jih je predlagalo nabore meritev, ki merijo tiste atribute, ki jih je smiselno meriti [SEL95, Pfl90, Gra94, Pau94]. Ker mora biti nabor za merjenje primerno majhen, ta običajno obsega nekaj glavnih meritev: velikost, obseg dela, stroške in število napak. Lepota tega pristopa je enostavnost, s katero določimo »primerne« meritve, vendar se pristop ne obnese, ker organizacija ne more prilagajati merjenja svojim potrebam. Učinkovito merjenje mora temeljiti na ciljih organizacije (glej razdelek 2.1.6.2).

c) Izbor meritev na podlagi ciljev organizacije, vendar iz množice primernih meritev
Tretja rešitev je nekje vmes. Bogate izkušnje drugih je vsekakor smiselno uporabiti, vendar je treba omogočiti tudi upoštevanje ciljev organizacije. Ker je vseh predlaganih meritev veliko, je treba določiti jasno in enostavno metodo, ki omogoča postopoma izbrati pravo meritev za vsak merski cilj (izhodišče) organizacije. Običajno na podlagi izkušenj izberemo nabor učinkovitih meritev, ki jih združujemo v skupine glede na določene skupne značilnosti. Te skupine so povezane z določenimi področji, v katera sodijo merski cilji. Ta rešitev zelo pomaga pri izbiri meritev na podlagi merskih ciljev, vendar odločitev vseeno zahteva izkušnje. Pristop PSM uporablja tako rešitev, ki bo natančneje opisana v nadaljevanju.

Rešitev pristopa PSM z razvrščanjem na podlagi splošnih merskih izhodišč je primerna tudi za naš pristop, vendar bi jo želeli še poenostaviti. Nekatera izmed sedmih splošnih merskih izhodišč se nam ne zdijo potrebna. Poleg tega pristop PSM v naslednjih korakih določa trinivojsko strukturo za razvrščanje meritev: vsakemu splošnemu merskemu izhodišču pripada več razredov meritev, vsakemu razredu pa meritve (glej razdelek 4.2.2.2).

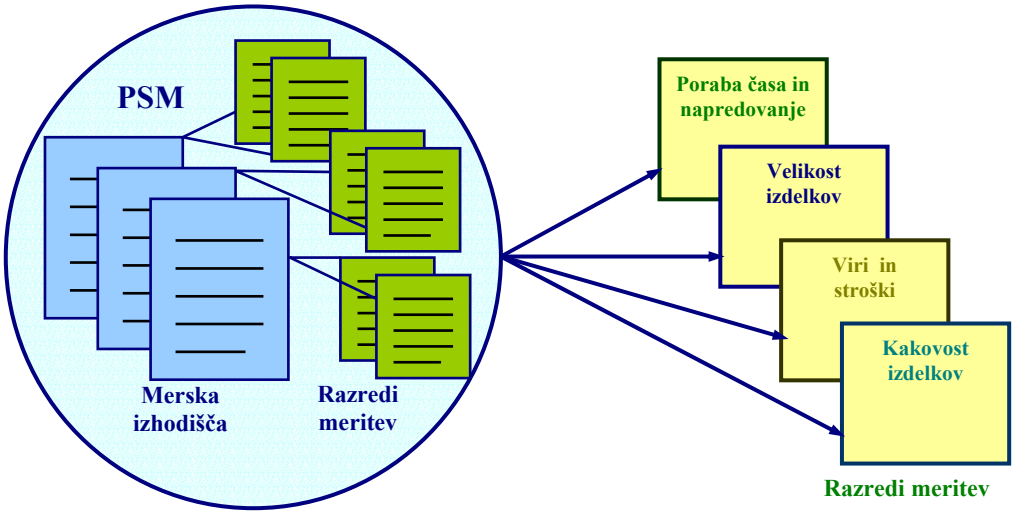
V našem primeru za razvrščanje predvidimo dvonivojsko hierarhijo. Splošna merska izhodišča in razrede meritev iz pristopa PSM tako združimo v eno, kar zaradi razumljivosti poimenujemo s terminom **razred meritev**. Naš pristop določa štiri razrede meritev (slika 4.6):

- **porabo časa in napredovanje,**
- **vire in stroške,**
- **velikost izdelkov in**
- **kakovost izdelkov.**

Ostala splošna merska izhodišča pristopa PSM lahko opustimo. Učinkovitost razvoja je namenjena spremljanju produktivnosti in zrelostne stopnje v skladu z zahtevami CMM. Prva je izpeljana meritev, določena na podlagi velikosti in porabe časa, zato jo lahko pripišemo v drugi razred. Merjenje zmožnosti v majhni organizaciji ni potrebno, saj ni podizvajalcev.

Tehnična primernost je smiselna za organizacije, ki uporabljajo različne tehnične pristope. V našem primeru je tehnološki proces določen, zato te skupine ne potrebujemo. Tudi zadovoljstvo naročnika je mogoče uvrstiti v obstoječe skupine, zato posebne skupine zanj ne uvajamo. Ker razredi meritev pristopa PSM samo podrobneje opredeljujejo posamezna splošna merska izhodišča, zato jih lahko brez škode opustimo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.6: Razredi meritev v našem merskem pristopu

Izbiro teh štirih razredov meritev potrjujejo tudi rezultati različnih raziskovalcev. [SEL95] recimo predlaga podoben nabor osnovnih meritev, enako tudi [Car92], ki posamezne meritve podrobno definira [Par92, Goe92, Flo92].

Določanje prioritete merskih izhodišč

Pristop PSM ne omejuje števila merskih izhodišč, zato je lahko teh veliko. Če vsakemu merskemu izhodišču določimo eno (ali več) meritev, lahko to zahteva velik obseg dela za merjenje, kar vpliva na drugo delo v okviru projekta. Poleg tega preveč meritev razprši pozornost merskih analitikov, kar ima za posledico manjšo pozornost pri pomembnih meritvah.

Pristop PSM rešuje to težavo z določanjem prioritet. Na ta način določijo najpomembnejša merska izhodišča, ki jim lahko posvetijo tudi več meritev. Nepomembna merska izhodišča v tem primeru opustijo. Prioriteto lahko določajo na podlagi mnenj razvijalcev ali s formalnimi metodami, recimo preko določanja verjetnosti pojavljanja določenega merskega izhodišča (če je to določeno na podlagi tveganja) in njegovim vplivom na potek projekta.

V našem primeru to težavo rešujemo drugače. Stalna merska izhodišča predpisujejo stalen del merskega spremljanja v okviru vsakega projekta in so obvezna. Pri posebnih merskih izhodiščih se sproti določimo, katera so pomembna in tista realiziramo.

Da bi bralec lažje spremljal opis našega merskega pristopa je potrebno, da na tem mestu idejno predstavimo glavne merske principe našega pristopa. Ta je bil deloma že predstavljen v okviru opisa procesa razvoja PO. Principi so naslednji:

- Izhajajoč iz omejenih virov majhne organizacije skušamo merjenje čim bolj poenostaviti. Tako skušamo določiti čim manjši obseg stalnih merskih izhodišč, vendar upravljan proces že sam po sebi zahteva precej merjenja vseh entitet v vsakem projektu. To pomeni, da na količino zbiranja merskih podatkov lahko vplivamo samo z zmanjševanjem števila posebnih merskih izhodišč. Tako sicer lahko merjenje zmanjšamo, vendar bo v vsakem primeru merjenja v projektu precej.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Enostavnost zato uvedemo na drugačen način: poenostaviti skušamo predvsem postopek merjenja, z obsegom pa se spopademo s pomočjo avtomatizacije skupnega repozitorija. V skladu z zahtevami prilagodljivega procesa razvoja PO moramo zbirati natančne merske podatke o vsaki aktivnosti na najnižji ravni. Zato določimo, da merske podatke zbirajo vsi posamezni izvajalci za dele aktivnosti, ki jih sami opravijo. Ker je struktura delitve dela jasna, lahko preko jasnih združevalnih pravil seštevamo rezultate, ki sodijo skupaj – recimo po izvajalcu, izdelku, delu procesa ali drugače. Na ta način dobimo rezultate za cele aktivnosti, korake ali faze. Posebnih merskih aktivnosti tako skoraj ni treba določiti.

- Enostavnost pristopa se skriva v tem, da je merjenje za izvajalce projektnih aktivnosti vedno enako. Kot bomo videli kasneje se mersko spremlja samo stalen nabor osnovnih meritev (velikost, porabo časa, obseg dela, stroške in kakovost oziroma število napak), ki so neposredno povezane z aktivnostjo, virom ali izdelkom. Merjenje je zato neposredno in rutinsko. V proces razvoja PO je uvedeno kot del izvajanja vsakega koraka: enostavnega ali zahtevnejšega (glej razdelek 3.3.2 ter dodatka A.2.10 in A.2.11). Merjenje se tako skrije v običajen način dela, ki je utemeljen na ustreznem inženirskem delu vsakega člana projektne skupine. Model PSP, ki definira učinkovit proces dela inženirja, predpisuje merjenje v podobni obliki [Hum95]. Lahko bi rekli, da naš pristop zahteva zgolj primeren način dela razvijalcev, pri čemer so zbrani merski podatki koristen stranski rezultat.

- Tak pristop ima še druge pozitivne učinke. Aktivnosti posameznikov so kratke, zato jih je mogoče natančno meriti. Tudi ob hkratni napaki vseh izvajalcev aktivnosti obstaja velika verjetnost, da bo ta v skupnem rezultatu manjša, kot bi bila ob enkratni nenatančni meritvi večjega obsega.

- Pristop z zbiranjem vseh merskih podatkov posameznikov je učinkovitejši zaradi majhnosti organizacije in majhnosti projektov, ki jih ta razvija (glej razdelek 2.3.1). Pristop omogoča tudi izvedbo novih analiz nad podatki o preteklih projektih.

- Predvidevamo, da je zajem merskih podatkov avtomatiziran, podobno pa tudi analiza in prikaz rezultatov. Vse temelji na skupnem repozitoriju, v katerem so poleg merskih podatkov tudi vsi meta podatki o merjenju, projektu, izvajalcih, virih, izdelkih in drugem. To omogoča povezovanje rezultatov in veliko načinov analize.

Končni rezultat prve podaktivnosti v okviru planiranja meritev je seznam stalnih in posebnih merskih izhodišč. Prva so bila na ravni organizacije določena enkrat in se uporabljajo vedno enako. Planiranje je zares uporabljeno samo za posebna merska izhodišča projekta. Vsako mersko izhodišče sodi v enega izmed štirih razredov merjenja.

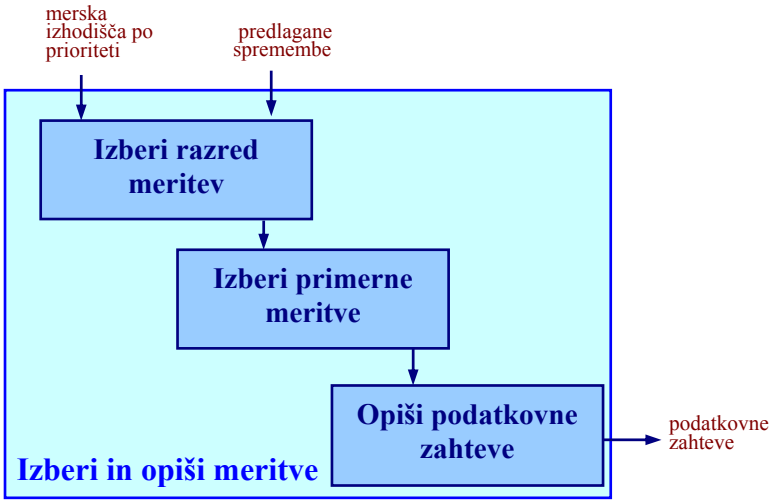
4.2.2.2 Izbira in opis meritev

V okviru te podaktivnosti dokončno izberemo in določimo meritve. Izbiro meritev po pristopu PSM opravimo postopoma, preko določanja razreda meritev. Slika 4.7 prikazuje posamezne korake te podaktivnosti.

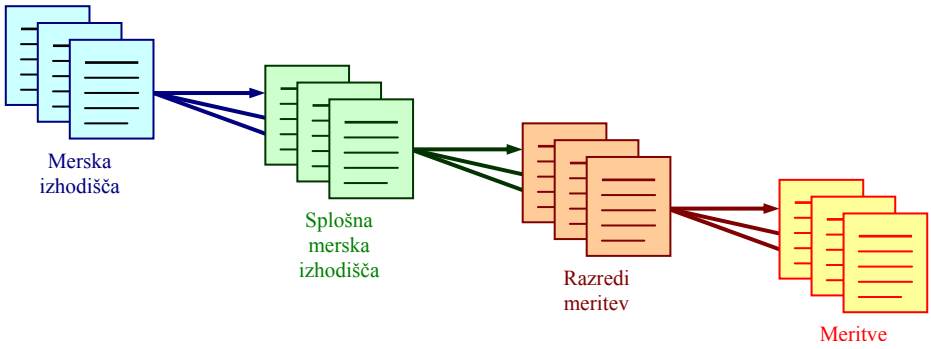
Izbira razreda meritev

Pristop PSM omogoča izbiro učinkovitih meritev iz množice primernih meritev, pri čemer za izhodišče vzame cilje projekta. Mehanizem izbire obsega tri korake (glej sliko 4.8):

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.7: Druga podaktivnost v okviru »Planiranja merjenja« (pristop PSM)



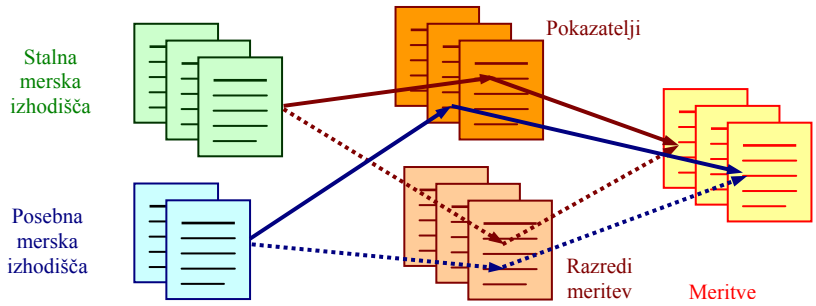
Slika 4.8: Način izbire meritev v pristopu PSM

- Določitev enega izmed sedmih splošnih merskih izhodišč za vsako definirano mersko izhodišče.
- Določitev razreda meritev znotraj splošnega merskega izhodišča. Vsako splošno mersko izhodišče lahko predstavimo z različnih vidikov in te predstavljajo razredi meritev. Vsakemu razredu pripada množica meritev, ki naslavljajo vidik razreda. Splošno mersko izhodišče običajno vsebuje dva ali tri razrede meritev.
- Določitev najprimernejših meritev znotraj razreda meritev. Izbira meritev pomeni tudi odločitev o merskih podatkih, ravni zbiranja in drugem.

Pristop PSM predvideva, da je množice meritev mogoče po potrebi tudi dopolniti. Projekt ne sme implementirati vseh meritev, temveč zgolj tiste, ki najbolj ustrezajo njegovim zahtevam. Da bi bila izbira razreda meritev enostavnejša, pristop PSM pripisuje vsakemu razredu tudi vprašanja, na katera meritve iz tega razreda odgovarjajo. Na tak način se je mogoče enostavneje odločiti med podobnimi razredi.

Čeprav smo tudi v našem primeru uporabili izbiro meritev v treh korakih, je princip popolnoma drugačen. Izbira meritev namreč poteka preko izbire pokazateljev, ki predstavljajo vezni člen med (stalnim ali posebnim) merskim izhodiščem in meritvijo. Postopek sodi v kombinirano metodo za določanje meritev in je podrobno opisan v razdelku 5.4.3.3. Slika 4.9 prikazuje skico rešitve.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.9: Način izbire meritev v našem merskem pristopu

Stalna merska izhodišča so znana vnaprej. Določena so v organizacijskem merskem procesu, ki ga mora uporabljati vsak projekt. Zaradi tega so znani tudi ustrezni razredi meritev in najprimernejše meritve. Za vsako mersko izhodišče je tako natančno določen razred meritev in tudi meritve. Neposredne meritve so določene kar s stalnim naborom (osnovnih) meritev. Pri posebnih merskih izhodiščih lahko dodamo še katero posredno meritev. Ker stalna merska izhodišča pokrivajo širok spekter meritev, pogosto drugega sploh ne bomo merili.

Osnovne meritve so vnaprej določene, tako da se meritev nanaša predvsem na združene meritve, izpeljane meritve in pokazatelje. S tem dejansko vplivamo samo na večje možnosti analiz. Ne glede na to pa težimo k majhnemu naboru (vseh) meritev. Posamezne meritve in postopek njihovega vključevanja v nabor bo podrobno predstavljen v okviru 5. poglavja.

Izbira primernih meritev

- Vsak razred meritev vsebuje množico meritev in v tem koraku izmed teh izberemo eno ali več meritev, ki ustrezajo merskim izhodiščem. Vsaka meritev je primerno opisana, tako da omogoča izbiro po različnih kriterijih.

Naš pristop izbire meritev temelji na postopnem izbiranju pokazateljev (glej razdelek 5.4.3.3). Neformalno gledano pa ponuja tudi možnost s pomočjo uporabe repozitorija, v katerem so meta podatki o vseh merljivih entitetah. Tako lahko razberemo razmerja med posameznimi aktivnostmi, izdelki in fazami razvoja, plani opravil pa za aktivnosti določijo še izvajalce. Če želimo meriti določene značilnosti izbrane faze, lahko na podlagi izbrane faze določimo njej pripadajoče aktivnosti in izdelke, iz plana opravil pa posamezne izvajalce. Če se določena meritev nanaša na izdelek, ki ne pripada izbrani fazi, potem ni primerna za merjenje tega merskega izhodišča.

Opis podatkovnih zahtev

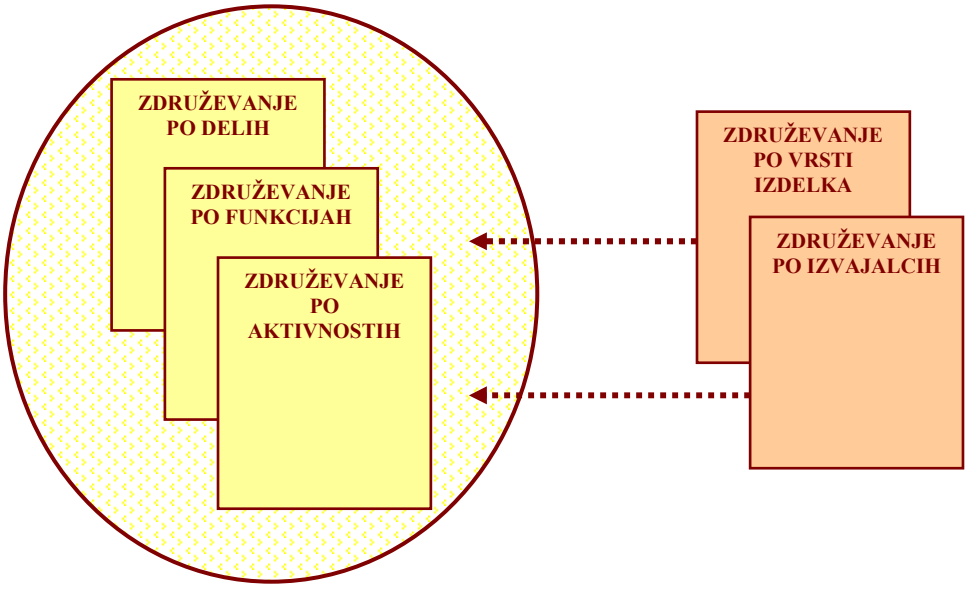
Pristop PSM v tem koraku natanko opiše posamezne meritve, s čemer določi merske podatke, stopnjo podrobnosti, pogostost zbiranja, format podatkov in podobno. V okviru tega koraka izbere tudi strukturo združevanja, ki je lahko naslednjih vrst:

- **združevanje po delih**: združimo vse dele določene entitete, recimo vse module združimo v celotno programsko kodo;
- **združevanje po funkcijah**: združimo aktivnosti, ki izvajajo isto funkcijo, recimo vsa testiranja (po delih, uporabniško testiranje ipd.);
- **združevanje po aktivnostih**: združimo aktivnosti v okviru posameznih faz ali ciklov, recimo vse aktivnosti iz faze zajema zahtev.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

V našem primeru je večji del opisa vnaprej določen v repozitoriju. Pravilo določa, da zbirajo podatke posamezniki na najnižji ravni in sicer ob izvedbi vsake aktivnosti. Podatke sproti vnašajo v merski del repozitorija ob podpori IS repozitorija. Format merskih podatkov je določen na podlagi opisa v repozitoriju. V repozitoriju so določene tudi strukture združevanja. Poleg opisanih treh vrst v našem primeru določimo še dve (slika 4.10):

- **združevanje po vrsti izdelka**: združimo izdelke, ki ustrezajo procesnemu ali podatkovnemu opisu končnega izdelka in
- **združevanje po izvajalcih**: združimo aktivnosti, ki jih je izvedel določen razvijalec.



Slika 4.10: Združevalne strukture v našem merskem pristopu

Pri izvajanju koraka moramo preveriti ujemanje izbranih meritev in podatkov o meritvah, ki so zbrane v repozitoriju. To bo razvidno iz meta modela (glej podpoglavje 5.3).

Ob zaključku druge podaktivnosti imamo izbran in opisan nabor meritev, ki ustrezajo merskim izhodiščem. Poleg vseh osnovnih meritev, ki predstavljajo stalen del nabora, bo zraven še več združenih in nekaj izpeljanih meritev.

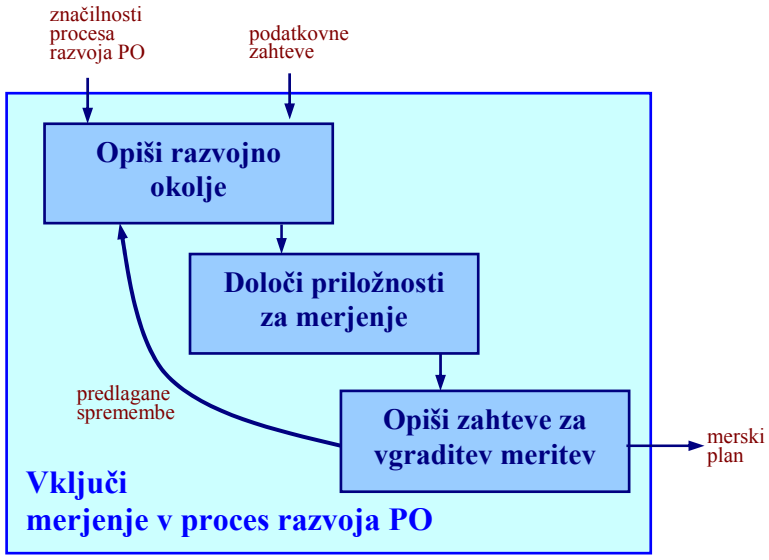
4.2.2.3 Vključevanje merjenja v proces razvoja PO

Izbrane meritve moramo na primeren način vključiti v obstoječi proces razvoja PO. To je lahko zahteven postopek, ki zahteva prilagajanje pravkar določenega nabora meritev. Na sliki 4.11 so prikazani posamezni koraki te podaktivnosti. V prvem koraku preverimo razvojno okolje (proces razvoja PO), nato pa še možnosti za vključitev meritev vanj. Na koncu meritve vgradimo, kar opišemo v merskem planu.

Opis razvojnega okolja

Proces razvoja PO v veliki meri odloča o uspešnosti merjenja, zato ga je treba pred vgraditvijo meritev dodobra spoznati. Pomembni dejavniki so življenjski cikel, struktura izdelkov, tehnologija, obstoječi standardi in meritve, ki se uporabljajo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.11: Tretja podaktivnost v okviru »Planiranja merjenja« (pristop PSM)

V našem primeru je bil proces razvoja PO opisan v 3. poglavju. Njegova glavna posebnost je možnost dinamičnega prilagajanja zahtevam projekta, ki jo je treba ob uvajanju merjenja smiselno upoštevati. Na izvajanje merjenja pomembno vpliva naš pristop s planiranji ob začetku in analizami ob zaključku ciklov.

Določanje priložnosti za merjenje

Kadar vgrajujemo merjenje v obstoječ proces razvoja PO, ki je že vseboval merjenje v določeni obliki, je pomembno ohraniti zbrane merske podatke. Ti se lahko nahajajo v različnih oblikah. Pristop PSM loči med tremi vrstami merskih podatkov:

- **zgodovinski podatki**: (običajno) osnovni podatki iz preteklih projektov;
- **planski podatki**: podatki iz planov (ocene) oziroma s strani vodstva potrjene ocene;
- **dejanski podatki**: izmerjeni podatki, ki odražajo resnične rezultate.

V našem primeru ne moremo govoriti o obstoječem merjenju, ker tako proces razvoja kot merski proces na novo določamo. Ta korak v našem primeru torej ni potreben.

Naš merski pristop zbira vse tri vrste merskih podatkov. Dejanski podatki se sproti zbirajo za vse projekte organizacije na enak način. Ko je projekt končan, postanejo del zgodovinskih podatkov. Ker se dejanski merski podatki zbirajo za vsako projektno aktivnost (korak ali del koraka) na ravni posameznih izvajalcev, bodo taki tudi zgodovinski podatki. Določeni podatki pa bodo še zbrani na način, ki bo omogočal učinkovitejše kasnejše analize. Za vsak projekt in posamezne cikle se shranjujejo tudi ustrezni planski podatki, predvsem ocene.

Poleg tega naš pristop trem vrstam merskih podatkov doda še četrto: **napovedi**. Napoved definiramo kot najboljšo oceno izbranega atributa, ki jo v nekem trenutku lahko damo. Napovedovali bomo samo ključne atribute, kot so recimo velikost končnega izdelka ali potreben obseg dela za celotne cikle. Na začetku bo napoved kar enaka planirani vrednosti, potem pa se bo postopoma z razvojem popravljala na podlagi novih znanih dejstev. Ob koncu projekta bo napoved enaka dejanski izmerjeni vrednosti. Pomembna značilnost napovedi je v tem, da se ne shranjuje (kot recimo ocene), temveč obstaja le trenutna vrednost. Podrobneje je napovedovanje predstavljeno v razdelku 4.3.4.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Opis zahtev za vgraditev meritev

V tem koraku natančno določimo način izvajanja meritev, kar opišemo v merskem planu. Pristop PSM določa merski plan za projekt, ki je lahko formalno ali neformalno določen. Slika 4.12 prikazuje primerek merskega plana v skladu z zahtevami pristopa PSM.

Plan merjenja PO

1. Uvod
- Namen
- Cilj

2. Opis projekta
- Značilnosti tehnološkega razvoja
- Značilnosti upravljanja projekta

3. Merski pristop
- Vključitev merjenja v tehnološki in upravljalski proces
- Zbiranje in uporaba podatkov
- Odgovornosti za zbiranje
- Komunikacijska struktura v organizaciji

4. Opis merskih izhodišč
- Seznam vseh merskih izhodišč po prioriteti

5. Meritve PO in njihove specifikacije (za vsako meritev)
a) Ime meritve
b) Mersko izhodišče, ki ga rešuje
c) Podatkovni elementi
d) Atributi
e) Združevalne strukture
f) Raven podrobnosti
g) Kriterij za štetje
h) Definicija podatkov
i) Metoda ocenjevanja
j) Mehanizem zbiranja in poročanja
k) Vir podatkov
l) Pogostost zbiranja in poročanja

6. Združevalne strukture
- Združevalna struktura po delih
- Združevalna struktura po funkcijah
- Združevalna struktura po aktivnostih

Slika 4.12: Primer merskega plana (pristop PSM)

Merski plan na ravni organizacije predvideva PSM samo v primeru, če gre za organizacijo, ki izvaja enake vrste projektov z enakim procesom in tehnologijo. V tem primeru omenja potrebo po normalizaciji podatkov ter naslavlja potrebe vodstva organizacije: njihove merske zahteve in poročanje vodstvu.

V našem primeru bomo merski plan okvirno opredelili v okviru opisa merskega informacijskega modela v 5. poglavju. Pri tem se bomo predvsem posvetili ustrezni definiciji vsebine merjenja (recimo z meta modelom) oziroma popolnosti predstavitve le-te, manj pa formalni predstavitvi merskega plana. Določili bomo torej vse pomembne dele merjenja, a jih ne bomo posebej sestavljali v merski plan. S tega stališča lahko predvidevamo, da je merski plan v podobni obliki, kot ga določa pristop PSM.

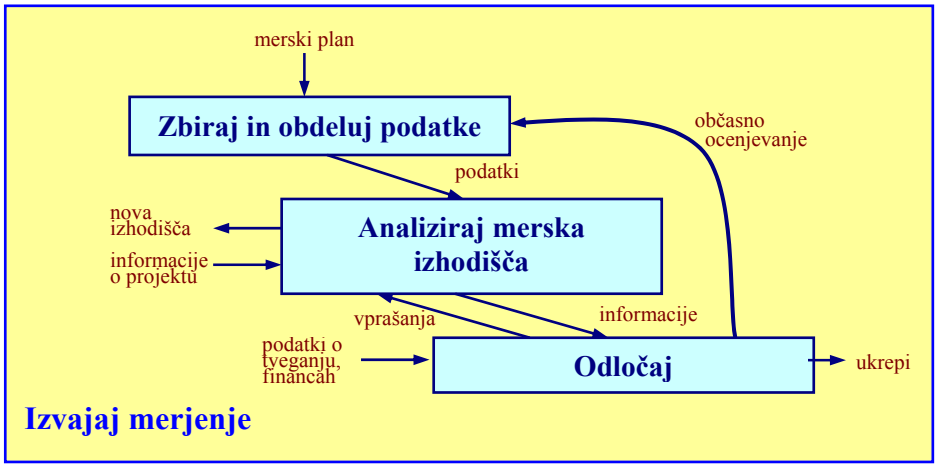
4.2.3 Izvajanje merjenja

Izvajanje merjenja je tista glavna aktivnost v okviru pristopa PSM, ki v praksi izvaja zastavljeni merski plan. Je dokaj natančno opredeljena in obsega tri podaktivnosti: zbiranje podatkov, njihovo analizo ter ukrepanje (glej sliko 4.13). Podaktivnost zbiranje podatkov je pomemben del izvajanja meritev, vendar je predvsem analiza tista, ki ji je treba posvetiti posebno pozornost. Izvajanje merjenja se ponavlja in postopoma spreminja s spremembami zahtev in merskih planov.

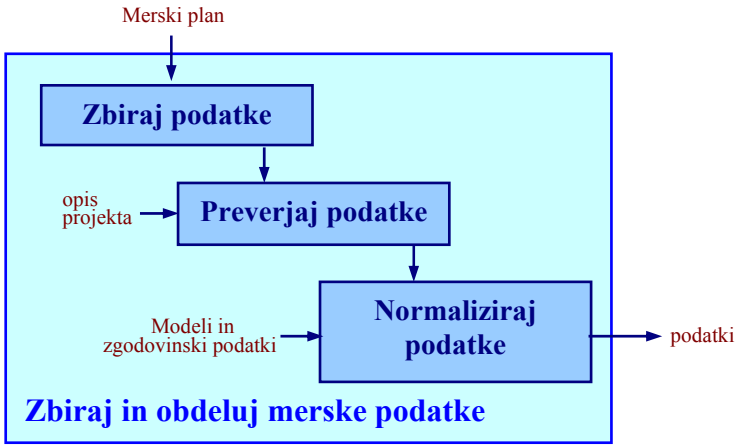
4.2.3.1 Zbiranje in obdelava podatkov

Točni in popolni merski podatki so predpogoj za kakršnokoli odločanje. V tej podaktivnosti poskrbimo za to. Na podlagi merskega plana zbiramo podatke, jih preverjamo in po potrebi še normaliziramo (glej sliko 4.14). Vsak korak bo podrobneje predstavljen v nadaljevanju.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.13: »Izvajanje merjenja« v pristopu PSM



Slika 4.14: Prva podaktivnost v okviru »Izvajanja merjenja« (pristop PSM)

Zbiranje podatkov

V okviru tega koraka zbiramo »surove« podatke. Glavnina podatkov bo prišla s strani razvijalcev, predvsem iz planov opravil in poteka dela, projektne dokumentacije in opisov konfiguracije oziroma izdelkov. Zbiranje podatkov vzpostavlja mehanizem za dostop do podatkov, ki določa:

- kako se bodo podatki zbirali, shranjevali ter kako bo mogoč varen dostop do njih ter
- kdaj in kako pogosto se dostopa do podatkov.

V našem primeru se podatki zbirajo v centralnem repozitoriju, ki služi celotni organizaciji. Odgovornost za točnost in popolnost podatkov je zaupana posameznim izvajalcem. Zajem podatkov je avtomatiziran, vsak podatek se ob zajemu opremi s časovno oznako. Razvijalci so dolžni oddati podatke neposredno po opravljeni aktivnosti. Vodja procesa in merjenja, ki skrbi tudi za planiranje, zbiranje in analizo merskih podatkov, izvaja občasna preverjanja. Delne analize izvaja tedensko, obsežnejše pa ob zaključkih ciklov in ob zaključku projekta.

Dostop do podatkov imajo vsi tisti člani projekta ali organizacije, ki jim to po vlogi pripada. Podatke si lahko ogledajo v »surovi« obliki ali s pomočjo ustreznih združenih in izpeljanih meritev ali pokazateljev.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Preverjanje podatkov

Poseben korak je namenjen preverjanju kakovosti podatkov. Preverja se točnost in ustreznost zapisa vnešenih podatkov. Preverjamo predvsem naslednje stvari:

- usklajenost podatkov s trenutnim stanjem,
- primernost podatkov za združevanje v okviru združevalnih struktur,
- enolično upoštevanje enot v vseh delih,
- tip in obseg vnešenih podatkov,
- popolnost podatkov,
- vpliv na že zbrane podatke (recimo nadomeščanje) in
- splošen vtis o pravilnosti podatkov.

Normalno je, da med zbiranjem merskih podatkov prihaja do določenih napak in te skušamo odpraviti. Izkušnje kažejo, da so preveč pravilni podatki ob zajemu sumljivi, saj pogosto ne odražajo resničnosti. To kaže na hujši problem: neustrezno zastavljeno merjenje.

Naš pristop namenja kakovosti merskih podatkov veliko pozornosti, saj to zahteva že nizka raven zbiranja podatkov. Pri tem vključuje enaka preverjanja kakovosti vnešenih podatkov kot pristop PSM na dva načina:

- avtomatsko preverjanje tipa, obsega, enote in popolnosti merskih podatkov na podlagi meta podatkov o vrsti podatka, ki se zbira, ter
- preverjanje vseh drugih vidikov ustreznosti podatkov s strani vodje procesa in merjenja.

Avtomatsko preverjanje je namenjeno formalnemu pregledu pravilnosti podatkov. Podatki se preverjajo neposredno ob vnosu, tako da napačnih vrednosti ni mogoče vnesti. Vodja procesa preverja bolj vsebinski vidik kakovosti podatkov. V ta namen uporablja določena poročila, ki mu jih ponuja IS repozitorija, poslužuje pa se tudi neformalnih metod (recimo [Rož00]). Primer ocene kakovosti podatkov je prikazan v [Mah01], [Str01] pa prikazuje zanimivo metodo za reševanje težav ob manjkajočih podatkih. Vodja procesa in merjenja po potrebi uporablja tudi takšne prijeme.

Normalizacija podatkov

Pristop PSM definira normalizacijo podatkov predvsem s stališča pretvarjanja enot posameznih izmerjenih vrednosti, da bi tako omogočili združevanje podatkov. Normalizacija je lahko enostavna ali zahtevna. Primer enostavne normalizacije je recimo pretvorba vrednosti zapisane v minutah v vrednost zapisano v urah, zahtevna normalizacija pa bi bila pretvorba velikosti programa zapisanega v številu vrstic iz programskega jezika C v Ado. V tem primeru je treba upoštevati različni vrsti pravil za štetje vrstic.

Naš pristop obravnava normalizacijo nekoliko drugače, bolj v funkciji napovedovanja. Nekatere pomembne atribute ključnih merskih entitet (recimo velikost končnega izdelka ali celoten obseg dela projekta) ves čas razvoja spremljamo z napovedmi. Iz združevalne strukture je razvidno, da so določene delne meritve odvisne od meritve končnega izdelka. Med napovedovanjem si s temi delnimi meritvami pomagamo do čim boljše ocene. Pri tem moramo zaradi primerljivosti pretvarjati izmerjene vrednosti delnih meritev v ustrezne vrednosti, ki so izražene v enoti glavne. To pretvorbo imenujemo normalizacija. Pri tem ne gre zgolj za oceno, temveč za zahtevno pretvorbo na podlagi zgodovinskih podatkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Primer:

Ob zaključku cikla Raziskava zahtev je opis končnega izdelka podan v specifikaciji zahtev, ki obsega 4 pomembne dele: seznam zahtev, podatkovni model zahtevanega sistema (ER diagram), procesni model zahtevanega sistema (diagram podatkovnih tokov) in diagram hierarhije funkcij (funkcionalna dekompozicija). Specifikacija zahtev predstavlja opis končnega izdelka in z njeno pomočjo lahko ocenimo velikost končnega izdelka (procesnega ali podatkovnega dela). Znana je velikost vseh štirih delov specifikacije zahtev:

- seznam zahtev: število posameznih zahtev (lahko tudi število vrstic zahtev);
- ER diagram: število entitet in razmerij (lahko tudi atributov);
- diagram podatkovnih tokov: število elementarnih procesov, zunanjih entitet, skladišč podatkov in podatkovnih tokov na najnižji ravni;
- diagram hierarhije funkcij: število funkcij (lahko tudi število ravni).

Velikost vsake od predstavljenih vrednosti normaliziramo tako, da s pomočjo podatkov preteklih projektov določimo, kakšna velikost končnega izdelka bi ustrezala vrednosti tega dela. Recimo: če želimo oceniti procesni del, je enota za velikost število vrstic programske kode. Če zgodovinski podatki kažejo, da v povprečju vsaka zahteva ustreza 300 vrsticam programske kode, potem lahko na podlagi 30 zahtev iz seznama zahtev sklepamo, da bo programska koda na koncu obsegala 9000 vrstic. Drugače rečeno, normalizacija pretvori 30 zahtev v 9000 vrstic programske kode.

Če bi to vrednost uporabili za napoved, bi verjetno precej zgrešili. Očitno je, da bi recimo normalizirana vrednost števila elementov diagrama podatkovnih tokov objektivneje napovedovala končno velikost procesnega dela. Naloga napovedovanja je presoditi, katera ali katere vrednosti (ne samo iz specifikacije zahtev) so glede na zgodovinske podatke najbolj povezane z izbranim vidikom velikosti programske kode. Utežene normalizirane vrednosti teh potem sestavi v končno napoved.

Naš pristop definira posebno tabelo v repozitoriju, v kateri so zbrani podatki, ki omogočajo normalizacijo in s tem ocenjevanje oziroma napovedovanje. Tabela tako vsebuje posamezne koeficiente za pretvorbo in uteži. Pomembno je, da se tabela avtomatsko ažurira na podlagi sprememb zgodovinskih podatkov. Podroben opis tabele je predstavljen v razdelku 4.3.1.3.

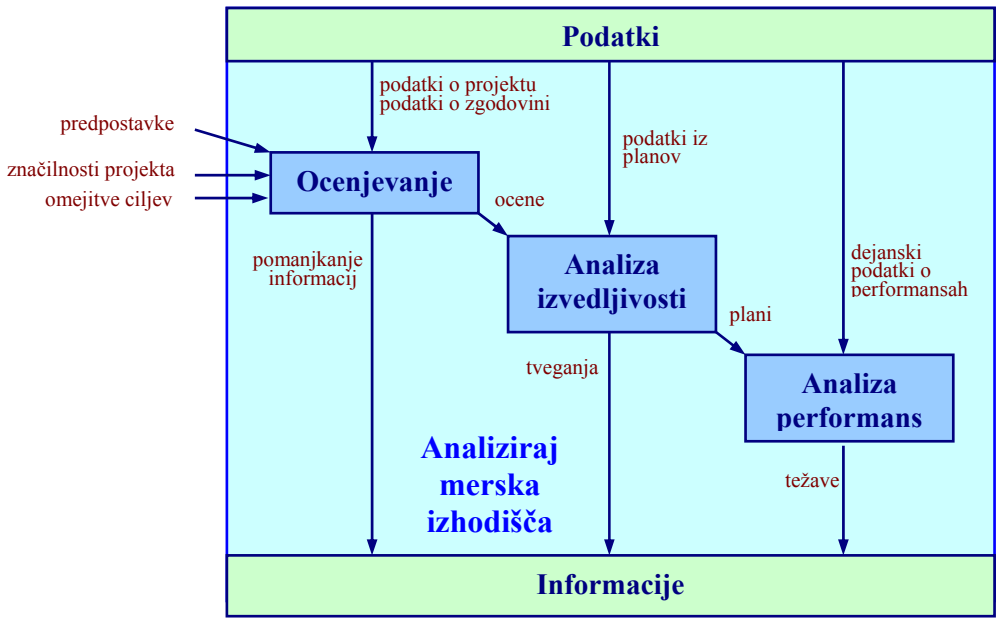
Naš pristop predvideva tudi pretvarjanja med različnimi enotami iste količine. Na ravni organizacije obstaja enoten standard, ki določa, kako se izvajajo meritve. Predvidi lahko več enot, med katerimi obstajajo stalna razmerja. Tako recimo porabo časa merimo predvsem v minutah, lahko pa tudi v urah (1 ura je 60 minut) ali dnevih (1 dan ustreza 8 delovnim uram oziroma 480 minutam). Podatki se sproti pretvarjajo in shranjujejo v najprimernejši enoti.

4.2.3.2 Analiza merskih izhodišč

Analiza je ključna aktivnost celotnega merjenja. Različni avtorji navajajo, da se pri merjenju PO v praksi pogosto preveč poudarka posveča zbiranju merskih podatkov in premalo analizi teh [Roz00, Fen99, Hum89]. Dobro zastavljen merski plan in natančno zbiranje merskih podatkov sta dejansko pogoja, ki omogočata analizi doseči glavni cilj merjenja: objektivno ugotoviti težave in možnosti za izboljšave. Pristop PSM definira tri vrste analiz (slika 4.15):

- ocenjevanje,
- analiza izvedljivosti in
- analiza performans.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.15: Druga podaktivnost v okviru »Izvajanja merjenja« (pristop PSM)

Ocenjevanje služi za določanje ciljnih vrednosti pri planiranju projekta. Na podlagi zgodovinskih podatkov in informacij o projektu pri pristopu PSM ocenimo velikost končnega izdelka, potreben obseg dela in potek dela. Ocenjevanje se vedno uporablja ob planiranju.

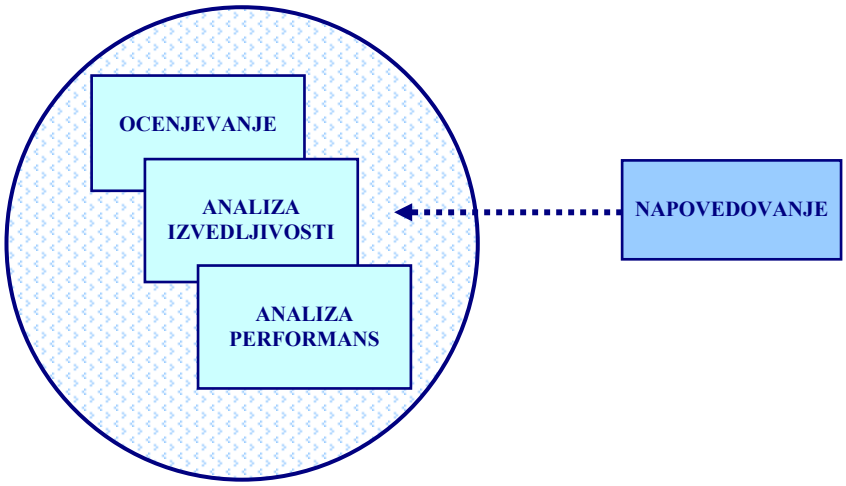
Analiza izvedljivosti služi za preverjanje zastavljenih planov in ciljev. S pomočjo podatkov o preteklih projektih, izkušenj in preverjanj usklajenosti ocenimo, če so zastavljeni cilji tehnično dosegljivi in realistični. Analiza izvedljivosti se običajno izvaja ob predložitvi planov vodstvu.

Analiza performans služi za ugotavljanje, v kolikšni meri dejanski rezultati ustrezajo planiranim vrednostim. Vhodni podatki so tako planski podatki in dejanske izmerjene vrednosti in tudi prikaz rezultatov je običajno primerjalen. S pomočjo analize performans odkrijemo težave, kar nam omogoča nadaljnje ukrepanje. Izvajamo jo kadarkoli med potekom projekta po potrebi.

Naš pristop vsebuje vse tri analize, ki pa se izvajajo na nekoliko drugačen način. Čeprav majhna organizacija deluje z omejenimi viri, analiz zaradi njihove pomembnosti ne omejujemo preveč. Osnovno vodilo je omogočiti kar največ z uporabo obstoječih podatkov, pri čemer se zelo naslanjamo na avtomatizacijo repozitorija.

Da bi olajšali izvedbo analiz naš pristop trem vrstam analize dodaja še četrto: **napovedovanje** (slika 4.16). Kot je bilo že nakazano, je napovedovanje namenjeno sprotnemu projeciranju nekaterih pomembnih merskih atributov do konca razvoja projekta. Napovedujemo recimo velikost glavnih izdelkov, obseg dela (skupen in po ciklih), doseganje glavnih mejnikov, stroške, kakovost in podobno. V bistvu gre za avtomatizirano ocenjevanje na podlagi trenutno dosegljivih dejanskih vrednosti in zgodovinskih podatkov, ki poteka neprenehoma in na najboljši način, kot ga trenutno zmoremo. Rezultat predstavlja visokonivojski pregled vseh pomembnih atributov, iz katerega lahko hitro ugotovimo, da obstajajo težave. Za podrobnejše odkrivanje vzrokov moramo nato izvesti ustrezne analize performans.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

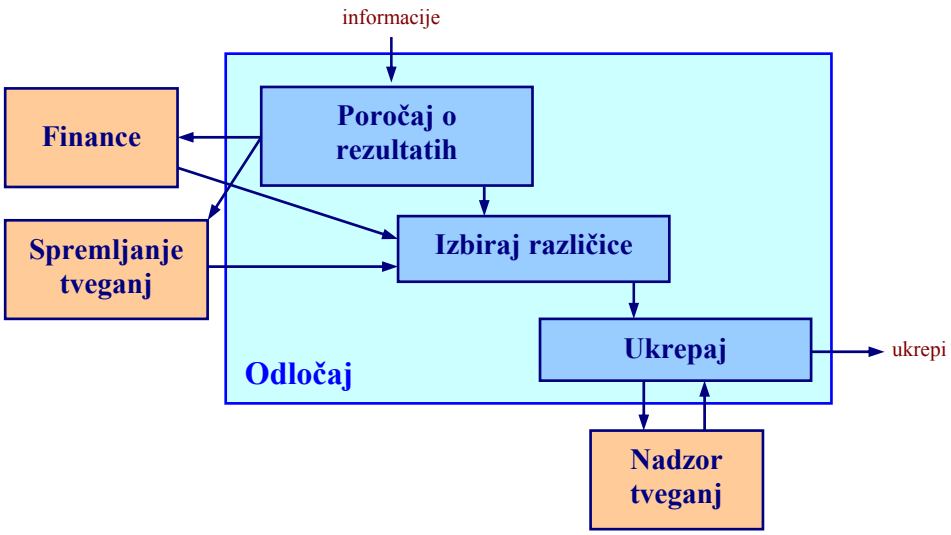


Slika 4.16: Vrste analiz v našem merskem pristopu

V primerjavi s pristopom PSM v našem pristopu temeljimo predvsem na ocenjevanju in napovedovanju, ki sta najbolj avtomatizirani analizi. Ti dve analizi pogosto nadomeščata ostali dve vrsti analiz, ki ju uporabljamo bolj po potrebi. Zaradi pomembnosti bomo analize podrobneje predstavili v posebnem podpoglavju 4.3.

4.2.3.3 Odločanje

Na sliki 4.17 je predstavljena zadnja podaktivnost v okviru izvajanja merjenja, ki določa način odzivanja na pridobljene informacije. Po opravljeni analizi moramo o rezultatih ustrezno poročati, pri čemer predstavimo stanje glede na zastavljena merska izhodišča in težave, ki smo jih na novo odkrili. Pristop PSM povezuje merjenje predvsem z aktivnostmi upravljanja tveganj [Boe91] in nadzora porabe financ, zato se na teh področjih običajno predlagajo možni ukrepi. Zadnji korak je namenjen izbiri optimalnih ukrepov, ki bodo kar najbolj povečali možnosti za uspeh projekta.



Slika 4.17: Tretja podaktivnost v okviru »Izvajanja merjenja« (pristop PSM)

Poročanje o rezultatih

Pristop PSM določa redno poročanje o štirih točkah:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- o celotni predstavitvi poteka projekta glede na znana merska izhodišča in cilje;
- o novih ovirah (težavah, tveganjih, pomanjkanju informacij) z vidika mesta pojavitve, vzroka in vpliva;
- o priporočilih, ki so bila domišljena med delom;
- o možnih novih merskih izhodiščih, ki so se pojavila.

Vse štiri točke vsebuje tudi naš pristop. Poročanje vodstvu se izvaja na šest načinov:

- na rednih tedenskih sestankih projektne skupine (vodja procesa in merjenja poroča vodji projekta, člani poročajo vodji projekta);
- na rednih tedenskih sestankih organizacije (vodja projekta poroča vodstvu organizacije, člani skupine SEPG poročajo vodstvu organizacije);
- na sestankih ob začetku (planiranje) in koncu (analiza) vsakega cikla (vodja projekta poroča vodstvu organizacije);
- na sestankih ob začetku (globalno planiranje) in koncu (zaključno poročilo) projekta (vodja projekta poroča vodstvu organizacije);
- preko IS repozitorija (vodstvo projekta in organizacije ima neposreden vpogled do podatkov opravljenih merskih analiz);
- neformalno, ob delu (člani poročajo vodji projekta ter vodji procesa in merjenja).

Pri vseh poročanjih je tako ali drugače vpleteno merjenje. Specifična poročila o opravljenih analizah daje vodja procesa in merjenja projektnemu vodji ali preko skupine SEPG direktno vodstvu organizacije. Za poročanje vodstvu je sicer zadolžen vodja projekta. Člani lahko obveščajo vodjo procesa in merjenja tudi s pomočjo predlogov za izboljšavo procesa.

Izbiranje ukrepov

Eden od rezultatov merjenja so tudi ugotovljene možnosti za ukrepanje. Če z merjenjem odkrijemo težave ali možnosti za izboljšavo, moramo identificirati oboje. V kolikor meritve dokazujejo, da izvajanje projekta poteka po planih, lahko tudi ne ukrepamo nič.

Predlogi za ukrepe prihajajo pri našem pristopu iz treh virov:

- Glavni vir je vodja procesa in merjenja, ki ima pregled nad vsemi merskimi rezultati in analizami. On je tudi neposredno zadolžen za predlaganje ukrepov.
- Ukrepe na podlagi merskih rezultatov lahko predlaga tudi vodja kakovosti, ko na podlagi meritev kakovosti ugotovi odstopanja na področju zagotavljanja kakovosti.
- Vsi ostali člani imajo tudi vpogled v del merskih rezultatov. Njihovi predlogi se nanašajo predvsem na aktivnosti, ki jih sami izvajajo.

Ukrepanje

Učinkovito ukrepanje je zadnji del izvajanja merjenja. V kolikor nikoli ne ukrepamo, nam merjenje ne pomaga učinkoviteje dosegati projektne cilje. Običajni ukrepi obsegajo spremembe planov, dodajanje razvijalcev, usposabljanje, uvajanje novih znanj in podobno. Pomembno je, da z ukrepi odpravimo vzroke, ne pa popravljamo posledice težav.

V kolikor merski podatki kažejo na potrebo po ukrepanju, omogoča prilagodljiv proces razvoja PO več aktivnosti, v okviru katerih to lahko storimo:

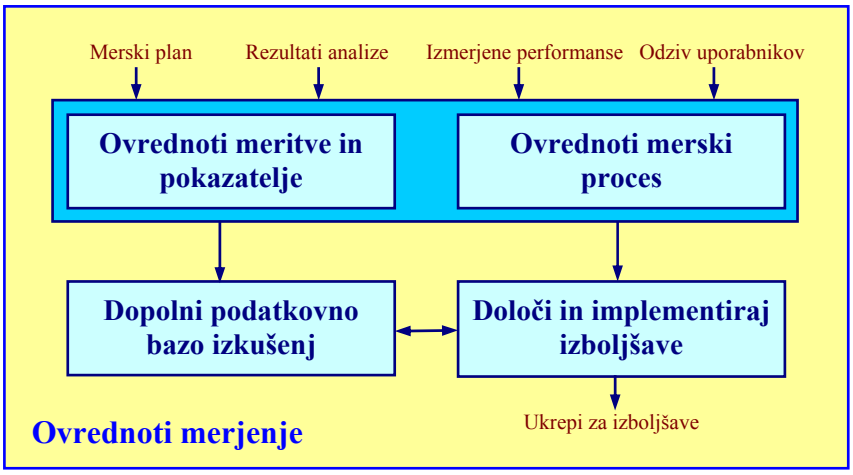
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Na najbolj nujne zadeve se lahko odzovemo v okviru vsakega tedenskega sestanka projekta. Tedaj vodja projekta skupaj s člani izbere način in ukrepi stopijo v veljavo takoj. Ti ukrepi ne obsegajo nekaterih večjih posegov (recimo spreminjanja planov).
- Glavna možnost za ukrepanje je ob koncu vsakega cikla. Po opravljeni analizi opravimo ustrezne ukrepe, ki zagotovijo ustrezen zaključek vseh aktivnosti cikla.
- Na podlagi analize ob zaključku cikla lahko ukrepamo tudi v naslednjem ciklu. Takrat lahko spremenimo cilje, popravimo plane ali izvedemo druge večje ukrepe.

4.2.4 Ovrednotenje merjenja

Zadnja glavna aktivnost merskega procesa je namenjena preverjanju kakovosti izvajanja merjenja in nenehnemu izboljševanju tako vsebine merjenja kot merskega procesa. Čeprav je ta aktivnost po obsegu krajša, je njena vloga za učinkovito merjenje zelo pomembna [Pfl94]. Aktivnost obsega štiri podaktivnosti (slika 4.18):

- ovrednotenje meritev in pokazateljev,
- ovrednotenje merskega procesa,
- dopolnitev PB izkušenj in
- določitev in implementacija izboljšav.



Slika 4.18: »Ovrednotenje merjenja« v okviru pristopa PSM

Vsaka od podaktivnosti bo kratko predstavljena v nadaljevanju.

4.2.4.1 Ovrednotenje meritev in pokazateljev

Namen te podaktivnosti je preveriti različne vidike kakovosti izbranega nabora meritev in pokazateljev, da bi ugotovili, v kolikšni meri ti ustrezajo zahtevam projekta (primera preverjanja kakovosti meritev predstavljata recimo [Jef93, Sch92]). V skladu s standardom ISO 15939 pristop PSM preverja naslednje vidike:

- pogostost uporabe merskih izdelkov,
- zaupanje merskim rezultatom,
- primernost meritev in pokazateljev (glede na namen merjenja),
- razumljivost merskih rezultatov,
- ustreznost za uporabo statističnih pristopov,
- točnost merjenja (manjkajoči podatki, nekonsistentnost) [Mah01] in
- zanesljivost merjenja (ponovljivost, večkratna uporabnost).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Naš pristop s pomočjo avtomatizacije in natančno določenih pravil merjenja na ravni organizacije poskrbi za večino vidikov. Tudi v okviru analiz ob zaključku vsakega cikla preverjamo vsakega od vidikov. Preverjanja so v glavnem subjektivna, izvaja pa jih vodja procesa in merjenja v sodelovanju z vodjem kakovosti in procesnim vodjem. Na sestanku ob koncu vsakega cikla se o teh vprašanjih razpravlja tudi s člani projektne skupine.

4.2.4.2 Ovrednotenje merskega procesa

Ustrezno izbrane meritve in pokazatelji ne zadoščajo, če je merski proces neustrezen. Pristop PSM preverja proces po treh kriterijih:

- po **dosežkih**,
- po **ustreznosti** in
- po **zmožnosti**.

Pri **dosežkih** izmeri merski analitik proces enkrat letno s štirih vidikov:

- pravočasnost (pogostost poročanja, trajanje analiz),
- učinkovitost procesa (dobiček v primerjavi s stroški),
- izločanje napak (skrb za minimalni vnos napak) in
- zadovoljstvo uporabnikov.

Ustreznost se določa dvakrat letno na podlagi treh ocen:

- preverjamo prilagajanje meritev določenim izhodiščem,
- preverjamo zbiranje in uporabo merskih podatkov in
- preverjamo implementacijo procesa.

Kriterij **zmožnosti** preverjamo vsako leto ali dve. V skladu z modelom CMMI in standardom ISO/IEC 15504 [Bsi02] ocenjujemo zmožnost merskega procesa.

Vsi trije kriteriji so zahtevni za preverjanje in zahtevajo veliko podatkov. Vsa ocenjevanja razen ocenjevanja performans izvajajo zunanji analitiki.

Naš pristop predvideva ocenjevanje merskega procesa v okviru postopkov izboljšave organizacijskega procesa razvoja PO. To je dejavnost, ki se neprenehoma izvaja na ravni organizacije, zanjo pa je zadolžena skupina SEPG. Pri ovrednotenju merskega procesa preverjamo predvsem merske dosežke, ostala kriterija pa preverjamo delno.

4.2.4.3 Dopolnjevanje podatkovne baze izkušenj

Na podlagi ugotovitev in drugih izkušenj z merjenjem lahko dopolnimo in obogatimo znanje kolektiva. Pristop PSM v ta namen predvideva PB izkušenj. Ta je lahko zgolj kratek dokument ali elektronsko podprt sistem sporočil. Tipična PB obsega poleg opisa postopkov, meritev, pokazateljev in raznih poročil o izvedenih ocenjevanjih procesa ali meritev tudi tipične probleme merjenja, (ne)uspešne rešitve in ocene orodij.

Za vse podatke o procesu razvoja (in merskem procesu) je v repozitoriju predviden prostor. Tja sodijo tudi podatki o izkušnjah. Za dokumentiranje teh so zadolženi člani skupine SEPG.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

4.2.4.4 Določanje in implementacija izboljšav

Zbrane izkušnje so brez pomena, če jih ne uporabimo. Nekatere so takšne, da jih lahko takoj vgradimo v merski proces. V tem primeru proces spremenimo v okviru naslednjega planiranja merjenja ali s spremembo procesa. Tipični ukrepi se nanašajo na glavne postopke: določanje meritev, postopke analize, zbiranje podatkov in podobno.

Tudi naš pristop predvideva te postopke. V okviru procesa projekta implementiramo merske spremembe ob začetkih ciklov, pri čemer stalna merska izhodišča ostajajo vedno enaka.

4.2.5 Razmerje med merskim procesom in procesom razvoja PO

S tem smo zaključili predstavitev aktivnosti merskega procesa, ki so v krajši obliki predstavljene tudi v dodatku B.1.1. Uporabljen način predstavitve nam je omogočil, da smo ustrezno določili vse potrebne aktivnosti našega merskega pristopa. To nam zagotavlja uporaba pristopa PSM. Vendar ima tak način opisa merskega pristopa tudi pomanjkljivost: iz njega je težje prepoznati posebnosti našega pristopa, ker opis procesa ni podan celovito. V tem razdelku želimo to težavo odpraviti ter merski proces umestiti v prilagodljiv proces razvoja PO.

Osnovne značilnosti našega merskega pristopa bi lahko povzeli v naslednjih točkah:

- merjenje je pretežno določeno na ravni organizacije, projekti pa ga povzemajo podobno kot povzemajo organizacijski proces ob določanju projektnega procesa razvoja PO;
- v skladu s tem obstaja del merjenja, ki je enak za vse, in del za posebne zahteve projektov;
- operativno se merjenje izvaja preko merjenja vsake posamezne aktivnosti posameznika;
- vsi rezultati se zbirajo v skupnem repozitoriju, ki je avtomatiziran in
- merski proces posveča posebno pozornost analizam merskih podatkov.

Formalno gledano naš merski pristop izvaja enake aktivnosti kot jih določa pristop PSM, vendar na za nas primernejši način. Predvsem je tu pomembno razlikovati med aktivnostmi na ravni organizacije in projekta. Pristop PSM predvideva izvedbo vseh aktivnosti na ravni projekta [McG01], medtem naš pristop velik del aktivnosti merskega procesa izvaja na ravni organizacije. To sta recimo celotni glavni aktivnosti »Uvajanje merjenja« ter »Ovrednotenje merjenja«, za kateri je v majhni organizaciji zadolžena skupina SEPG, pomaga pa ji tudi skupina SQA.

Tudi glavnina glavne aktivnosti »Planiranja merjenja« se izvaja v okviru aktivnosti skupine SEPG na organizacijski ravni. Stalna merska izhodišča ter stalni nabori meritev in pokazateljev so predpisani na tej ravni. Na ravni projekta se določajo samo posebna merska izhodišča in meritve. To se v prilagodljivem procesu razvoja PO lahko zgodi v okviru faze »Priprava projekta « v prvem ciklu ali ob vsakem planiranju na začetkih ciklov.

Najpomembnejša glavna aktivnost »Izvajanje merjenja« se seveda mora izvajati na ravni projekta. Podaktivnost »Zbiranje in obdelava podatkov« se izvaja neprenehoma, zanjo pa v okviru prilagodljivega procesa skrbi dodatna aktivnost »Zbiranje merskih podatkov«. Prvi dve analizi (ocenjevanje in analiza izvedljivosti) se izvajata ob ali neposredno po planiranju – torej bodisi ob začetku projekta ali ob začetkih posameznih ciklov. Za drugi dve analizi (analizo performans in napovedovanje) je drugače: načeloma se izvajata kadarkoli, vendar še posebno v času analiz – torej ob zaključku vsakega cikla ter predvsem ob zaključku projekta.

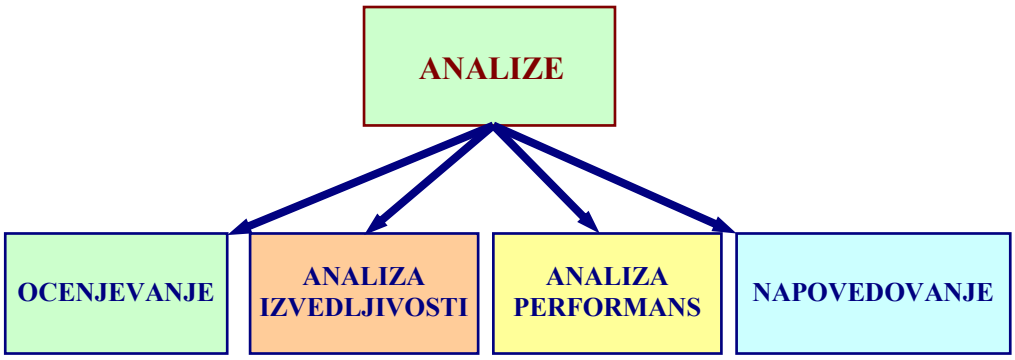
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Merski proces predvideva v tem delu še podaktivnost »Odločanje«, ki zajema poročanje vodstvu in ukrepanje. Prilagodljiv proces razvoja PO predvideva za poročanje vsa obdobja po planiranju in izvedbi analiz, torej začetek in konec projekta ter vse začetke in konce ciklov. Ukrepi se seveda sprejemajo na sestankih, katerim je na ravni projekta posvečena posebna faza v okviru »Podpornih aktivnosti«. Na ravni organizacije je za tedenske sestanke zadolženo vodstvo organizacije.

Umeščenost merskega procesa v proces razvoja PO je predstavljena tudi v dodatku B.1.2.

4.3 Analize merskih podatkov

Analiza merskih podatkov je tisti del postopka, ki odločilno vpliva na uspešnost merjenja. Čeprav je pomembna ustrezna izvedba vsake aktivnosti v okviru merskega procesa, se je treba zavedati, da so ostale aktivnosti namenjene zgolj pripravi pogojev za uspešno izvedbo analize in kasnejše ukrepanje (glej razdelek 2.1.6.2). Zato kljub omejenim virom analize v našem primeru bogato določimo. Definiramo štiri vrste analiz, ki so predstavljene na sliki 4.19.



Slika 4.19: Štiri vrste analiz našega merskega pristopa

Analize tipično obsegajo štiri vrste aktivnosti:

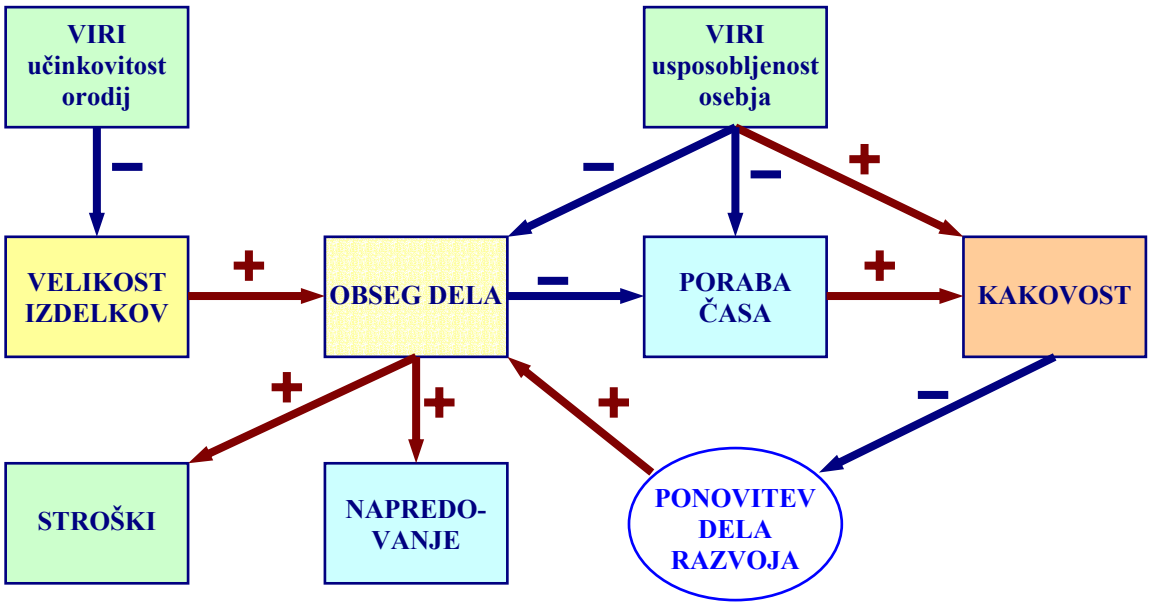
- izvedbo jedra analize (ocenjevanja, primerjanja vrednosti, napovedovanja ali podobno),
- ustrezno predstavitev ali uporabo analize (v planih, s pokazatelji),
- prepoznavanje težav in
- raziskavo možnosti za rešitev težav.

Med merjenimi vrednostmi obstajajo tipična razmerja, ki nam lahko veliko pomagajo pri izvedbi analiz [DoD02]. Če za izhodišče vzamemo nabor razredov meritev (glej razdelek 4.2.2.1), potem lahko ta razmerja predstavimo s sliko 4.20.

Razmerja so na sliki prikazana s puščicami. Puščice s plusom predstavljajo sorazmerno povečevanje vrednosti, medtem ko puščice z minusom predstavljajo obratno sorazmerje povečevanja. Iz slike lahko razberemo naslednje:

- Povečanje učinkovitosti orodij ali virov omogoča, da lahko v istem času izdelamo večji izdelek.
- Večja usposobljenost osebja nad istim delom se kaže v manjšem obsegu dela, krajšem času razvoja in večji kakovosti izdelkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.20: Tipična razmerja med merjenimi vrednostmi

- Večji izdelek zahteva tudi večji obseg dela.
- Večji obseg dela nad istim obsegom dela ima za posledico manjšo porabo časa, hitrejše napredovanje in večje stroške.
- Večja poraba časa za isto delo pomeni običajno večjo kakovost.
- Večja kakovost pomeni manjšo potrebo po ponovitvi dela razvoja.
- Če ponavljamo določene dele razvoja to zahteva dodaten obseg dela.

Ta razmerja so uporabna pri vseh vrstah analiz. Ocenjevanje uporabljamo pri planiranju. Če recimo ocenimo, da bo končni izdelek večji kot smo prvotno predvideli, lahko posledično pričakujemo tudi večji obseg dela in večje stroške. Pri analizah izvedljivosti lahko predvidimo, kateri atributi so povezani med seboj. Če planiramo kakovostne izdelke, moramo razvoju nameniti bodisi dovolj osebja ali dodatno poskrbeti za večjo usposobljenost osebja. Razmerja so odlično vodilo za raziskavo vzrokov pri analizi performans. Če smo recimo ugotovili, da izvedba projekta zamuja, lahko sklepamo, da so se bodisi spremenile zahteve in razvijamo večji izdelek ali pa nimamo primerno usposobljenih razvijalcev. Če napovedi kažejo, da napoved obsega dela presega predvidene vrednosti, lahko pričakujemo večje stroške.

Vse vrste analiz so podrobneje predstavljene v nadaljevanju, povzete pa v dodatku B.1.3.

4.3.1 Ocenjevanje

Ocenjevanje oziroma nasploh napovedovanje je zelo pomemben princip v merjenju PO [DeM86, Fen97, Hum95, McG01], zato se na tem področju aktivno raziskuje (recimo [Zha98, Ohl98, Str01]). V splošnem se ocenjevanje izvaja na dva načina. Bodisi:

- uporabljamo rezultate ene meritve za napoved rezultatov druge ali
- uporabljamo več rezultatov za napoved bodočih rezultatov druge množice iste vrste.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pojem *cenilka* (ang. estimator) predstavlja posebno vrsto pokazatelja, ki prikazuje razmerje med dvema meritvama. Na to razmerje vplivajo *performančni dejavniki* (ang. performance factors), ki predstavljajo vpliv okolja, v katerem izvajamo ocenjevanje [DoD02]. Pristop PSM predvideva ocenjevanje štirih vrednosti:

- velikosti PO,
- obsega dela,
- poteka dela in
- kakovosti.

Cenilke za te vrednosti so naslednji:

- na podlagi funkcionalnosti napovedujemo velikost izdelka;
- na podlagi funkcionalnosti napovedujemo obseg dela;
- na podlagi velikosti izdelka napovedujemo obseg dela;
- na podlagi obsega dela napovedujemo potek dela;
- na podlagi velikosti izdelka napovedujemo število napak (kakovost).

Na cenilke vplivajo performančni dejavniki, kot so recimo produktivnost razvijalcev, učinkovitost uporabljene tehnologije in podobno.

Modeli in tehnike za ocenjevanje PO uporabljajo različna razmerja med cenilkami in performančnimi dejavniki. Pri ocenjevanju poznamo štiri vrste pristopov [Fen97, DoD02]:

- **Parametrični modeli**: ti modeli upoštevajo, da med velikostjo, obsegom dela, potekom dela in kakovostjo obstaja matematično razmerje, na katerega vplivajo performančni dejavniki. Razmerje je zapisano v obliki enačbe, v kateri dejavniki vplivajo na določitev konstant. Najbolj znan model te vrste je COCOMO [Boe81, Boe95].
- **Ocenjevanje na podlagi dekompozicije aktivnosti**: za vsako aktivnost na nižji ravni določimo vrednosti izbrane značilnosti. Nato vse vrednosti združimo v skupni rezultat na višji ravni. Ker so ocene posameznih delov natančnejše, je tudi skupen rezultat natančnejši.
- **Ocenjevanje na podlagi podobnosti**: oceno določimo na podlagi znanih rezultatov meritev podobnega projekta. Pri tem skušamo ustrezno upoštevati razlike.
- **Ocenjevanje s pomočjo preprostih razmerij oziroma izkušenj**: na podlagi zgodovinskih podatkov določimo preprosta razmerja med meritvami (recimo obsegom dela in funkcionalnostjo), s pomočjo katerih ocenjujemo vrednosti. Ta način ocenjevanja je uporaben samo v okviru iste organizacije in istovrstnih projektov.

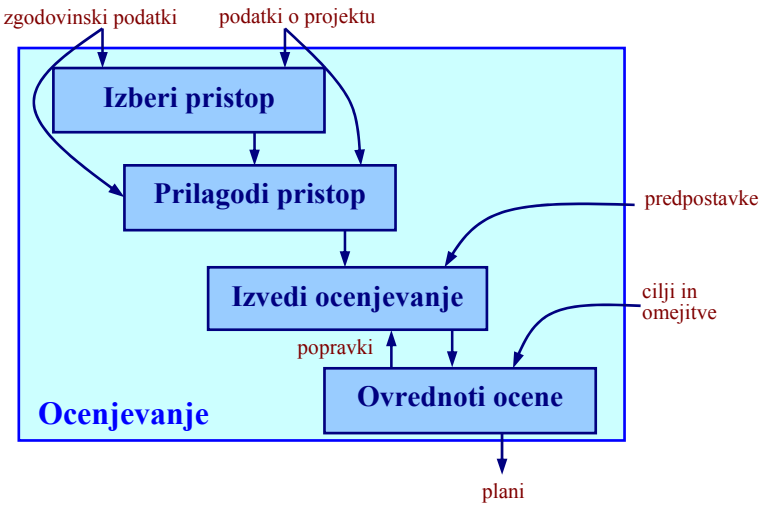
Omenjenim pristopom lahko dodamo še enega, ki za ocenjevanje uporablja principe umetne inteligence (recimo naivni Bayesov kvantifikator [Fen00] ali nevronske mreže [Wit97]).

Slika 4.21 predstavlja izvajanje ocenjevanja v okviru pristopa PSM. To obsega štiri korake: izbiro pristopa, prilagoditev pristopa, izvedbo ocenjevanja in ovrednotenje ocene.

4.3.1.1 Izbira pristopa

Pristop PSM omogoča izbiro poljubnega pristopa za ocenjevanje, vendar se je treba zavedati, da niso vsi pristopi enako primerni v vseh razmerah. Tabela 4.1 prikazuje primernost posameznih pristopov glede na stopnjo razumevanja problema, obseg zgodovinskih podatkov in zahtevnost matematične analize.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.21: Izvedba ocenjevanja v okviru pristopa PSM

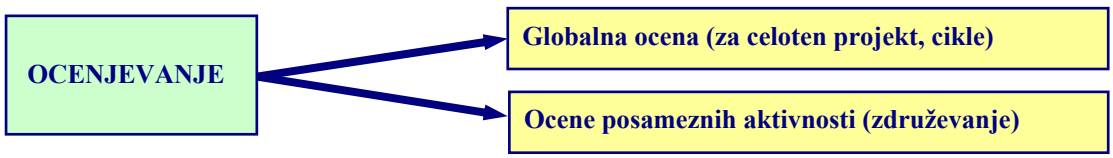
	Potrebna stopnja razumevanja	Obseg zgodovinskih podatkov	Zahtevnost matematične analize
Parametrični modeli	Osnovni opis	Veliko projektov	Zahtevne statistične metode
Ocenjevanje na podlagi aktivnosti	Podroben opis procesa	Zelo natančni podatki za nekaj projektov	Aritmetične operacije
Ocenjevanje na podlagi podobnosti	Podroben opis izdelkov	Vsaj en podoben projekt	Aritmetične operacije
Ocenjevanje s pomočjo preprostih razmerij	Osnovni opis	Veliko projektov	Enostavne statistične metode

Tabela 4.1: Primernost različnih pristopov ocenjevanja

Naš pristop obravnava ocenjevanje na podoben način kot pristop PSM. Uporablja ga za ocenjevanje velikosti, obsega dela, poteka dela in kakovosti ter dodatno še stroškov, česar PSM ne predvideva. Enako kot pristop PSM ne ocenjujemo posebej virov (osebja), ker predvidevamo, da so ti za projekt fiksni. Ocenjevanje je del globalnega planiranja ter del planiranja v vseh začetnih fazah ciklov, kjer se določajo podrobni plani opravil, poteka dela in kakovosti. Glede na proces razvoja PO se ocene določajo v fazi planiranja projekta (1. cikel) in nato še v fazah priprave cikla pri tretjem, četrtem in petem ciklu.

V našem pristopu ne dopuščamo poljubne izbire pristopa ocenjevanja, temveč ga natanko določimo. Zdi se nam smiselno standardizirati učinkovit postopek ter ga avtomatizirati, poleg tega pa zahteve merjenja na 4. stopnji CMM zahtevajo obravnavo merskih podatkov s statističnimi metodami [Fen97, Hum95, Ray99], kar zahteva večjo določenost.

Ocene določamo na dva načina (slika 4.22):



Slika 4.22: Dva načina izvedbe ocenjevanja v našem merskem procesu

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Ocene za celoten projekt, celotne cikle in ponekod tudi faze določamo na podlagi znanih izmerjenih podatkov in zgodovinskih podatkov. Pri tem si pomagamo s statističnimi metodami, s katerimi določimo tudi območje (običajno 90%) zaupanja oceni. Tako rezultat ni zgolj boljša ali slabša ocena, temveč statistično utemeljeno območje, za katero lahko z 90% verjetnosti trdimo, da bo v njem tudi izmerjen rezultat.
- Glede na to, da zbiramo podatke na najnižji ravni, uporabljamo pri ocenjevanju tudi natančnejši pristop na podlagi aktivnosti. Pri tem ocenimo vrednosti posameznih aktivnosti ali izdelkov, za kar tudi običajno uporabimo zgodovinske podatke. Ta del ocenjevanja običajno prepustimo posameznim razvijalcem, s čemer pridobimo tudi njihovo subjektivno oceno. Običajno se to ocenjevanje izvaja samo v okviru enega cikla.

Oba načina ocenjevanja uskladimo in združimo v končno oceno, ki je podlaga za planiranje.

Pomemben vidik našega pristopa je sprotno upoštevanje že znanih dejstev. Tako na začetku ocenjujemo predvsem na podlagi zgodovinskih podatkov, ob planiranjih naslednjih ciklov pa že izmerimo določene vrednosti in lahko ocenjujemo natančneje. Tako naše ocene postajajo vse natančnejše, kar nam omogoča učinkovit nadzor nad izvajanjem projekta.

Kot bomo videli v nadaljevanju je izhodišče za ocene različnih vrednosti ocena velikosti končnega izdelka. Ker je ta sestavljen iz procesnega in podatkovnega dela, imamo tudi dve oceni. Za obe določimo tudi območje zaupanja, ki ga upoštevamo tudi pri nadaljnjih ocenah.

4.3.1.2 Prilagoditev pristopa

Prilagajanje pristopa ocenjevanja obsega dva dela. V prvem delu prilagodimo pristop značilnostim projekta, v drugem pa ustrezno »nastavimo« model na podlagi lokalnih zgodovinskih podatkov. Prvi del upošteva posebnosti projekta, kot so recimo prilagoditev življenjskega cikla, povečan obseg dokumentacije in podobno. Nastavitev se nanaša predvsem na parametrične modele, kjer je treba pred uporabo modela na podlagi zgodovinskih podatkov primerno nastaviti konstante.

V našem primeru je pomemben predvsem prvi del koraka. Dinamično prilagodljiv proces ima to značilnost, da natančnih planov običajno vnaprej ne moremo dokončno določiti, temveč jih določamo postopoma (glej razdelek 3.3.3.6). To pomembno vpliva na uporabo pristopa, saj na začetku (običajno) ne znamo natanko določiti vseh aktivnosti, temveč samo glavne. Temu primerno bodo ocene na začetku manj natančne kot kasneje. Kasneje bodo ocene natančnejše, pristop pa bo upošteval tudi nekatere dejansko izmerjene podatke.

Posebnih nastavitev naš pristop ne predvideva, predvideva pa upoštevanje vseh zgodovinskih podatkov. Ko izvedemo del razvoja, so izmerjeni podatki tega dela že znani. Ob naslednji oceni je treba poleg ostalih zgodovinskih podatkov upoštevati tudi te.

4.3.1.3 Izvedba ocenjevanja

Pristop PSM določa ocenjevanje štirih vrednosti. Te so del železnega repertoarja številnih naborov meritev (recimo [SEL95, Pfl90, Gra94, Car92]). Ocenjuje se:

a) velikost končnega izdelka,
b) skupen obseg dela,
c) potek dela ter
d) število napak oziroma kakovost.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

a) Pristop PSP določa merjenje tako funkcionalnosti kot fizične **velikosti izdelka**. Prvo meri s funkcijskimi točkami (FP) [Alb83], medtem ko za merjenje fizične velikosti uporabi število vrstic programske kode [Par92]. Obe meritvi nista natančni na začetku, zato je treba ocenjevanja večkrat ponoviti.

b) Izhodišče za določanje **obsega dela** je ocena velikosti. V ta namen PSM predlaga uporabo parametričnega pristopa, recimo model COCOMO [Boe81, Fen97]. Ta določa obseg dela s pomočjo razmerja:

$$obseg\ dela = a * velikost^{b}$$

a in *b* sta konstanti, ki izhajata iz značilnosti PO. Model je bil nadgrajen v COCOMO 2.0 [Boe95], ki natančneje upošteva različna obdobja razvoja. Uporaba in alternativni pristopi za merjenje obsega dela so prikazani tudi v [Ohl98, Wit97].

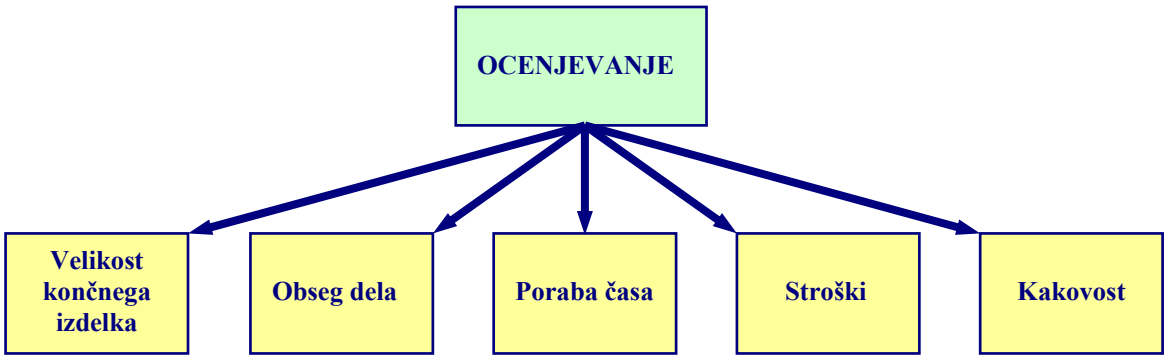
c) Pri določanju **poteka dela** izhajamo iz skupne velikosti izdelka in obsega dela, ki ju razdelimo med posamezne aktivnosti [Fen97]. Pri tem je pomembno upoštevati kritične poti in razporeditev osebja po časovnem obdobju.

d) Pristop PSM za določanje **kakovosti** uporablja zgodovinske podatke o napakah in oceno velikosti. Predlaga dva pristopa za določanje kakovosti na podlagi napak [DoD98, DoD02]:

- uporabo modelov, ki temeljijo na določanju povprečnega časa do naslednje napake in
- uporabo modelov, ki določajo stopnjo vnašanja in izločanja napak v posameznih fazah življenjskega cikla.

Pristop za določanje kakovosti z merjenjem napak je pogost (recimo [Flo92, Pfl90, Gra94, Hum95, Zha98]), ker je enostaven in neposreden. Alternativni pristop določa, da je kakovost sestavljena iz več komponent. Klasična modela [McC77, Boe78] tako določata hierarhijo več deset zaželjenih lastnosti in merita vsako posebej. Na podobni podlagi je določen tudi standard ISO 9126 [BSi02], po katerem kakovost sestavlja 6 lastnosti: funkcionalnost, zanesljivost, učinkovitost, uporabnost, vzdrževalnost in prenosljivost. Vsako lastnost merimo posebej, recimo zanesljivost [Bro92] vzdrževalnost [Col94, Hen94, Men98] in podobno. Znana so tudi drugačna pojmovanja kakovosti [Dah97, Pig98].

Naš pristop zastavlja ocenjevanje nekoliko širše (ocenjujemo tudi stroške) in ga predvsem bolj natančno določa (slika 4.23). Ocenjevanje različnih merskih atributov si bomo zato ogledali podrobneje.



Slika 4.23: Ocenjevanje po našem merskem pristopu

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Ocenjevanje velikosti končnega izdelka

Podobno kot pri pristopu PSM je prvi korak ocenjevanja v našem pristopu ocena velikosti končnega izdelka. Velikost ostalih izdelkov lahko ocenjujemo s pomočjo posebne tabele razmerij, kar bo natančneje predstavljeno v okviru napovedovanja (razdelek 4.3.4). Končni izdelki so sestavljeni iz treh delov:

- procesnega dela, ki obsega programsko kodo za realizacijo vseh modulov;
- podatkovnega dela, ki predstavlja nameščeno PB;
- dokumentacije (za procesni in podatkovni del).

Teoretično je velikost PO sestavljena iz treh razsežnosti [Fen97]: fizične velikosti, funkcionalnosti in kompleksnosti. Fizično velikost PO običajno merimo v vrsticah programske kode, ki je natanko standardizirana [Par92]. Standardna mera funkcionalnosti so funkcijske točke (FP) [Alb83, Dol97], za katere obstaja več izboljšav [Sym91, Hor98]. Poleg ciklomatskega števila [McC76] je tudi mer kompleksnosti več (recimo [Mun92]). Celovit prikaz merjenja velikosti je v [Rož01].

Naš pristop predvideva merjenje (in ocenjevanje) velikosti programske kode **samo s fizično velikostjo**, da bi tako ohranili čim večjo enostavnost postopkov. Čeprav imajo nekatera orodja (recimo Oracle [Dor97]) merjenje funkcionalnosti po metodi FP že vgrajeno, našega pristopa ne želimo vezati na nobeno orodje in se merjenju funkcionalnosti raje izognemo. FPA bi sicer lahko izpeljali tudi iz števila vrstic programske kode [Dek00], vendar je takšna pretvorba nenatančna, predvsem pa konvergenca med številom vrstic in FP [Dol97] kaže, da na tak način ne bi zvedeli nič novega. Tudi alternativne mere velikosti (recimo vektorska velikost [Has01]) za našo vrsto PO žal niso primerne.

Podatkovni del merimo (in ocenjujemo) s številom elementov PB, ki nam predstavlja kar enostaven seštevek tabel, atributov, pogledov, indeksov in sekvenc, ki bodo na koncu projekta realizirani v PB. Podoben pristop predlaga [DeM86]. Alternativen pristop bi uporabil oceno velikosti kode. Med velikostjo podatkovnega dela in programsko kodo namreč obstaja tipično razmerje, ki ga določimo na podlagi zgodovinskih podatkov. Vendar je treba upoštevati, da med projekti obstajajo razlike, zato je tako ocenjevanje primerno samo za grobe ocene. Včasih je pomembno oceniti tudi velikost vsebine PB, kar merimo z velikostjo zasedenosti zunanjega pomnilnika (recimo v MB).

Dokumentacijo merimo s številom strani dokumentacije, ker je to enostavno, čeprav obstajajo tudi alternativne mere (recimo [Lev00]). Velikost strani oziroma število vrstic na stran in druge podrobnosti so natanko določene v standardu. Med številom strani dokumentacije, številom vrstic programske kode in številom elementov podatkovnega dela obstajajo tipična razmerja, ki se jih določi na podlagi zgodovinskih podatkov. Del dokumentacije izhaja iz procesnega dela in tega lahko ocenimo na podlagi velikosti programske kode. Drugi del je vezan na podatkovni del in ga določimo iz velikosti podatkovnega dela.

Ko izvajamo ocenjevanja (v fazi planiranja projekta in ob začetkih ciklov) ocenjujemo ločeno procesni in podatkovni del. Pri tem za podlago vzamemo že izmerjene podatke izdelkov, ki se nanašajo na procesni ali podatkovni vidik. Ob prvem ocenjevanju običajno še nimamo dovolj informacij za natančno oceno, saj je dokončan samo Inicializacijski in Strateški dokument. Iz teh lahko razberemo okvirno število zahtev ter visokonivojske diagrame (entiteta-razmerje, diagram podatkovnih tokov in funkcijsko dekompozicijo) obstoječega sistema. Vse podatke upoštevamo v smislu zgodovinskih podatkov in podobnosti. V ocenah podatkovnega in procesnega dela je vsebovana tudi velikost dokumentacije, zato te posebej ne ocenjujemo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pri procesnem delu izhajamo predvsem iz:

- obsega zahtev v Inicializacijskem dokumentu;
- števila zahtev, ki so vezane na procesiranje;
- števila različnih elementov diagramov podatkovnih tokov (DFD);
- števila funkcij in ravni funkcijske dekompozicije (FD);
- števila elementov konceptualnega diagrama, ki opisuje sestavo modulov;
- števila modulov in drugega.

Pri podatkovnem delu lahko uporabimo naslednje podatke:

- obseg zahtev v Inicializacijskem dokumentu;
- število zahtev, ki so vezane na podatkovni del;
- število različnih elementov diagramov entiteta razmerje (ERD);
- število nekaterih elementov iz diagramov podatkovnih tokov (DFD);
- število različnih elementov fizičnega podatkovnega modela (fPM) in drugo.

Nekateri podatki so povezani samo s procesnim ali podatkovnim vidikom, drugi pa z obema. Podatki se ponavadi nanašajo na preštete elemente, lahko pa so izraženi tudi v drugih enotah, kar ustrezno obravnavamo. Vedno upoštevamo samo tiste podatke, ki so že izmerjeni. Ker so izdelki med seboj odvisni in novejši izdelki bolje napovedujejo končno velikost, pri izračunu upoštevamo samo nekaj koeficientov, ki so najbolj povezani s končnim izdelkom.

Velikost končnega izdelka določimo na podlagi linearne enačbe, ki je določena na podlagi tipičnih razmerij iz podatkov vseh preteklih projektov organizacije. Razmerja so izračunana vnaprej in zapisana v normalizacijski tabeli repozitorija (opisana bo v nadaljevanju), zato je ocenjevanje z enačbo enostavno. Razmerja so podana s štirimi koeficienti:

- A_i je koeficient, ki predstavlja razmerje med enoto izmerjene vrednosti in velikostjo končnega izdelka izraženega v številu vrstic glede na zgodovinske podatke organizacije.
- B_i predstavlja utežni koeficient, ki predstavlja razmerje med posameznimi podatki za procesni del. Podatki, ki se pojavijo kasneje v razvoju morajo imeti ustrezno večjo težo, ker so bolj povezani s končno velikostjo izdelka. Pri izračunu vedno upoštevamo samo nekaj (recimo 3) najbolj primernih podatkov. Rezultat moramo vedno normirati glede na podatke, ki jih dejansko uporabimo.
- C_i je koeficient, ki predstavlja razmerje med enoto izmerjene vrednosti in velikostjo končnega izdelka izraženega v številu elementov PB glede na zgodovinske podatke organizacije.
- D_i predstavlja utežni koeficient, ki predstavlja razmerje med posameznimi podatki za podatkovni del. Enako kot je koeficient B_i povezan z A_i, je koeficient D_i povezan s C_i.

Linearni enačbi za izračun velikosti sta za procesni in podatkovni del enaki, le ustrezne koeficiente je treba upoštevati. Enačbi sta naslednje oblike:

$$\textit{ocena velikosti procesnega dela [v številu vrstic prog. kode]} = \frac{\sum_{3} A_i * B_i * vrednost_i}{\sum_{3} B_i}$$

$$\textit{ocena velikosti podatkovnega dela [v številu elementov PB]} = \frac{\sum_{3} C_i * D_i * vrednost_i}{\sum_{3} D_i}$$

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Indeks i zavzame tiste vrednosti, ki pripadajo trem koeficientom B_i oziroma D_i z največjimi vrednostmi in se nanašajo na podatke, ki so v danem trenutku dejansko izmerjeni.

Ocenjevanje se izvaja v določenih fazah in v vsaki od teh lahko določimo, katere vrednosti naj bi bile tedaj znane. Ob koeficientih je zato označeno, od katere faze naprej jih lahko uporabljamo. Dejansko je potrebno pred uporabo preveriti, če določeni izmerjeni podatki zares obstajajo, saj lahko zaradi izpuščanja korakov ali faz določeni merski podatki manjkajo. Ustrezno določena razmerja med velikostmi utežnih koeficientov poskrbijo, da vedno lahko izberemo tri najustreznejše podatke. V kolikor je vseh primernih vrednosti manj, upoštevamo manjše število podatkov. Primer dela normalizacijske tabele predstavlja nekaj koeficientov (tabela 4.2). Velikosti koeficientov niso realne, temveč so podane samo kot primer.

Indeks	Naziv	Faza	A_i	B_i	C_i	D_i
1	Obseg zahtev v številu vrstic (incializacijski dok.)	1.1	390	8	4	9
2	Število zahtev za procesni del (strateški dokument)	1.2	912	54	0	0
3	Število zahtev za podatkovni del (strateški dokument)	1.2	0	0	17.8	27
4	Število elem. procesov vnDFD obstoječega sistema	1.2	460	71	0	0
5	Število zunanjih entitet vnDFD obstoječega sistema	1.2	3720	6	25.2	8
6	Število podat. skladišč vnDFD obstoječega sistema	1.2	3092	9	18	41
7	Število podat. tokov vnDFD obstoječega sistema	1.2	201	53	2	44
8	Število entitet vnERD obstoječega sistema	1.2	0	0	15	56
9	Število razmerij vnERD obstoječega sistema	1.2	922	7	6	70
10	Število atributov vnERD obstoječega sistema	1.2	158	2	2.3	81
11	Število ravni vnFD obstoječega sistema	1.2	2645	22	42	14
12	Število funkcij vnFD obstoječega sistema	1.2	1184	62	9.2	32
13	Število elem. procesov lDFD (specifikacije zahtev)	2.3	416	88	0	0
14	...					

Tabela 4.2: Primer normalizacijske tabele s tipičnimi razmerji velikosti

Primer:
Denimo, da ocenjujemo velikost v fazi planiranja projekta. Za nami je sestava projektne skupine in opravljena faza strateškega načrtovanja. Pri tem smo namerili vrednosti, ki so predstavljene v tabeli 4.3. Iz tabele je razvidno, da pri izvedbi strateškega načrtovanja nismo podrobneje specificirali zahtev, zato ustrezni podatki manjkajo.

	Indeks	Naziv	Vrednost	Velikost proc. dela	Velikost pod. dela
	1	Obseg zahtev v številu vrstic	51	19890	204
#	2	Število zahtev za procesni del	-	-	-
	3	Število zahtev za podatkovni del	-	-	-
1	4	Število elem. procesov vnDFD	39	17940	0
	5	Število zunanjih entitet vnDFD	5	18600	126
	6	Število podat. skladišč vnDFD	9	27828	162
3	7	Število podat. tokov vnDFD	89	17889	178
3	8	Število entitet vnERD	12	0	180
2	9	Število razmerij vnERD	28	25816	168
1	10	Število atributov vnERD	72	11376	165.6
	11	Število ravni vnFD	4	10580	168
2	12	Število funkcij vnFD	14	16676	128.8

Tabela 4.3: Zgled uporabe normalizacijske tabele za ocenjevanje velikosti

Če bi izračunali velikost procesnega in podatkovnega dela končnega izdelka neposredno iz posameznih podatkov, bi dobili vrednosti v desnih stolpcih. Z upoštevanjem normalizacijske tabele oziroma koeficientov B_i in D_i izberemo vrednosti, ki so obkrožene. Z uporabo enačbe določimo naslednji oceni:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

$$\text{velikost [št.vrstic]} = \frac{17940*71 + 16676*62 + 17889*53}{71 + 62 + 53} = \frac{3255769}{186} = \mathbf{17504.13}$$

$$\text{velikost [št.elementov PB]} = \frac{165.6*81 + 168*70 + 180*56}{81 + 70 + 56} = \frac{35253.6}{207} = \mathbf{170.30}$$

Ta ocena temelji na zgodovinskih podatkih, saj so vsi koeficienti določeni iz merskih podatkov o preteklih projektih na naslednji način:

- vrednosti koeficientov A_i in C_i so določene kot količnik med skupno velikostjo vseh končnih izdelkov in skupno velikostjo vseh vrednosti, za katere določamo koeficiente.

$$A_i = \frac{\sum velikost\ programske\ kode_i}{\sum vrednost_i}$$

$$C_i = \frac{\sum število\ elementov\ PB_i}{\sum vrednost_i}$$

  Recimo: A_1 določimo tako, da vsoto vseh vrstic programske kode vseh projektov delimo z vsoto vseh vrstic opisov zahtev iz vseh Inicializacijskih dokumentov

- vrednosti koeficientov B_i in D_i so določene na podlagi stopnje korelacije[24] r med velikostjo in izbranimi vrednostmi za vse dokončane projekte. Zavzamejo lahko vse vrednosti med 0 in 100. Določene so po naslednji formuli:

$$B_i\ (ali\ D)_i = \begin{cases} 0 & če\ r^2 <= 0.5 \\ \lceil ((r^2(velikost, vrednost_i) - 0.5)*20)^2 \rceil & sicer \end{cases}$$

  Primer: če je vrednost korelacije med velikostjo programske kode in številom vrstic opisa zahtev enaka 0.8, potem velja, da je $B_1 = \lceil (0.64 - 0.5)*20)^2 \rceil = \lceil 2.8^2 \rceil = 7.84 = 8$.

Za oceno velikosti uporabimo kombinacijo treh podatkov, ki imajo v danem trenutku največjo korelacijo s končnim rezultatom. Čeprav ocenjujemo na podlagi zgodovinskih podatkov, težko presodimo, kako dobra je ocena. To je pomembno, ker je velikost izhodišče za vse ostale ocene. Kakovost ocene zato ovrednotimo s pomočjo podatkov o preteklih ocenah velikosti in dejanskih izmerjenih velikostih končnih izdelkov. Pri tem predpostavimo:

- da imamo dovolj zgodovinskih podatkov (vsaj 3);
- da med ocenami in velikostjo obstaja ustrezna povezava (r^2>=0.7);
- da uporabljamo zgodovinske podatke, ki so bili določeni na enak način: če preverjamo kakovost ocen v fazi planiranja projekta, moramo uporabljati zgodovinske podatke o ocenah iz iste faze.

Pri tem uporabimo statistično metodo imenovano *linearna regresija*. Z uporabo linearne regresije lahko določimo *območje*, za katerega lahko z določeno verjetnostjo (*stopnjo zaupanja)* verjamemo, da vsebuje tudi končno velikost procesnega ali podatkovnega dela.

[24] Postopek izračuna korelacije si lahko bralec ogleda v statističnih ali matematičnih priročnikih. Kratek povzetek je tudi v [Hum95] in [Fen97] .

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Običajno se uporabljajo stopnje zaupanja nad 70% [Hum95], v našem primeru uporabljamo 90%. V nadaljevanju na kratko predstavljamo postopek, ki ga bomo uporabljali tudi drugod.

Linearna regresija opisuje razmerje med ocenami (*o*) in izmerjenimi vrednostmi (*v*) z linearno enačbo oblike:

$$v_k = \beta_0 + \beta_1 * o_k$$

Če si vrednosti vektorjev *o* in *v* predstavljamo kot točke v ravnini, metoda določi enačbo take premice, za katero velja, da je vsota kvadratov razdalj točk od premice najmanjša (glej sliko 4.24). Parametra β_0 in β_1 sta določena na podlagi vrednosti ocen *o* in izmerjenih velikosti *v* iz *n* zaključenih projektov na naslednji način:

$$\beta_1 = \frac{\sum_{i=1}^{n} v_i * o_i - n * o_{pov} * v_{pov}}{\sum_{i=1}^{n} o_i^2 - n * o_{pov}^2}$$

$$\beta_0 = v_{pov} - \beta_1 o_{pov}$$

Preden določimo območje zaupanja, moramo določiti še standardno deviacijo:

$$\sigma = \sqrt{\frac{1}{n-2} \sum_{i=1}^{n} (v_i - \beta_0 - \beta_1 * o_i)^2}$$

Območje za oceno velikosti o_k določimo po enačbi:

$$območje = t(\alpha/2, n-2) * \sigma * \sqrt{1 + \frac{1}{n} + \frac{(o_k - o_{pov})^2}{\sum_{i=1}^{n} (o_i - o_{pov})^2}}$$

Vrednost $t(\alpha/2, n-2)$ je konstanta, ki jo razberemo v tabeli porazdelitve *t*. To dobimo v statističnih priročnikih (tudi v [Hum95]). Če želimo določiti 90% območje zaupanja ($\alpha/2$ = 90) in določamo območje na podlagi zgodovinskih podatkov 10 projektov (n=10), potem je ta vrednost 1.86.

Slednjič določimo območje po enačbah:

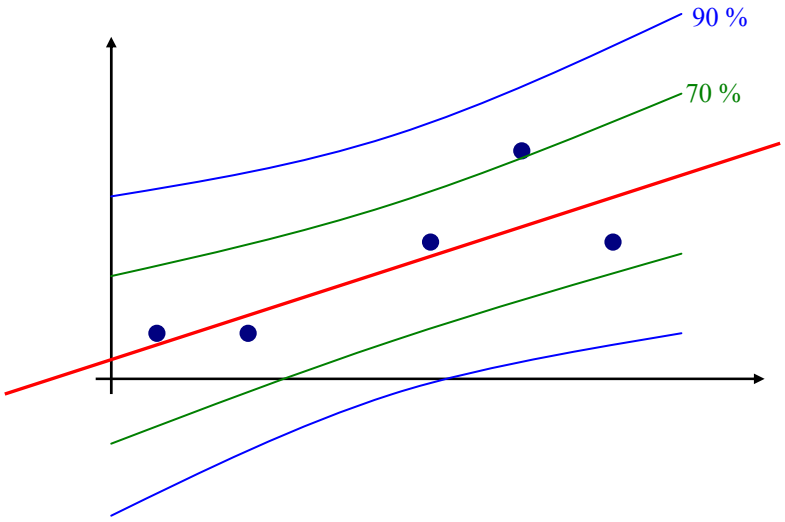
$$ocena\ velikosti_{spodnja\ meja} = ocena\ velikosti - območje$$

$$ocena\ velikosti_{zgornja\ meja} = ocena\ velikosti + območje$$

Slika 4.24 prikazuje grafični prikaz regresijske premice in območij 90% in 70% zaupanja, ki je bilo določeno na podlagi petih parov podatkov. Iz slike je lepo razvidno, da majhno število podatkov pomeni široko območje zaupanja. Podoben učinek ima tudi visoka stopnja zaupanja. Težiti je treba k temu, da regresijo uporabljamo nad dovolj velikim številom podatkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Območje določimo tako za oceno velikosti procesnega kot podatkovnega dela. Ker smo pri zajemu podatkov natančni (zajemajo jih posamezniki) lahko pričakujemo, da bodo tudi ocene kakovostne, kar se bo odražalo v ozkosti območja.



Slika 4.24: Grafični prikaz regresijske premice in dveh območij zaupanja

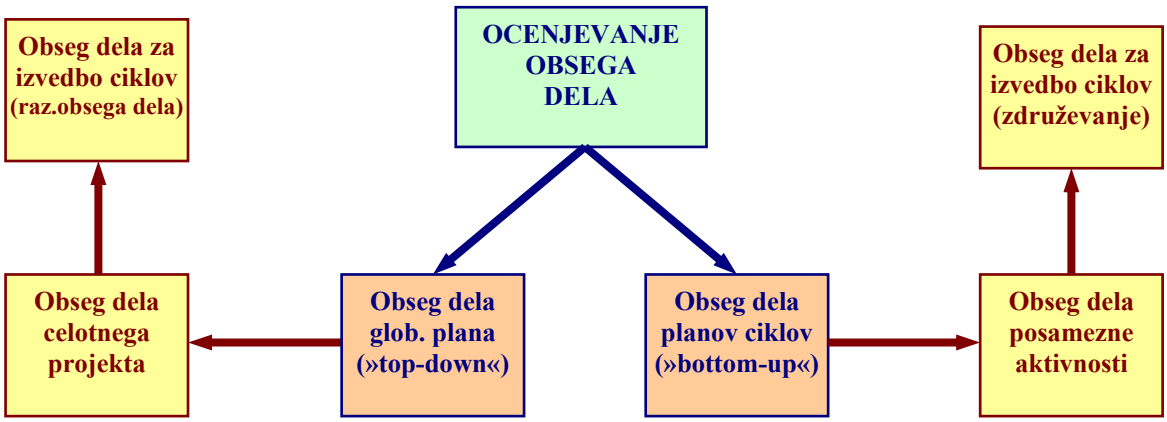
Ocenjevanje obsega dela

Postopek linearne regresije je avtomatiziran, zato ga uporabimo tudi za oceno obsega dela. V tem primeru za spremenljivke namesto ocen velikosti in izmerjene velikosti uporabimo razmerje med velikostjo in obsegom dela. Pri določanju obsega dela ločimo dva primera (slika 4.25):

- oceno obsega dela v okviru globalnega plana in
- oceno obsega dela za posamezne aktivnosti v okviru plana cikla.

Ocenjevanje obsega dela v okviru globalnega plana obsega dva dela:

- najprej ocenimo obseg dela za celoten projekt in
- na podlagi tega še obseg dela za izvedbo posameznih ciklov.



Slika 4.25: Ocenjevanje obsega dela po našem merskem pristopu

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pri **ocenjevanju obsega dela za celoten projekt** uporabimo oceno velikosti obeh vidikov končnega izdelka in linearno regresijo. Z njeno pomočjo določimo dve linearni enačbi, ki določata, kako na podlagi znane ocene velikosti določimo skupni obseg dela za procesni oziroma podatkovni del. Za obe oceni po enakem postopku kot pri oceni končne velikosti določimo še območje 90% zaupanja. Tak pristop je blizu uveljavljenem modelom COCOMO [Boe81, Boe95], ker obseg dela določa le na podlagi velikosti izdelka.

Na podlagi tega rezultata lahko ocenimo, kolikšen obseg dela bo potreben za izvedbo posameznih ciklov in faz. Iz zgodovinskih podatkov je znano tipično razmerje aktivnosti, ki ga lahko uporabimo za izračun. Tudi pri ocenah obsega dela za posamezne cikle in faze določimo zgornjo oziroma spodnjo mejo območja zaupanja.

Pomudimo se nekoliko pri tipičnih razmerjih med aktivnostmi. Tipično razmerje med vsemi aktivnostmi je shranjeno v določeni tabeli repozitorija organizacije. Določa odstotne deleže obsega dela za vsako fazo in cikel glede na celoten obseg dela projekta. Vsota deležev vseh ciklov je 100%, vsota deležev obsega dela faz v izbranem ciklu pa je enaka deležu cikla. Tabela se ob spremembah merskih podatkov kateregakoli projekta sproti ažurira, tako da vedno predstavlja primerno izhodišče. Tabela predstavlja primer z izmišljenimi podatki.

	Naziv cikla	Delež v %	Faza	Naziv faze	Delež v %
0	DODATNE AKTIVNOSTI	6.3	0.1	Tedenski sestanki	0.7
			0.2	Zbiranje merskih podatkov	3.1
			0.3	Upravljanje konfiguracije	1.0
			0.4	Upravljanje sprememb zahtev	0.8
			0.5	Nadgradnja procesa	0.3
			0.6	Spremljanje tveganj	0.4
1	ZAČETEK PROJEKTA	6.9	1.1	Priprava projekta	0.3
			1.2	Strat. načrtovanje in idejna rešitev	2.1
			1.3	Planiranje projekta	3.4
			1.4	Vodenje konfiguracij in zahtev	0.7
			1.5	Poročilo vodstvu organizacije	0.1
			1.6	Zaključek priprave projekta	0.3
2	RAZISKAVA ZAHTEV	22.2	2.1	Priprava raziskave zahtev	1.1
			2.2	Zajem zahtev	9.3
			2.3	Analiza zahtev in logični model	8.2
			2.4	Obseg rešitve in specifikacija zahtev	3.2
			2.5	Zaključek raziskave zahtev	0.4
3	NAČRTOVANJE	19.5	3.1	Priprava načrtovanja	2.1
			3.2	Konceptualni načrt in prototipi	4.7
			3.3	Teh. rešitev in strat. Implementacije	3.7
			3.4	Fizični načrt PB	3.1
			3.5	Fizični načrt IS	2.9
			3.6	Preverjanje načrtov in spec. načrt.	2.5
			3.7	Zaključek načrtovanja	0.5
4	IZGRADNJA	28.8	4.1	Priprava izgradnje	2.1
			4.2	Izgradnja PB	5.7
			4.3	Podroben načrt modulov (po delih)	5.1
			4.4	Izgradnja modulov (po delih)	10.7
			4.5	Testiranje (po delih)	4.5
			4.6	Zaključek izgradnje	0.7
5	TESTIRANJE	11.2	5.1	Priprava testiranja	1.5
			5.2	Sistemsko testiranje	2.7
			5.3	Priprava dokumentacije in usposabljanja	2.4
			5.4	Testiranje izgleda in dokumentacije	1.2
			5.5	Uporabniško testiranje	2.3
			5.6	Zaključek testiranja	1.1
6	UPORABA	3.8	6.1	Priprava uporabe	1.1
			6.2	Namestitev	1.0
			6.3	Vzdrževanje	1.3
			6.4	Zaključek uporabe	0.4
7	ZAKLJUČEK PROJEKTA	1.3	7.1	Arhiviranje projekta	0.3
			7.2	Analiza projekta	0.7
			7.3	Poročilo vodstvu in zaključek projekta	0.3
		100			100

Tabela 4.4: Primer tipičnih razmerij aktivnosti prilagodljivega procesa

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Uporaba tabele omogoča enostavno delitev globalnih projektnih količin med posamezne faze in cikle. Vendar moramo biti pri tem previdni. Razmerja so določena za vse faze in cikle, mi pa moramo zaradi izpuščanja aktivnosti izbranega projektnega procesa razvoja upoštevati samo tiste, ki jih bomo zares izvajali. Te lahko poiščemo v planu opravil projekta. Možni so naslednji primeri:

- Optimalno je, če obstaja podroben plan za vse cikle in faze.
  - Če izvajamo čisto vse faze in cikle, uporabimo tabelo brez sprememb.
  - Običajno izpustimo določene cikle ali faze. V tem primeru upoštevamo samo izvedene cikle in faze, njihove deleže pa normiramo glede na vsoto deležev izvedenih faz.
- Pri večcikličnem izvajanju procesa razvoja obstajajo usklajeni plani na dveh ravneh: globalni za celoten projekt in podrobni za izvedbo cikla.
  - V večini primerov je globalni plan določen do ravni faz posameznih ciklov, zato z uporabo tabele ni težav (prvi primer).
  - Vendar prilagodljivi proces dopušča tudi globalne plane, kjer planiramo samo na ravni cikla. Tedaj obstaja poleg globalnega plana še vsaj en plan cikla (lahko tudi več), kjer so določene tudi faze in koraki. Tabelo razmerij uporabljamo na dveh ravneh:
    - Najprej ugotovimo, kateri cikli se bodo izvajali in normiramo podatke glede na vsoto vseh deležev izvajajočih ciklov. S tem se bodo odstotni deleži izvajajočih ciklov povečali.
    - Nato upoštevamo podrobne plane tistih ciklov, ki jih poznamo, in razdelimo spremenjeni odstotni delež cikla med tiste faze, ki se bodo izvajale. Če se v določenem ciklu izvaja samo ena faza, bo celoten odstotni delež cikla enak deležu faze.
- Ker prvič izvajamo ocenjevanje še pred planiranjem, plani še ne obstajajo. V tem primeru predvidimo, da se bodo čisto vse faze izvajale in upoštevamo celotno tabelo.

Zgled:

Denimo, da izvajamo enostaven projekt, ki se izvaja v več ciklih. Vse zahteve so bile že vnaprej popolnoma znane in opisane z logičnim modelom. Projekt zahteva samo načrtovanje, izgradnjo in testiranje procesnega dela. Ob predpostavki, da se trenutno nahajamo pred fazo priprave izgradnje in je globalni plan določen samo na ravni ciklov, nam podrobnosti izvedbe faz izgradnje in testiranja še niso znane. Če uporabimo za izhodišče tabelo 4.4, bi izračun razmerja za fazo 3.5 (Fizični načrt IS) izvedli na naslednji način:

novi delež 3. cikla =
stari delež 3. cikla /(1 - deleži neizvedenih ciklov) =
0.195 / (1 - 0.222 - 0.038) = 0.26351 = **26.35 %**

novi delež faze 3.5 =
(stari delež faze 3.5) / (1 - deleži vseh neizvedenih faz) =
(stari delež faze 3.5 * novi delež 3. cikla) / (stari delež 3.cikla - deleži neizvedenih faz 3.cikla) =
(0.029 * 0.2635) / (0.195 - 0.037 - 0.031) = 0.000764 / 0.127 = 0.0060157 = **6.02 %**

Vsa razmerja so predstavljena v tabeli 4.5 na dve decimalki natančno.

Drugi del ocenjevanja obsega dela je **ocenjevanje v okviru posameznih ciklov**. To se običajno izvaja v fazah priprave cikla. V fazi planiranja projekta se pri večciklični izvedbi izvede samo za cikel raziskave zahtev, pri izvedbi v enem ciklu pa v celoti za vse aktivnosti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Če uporabljamo pri oceni celotnega obsega dela in obsega dela ciklov pristop od zgoraj navzdol, uporabimo pri določanju obsega dela za posamezne aktivnosti pristop od spodaj navzgor. Pri tem si pomagamo s planom opravil (izvedbe aktivnosti) in tipično učinkovitostjo posameznikov, ki jo lahko razberemo iz zgodovinskih podatkov. Ocenjevanje te vrste obsega dela je tesno povezano tudi s planiranjem porabe časa.

	Naziv cikla	Delež v %	Faza	Naziv faze	Delež v %
0	DODATNE AKTIVNOSTI	6.3 8.51	0.1	Tedenski sestanki	0.7 0.95
			0.2	Zbiranje merskih podatkov	3.1 4.19
			0.3	Upravljanje konfiguracije	1.0 1.35
			0.4	Upravljanje sprememb zahtev	0.8 1.08
			0.5	Nadgradnja procesa	0.3 0.40
			0.6	Spremljanje tveganj	0.4 0.54
1	ZAČETEK PROJEKTA	6.9 9.32	1.1	Priprava projekta	0.3 0.58
			1.2	Strat. načrtovanje in idejna rešitev	2.1
			1.3	Planiranje projekta	3.4 6.60
			1.4	Vodenje konfiguracij in zahtev	0.7 1.36
			1.5	Poročilo vodstvu organizacije	0.1 0.19
			1.6	Zaključek priprave projekta	0.3 0.58
2	RAZISKAVA ZAHTEV	22.2	2.1	Priprava raziskave zahtev	1.1
			2.2	Zajem zahtev	9.3
			2.3	Analiza zahtev in logični model	8.2
			2.4	Obseg rešitve in specifikacija zahtev	3.2
			2.5	Zaključek raziskave zahtev	0.4
3	NAČRTOVANJE	19.5 26.35	3.1	Priprava načrtovanja	2.1 4.36
			3.2	Konceptualni načrt in prototipi	4.7 9.75
			3.3	Teh. rešitev in strat. implementacije	3.7
			3.4	Fizični načrt PB	3.1
			3.5	Fizični načrt IS	2.9 6.02
			3.6	Preverjanje načrtov in spec. načrt.	2.5 5.19
			3.7	Zaključek načrtovanja	0.5 1.04
4	IZGRADNJA	28.8 38.92	4.1	Priprava izgradnje	2.1 2.84
			4.2	Izgradnja PB	5.7 7.70
			4.3	Podroben načrt modulov (po delih)	5.1 6.90
			4.4	Izgradnja modulov (po delih)	10.7 14.46
			4.5	Testiranje (po delih)	4.5 6.08
			4.6	Zaključek izgradnje	0.7 0.95
5	TESTIRANJE	11.2 15.14	5.1	Priprava testiranja	1.5 2.03
			5.2	Sistemsko testiranje	2.7 3.65
			5.3	Priprava dokumentacije in usposabljanja	2.4 3.24
			5.4	Testiranje izgleda in dokumentacije	1.2 1.62
			5.5	Uporabniško testiranje	2.3 3.11
			5.6	Zaključek testiranja	1.1 1.49
6	UPORABA	3.8	6.1	Priprava uporabe	1.1
			6.2	Namestitev	1.0
			6.3	Vzdrževanje	1.3
			6.4	Zaključek uporabe	0.4
7	ZAKLJUČEK PROJEKTA	1.3 1.76	7.1	Arhiviranje projekta	0.3 0.41
			7.2	Analiza projekta	0.7 0.95
			7.3	Poročilo vodstvu in zaključek projkta	0.3 0.41
		100			100

Tabela 4.5: Primer uporabe tabele razmerij aktivnosti

Preden ocenimo obseg dela posameznih aktivnosti, moramo najprej na podlagi znanih zahtev za izvedbo cikla natančno določiti faze in korake v okviru cikla, ki jih bo treba izvesti. Za vsako fazo so znani izdelki oziroma rezultati. Naš proces razvoja PO določa tudi izvajalca , ki bo izvedbo koraka opravil ali vodil. Ocena obsega dela je nato prepuščena tem izvajalcem.

Vsak posameznik mora oceniti potreben obseg dela za izvedbo aktivnosti, za katero je zadolžen. Običajno to stori na podlagi ocene končnega izdelka in razmerja, ki ga v primerjavi s tem predstavlja rezultat njegove izvedbe aktivnosti. V ta namen lahko v obratni smeri uporabi že predstavljeno normalizacijsko tabelo. Pri oceni lahko uporabi zgodovinske podatke o svoji produktivnosti pri enakih aktivnostih, podobnem delu ali približno enako velikih projektih, ki jih tudi lahko razbere iz repozitorija. V kolikor teh podatkov ni, lahko uporabi tudi podatke, ki veljajo za vse člane organizacije. Nekatere aktivnosti niso odvisne od velikosti in pri teh bo ocena temeljila na izkušnjah in podobnosti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

S tem, ko ocenjevanje aktivnosti prepustimo posameznikom, dosežemo dvoje:

- posamezniki so sami aktivno vključeni v postopek ocenjevanja, zato se čutijo zavezane in odgovorne. Njihovi rezultati so odraz njihovega videnja izvedbe aktivnosti.
- ker ocenjevanje obsega dela porazdelimo na mnogo delov, celota ni vidna. Ocene so natančnejše in ob združitvi ocen obsega dela v okviru faze ali cikla te objektivneje odražajo potreben obseg dela. V bistvu posameznikom ni potrebno formalno predpisovati načina ocenjevanja obsega dela, dovolj jim je ponuditi vse potrebne podatke in jih prepustiti njihovi lastni metodi ocenjevanja.

Obseg dela za izvedbo posameznih aktivnosti je tako ocenjen za vsak korak posebej. S tem še ni določeno, da bodo posamezniki sami izvajali korake, ki so jih ocenjevali. Pogosto jih bodo samo vodili. Natančnejši način izvedbe korakov določimo pri planiranju porabe časa, kjer bomo ocenjevali porabo časa in zadolžitve za vsako aktivnost posebej.

Ocene za posamezne aktivnosti seštejemo v skladu z združevalno strukturo in jih uporabimo za preverjanje prej ocenjenega obsega dela celotnega cikla in posameznih faz. Pri tem upoštevamo, da so nekateri koraki povezani z razvojem procesnega dela končnega izdelka, drugi s podatkovnim delom, tretji pa z obojim. Obseg dela za tretji del delimo med procesni in podatkovni del v enakem razmerju, kot je razmerje med obsegom dela prvih dveh delov. Če je proces razvoja PO ustrezno izvajan, bi se morali združeni podatki in oceni celotnega cikla približno ujemati. Ocene med seboj uskladimo, pri čemer dajemo enako težo vsem ocenam.

Pri ocenjevanju obsega dela v okviru priprav posameznih ciklov upoštevamo tudi izmerjene dejanske podatke o obsegu dela. Ti so zbrani ločeno za procesni in podatkovni del. Oceno celotnega obsega dela in obsega dela ciklov v tem primeru izvedemo na naslednji način:

- Obseg dela za celoten projekt ocenimo na enak način kot prvič, le da uporabimo popravljeni oceni velikosti, ki sta določeni na podlagi aktualnih podatkov.
- Oceni celotnega obsega dela razdelimo na dva dela:
  - del, ki v skladu s tabelo razmerij pripada že opravljenim fazam in
  - del, ki pripada fazam, ki jih moramo še opraviti.

  Delitev na opravljeni in neopravljeni del v skladu s tabelo razmerij upoštevamo tudi pri aktivnostih, ki se neprenehoma izvajajo.
- Del ocene obsega dela za opravljeno delo primerjamo z dejanskim obsegom dela za procesni in podatkovni del. Določimo razmerji med njima.
- Predvidimo, da bo razmerje še naprej ostalo enako in v skladu z njim spremenimo oceno za del aktivnosti, ki nas še čakajo.
- V skladu s tabelo razmerij določimo potreben obseg dela za izvedbo posameznih ciklov ali faz, ki nas še čakajo.

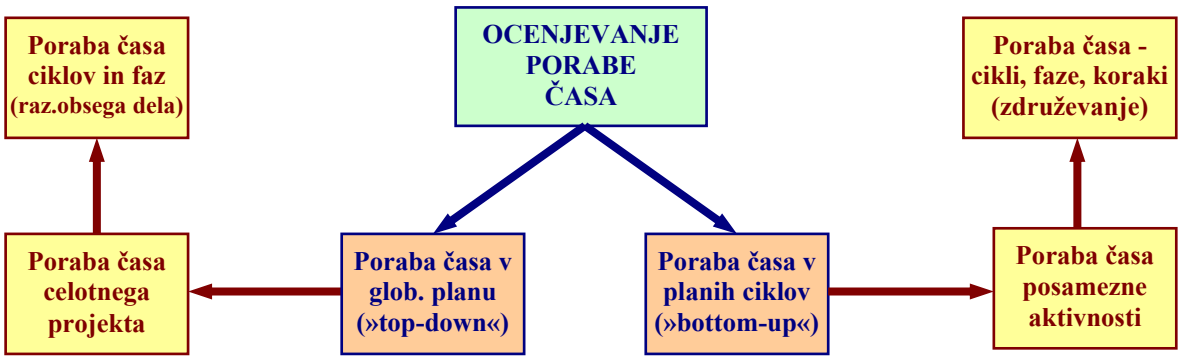
Pri popravljanju ocene obsega dela ne predvidimo uporabe linearne regresije, ker želimo popraviti rezultate glede na dejanske podatke za dani projekt. Pri določanju obsega dela po aktivnostih posamezniki postopajo na enak način kot prvič, le da upoštevajo aktualnejše podatke o velikosti in celotnem obsegu dela.

Ocenjevanje porabe časa

Ko poznamo obseg dela, lahko podrobno planiramo izvedbo projekta. To je neposredno povezano z ocenjevanjem porabe časa posameznih aktivnosti, korakov, faz, ciklov ali celotnega projekta. Tudi porabo časa ocenjujemo na dveh ravneh (slika 4.26):

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- globalno za celoten projekt ali posamezne cikle in
- natančno za vsak cikel posebej v fazi priprave cikla.



Slika 4.26: Ocenjevanje porabe časa po našem merskem pristopu

Globalno oceno porabe časa za celoten projekt lahko izvedemo na podlagi ocen obsega dela za procesni in podatkovni del, številčnosti projektne skupine in zgodovinskih podatkov. Ker predvidevamo, da se število razvijalcev ne spreminja, celoten obseg dela podatkovnega in procesnega dela delimo s številom razvijalcev. Nato uporabimo linearno regresijo nad podatki o obsegu dela (na člana) in trajanju preteklih projektov (v efektivnih dnevih). Z uporabo regresijske enačbe določimo število delovnih dni trajanja projekta ter z uporabo koledarja tudi končni datum. Ker imamo zbrane zgodovinske podatke o razmerju med ocenami časa trajanja in dejanskim trajanjem, določimo še območje 90% zaupanja. Z uporabo tipičnih razmerij med aktivnostmi na podoben način kot pri obsegu dela okvirno določimo dolžino trajanja posameznih ciklov in faz (v delovnih dnevih) oziroma določimo glavne mejnike (datume).

To ni dovolj natančno za planiranje porabe časa cikla, ki ga izvajamo v fazi priprave cikla. **Natančnejše ocenjevanje porabe časa** zahteva upoštevanje obsega dela za posamezne aktivnosti, pri čemer upoštevamo, da določene aktivnosti sodijo v procesni in druge v podatkovni del razvoja. Prvi del je bil opravljen že s tem, ko smo na podlagi procesa razvoja PO posameznim aktivnostim dodelili izvajalce in so ti ocenili obseg dela za svoje aktivnosti. Njihove ocene obsega dela moramo uskladiti, jim po potrebi določiti pomočnike in na tej podlagi določiti optimalen razpored opravil. Pri tem izhajamo iz naslednjih predpostavk:

- Pri ocenjevanju porabe časa predvidevamo, da se število članov projektne skupine ne spreminja oziroma se ne spreminja število ključnih razvijalcev. To je glede na majhnost organizacije in projektov realna predpostavka.
- Glede na določila naše organizacije lahko vsak član projektne skupine izvaja vse aktivnosti. Pri tem je sicer treba uporabljati logično presojo, vendar predvidevamo, da vsi člani projektne skupine sodelujejo ves čas izvajanja projekta.
- V skladu s tem je idealna tista razporeditev, po kateri so vsi člani projektne skupine ves čas enakomerno obremenjeni. Tega v popolnosti ni mogoče doseči, vendar postopki v okviru ocenjevanja in planiranja porabe časa težijo k temu.
- Pri določanju potrebnega časa za izvedbo posamezne aktivnosti ali koraka predvidimo dovolj časa. To pomeni, da potreben čas za izvedbo aktivnosti z določenim obsegom dela ni najkrajši možen čas ob optimalnem poteku aktivnosti, temveč čas, ki ga potrebujemo ob povprečnem poteku izvajanja. Ta predvidi tudi nekaj zapletov in zamud [DeM86].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Pri ocenjevanju porabe časa upoštevamo formalne roke izvedbe ali zahteve po čim krajši izvedbi, če je to smiselno. Ocena porabe časa za projekt ali izbran cikel pa ni obvezujoča in jo v bistvu preverjamo.
- Pri planiranju najprej opredelimo ključne aktivnosti. Te so običajno tiste, ki zahtevajo največ obsega dela in v kolikor se ne izvedejo po planih, zamaknejo celotno izvedbo projekta (kritične poti). Ostale aktivnosti so premakljive.
- Med aktivnostmi obstajajo medsebojne odvisnosti. Za ustrezen vrstni red poskrbi že proces razvoja PO, vendar so nekatere aktivnosti veliko krajše od drugih in zato lahko prihaja do vzporednega izvajanja. Naš pristop skuša izvajati aktivnosti vzporedno povsod tam, kjer je to mogoče. Ta se predvsem nanaša na razvoj izdelkov procesnega in podatkovnega dela, ki je večinoma dokaj neodvisen.
- Ko pride na vrsto, se večina aktivnosti izvede naenkrat. Obstajajo pa tudi aktivnosti, ki se izvajajo neprenehoma ali pa se sprožijo zgolj na podlagi nepredvidljivih dogodkov, do katerih lahko pride večkrat ali nikoli. Za oboje lahko določimo potreben čas samo približno na podlagi izkušenj, pri čemer pazimo, da predvidimo dovolj rezerve.

Na podlagi teh predpostavk zastavimo oceno porabe časa (recimo v minutah) za vsako aktivnost, ki je neposredno povezana s planom opravil cikla in planom porabe časa cikla. Predvidimo tudi vse izvajalce posameznih korakov, čeprav se to po potrebi lahko še spremeni. Aktivnosti nato z upoštevanjem združevalne strukture združimo v celotne korake, faze in cikel. Na podlagi porabe časa v minutah določimo ocene porabe časa za faze, ki so izražene v delovnih dneh. Ker naš proces razvoja predvideva jasne rezultate razvoja vsake faze, za mejnike določimo datume zaključka posameznih faz. Rezultati teh faz so sredstvo, s katerim preverjamo doseganje mejnikov.

Globalni plan porabe časa in podrobni plan izvedbe izbranega cikla bi se morala pri zrelem procesu razvoja PO približno ujemati. V kolikor se ne, globalni plan popravimo tako, da upošteva plan izvedbe cikla. To lahko povzroči zamik izvedbe naslednjih ciklov.

Ocenjevanje globalne porabe časa izvajamo ob vsakem začetku cikla in upoštevamo dejanske podatke. Najprej izvedemo enak postopek ocenjevanja kot prvič, le da izhajamo iz obsega dela, ki ga še moramo vložiti do konca projekta. Nato moramo upoštevati še zamike, do katerih je prišlo glede na izmerjene podatke. Glede na trenutno stanje projekta ugotovimo, kateri je bil zadnji mejnik, ki je bil dosežen in določimo razliko med našo ceno in dejanskim doseganjem mejnika. Nato vse prihodnje mejnike (z upoštevanjem koledarja) zamaknemo za to razliko.

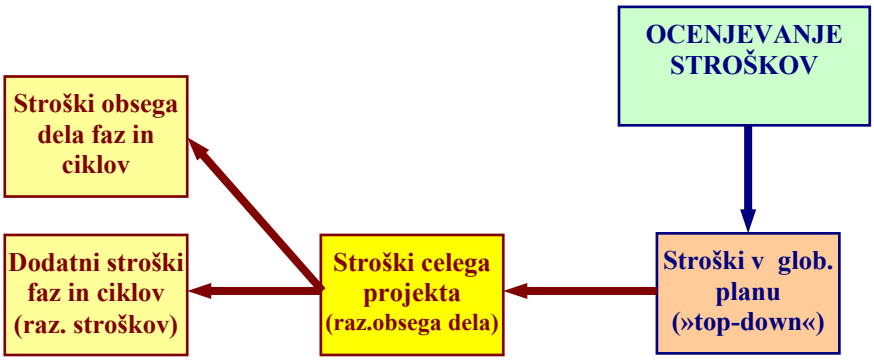
Natančno ocenjevanje porabe časa vedno izvajamo za različne cikle in ga ne ponavljamo, zato upoštevanje dejanskih podatkov ni potrebno. Izjemoma bi se to lahko zgodilo le tedaj, ko bi analiza ob koncu cikla pokazala, da celoten cikel ni bil ustrezno izveden. Vendar tudi tedaj ocenjujemo celoten cikel brez upoštevanja dejanske porabe časa za posamezne aktivnosti ob prvem izvajanju.

Naročnik ali vodstvo organizacije pogosto omeji rok za izvedbo projekta ali pa zahteva čim krajši rok. To lahko povzroči težnje po izpuščanju določenih korakov ali faz, ki bi jih sicer morali izvesti. Včasih je to smiselno, pogosto pa ima to za posledico manjšo kakovost izdelkov. Naš pristop z dvojnim določanjem obsega dela in porabe časa – globalno in z združevanjem ocen posameznih aktivnosti – pripomore k temu, da lahko utemeljeno zagovarjamo svoje plane pred vodstvom podjetja ali naročnikom. Posledica tega so realnejši plani, ki so v interesu vseh.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Ocenjevanje stroškov

Naš pristop za razliko od pristopa PSM predvideva tudi spremljanje stroškov, kar je zahteva modela CMM. Stroškov ne ocenjujemo za posamezne aktivnosti, temveč samo za celoten projekt, posamezne cikle in faze. Zato jih tudi določamo samo po pristopu od zgoraj navzdol: iz celotnih ocenimo stroške posameznih faz in ciklov (slika 4.27).

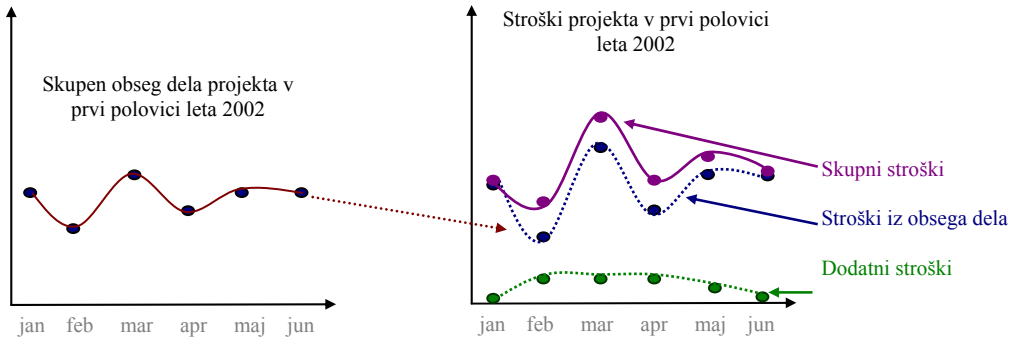


Slika 4.27: Ocenjevanje stroškov po našem merskem pristopu

Za merjenje stroškov so znani modeli, ki izhajajo iz velikosti izdelkov (recimo SLIM [Put78, Fen97, Hei99]). V našem primeru pri ocenjevanju skupnih stroškov projekta izhajamo iz ocene obsega dela (skupaj seštejemo obseg dela za procesni in podatkovni del) in zgodovinskih podatkov o razmerju med obsegom dela in stroški. Obseg dela je boljše izhodišče kot velikost zaradi tega, ker upošteva tudi subjektivne poglede razvijalcev. Za ocenitev stroškov uporabimo linearno regresijo. Na podlagi preteklih ocen in dejanskih stroškov po standardni poti določimo tudi območje zaupanja.

Če gledamo s stališča celotnega projekta je skupna višina stroškov še najbolj odvisna od celotnega obsega dela [Hum95] in naš pristop je zato pravilen. Če pa pogledamo na raven posameznih faz, imajo določene faze tudi dodatne stroške, ki niso neposredno odvisni od obsega dela posameznih faz. Čeprav predstavljajo manjši del stroškov projekta, ti spreminjajo razmerja med celotnimi stroški posameznih faz. Nekatere faze recimo zahtevajo sodelovanje z uporabniki, kar je povezano z večjimi stroški (recimo faze raziskave zahtev, namestitve ali vzdrževanja), druge pa so pretežno avtomatizirane in skoraj nimajo dodatnih stroškov. Naš pristop zajema podatke o dodatnih stroških stroškov ob izvedbi vseh aktivnosti posameznika.

Govorimo torej o dveh vrstah stroškov: o stroških iz obsega dela ter dodatnih stroških (slika 4.28).



Slika 4.28: Dve vrsti stroškov: stroški iz obsega dela in dodatki stroški

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Tabela tipičnih razmerij obsega dela za aktivnosti predstavlja razmerja med tistimi deli stroškov faz, ki izhajajo neposredno iz obsega dela. Če želimo biti natančnejši, moramo za delitev stroškov po fazah in ciklih določiti še posebno tabelo tipičnih razmerij dodatnih stroškov. Ta je določena na popolnoma enak način kot tabela tipičnih razmerij obsega dela aktivnosti, le da se nanaša na dodatne stroške posameznih aktivnosti. Poleg tega obstaja v repozitoriju še podatek o tipičnem razmerju organizacije med celotnimi stroški iz obsega dela in celotnimi dodatnimi stroški. Tabela in razmerje stroškov se ob vsakem dokončanju projekta spreminja, tako da odraža celotne zgodovinske podatke.

Ko določamo stroške posameznih faz, uporabimo podatek o tipičnem razmerju in obe tabeli. Najprej na podlagi tipičnega razmerja določimo oba deleža stroškov: prvega, ki izhaja iz obsega dela, in drugega, ki se nanaša na dodatne stroške. Pri prvem nato uporabimo tabelo tipičnih razmerij obsega dela posameznih aktivnosti in določimo deleže stroškov za vse faze in cikle, ki se bodo izvajali. Enako ravnamo v drugem primeru, le da stroške posameznih faz in ciklov določimo na podlagi tabele tipičnih razmerij dodatnih stroškov. Za vsako fazo (in cikle) oba dela stroškov seštejemo in ocenjevanje je zaključeno.

Ko ocenjujemo globalne stroške ob vsakem začetku cikla, moramo upoštevati tudi dejanske podatke o obsegu dela in dodatnih stroških izvedenih aktivnosti. Pri tem uporabimo podoben pristop kot pri določanju globalnega obsega dela:

- Stroške za celoten projekt ocenimo na enak način, le da uporabimo vsoto popravljenih ocen obsega dela (za procesni in podatkovni del), ki sta določeni iz aktualnih podatkov.
- Oceno celotnih stroškov razdelimo na dva dela: stroške iz obsega dela in dodatne stroške.
- Oceno stroškov iz obsega dela še naprej delimo na dva dela:
  - del, ki v skladu s tabelo razmerij obsega dela pripada že opravljenim fazam in
  - del, ki pripada fazam, ki jih moramo še opraviti.

  Delitev na opravljeni in neopravljeni del v skladu s tabelo razmerij upoštevamo tudi pri aktivnostih, ki se neprenehoma izvajajo.
- Enako ravnamo pri oceni dodatnega obsega dela. Razdelimo ga na dva dela:
  - del, ki v skladu s tabelo razmerij dodatnih stroškov pripada že opravljenim fazam in
  - del, ki pripada fazam, ki jih moramo še opraviti.

  Tudi tu upoštevamo enako razmerje za aktivnosti, ki se neprenehoma izvajajo.
- Del ocene stroškov iz obsega dela za opravljene faze primerjamo s celotnim dejanskim obsegom dela za ta del. Določimo razmerje med njima. Predvidimo, da bo razmerje še naprej ostalo enako in v skladu z njim spremenimo oceno za del prihodnjih aktivnosti.
- Enako ravnamo z dodatnimi stroški. Primerjamo dejanske dodatne stroške in oceno stroškov za že dokončane faze in na podlagi razmerja med njima spremenimo ocene dodatnih stroškov za tiste faze, ki se šele morajo izvesti.
- Seštejemo obe vrsti obsega dela za že izvedene in prihodnje faze. Na ta način določimo oceno celotnih stroškov.
- V skladu z razmerjem med stroški iz obsega dela in dodatnimi stroški ter obema tabelama razmerij določimo stroške za izvedbo posameznih ciklov in faz, ki nas še čakajo.

Ocenjevanje kakovosti

Kakovost končnih izdelkov si lahko predstavljamo kot zbirko lepih lastnosti (kot so pravilno delovanje, uporabnost, primernost za vzdrževanje in podobno) [McC77, Boe78], ki pa jih običajno ni mogoče direktno meriti [Fen97]. Zato se pri spremljanju kakovosti osredotočimo na edino direktno meritev: spremljanje napak. Na podlagi napak skušamo čim bolje presoditi tudi druge lastnosti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Običajen pristop pri ocenjevanju kakovosti uporablja podatke o odkritih in odpravljenih napakah ter velikost preteklih projektov. Glede na oceno velikosti končnega izdelka trenutnega izdelka tako lahko predvidimo, koliko napak bomo naredili in odpravili v trenutnem projektu. Podobno kot drugod obstajata tudi tu dve ravni ocenjevanja:

- globalno ocenjevanje kakovosti končnega izdelka in
- ocenjevanje kakovosti posameznih faz oziroma ciklov.

Glede na pomembnost spremljanja kakovosti želimo biti v obeh primerih čim bolj določeni. Pri tem lahko uporabimo principe združevanja, ki so določeni v merskem modelu. Na podlagi le-teh, zgodovinskih podatkov, planov in dejanskih podatkov lahko določimo več razmerij. Združevalne strukture nam omogočajo združevati podatke na naslednje načine:

- po delih (kateri izdelki sestavljajo druge izdelke);
- po funkcijah (vse aktivnosti z isto funkcijo, recimo vsa preverjanja kakovosti);
- po aktivnostih (koraki sestavljajo faze, faze sestavljajo cikle);
- po vidiku uporabe (podatkovni in procesni vidik);
- po uporabnikih (aktivnosti, ki jih izvaja isti uporabnik).

Zgodovinski podatki obsegajo podatke o več projektih in nam omogočajo natančno določiti naslednja razmerja:

- Za vsak projekt natanko vemo, kateri cikli, faze in koraki so bili izvršeni.
- Za vsak korak natanko vemo, kdo je vodil njegovo izvedbo in kdo je pri njem sodeloval.
- Za vsakega izvajalca aktivnosti natanko poznamo rezultate oziroma njihovo velikost, obseg dela, čas začetka in zaključka, dodatne stroške in število odkritih in odpravljenih napak, tipe teh napak ter obseg dela, ki je bil za odpravljanje potreben.

Iz planov lahko določimo:

- katere aktivnosti (koraki, faze, cikli) so planirane,
- kdo jih bo izvajal (vodil in sodeloval pri njih) in
- oceno končne velikosti, obsega dela po posameznih aktivnostih, porabo časa in stroškov.

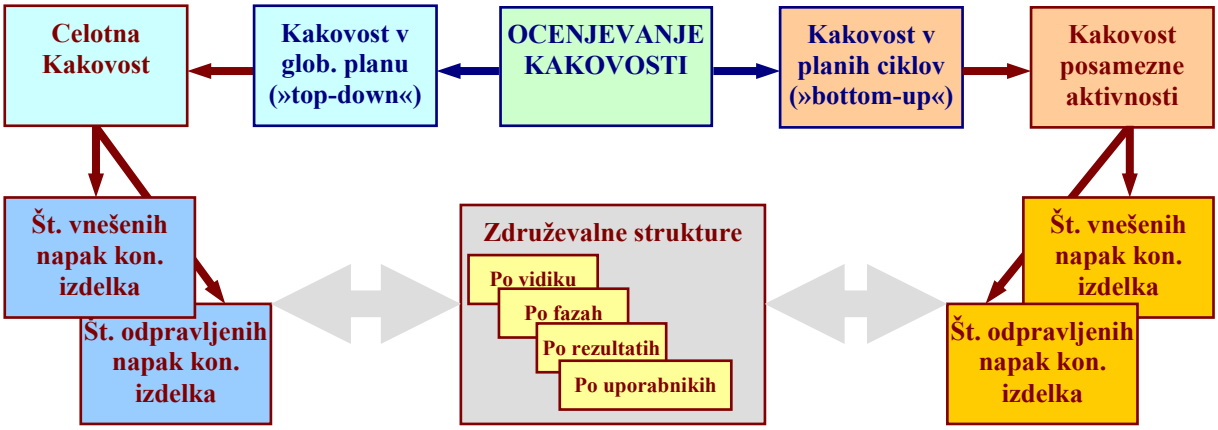
Dejanski podatki predstavljajo velikost rezultatov, potreben obseg dela, stroške in kakovost za aktivnosti, ki so že bile izvedene.

Ker zbiramo podrobne osnovne podatke, lahko preverjamo število vnešenih in odpravljenih napak po različnih kriterijih:

- po procesnem ali podatkovnem vidiku,
- po posameznih fazah ali ciklih,
- po posameznih rezultatih in
- po posameznih izvajalcih.

Ustrezno ocenjevanje kakovosti zagotovimo z večkratnim ocenjevanjem in uporabo različnih kriterijev. Prvi kriterij uporabimo pri globalnem ocenjevanju, ostale pa pri natančnem. Na koncu obe ocenjevanji združimo in uskladimo. Slika 4.29 prikazuje izvedbo ocenjevanja kakovosti po našem merskem pristopu.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.29: Ocenjevanje kakovosti po našem merskem pristopu

Globalno ocenjevanje izvedemo z uporabo linearne regresije. Pri tem določimo štiri regresijske enačbe:

- število vnešenih napak v procesni del končnega izdelka je določeno z razmerjem med velikostjo procesnega dela končnega izdelka in številom vnešenih napak v preteklih projektih;
- število vnešenih napak v podatkovni del končnega izdelka je določeno z razmerjem med velikostjo podatkovnega dela končnega izdelka in številom vnešenih napak v preteklih projektih;
- število odpravljenih napak v procesnem delu končnega izdelka je določeno z razmerjem med velikostjo procesnega dela končnega izdelka in številom odpravljenih napak v preteklih projektih;
- število odpravljenih napak v podatkovnem delu končnega izdelka je določeno na podlagi razmerja med velikostjo podatkovnega dela končnega izdelka in številom odpravljenih napak v preteklih projektih.

Za vsako od globalnih ocen lahko določimo tudi območja zaupanja. Pri tem izhajamo iz zgodovinskih podatkov o razmerju med isto vrsto ocene napak in izmerjenim številom napak.

Globalne ocene kakovosti niso natančne, zato jih popravimo z izvedbo natančnega planiranja kakovosti. Ker iz plana cikla natanko poznamo aktivnosti in odgovorne razvijalce zanje, izvedemo ocenjevanje za vsako aktivnost posebej. Pri tem ne izhajamo iz velikosti končnega izdelka, temveč iz predvidenega obsega dela za izvedbo posameznih aktivnosti. Postopek ocenjevanja je natančno določen:

Uporabimo linearno regresijo in določimo dve linearni enačbi. Prva določa, kako na podlagi obsega dela izbrane aktivnosti izračunamo oceno števila vnešenih napak med izvedbo aktivnosti. Druga določa izračun števila odpravljenih napak glede na obseg dela aktivnosti.

Pri določanju enačb uporabimo zgodovinske podatke o razmerjih med vnešenimi (oziroma odpravljenimi) napakami in obsegom dela aktivnosti, pri čemer skušamo uporabiti čim bolj primerne zgodovinske podatke:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- V kolikor je mogoče uporabimo zgodovinske podatke za projekte, kjer je isti izvajalec izvajal isto aktivnost. Obstajati morajo vsaj trije taki primeri.
- Če to ni mogoče, uporabimo zgodovinske podatke, kjer je isti izvajalec izvajal aktivnost, ki sodi v isto funkcijsko skupino. Obstajati morajo vsaj trije taki primeri.
- Če tudi to ni mogoče, uporabimo zgodovinske podatke za vse izvajalce v organizaciji, ki so izvajali isto aktivnost. Obstajati morajo vsaj trije taki primeri.
- Če je zbranih podatkov res zelo malo se lahko zgodi, da moramo uporabiti povprečja celotne organizacije za katerokoli aktivnost.

Ko določimo obe oceni za vsako aktivnost posebej, ocene združimo s pomočjo združevalnih struktur po vseh štirih kriterijih:

Po procesnem ali podatkovnem vidiku:

- seštejemo ločeno števila vnešenih in odpravljenih napak za vse aktivnosti, ki se nanašajo na procesni vidik razvoja projekta;
- seštejemo ločeno števila vnešenih in odpravljenih napak za vse aktivnosti, ki se nanašajo na podatkovni vidik razvoja projekta;
- za splošne aktivnosti enako ločeno seštejemo števila vnešenih in odpravljenih napak in jih prištejemo vsakemu vidiku pol. Predvidoma te aktivnosti ne vnašajo (ali odpravljajo) veliko napak.

Po fazah izbranega cikla:

- za vsako fazo v okviru cikla ločeno seštejemo števila vnešenih in odpravljenih napak za vse aktivnosti, ki pripadajo fazi.

Po rezultatih izbranega cikla:

- iz podatkov o merjenju razberemo glavne rezultate cikla;
- za vsak glavni rezultat v okviru cikla ločeno seštejemo števila vnešenih in odpravljenih napak za vse aktivnosti, ki so vezane na izbran rezultat.

Po uporabnikih:

- po posameznih članih projektne skupine ločeno seštejemo števila vnešenih in odpravljenih napak za vse aktivnosti.

Čeprav ne ocenjujemo istih atributov kot pri globalnih ocenah lahko ugotovimo, da so združene ocene natančnejše od globalnih ocen kakovosti. Ob primerni izvedbi ocenjevanja namreč natančno upoštevajo posebnosti posameznih aktivnosti in izvajalcev, ki jih izvajajo. Če nič drugega z upoštevanjem ocen obsega dela posameznih aktivnosti nekoliko upoštevamo tudi subjektivne ocene posameznikov.

Ocene kakovosti ob globalnem planiranju projekta nam omogočajo planirati kakovost. Na njihovi podlagi lahko predvidimo faze, izdelke in izvajalce, kjer je treba zagotavljanju kakovosti posvetiti več pozornosti. Za dinamično spremljanje kakovosti so namenjene druge analize. Prava uporabnost ocenjevanja kakovosti po več kriterijih se tako pokaže pri napovedovanju in analizi performans.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

V skladu s to ugotovitvijo je treba razumeti tudi ocenjevanje kakovosti ob začetkih ciklov. Tedaj sicer upoštevamo dejanske izmerjene vrednosti, vendar to še ne omogoča pravega vpogleda.

Pri določanju štirih globalnih ocen uporabimo popravljeno oceno velikosti končnega izdelka in na ta način bolje ocenimo števila napak obeh delov. Ker poznamo števila odkritih in odpravljenih napak v že dokončanem delu, lahko te napake odštejemo od celotnega števila predvidenih napak in predvidimo, koliko napak nas še čaka. Taka ocena je dokaj nenatančna, a je največ, kar lahko storimo. Globalne ocene kakovosti se namreč nanašajo na celoten projekt in nimamo učinkovitega mehanizma, kako bi izločili del, ki ustreza že opravljenemu (in izmerjenemu) delu projekta. Kot rečeno v ta namen služijo napovedi in analiza performans.

Natančno planiranje ob začetkih ciklov je namenjeno ocenjevanju kakovosti v trenutnem ciklu, torej si z izmerjenimi podatki preteklega cikla nič ne moremo pomagati. Upoštevamo lahko le natančneje določen obseg dela.

4.3.1.4 Ovrednotenje ocen

Ovrednotenje ocen pri pristopu PSM obsega tri vidike:

- kakovost ocen: ocene morajo biti usklajene, popolne in zanesljive,
- upoštevanje omejitev: ocene morajo biti v dovoljenih mejah in
- dokumentiranje ocen: vse okoliščine ocenjevanja morajo biti dokumentirane.

Pri našem pristopu je kakovost ocen zagotovljena na dva načina:

- Pri določanju ocen uporabljamo standardizirane postopke, ki temeljijo na preverjenih statističnih metodah in realnih podatkih preteklih projektov. Postopki se uporabljajo vedno na enak način. Uporabljeni zgodovinski podatki se nanašajo na podatke ustrezne vrste.
- Poleg tega pri ocenjevanju vedno skušamo upoštevati različne pristope. Tako kombiniramo pristope od zgoraj navzdol in od spodaj navzgor. Pri ocenjevanju upoštevamo učinkovite prakse, izkušnje preteklih projektov in vidik razvijalcev. Ocene uskladimo z vodstvom.

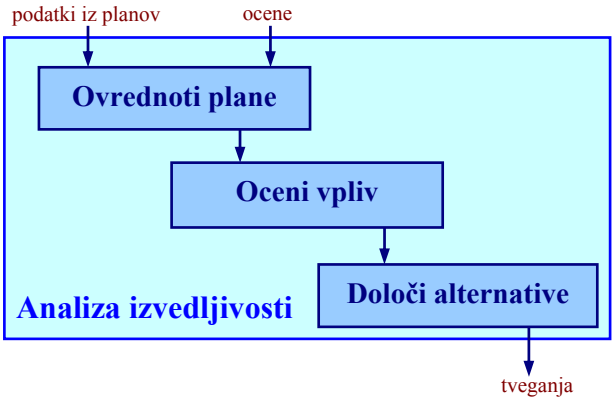
Uporabljena načela pri ocenjevanju zagotavljajo tudi upoštevanje dovoljenih meja in dejansko omogočajo tudi objektivno pogajanje o planih z vodstvom organizacije.

Predvidevamo, da je za dokumentiranje ocen poskrbljeno s pomočjo repozitorija. IS repozitorija poskrbi za ustrezen vnos podatkov. Vsi potrebni izračuni (recimo linearna regresija) so v IS vgrajeni. Merski podatki so bodisi shranjeni ali pa jih je mogoče z uporabo istih postopkov iz izmerjenih vrednosti znova določiti. Glede na to, da je organizacija majhna in izvaja majhne projekte, količina podatkov ni prevelika. Zato je ustaljena praksa, da ocene shranimo tudi potem, ko ocenjene vrednosti izmerimo ali bolje ocenimo.

4.3.2 Analiza izvedljivosti

Pomemben vzrok za neuspeh projektov so nerealni plani, do katerih lahko pride iz različnih vzrokov. Analiza izvedljivosti (slika 4.30) je namenjena pravočasnemu odkrivanju te težave.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.30: Koraki analize izvedljivosti

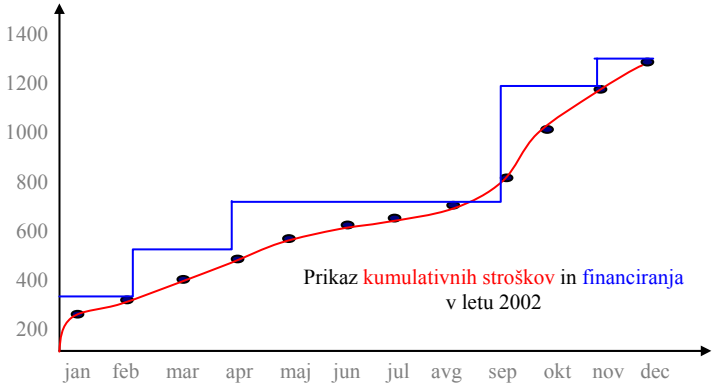
Uporablja se predvsem ob spremembah planov, pa tudi ob drugih pomembnih dogodkih, kot so spremembe zahtev, tehnologije ali organizacije dela. V okviru analize izvedljivosti najprej ovrednotimo plane, ugotovimo njihov vpliv ter določimo alternative, s katerimi zmanjšamo tveganja. Analiza izvedljivosti v pristopu PSM preverja tri vrste primerov:

- posebej preveri vsa izhodišča, ki imajo visoko prioriteto (na visoki ravni in podrobno);
- posebej preveri elemente, ki so sestavni deli pomembnih izhodišč (recimo število osebja);
- primerjalno oceni različna izhodišča.

Preverjanja izvajamo s pomočjo ustreznih pokazateljev. Običajno ti bodisi primerjajo elemente planov med seboj ali pa elemente planov z zgodovinskimi podatki. Pri tem uporabljamo pokazatelje dveh vrst:

- pokazatelje trenda planov in
- pokazatelje doseganja omejitev.

Pokazatelji trenda planov prikazujejo spremembo določenih planiranih vrednosti s časom. Najlažje jih predstavimo z grafi, kjer abscisna os predstavlja časovno obdobje, na ordinatni osi pa planirane vrednosti. Prikazujejo lahko kumulativne količine (glej sliko 4.31) ali posamične planirane vrednosti (slika 4.32). Pogosto na istem grafu združimo vrednosti več pokazateljev. Primere združevanja več pokazateljev na istem grafu opisujejo recimo [Por90, Pfl92, Das92].



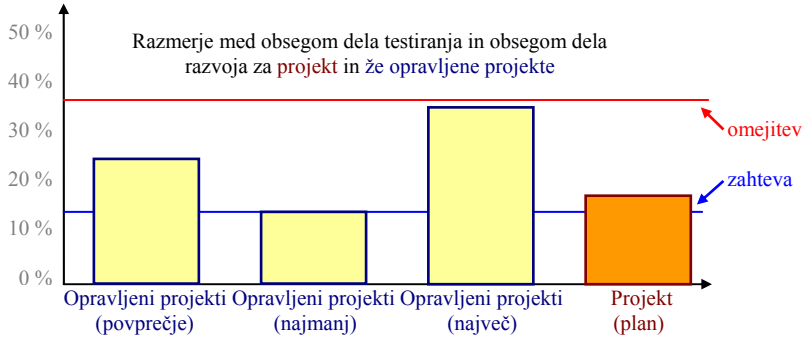
Slika 4.31: Primer pokazatelja kumulativnih količin pri analizah izvedljivosti

Pokazatelji doseganja omejitev ([DoD02]) prikazujejo trenutno stanje določenega atributa glede na omejene ali zahtevane vrednosti. Iz grafičnega prikaza je jasno razvidno, če so omejitve dosežene in koliko do njih manjka (slika 4.33).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 4.32: Primer pokazatelja posamičnih vrednosti pri analizah izvedljivosti



Slika 4.33: Primer pokazatelja doseganja omejitev

V našem primeru je analiza izvedljivosti namenjena predvsem vodstvu organizacije, ki na ta način preveri izvedljivost planov. IS repozitorija omogoča neposreden vpogled in izvajanje določenih preverjanj.

Analiza izvedljivosti se razlikuje od ocenjevanja. Pri ocenjevanju je pomemben postopek, ki zagotovi kakovostne ocene, medtem ko je bistvo analize izvedljivosti čim bolj nazoren prikaz zbranih podatkov. V okviru stalnega dela merjenja, ki je opredeljeno s stalnimi merskimi izhodišči, ne predvidevamo posebnih pokazateljev za analize izvedljivosti. Namesto teh si pomagamo z ocenami in napovedmi, ki so učinkovitejše in zadoščajo za naše potrebe.

To seveda ne pomeni, da opisanih pokazateljev izvedljivosti ne uporabljamo. Definiramo jih lahko v okviru posebnih merskih potreb projekta. Vendar za osnovno preverjanje izvedljivosti zadošča tisto, kar ponujajo pokazatelji napovedi.

4.3.2.1 Ovrednotenje planov

V tem koraku preverimo izvedljivost planov. Izhodišče so planski podatki in ocene, uporabimo pa tudi zunanje omejitve in pričakovana razmerja, ki jih razberemo iz zgodovinskih podatkov. Pri preverjanju pristop PSM uporablja naslednje kriterije:

- ustreznost, utemeljenost ocen,
- primernost upoštevanja vpliva okolja,
- popolnost plana,
- usklajenost elementov plana,
- primernost uporabljenega načina planiranja glede na izkušnje,
- možnosti sprememb predpostavk ali okolja, ki lahko vplivajo na plane in
- ustreznost plana glede na podobne projekte.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

V našem primeru se ovrednotenje planov izvaja v fazi planiranja projekta in v vseh fazah priprave ciklov. Po pripravi plana (globalnega ali plana cikla) je dolžnost projektnega vodje predstaviti plane vodstvu organizacije. Plani se nahajajo v repozitoriju in člani vodstva lahko po potrebi izvajajo analize. IS omogoča avtomatski prikaz ustreznih pokazateljev, tako da sodelovanja vodje procesa in merjenja običajno ne potrebujejo. Kot rečeno v ta namen ne določamo posebnih pokazateljev, temveč uporabimo napovedi.

Večji del kriterijev je načeloma izpolnjen že z uporabo standardiziranih postopkov, katerih izvedbo deloma nadzoruje tudi IS repozitorija. Med take lahko štejemo recimo utemeljenost ocen, popolnost plana, primernost uporabljenega načina planiranja in ustreznost glede na podobne projekte. Glede na predpostavko, da organizacija razvija samo PO izbranega tipa, se upoštevanje okolja navezuje predvsem na značilnosti posameznega projekta. Te so upoštevane z ustrezno izbiro projektnega procesa razvoja PO. Neusklajenosti elementov plana bi glede na nadzor IS repozitorija načeloma ne smelo biti. Možna težava je neupoštevanje vseh potrebnih vidikov v planih, do katerega lahko pride zaradi tega, ker projektna ekipa nima širšega pogleda na delovanje organizacije in vlogo projekta v njem.

4.3.2.2 Ocenjevanje vpliva

Pristop PSM namenja poseben korak presoji odkritih težav. V planiranje je bilo namreč vloženega že veliko dela in vsaka sprememba pomeni še dodaten obseg dela. Poleg tega spremembe planov niso neodvisne - pogosto zahtevajo še druge spremembe, ki lahko popolnoma porušijo koncept plana. Zato je potrebno presoditi, če je smiselno plane popravljati, oziroma kaj vse je treba popraviti.

Naš prilagodljiv proces razvoja PO predvideva izvedbo ocenjevanja vpliva skupaj z izvedbo analize izvedljivosti, torej pred in med predstavitvijo planov vodstvu. Pri tem sodeluje tudi vodja projekta, ki zagovarja pripravljene plane projektne skupine.

4.3.2.3 Določanje alternativ

Posledica presoje težav so predlogi, kako težave odpraviti ali vsaj zmanjšati njihov vpliv. S tem samo pripravimo predloge ukrepov, za katere se lahko v okviru naslednjih korakov izvedbe analiz odločimo. Običajno so dobro vodilo uspešne izkušnje preteklih projektov.

Naš pristop predvideva, da se o možnostih pogovori na sestanku ob predstavitvi planov. Ob tem pride tudi do odločitve o najprimernejših ukrepih. Ko se plani popravijo, se analiza izvedljivosti zopet izvede.

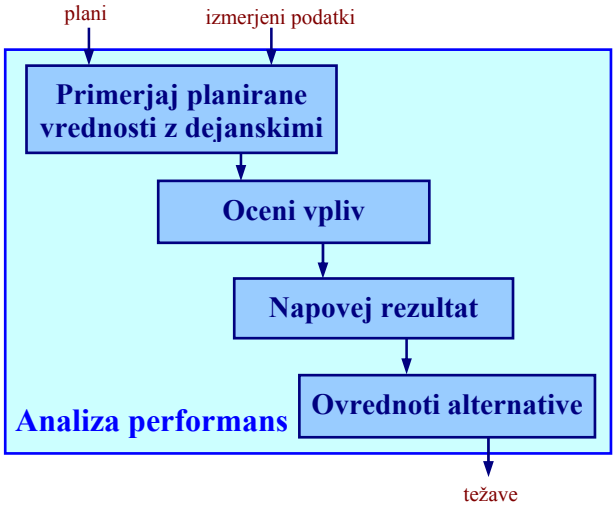
4.3.3 Analiza performans

Ko so predlagani plani preverjeni, usklajeni in potrjeni s strani vodstva, postane njihova vsebina vodilo in cilj, ki ga je treba doseči. Izmerjeni merski podatki povedo, v kolikšni meri plane zares dosegamo. Analiza performans je namenjena učinkovitemu prikazu razmerij med izmerjenimi podatki in planiranimi vrednostmi, ki omogoča pravočasno ukrepanje. Izvajamo jo namreč ves čas od trenutka potrditve planov do konca projekta.

Pristop PSM določa štiri korake analize performans (slika 4.34). Primerjava med planiranimi in dejanskimi vrednostmi se izvaja neprenehoma, ostale tri pa se izvedejo takrat, ko odkrijemo

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

težave. Podobno kot pri analizi izvedljivosti vedno najprej ocenimo vpliv odkrite težave. Nato predstavimo napoved (neugodnega) končnega rezultata do katerega bi prišlo, če težav ne odpravimo. Nazadnje predvidimo in ovrednotimo še več možnih rešitev problema.



Slika 4.34: Koraki analize performans

Analiza performans je po načinu izvedbe podobna analizi izvedljivosti. Izvajamo jo namreč s preverjanjem vrste različnih pokazateljev, ki naslavljajo vsa pomembna področja razvoja PO. Tudi uporabljeni vrsti pokazateljev sta enaki kot analizi izvedljivosti: pokazatelji trenda planov in pokazatelji doseganja omejitev, dodatno pa poznamo še pokazatelje statusa. Za analizo performans so tipični predvsem tisti pokazatelji, ki na istem grafu prikazujejo planske in izmerjene podatke glede na pretečen čas.

Naš pristop predvideva celovito ali delno izvedbo analize performans v treh obdobjih:

- delno izvedbo analize redno vsak teden pred tedenskimi sestanki projektne skupine,
- celovito analizo performans cikla in projekta ob zaključkih posameznih ciklov in
- celovito analizo celotnega projekta ob koncu projekta (v fazi analiza projekta).

Redno tedensko spremljanje performans se izvaja predvsem s pomočjo napovedi. Napovedi so po svoji tehnični izvedbi zelo podobne ocenam, vendar po namenu sodijo bolj v analizo performans. Z njihovo pomočjo lahko za nekaj glavnih atributov v vsakem trenutku presodimo, kakšen končni rezultat se nam obeta. Podrobno analizo performans uporabimo občasno in vedno takrat, ko napovedi pokažejo odklone od normalnega stanja ali sum, da nekaj ni v redu.

Pri celovitih analizah performans ob koncih ciklov ali projekta sistematično preverimo vse pomembne vidike razvoja PO. S tem tudi zberemo analizne podatke (imenujemo jih informacijski izdelki), ki so pomemben del poročil vodstvu.

Analize performans izvaja vodja procesa in merjenja, ki o stanju redno poroča vodji projekta. Ta po potrebi odredi še dodatne izvedbe analize performans ali kako drugače ukrepa.

Kot rečeno naš merski pristop poleg opisanih dveh vrst pokazateljev predvideva še tretjo: **prikaz statusa**. Prikaz statusa ni grafični pokazatelj, temveč avtomatiziran izpis vseh

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

podatkov, ki jih o določeni merski entiteti (izdelku, aktivnosti ali viru) poznamo. V bistvu se za pokazateljem ne skriva zahtevna analiza, temveč zgolj zbiranje vseh povezanih (merskih in drugih) podatkov ter ustrezen prikaz.

4.3.3.1 Primerjava planiranih vrednosti z dejanskimi

Prvo opravilo tega koraka je primerjava izmerjenih vrednosti osnovnih pokazateljev s planiranimi. Če razlika presega določene meje, moramo ukrepati. Težave napoveduje tudi neugoden trend, tudi če dejanski rezultati niso prekoračili meje.

Pri tem je treba analize vedno izvajati celovito. Preverjanje enega samega pokazatelja ne pove dovolj, treba je preveriti tudi z njim povezane pokazatelje. Pri tem veljajo naslednja pravila:

- Določeni pokazatelji napovedujejo obnašanje drugih v prihodnosti, recimo veliko sprememb zahtev pomeni povečan obseg dela.
- Posebej je treba preveriti kritične dele razvoja. Preveč usklajeni podatki so sumljivi.
- Dva različna pokazatelja lahko prikazujeta neusklajene rezultate določenega področja. To običajno ne pomeni napačnih meritev, temveč opozorilo. Neusklajeni rezultati običajno napovedujejo težave.
- Včasih se pojavijo rezultati, ki popolnoma odstopajo od vseh ostalih. Take primere je treba posebej preveriti.

Ko odkrijemo težavo, jo moramo podrobneje raziskati. To zahteva podrobnejše analize, ki se izvajajo nad podrobnejšimi podatki. Vedno se moramo zavedati, da pokazatelji kažejo posledice, naš namen pa je odkriti (in odpraviti) vzroke težav.

V našem primeru je pretežen del osnovne primerjave opravljen z napovedmi, vendar je treba občasno izvajati tudi analize določenih vidikov performans. Podobno kot pri drugih analizah skušamo tudi tu iz zbranih podatkov iztržiti čim več, pri čemer izkoriščamo prednosti avtomatizacije repozitorija.

Pokazatelji, ki jih uporabljamo pri analizi performans, se nanašajo na vse potrebne vidike procesa, izdelkov, virov in projekta. Z njihovo pomočjo preverjamo doseganje planiranih vrednosti za vsa izbrana merska izhodišča. Nekaj pokazateljev za analizo performans je podrobneje predstavljenih v okviru razdelka 5.5.4 in v dodatku B.2.5.1.

4.3.3.2 Ocenjevanje vpliva

Ko odkrijemo težave in njihov vzrok, je treba presoditi vpliv, ki ga imajo te na potek dela in rezultate. Zaradi medsebojne odvisnosti med značilnostmi procesa in izdelkov imajo včasih majhna odstopanja od planiranih vrednosti velike posledice in v takih primerih je ukrepanje nujno. V nasprotnem primeru se lahko zgodi, da so razlike med planiranimi in dejanskimi vrednostmi posledica neustreznih postopkov merjenja: recimo slabega ocenjevanja ali neusklajenih postopkov zbiranja merskih podatkov. V tem primeru ukrep pri izvedbi procesa ni potreben, treba pa je urediti merski proces.

Naš pristop nalaga odgovornost za ocenjevanje vpliva predvsem vodji procesa in merjenja. On ima popoln pregled nad merskimi podatki in v kolikor odkrije težavo, mora presoditi njen pomen. O tem poroča na rednih tedenskih sestankih, kjer se člani projektne skupine odločajo

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

o nadaljnjih akcijah. Ker tudi drugi člani skupine vidijo napovedi, ga lahko na podlagi teh ali drugih ugotovitev opozarjajo na možne napake.

4.3.3.3 Napovedovanje rezultatov

Ugotovljen vpliv težav najlažje prikažemo tako, da napovemo, kako bi se težave odražale na končnih rezultatih. Pristop PSM predvideva predvsem enostavne tehnike napovedovanja, kot so projekcija trenutnega trenda v prihodnost in podobno.

Pristop PSM predvideva napovedovanje samo v primeru odkritih težav, medtem ko naš pristop s stalnim napovedovanjem to vrsto analize omogoča že vnaprej. Čeprav naše napovedi spremljajo samo nekaj glavnih atributov merskih entitet, so te s pomočjo dejanskih in zgodovinskih podatkov določene na tak način, da zagotavljajo natančne napovedi. Napovedovanje bo podrobno predstavljeno v razdelku 4.3.4.

4.3.3.4 Ovrednotenje alternativ

Zadnji korak vsake analize performans je predlagati več rešitev odkritih težav. Pri tem upoštevamo vse zbrane informacije, po potrebi pa se tudi posvetujemo z drugimi razvijalci. Predlagane rešitve morajo upoštevati čim širši vidik merjenja.

Naš pristop določa, da prve predloge ukrepov pripravi vodja procesa in merjenja, o njih pa se razpravlja na sestankih projektne skupine. V svojih predlogih vodja procesa in merjenja opiše rešitve predvsem s tehničnega stališča, naloga projektnega vodje in drugih članov pa je prispevati celovit pogled. Tak način zagotavlja učinkovito izvedbo sprejetih odločitev, ker je dogovor na sestanku projektne skupine zaveza za vse člane.

4.3.4 Napovedovanje

Naš pristop poleg treh vrst analiz uvaja še napovedovanje. Smisel napovedovanja je predvsem poenostavitev izvajanja analiz merskih podatkov, saj se ta analiza izvaja avtomatsko in neprenehoma. Vodji procesa in merjenja omogoča, da ima brez truda pregled nad najpomembnejšimi značilnostmi projekta.

Napovedovanje je križanec med ocenjevanjem in analizo performans, ki dodaja našemu pristopu dodatno funkcionalnost. Glede tehnične izvedbe je napovedovanje zelo podobno ocenjevanju, vendar je po namenu bližje analizi performans. Razmerje med ocenjevanjem, analizo performans in napovedovanjem je prikazano na sliki 4.35.



Slika 4.35: Razmerje med ocenjevanjem, analizo performans in napovedovanjem

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Razlike in podobnosti med ocenjevanjem in napovedovanjem so naslednje:

- Ocenjevanje se izvaja samo v fazah planiranja, medtem ko se napovedovanje izvaja neprenehoma od začetka projekta naprej.
- Ocene pa se zapišejo v različne plane in po zaključku projekta postanejo del zgodovinskih podatkov. Trenutne napovedi projekta so zapisane na določenem mestu v repozitoriju. Ko se stanje projekta spremeni, se spremenijo tudi napovedi v repozitoriju. Pri tem se prejšnje vrednosti pozabijo. Po zaključku projekta napovedi niso več smiselne in ne obstajajo več.
- Čeprav sta tako ocenjevanje kot napovedovanje avtomatizirani, je ocenjevanje zahtevnejše za izvedbo, ker upošteva tudi vidike posameznih razvijalcev. Napovedi se natančno izvajajo samo na globalni ravni.
- Po načinu izvedbe in značilnostih, ki jih naslavlja, je napovedovanje zelo podobno globalnemu delu ocenjevanja, le da pokriva več značilnosti.

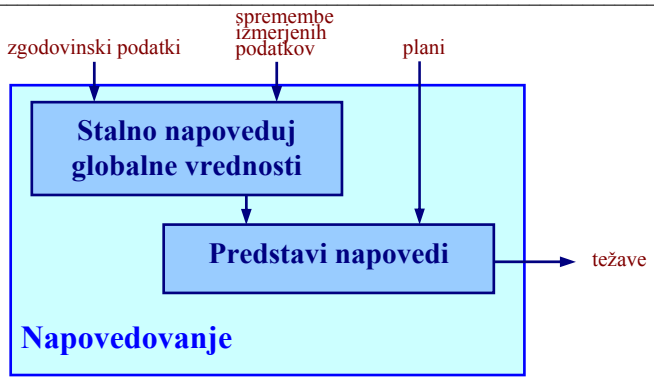
V primerjavi z analizo performans lahko napovedovanje opredelimo na naslednji način:

- Tako napovedovanje kot analiza performans sta namenjena ugotavljanju trenutnega stanja projekta glede na planirane cilje.
- Razlikujeta se po pristopu: analiza performans prikazuje stanje z razmerjem med izmerjenimi in planiranimi vrednostmi, medtem ko napovedovanje na podlagi zgodovinskih in trenutnih vrednosti napoveduje končni izdelek. Ustrezna projekcija rezultata v prihodnost poudari odstopanja in omogoča lažje odkrivanje težav.
- Analiza performans obsega več pokazateljev in je lahko dokaj natančna. Po potrebi jo lahko izvedemo še z natančnejšimi podatki. Napovedovanje obsega stalen nabor globalnih značilnosti. Če te kažejo na težave, izvedemo analizo performans z določenega vidika.
- Ne glede na to napovedi ustrezajo zahtevam merjenja na 4. stopnji modela CMM, saj jih določamo na podlagi statističnih pokazateljev nad zgodovinskimi rezultati.
- Analize performans se izvajajo na zahtevo, običajno pred tedenskimi sestanki projektne skupine ali ob zaključku ciklov. Napovedi so avtomatizirane in se spreminjajo dinamično ob spremembah merskih podatkov.
- Za analize performans je zadolžen vodja procesa in merjenja, ki jih izvaja, ugotavlja težave in o njih poroča (čeprav so rezultati analiz dostopni v repozitoriju). Napovedi so dostopne in namenjene vsem, predvsem pa vodji procesa in merjenja ter vodji projekta.
- Analize performans izhajajo iz izmerjenih in planskih podatkov, medtem ko napovedi temeljijo predvsem na izmerjenih in zgodovinskih podatkih, planske podatke pa uporabljajo posredno. Napovedovanje ne bi bilo mogoče brez skupnega repozitorija, kjer poleg trenutnih in zgodovinskih merskih podatkov obstajajo še meta podatki o projektu.

Slika 4.36 prikazuje oba koraka napovedovanja. Napovedovanje izbranih globalnih vrednosti se izvaja neprenehoma. Ko so vrednosti določene, se jih na nazoren način predstavi. To s stališča procesa razvoja PO ne zahteva posebnih aktivnosti. Vodja procesa in merjenja napovedi sproti preverja v okviru aktivnosti za spremljanje merjenja ter še posebej ob izvedbi analiz ob koncu ciklov in projekta.

Poudariti velja, da smo zaradi učinkovitosti prisiljeni k poenostavljanju postopka napovedovanja, čeprav v splošnem težimo k najboljšim napovedim. Ocenjevanje se namreč izvaja nekajkrat v projektu in takrat si lahko privoščimo zahtevnejše postopke. Napovedovanje se dinamično spreminja ob vsakem dodatnem merskem podatku, zato po zahtevnih postopkih določimo le nekaj glavnih napovedi za celoten projekt. Te potem s pomočjo ustreznih tabel v razmerju prikažemo (projeciramo) tudi za posamezne komponente.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

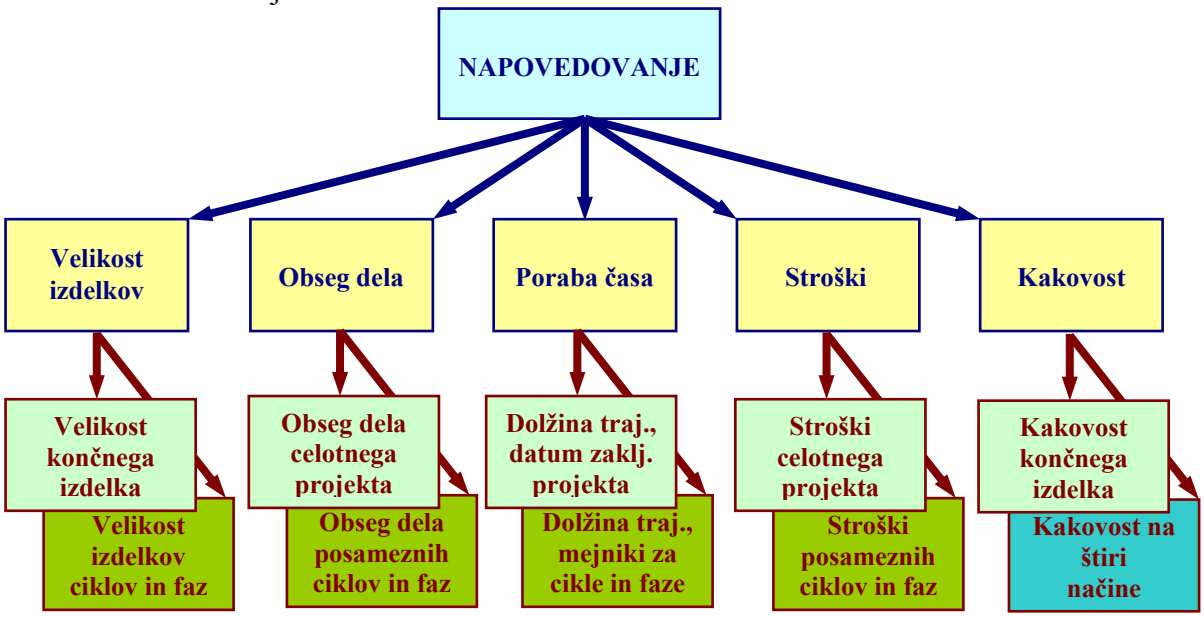


Slika 4.36: Koraki analize napovedovanja

4.3.4.1 Stalno napovedovanje globalnih vrednosti

Ta korak predstavlja jedro napovedovanja in je popolnoma avtomatiziran. Naš pristop napoveduje naslednje vrednosti (glej sliko 4.37):

- velikost končnega izdelka;
- velikost ciklov in faz (projekcija);
- obseg dela celotnega projekta;
- obseg dela posameznih ciklov in faz (projekcija);
- dolžino trajanja in datum zaključka projekta;
- dolžino trajanja in mejniki posameznih ciklov in faz;
- stroške projekta;
- stroške posameznih faz in ciklov (projekcija);
- kakovost končnega izdelka;
- kakovost na štiri načine:
  - z vidika procesnega in podatkovnega dela,
  - z vidika posameznih faz in ciklov,
  - z vidika posameznih rezultatov in
  - z vidika izvajalcev.



Slika 4.37: Napovedovanje po našem merskem pristopu

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Vrednosti napovedujemo od tedaj, ko je na podlagi znanih podatkov to mogoče, pa do trenutka, ko je napoved smiselna. V primeru izdelkov je smiselno napovedovati od začetka drugega cikla do zaključka cikla testiranja, ko je končni rezultat izdelan. Aktivnosti in vire je smiselno napovedovati ves čas izvedbe projekta.

Velikost končnega izdelka

Izhodišče za vse napovedi je najboljša napoved velikosti, kot jo v danem trenutku zmoremo. Na podlagi te določimo večino ostalih napovedanih vrednosti na podoben način kot pri globalnem ocenjevanju.

Napoved celotne velikosti projekta določimo na popolnoma enak način kot pri ocenjevanju. Na podlagi že zbranih merskih podatkov in normalizacijske tabele (glej razdelek 4.3.1.3) presodimo, kateri podatki so najbolj povezani s končno velikostjo. Na podlagi treh najbolj primernih določimo napoved končne velikosti za oba dela: procesni in podatkovni.

Natančna presoja območja zaupanja za napovedi žal ni mogoča, ker bi zanjo potrebovali ustrezne zgodovinske podatke. Zbirati bi morali vse napovedi o preteklih projektih, česar zaradi prevelike količine podatkov ne počnemo. Težava je predvsem v tem, da bi morali uporabiti zgolj podatke tistih napovedi, ki temeljijo na enakih treh vrstah izmerjenih podatkov (posebej za procesni in podatkovni del). Možnih kombinacij najprimernejših izdelkov je enostavno preveč.

Zato približno ocenimo območje zaupanja za oba dela. V ta namen uporabimo najprimernejše zgodovinske podatke o ocenah velikosti in dejanskih končnih velikostih. Glede na trenutno stanje projekta lahko presodimo, kateri nabor ocen je najprimernejši za nas (recimo če smo trenutno v fazi 4.3, lahko uporabimo ocene iz četrtega cikla). Tako določeno območje zaupanja je (verjetno) nekoliko širše. Ožje ne more biti, ker je določeno na podlagi zgodovinskih ocen, ki so slabše ali kvečjemu enake trenutni napovedi.

Velikosti izdelkov ciklov

Napoved velikosti končnega izdelka omogoča, da proporcionalno napovemo tudi velikosti glavnih izdelkov faz. V ta namen bi lahko izvedli linearno regresijo za vsako komponento teh izdelkov, vendar je izdelkov in atributov veliko in bi to zahtevalo preveč dela.

Enostavneje je uporabiti normalizacijsko tabelo v obratni smeri. V tem primeru lahko na podlagi napovedane končne velikosti presodimo, kolikšna bi bila vrednost določenega atributa, če bi veljalo isto razmerje. Ker imamo dve napovedi velikosti (na podlagi procesnega in podatkovnega vidika), vedno uporabimo primernejšo – za nekatere atribute bo primerneje izhajati iz končnega števila vrstic, za druge iz končnega števila elementov PB. Primernost lahko razberemo iz razmerja med koeficientoma B_i in D_i. Kjer je večji prvi, uporabimo velikost procesnega dela, sicer upoštevamo velikost podatkovnega dela. Ker se normalizacijska tabela sproti prilagaja celotni količini zgodovinskih podatkov in uporabljamo dobro oceno končne velikosti, lahko pričakujemo, da projekcija ustrezno napoveduje vrednosti. Pri vseh vrednostih projeciramo tudi obe meji območja zaupanja.

Na prvi pogled imamo še eno možnost, da na enostaven način popravimo ocene posameznih atributov. Napovedovanje velikosti je smiselno za tiste izdelke, ki še niso bili dokončani. Pri dokončanih je ocena brezpredmetna. Vendar lahko iz razlike med izmerjeno vrednostjo in napovedjo po naši metodi presodimo, kako se napovedi skladajo z dejanskim stanjem. Ker je takih atributov več, se postavlja vprašanje, na kakšen način upoštevati razliko.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Žal se pokaže, da na tak način ne moremo nič pridobiti oziroma to zahteva dodatne informacije. Če upoštevamo tri najprimernejše atribute (posebej za procesni in podatkovni del) v ustreznem razmerju normalizacijske tabele (enako kot pri ocenjevanju v razdelku 4.3.1.3), bomo dobili enake napovedi, saj smo na tem utemeljili našo napoved. Če bi želeli več, bi morali poznati odvisnosti med posameznimi atributi različnih izdelkov, kar je preveč podrobno. Zavedati se moramo, da veliko pripomoremo k točnosti rezultata že s tem, ko upoštevamo ločeno oba vidika velikosti končnega rezultata.

Naš pristop določa, da je nabor izdelkov (in atributov) fiksno določen, nanašajo pa se na rezultate od drugega do šestega cikla. Prej ni smiselno napovedovati, kasneje pa z vidika izdelkov nimamo kaj. V naboru so glavni izdelki posameznih faz in ciklov iz procesnega in podatkovnega dela, ki opisujejo končni rezultat, ter končni rezultati. Vsi so seveda navedeni v normalizacijski tabeli, nabor pa seveda ne vsebuje celotne normalizacijske tabele. V vsakem ciklu je to vsaj en izdelek procesnega dela in en izdelek podatkovnega dela, vsak pa je opredeljen z vsaj enim glavnim atributom. Nabor izdelkov je natančneje predstavljen v dodatku A.3.3.

Obseg dela celotnega projekta

Tudi napoved obsega dela za izvedbo celotnega projekta določimo na enak način kot pri ocenjevanju. Najprej s pomočjo podatkov o razmerju med obsegom dela in končno velikostjo določimo linearno regresijsko enačbo. Uporabimo oceno celotne velikosti in izračunamo napoved celotnega obsega dela.

Nato uporabimo postopek, ki smo ga opisali pri ocenjevanju ob začetkih posameznih ciklov. Na podlagi že izmerjenega obsega dela za posamezne aktivnosti (procesnega in podatkovnega dela) in tabele tipičnih razmerij aktivnosti ugotovimo, v kolikšni meri se ocena sklada z dejanskimi izmerjenimi rezultati. Oceno napovedi nato ustrezno popravimo, pri čemer predvidimo, da se bo obseg dela bodočih aktivnosti spremenil v enakem razmerju kot se je obseg dela že izvedenih aktivnosti.

Z uporabo linearne regresije in podatkov o ocenah obsega dela in dejanskem skupnem obsegu dela lahko približno določimo območje, za katero z 90% stopnjo zaupanja predvidevamo, da zajema tudi končni skupni obseg dela projekta. Enako kot pri določanju območja za velikost tudi tu uporabimo najprimernejše ocene, saj se podatki o napovedih ne shranjujejo.

Obseg dela posameznih ciklov in faz

Napoved obsega dela za izvedbo posameznih ciklov in faz določimo na podlagi tabele tipičnih razmerij. Izračunani rezultat za končni obseg dela oziroma območje razdelimo na ustrezne dele, pri čemer upoštevamo samo faze, ki se zares izvajajo.

V določenem trenutku projektnega razvoja je znan globalni plan opravil in podrobni plan opravil naslednjega cikla. Enako je tudi z napovedjo: vedno je natančnejša za nekaj naslednjih faz, medtem za zadnje cikle lahko obstajajo samo napovedi obsega dela za celoten cikel.

Tudi pri napovedih obsega dela posameznih faz bi lahko bili natančnejši, če bi s pomočjo več regresijskih enačb upoštevali zgodovinska razmerja med obsegom dela posameznih faz. Tako bi recimo lahko odkrili, da nam veliko povečanje točno določene faze napoveduje težave v nadaljevanju, čeprav pristop z enostavnim izračunom tega ne pokaže, ker se pribitek obsega dela te faze skrije v povprečju vseh izvedenih faz. Vendar tak postopek zahteva veliko več dela in ga prepustimo analizi performans.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Dolžina trajanja in datum zaključka projekta

Pri napovedi časa predvidenega zaključka projekta izhajamo iz napovedi obsega dela za celoten projekt. Ker je število osebja projekta stalno, celoten obseg dela delimo s številom razvijalcev. Podobno kot v prejšnjih primerih s pomočjo znanih razmerij med obsegom dela(za enega razvijalca) in časom trajanja določimo regresijsko enačbo ter iz nje s pomočjo napovedi skupnega obsega dela določimo potreben čas v številu delovnih dni. Za napoved določimo nato še območje 90% zaupanja, kjer za izhodišče uporabimo zgodovinske podatke o (najprimernejših) ocenah porabe časa in dejanski porabi časa. Na koncu napovedi območje z uporabo koledarja pretvorimo v natančne datume.

Izhodiščna napoved obsega dela že upošteva dejansko izmerjene podatke in to se odraža na dolžini časa. Vendar je obseg dela absolutna vrednost, poraba časa pa upošteva koledar in to lahko povzroči dodaten zamik. Zato ugotovimo, kateri je bil zadnji dosežen mejnik. Z uporabo tabele tipičnih razmerij in napovedi porabe časa določimo, kdaj bi ta mejnik morali doseči po naši napovedi. Pri tem upoštevamo samo planirane faze in cikle. Razlika med dejansko in napovedano vrednostjo določa zamik končnega datuma projekta.

Dolžina trajanja in mejniki posameznih ciklov in faz

Napovedujemo lahko tudi doseganje mejnikov za posamezne faze in cikle. Pri tem uporabimo enak pristop kot pri obsegu dela: s pomočjo tabele tipičnih razmerij med aktivnostmi določimo čas trajanja posameznih faz in ciklov (v številu delovnih dni). Trajanje nato pretvorimo v napovedi datumov. Enako kot pri trajanju celotnega projekta tudi tu zamaknemo datume vseh bodočih aktivnosti za razliko med dejanskim in napovedanim datumom zadnje dokončane aktivnosti.

Stroški projekta ter stroški faz in ciklov

Naš pristop omogoča tudi napovedovanje stroškov projekta. Pri napovedovanju globalnih stroškov enako kot pri ocenjevanju izhajamo iz napovedi obsega dela (vsote za procesni in podatkovni del). Na podlagi zgodovinskih podatkov o skupnem obsegu dela in celotnih stroških projekta določimo regresijsko enačbo. Za enačbo lahko na enak način kot pri drugih ocenah določimo območje 90% zaupanja, pri čemer izhajamo iz razmerja med najprimernejšimi ocenami in dejanskimi stroški.

Stroške posameznih ciklov napovedujemo na podlagi zgodovinskih podatkov o razmerjih stroškov med posameznimi fazami ali cikli. V ta namen obstaja v repozitoriju več podatkov, ki se nanašajo na tipične vrednosti organizacije. Te so:

- tipično razmerje med stroški iz obsega dela in dodatnimi stroški,
- tabela tipičnih razmerij obsega dela za vse aktivnosti in
- tabela tipičnih razmerij dodatnih stroškov za vse aktivnosti.

Na podlagi teh podatkov ter dejanskih podatkov o izmerjenem obsegu dela in dodatnih stroških za vse izvedene faze najprej popravimo skupno oceno stroškov projekta, nato pa še določimo stroške posameznih faz in ciklov. Pri tem uporabimo enak postopek kot pri ocenjevanju (glej razdelek 4.3.1.3). Na enak način lahko projeciramo tudi območje zaupanja.

Kakovost končnega izdelka

Tudi kakovost podatkov napovedujemo z uporabo enakega pristopa kot pri globalnem ocenjevanju. Uporabimo štiri regresijske enačbe za naslednje atribute:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- skupno število vnešenih napak v procesni del končnega izdelka,
- skupno število odpravljenih napak v procesni del končnega izdelka,
- skupno število vnešenih napak v podatkovni del končnega izdelka in
- skupno število odpravljenih napak v podatkovni del končnega izdelka.

Vse štiri so določene na podlagi zgodovinskih podatkov o razmerjih med števili napak posameznega dela in velikostjo (procesnega ali podatkovnega) dela. Aktualna ocena velikosti procesnega ali podatkovnega dela je tudi izhodišče za napoved vsake od štirih vrednosti. Ker obstajajo med zgodovinskimi podatki tudi različne vrste ocen in dejanske izmerjene vrednosti, lahko za vsako določimo tudi območje 90% zaupanja. Pri tem izberemo najprimernejšo oceno na podoben način kot drugod.

Kakovost na štiri načine

Napoved kakovosti lahko izvedemo tudi z upoštevanjem razmerij med zadnjimi ocenami napak in dejanskim številom odkritih in odpravljenih napak. Tak način ni neposredno povezan z zgodovinskimi podatki, vendar upošteva izmerjene rezultate o kakovosti. Zato ga ponudimo kot alternativo že predstavljenemu prikazu. Napovedi predstavimo na štiri načine:

- po procesnem ali podatkovnem vidiku,
- po posameznih fazah ali ciklih,
- po posameznih rezultatih in
- po posameznih izvajalcih.

Za izhodišče vzamemo ocene, ki so bile določene na enake štiri načine. Bolje bi bilo, če bi namesto ocen uporabili podatke, ki bi jih na ustrezen način izpeljali iz globalnih napovedi, vendar bi za to potrebovali določanje kakovosti po posameznih aktivnostih od spodaj navzgor. Tak pristop bi bil zahteven in v nasprotju z našo težnjo po enostavnih in učinkovitih rešitvah.

Za vsakega od štirih načinov predstavitve ocen kakovosti velja, da so vsote odkritih (odpravljenih) napak vseh razredov enake končni vsoti odkritih (odpravljenih) napak. Ker so bile te ocene določene na podlagi najprimernejših zgodovinskih podatkov, lahko iz njih določimo verodostojna razmerja med posameznimi razredi.

Najprej moramo ugotoviti, kolikšen odmik od ocenjenih vrednosti predstavljajo izmerjeni podatki. Z uporabo meta podatkov repozitorija določimo, kateri izmed zbranih podatkov sodi v katero fazo, oziroma za kolikšen del razvoja so podatki že zbrani. Podatke seštejemo po dveh kategorijah:

- odkrite napake procesnega in podatkovnega dela že dokončanih aktivnosti in
- odpravljene napake procesnega in podatkovnega dela že dokončanih aktivnosti.

Iz zadnjih ocen kakovosti po fazah lahko razberemo, kolikšno naj bi bilo skupno število odkritih in odpravljenih napak v dokončanih fazah. Nato določimo dve razmerji:

- razmerje med skupno oceno števila odkritih napak v dokončanih fazah in vsoto dejansko odkritih napak ter
- razmerje med skupno oceno števila odpravljenih napak v dokončanih fazah in vsoto dejansko odpravljenih napak.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Ti razmerji sta temelj za popravljanje zadnjih ocen oziroma določanje napovedi v okviru vseh štirih načinov prikaza. Pri tem ravnamo na naslednji način:

a) Prikaz po podatkovnem in procesnem vidiku:
Vse štiri zadnje ocene kakovosti (odkrite in odpravljene napake po procesnem in podatkovnem vidiku) spremenimo v skladu s prvim ali drugim razmerjem. Razmerja med procesnimi in podatkovnimi napakami ne moremo uporabljati, ker razmerje procesnega in podatkovnega dela ni v vseh fazah enako. Te štiri vrednosti napovedujejo enake količine kot globalni prikaz kakovosti, a so določene na drugi način.

b) Prikaz po posameznih fazah in ciklih:
Ocene odkritih in odpravljenih napak prihodnjih faz in ciklov spremenimo v skladu s prvim ali drugim razmerjem.

c) Prikaz po posameznih rezultatih:
Napake rezultatov (izdelkov) ciklov in faz so povezane z napakami faz, le način prikaza je drugačen. Zato tudi v tem primeru ocene kakovosti posameznih izdelkov spremenimo v napovedi tako, da upoštevamo ustrezno razmerje med ocenami in izmerjenimi napakami.

d) Prikaz po razvijalcih:
Za vsakega razvijalca posebej seštejemo število izmerjenih napak (odkritih in odpravljenih). Nato določimo dve razmerji:

- razmerje med skupno oceno števila odkritih napak v dokončanih fazah in celotnim številom odkritih napak projekta ter
- razmerje med skupno oceno števila odpravljenih napak v dokončanih fazah in celotnim številom odpravljenih napak projekta.

S pomočjo teh lahko približno ugotovimo, kolikšen del ocene skupnega števila (odkritih ali odpravljenih) napak posameznika pripada že izvedenim fazam. Razmerje med tem delom ocene (odkritih ali odpravljenih) napak posameznika in izmerjenimi rezultati omogoča napoved, za koliko se bo njegovo skupno število napak spremenilo. Pri tem predvidimo enakomerno obremenitev članov v vseh fazah razvoja, kar v splošnem ni čisto res.

Za vse štiri načine prikaza velja, da so določeni na drugačen način kot globalna ocena kakovosti. Ta je zaradi neposredne uporabe zbranih podatkov o napakah sicer boljši, vendar zaradi zahteve po učinkovitosti uporablja enostavnejše pristope. Zato je treba vse rezultate kakovosti jemati kot celoto in se zavedati, na kakšen način so določeni.

4.3.4.2 Predstavitev napovedanih vrednosti

Pomembna prednost napovedovanja je sproten pregled glavnih značilnosti poteka projekta. K temu sodi tudi nazoren prikaz vseh napovedanih vrednosti. Naš pristop se pri tem naslanja na avtomatizacijo repozitorija oziroma na IS repozitorija.

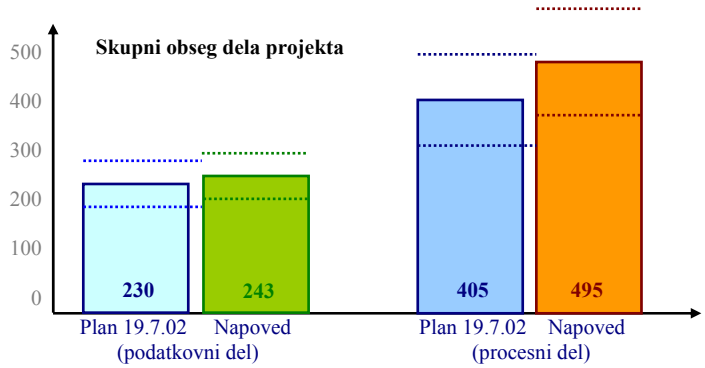
Aktualne napovedi so shranjene v posebnem delu repozitorija. Ob spremembah merskih podatkov se na novo določijo, pri čemer se stare vrednosti pozabijo.

Napovedi obstajajo na dveh ravneh:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- **Globalne napovedi** se nanašajo na celoten projekt in so primerljive z aktualnimi planiranimi vrednostmi. Običajno predstavljajo dva vidika: procesni in podatkovni. Tako zanje kot za ocene je določeno tudi območje 90% zaupanja.

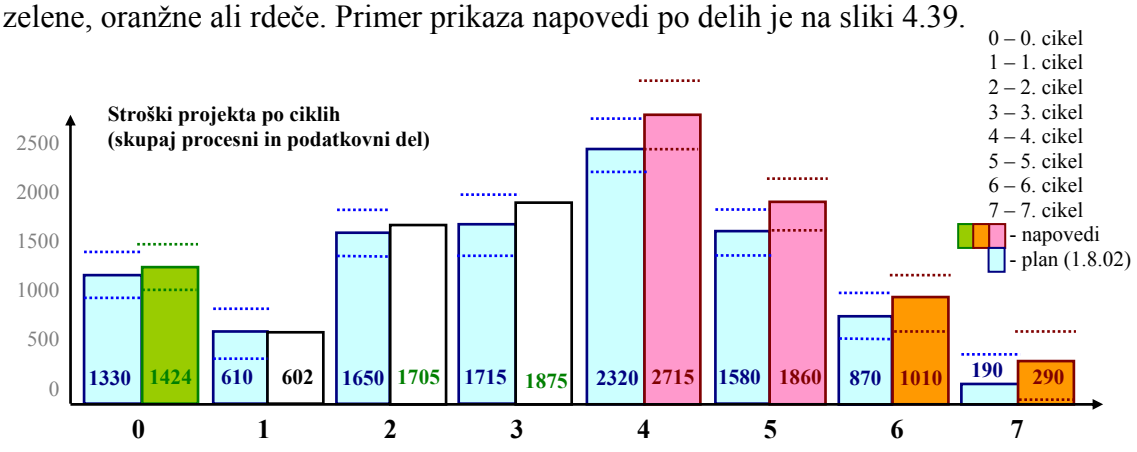
  Pri prikazu skušamo biti čim bolj nazorni, zato uporabljamo grafični prikaz s stolpci in številčni prikaz. Poleg napovedi v grafe vključimo tudi aktualne ocene iz ustreznih planov, ki jih prikazujemo ob napovedih, da je primerjava lažja. Na grafikonih so označena območja zaupanja ocen in napovedi. Ključne informacije poudarimo tudi z uporabo barv, predvsem osnovnih treh barv semaforja: zelene, oranžne in rdeče. Prva prikazuje pričakovane vrednosti napovedi glede na planirane vrednosti, oranžna pa vrednosti, ki odstopajo iz srednje tretjine območja zaupanja planiranih vrednosti. Rdeča barva prikazuje vrednosti napovedi, ki odstopajo tudi iz mej območja zaupanja ustreznih ocen. Primer tovrstnega grafa je na sliki 4.38.



Slika 4.38: Primer globalne napovedi po našem merskem pristopu

- **Napovedi za dele procesa** se nanašajo na faze ali cikle. Obsegajo več razredov, zato jih moramo predstaviti v več delih ali na več ravneh, recimo vsi cikli projekta ali vse faze izbranega cikla. Tudi zanje poznamo ocene in običajno tudi območja zaupanja. Če spremljamo napovedi med razvojem, imamo običajno za nekaj razredov že zbrane dejanske podatke. Tedaj namesto napovedi ob ocenah predstavimo izmerjene podatke. Napovedi predstavimo samo za tisti del, ki še ni znan.

  Tudi tu uporabimo enak pristop kot pri globalnih napovedih, le da je na grafu več informacij. Zato so stolpci ožji, uporabiti pa moramo tudi legendo. Med posameznimi deli prikazov predvidimo tudi povezave (recimo iz prikaza vseh ciklov projekta na prikaz izbranega cikla). Izmerjene podatke predstavimo s črno barvo, medtem ko so napovedi zelene, oranžne ali rdeče. Primer prikaza napovedi po delih je na sliki 4.39.



Slika 4.39: Primer napovedi za dele procesa po našem merskem pristopu

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

V nadaljevanju so našteti podatki, ki jih prikažemo pri posameznih napovedih:

a) Velikost končnega izdelka

- zadnja ocena velikosti procesnega dela končnega izdelka z območjem zaupanja;
- napoved velikosti procesnega dela končnega izdelka z območjem zaupanja;
- zadnja ocena podatkovnega dela končnega izdelka z območjem zaupanja;
- napoved velikosti podatkovnega dela končnega izdelka z območjem zaupanja.

b) Velikost ciklov in faz (projekcija)

Za vse izbrane atribute glavnih izdelkov posameznih ciklov:
- napoved velikosti atributa izdelka z območjem zaupanja.

c) Obseg dela celotnega projekta

- zadnja ocena obsega dela za izvedbo procesnega dela z območjem zaupanja;
- napoved obsega dela za izvedbo procesnega dela z območjem zaupanja;
- zadnja ocena obsega dela za izvedbo podatkovnega dela z območjem zaupanja;
- napoved obsega dela za izvedbo podatkovnega dela z območjem zaupanja.

d) Obseg dela posameznih ciklov in faz (projekcija)

Za vsak cikel:
- zadnja ocena obsega dela za izvedbo procesnega dela cikla z območjem zaupanja;
- napoved obsega dela za izvedbo procesnega dela cikla z območjem zaupanja;
- zadnja ocena obsega dela za izvedbo podatkovnega dela cikla z območjem zaupanja;
- napoved obsega dela za izvedbo podatkovnega dela cikla z območjem zaupanja.

Za vsako fazo, kjer je to mogoče:
- zadnja ocena obsega dela za izvedbo procesnega dela faze z območjem zaupanja;
- napoved obsega dela za izvedbo procesnega dela faze z območjem zaupanja;
- zadnja ocena obsega dela za izvedbo podatkovnega dela faze z območjem zaupanja;
- napoved obsega dela za izvedbo podatkovnega dela faze z območjem zaupanja.

e) Dolžina trajanja in datum zaključka projekta

- zadnja ocena dolžine trajanja projekta z območjem zaupanja;
- napoved dolžine trajanja projekta z območjem zaupanja;
- zadnja ocena datuma zaključka projekta z območjem zaupanja;
- napoved datuma zaključka projekta z območjem zaupanja.

f) Dolžina trajanja in mejniki posameznih ciklov in faz

Za vsak cikel:
- zadnja ocena dolžine trajanja z območjem zaupanja;
- napoved dolžine trajanja z območjem zaupanja.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Za vsako fazo, kjer je to mogoče:
- zadnja ocena dolžine trajanja z območjem zaupanja;
- napoved dolžine trajanja z območjem zaupanja.

Za vsak znan mejnik (ujema se z zaključki faz):
- zadnja ocena mejnika z območjem zaupanja;
- napoved mejnika z območjem zaupanja.

g) Stroški projekta

- zadnja ocena celotnih stroškov projekta z območjem zaupanja;
- napoved celotnih stroškov projekta z območjem zaupanja.

h) Stroški posameznih faz in ciklov (projekcija)

Za vsak cikel:
- zadnja ocena stroškov cikla z območjem zaupanja;
- napoved stroškov cikla z območjem zaupanja.

Za vsako fazo, kjer je to mogoče:
- zadnja ocena stroškov faze z območjem zaupanja;
- napoved stroškov faze z območjem zaupanja.

i) Kakovost končnega izdelka

- zadnja ocena skupnega števila vnešenih napak pri izvedbi procesnega dela projekta z območjem zaupanja;
- napoved skupnega števila vnešenih napak pri izvedbi procesnega dela projekta z območjem zaupanja;
- zadnja ocena skupnega števila odpravljenih napak pri izvedbi procesnega dela projekta z območjem zaupanja;
- napoved skupnega števila odpravljenih napak pri izvedbi procesnega dela projekta z območjem zaupanja;
- zadnja ocena skupnega števila vnešenih napak pri izvedbi podatkovnega dela projekta z območjem zaupanja;
- napoved skupnega števila vnešenih napak pri izvedbi podatkovnega dela projekta z območjem zaupanja;
- zadnja ocena skupnega števila odpravljenih napak pri izvedbi podatkovnega dela projekta z območjem zaupanja;
- napoved skupnega števila odpravljenih napak pri izvedbi podatkovnega dela projekta z območjem zaupanja.

j) Kakovost z vidika procesnega in podatkovnega dela

Izvedba je podobna kot pri kakovosti končnega izdelka, le napovedujemo na drugi podlagi:
- zadnja ocena skupnega števila vnešenih napak pri izvedbi procesnega dela projekta;
- napoved skupnega števila vnešenih napak pri izvedbi procesnega dela projekta;
- zadnja ocena skupnega števila odpravljenih napak pri izvedbi procesnega dela projekta;
- napoved skupnega števila odpravljenih napak pri izvedbi procesnega dela projekta;
- zadnja ocena skupnega števila vnešenih napak pri izvedbi podatkovnega dela projekta;
- napoved skupnega števila vnešenih napak pri izvedbi podatkovnega dela projekta;

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- zadnja ocena skupnega števila odpravljenih napak pri izvedbi podatkovnega dela projekta;
- napoved skupnega števila odpravljenih napak pri izvedbi podatkovnega dela projekta.

k) Kakovost z vidika posameznih faz in ciklov

Za vsakega izmed ciklov:
- zadnja ocena števila vnešenih napak cikla;
- napoved števila vnešenih napak cikla upanja;
- zadnja ocena števila odpravljenih napak cikla;
- napoved števila odpravljenih napak cikla.

Za vsako fazo, kjer je to mogoče:
- zadnja ocena števila vnešenih napak faze;
- napoved števila vnešenih napak faze;
- zadnja ocena števila odpravljenih napak faze;
- napoved števila odpravljenih napak faze.

l) Kakovost z vidika posameznih rezultatov

Za vsak rezultat (izdelek) faze ali cikla, kjer je to mogoče:
- zadnja ocena števila vnešenih napak pri izdelavi izdelka;
- napoved števila vnešenih napak pri izdelavi izdelka;
- zadnja ocena števila odpravljenih napak pri izdelavi izdelka;
- napoved števila odpravljenih napak pri izdelavi izdelka.

m) Kakovost z vidika izvajalcev

Za vsakega razvijalca oziroma člana projektne skupine:
- zadnja ocena skupnega števila vnešenih napak;
- napoved skupnega števila vnešenih napak;
- zadnja ocena skupnega števila odpravljenih napak;
- napoved skupnega števila odpravljenih napak.

4.4 Zaključek poglavja

V tretjem poglavju smo definirali prilagodljiv proces razvoja PO, ki ustreza zahtevam majhnih organizacij za razvoj IS. S tem smo definirali okvir za določitev merjenja.

Pregled raziskovalnega področja je pokazal, da se je na podlagi množice raziskav in izkušenj na področju merjenja izoblikoval enoten merski pristop, ki ga opisuje standard ISO/IEC 15939. Standard povzema tudi model CMMI, pristop PSM pa predstavlja primer njegove uspešne izvedbe. S tega stališča je primerno, da pristop PSM izberemo za vodilo pri določanju merjenja tudi v našem primeru.

Pristop PSM obsega dva dela: merski procesni model opisuje način izvajanja merjenja, medtem ko merski informacijski model opredeljuje vsebino merjenja. V četrtem poglavju smo na podlagi merskega procesnega modela opisali postopke merjenja v okviru našega merskega pristopa.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pri definiranju našega merskega procesa smo želeli natančno opredeliti merski proces. Zato smo uporabili merski procesni model pristopa PSM kot navodilo za to, katere aktivnosti mora naš merski pristop obsegati. S tega stališča naš merski proces formalno pokriva enake aktivnosti, vendar so izvedene drugače, na za nas primernejši način.

Poleg tega smo v izvedbo uvedli izvirne rešitve in pristope, s katerimi smo aktivnosti merskega procesa prilagodili našim potrebam in jih umestili v prilagodljivi proces razvoja PO. Nekatere izboljšave omogočajo tudi enostavnejšo in učinkovitejšo izvedbo merjenja.

To dejansko pomeni, da razvijamo naš merski proces v skladu s principom, ki smo ga predstavili ob zaključku tretjega poglavja. Če želimo v majhni organizaciji ustrezno izvajati merjenje, ga moramo tudi v tem primeru izvajati dosledno in ga ne smemo preveč poenostaviti. V vsakem primeru moramo poskrbeti za jasno definicijo merjenja, učinkovito zbiranje merskih podatkov, ustrezno analizo podatkov ter primerno ukrepanje. Kar lahko storimo je to, da:

- poenostavimo (ali opustimo) manj pomembne dele, kolikor je to mogoče,
- se osredotočimo na pomembne dele,
- te dele jasno, natančno in popolno definiramo,
- poskrbimo za učinkovito izvedbo teh delov, recimo z inovacijami in
- predvidimo avtomatizacijo za zahtevne dele postopkov.

S tem dosežemo, da je naša rešitev uporabna in učinkovita. V četrtem poglavju smo predstavili merski proces, ki te značilnosti dosega s pomočjo naslednjih pristopov:

Merjenje je (za razliko od pristopa PSM) določeno na organizacijski in projektni ravni:

- Merjenje je uvedeno kot poseben podporni proces.
- Definirano je na ravni organizacije in vsi projekti ga izvajajo na enak način.
- Osrednji del vsebine merjenja je vedno enak (stalna merska izhodišča), vendar projekti lahko merijo tudi na podlagi svojih informacijskih potreb (posebna merska izhodišča).
- Pomemben del podpornih aktivnosti merskega procesa se izvaja na ravni organizacije.

Uvedene so nekatere poenostavitve merskega procesnega modela pristopa PSM:

- Trinivojska struktura določanja meritev iz pristopa PSM (sedem splošnih merskih izhodišč, več razredov in veliko meritev) je poenostavljena v dvonivojsko (štirje razredi in več meritev).
- Merjenje funkcionalnosti je nadomeščeno z učinkovitejšim merjenjem velikosti.
- Ker se na ravni projekta določa merjenje samo za posebne potrebe, je planiranje in določanje meritev poenostavljeno.
- Postopek določanja meritev preko pokazateljev je učinkovitejši in daje koristne stranske rezultate.
- Analize so prilagojene potrebam majhnih organizacij.

Z nekaterimi izvirnimi rešitvami določimo zbiranje merskih podatkov na način, ki sovpada s prilagodljivim procesom razvoja PO:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Merske podatke zbirajo posamezniki ob izvedbi vsake aktivnosti.
- Posamezniki zbirajo stalen nabor merskih podatkov za vsako aktivnost.
- Zbiranje merskih podatkov je avtomatizirano.
- Podatki se zbirajo ob aktivnostih posameznika, podatki za večje entitete pa se nato določajo s pomočjo združevalnih struktur v repozitoriju.

Poseben poudarek je namenjen učinkovitemu izvajanju analiz:

- Možnosti analiz ne omejujemo, vendar poskrbimo za učinkovitost njihove izvedbe ter za avtomatizacijo.
- V analizah ločujemo podatkovni in procesni vidik aktivnosti in rezultatov, kar omogoča natančnejše merjenje razvoja IS.
- Postopki analize so sicer poenostavljeni in prilagojeni, vendar zelo natančno določeni in temeljijo na učinkovitih statističnih metodah.
- Če je le mogoče pri analizah (ocenjevanju) uporabimo dva pristopa: od spodaj navzgor (recimo oceno sestavimo iz delnih ocen) in od zgoraj navzdol (oceno izpeljemo na podlagi zgodovinskih podatkov).
- Dodatno uvedemo dve združevalni strukturi, kar omogoča boljšo izvedbo analiz.
- Dodatno uvedemo novo izvirno vrsto analize (napovedovanje), ki zelo poveča učinkovitost izvajanja analiz.

Avtomatiziran merski repozitorij podpira vse aktivnosti merjenja:

- Merski podatki projektov se zbirajo v centralnem skupnem repozitoriju organizacije.
- Aktivnosti repozitorija (zbiranje, pregledi in analize podatkov) so avtomatizirane.
- Merski repozitorij obsega različne vrste podatkov, ki omogočajo izvedbo različnih analiz.
- Repozitorij je mesto, kjer si uporabniki (v skladu s pristojnostmi) lahko ogledajo merske podatke in rezultate analiz.
- V repozitoriju obstajajo posebni združeni podatki preteklih projektov, ki omogočajo učinkovitejše izvajanje analiz.

Predstavljeni merski procesni model tako uvaja številne zanimive ideje, ki učinkovito nadgrajujejo prilagodljivi proces razvoja PO. Vendar merjenje ni jasno opredeljeno, dokler ne predstavimo še vsebinskega dela merskega pristopa. Tedaj se bo merski pristop zlil v celoto.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5. VSEBINSKI DEL MERSKEGA PRISTOPA: MERSKI INFORMACIJSKI MODEL

V prejšnjem poglavju smo opisali merski procesni model, ki predstavlja prvi del našega merskega pristopa. Procesni model predpisuje način izvajanja merjenja v majhni organizaciji za razvoj IS, ki zagotavlja učinkovito planiranje, izvajanje in nenehno izboljševanje merjenja. V tem poglavju je predstavljen drugi del merskega pristopa. Ta opisuje vsebino merjenja, torej odgovarja na vprašanje KAJ meriti.

5.1 Opis vsebine merjenja

V okviru tega poglavja je vsebina merjenja opisana na štirih ravneh:

a) **Merski informacijski model** določa, na kakšen način opišemo vsebino merjenja.
b) **Meta model** je glavni del opisa vsebine merjenja. Predstavlja glavne značilnosti našega pristopa: več ravni opisa merjenj, opis merskih podatkov in razmerij med njimi.
c) **Nabori stalnih merskih izhodišč, pokazateljev in meritev** predstavljajo vsebino ključnih delov meta modela. Za ustrezno določitev teh moramo identificirati konkretne merske cilje in uporabiti izviren postopek izpeljave.
d) **Merski repozitorij** je fizična implementacija meta modela.

a) Naš pristop za opis vsebine merjenja temelji na uporabi **merskega informacijskega modela**. Ta določa, katere dele vsebuje primeren opis vsebine merjenja in na kakšen način se ti dopolnjujejo v celoten merski plan. Pri tem smo merski informacijski model pristopa PSM nekoliko prilagodili, da bi lahko bolje opisali naš merski pristop.

b) V disertaciji želimo konkretno določiti osrednji del merskega informacijskega modela. Tega predstavimo z **meta modelom,** ki nam predstavlja opis vseh podatkov, ki predstavljajo merjenje na različnih ravneh. Iz meta modela so razvidni posamezni elementi in razmerja med njimi, iz teh pa lahko sklepamo o značilnostih našega merskega pristopa.

Značilnosti našega pristopa izhajajo iz zahtev, ki so povezane z vsebino podatkov. Glavne zahteve izhajajo iz začetnih izhodišč našega dela: četrte stopnje modela CMM in zahtev majhne organizacije za razvoj IS. Poleg tega poznamo značilnosti procesa razvoja PO in merskega procesa. Vsebina merjenja mora oba podpirati.

c) Če opredelimo glavne dele modela podatkov o merjenju, natanko določimo vsebino merjenja. Pri tem najprej izberemo in ustrezno opišemo **nabor (stalnih) merskih izhodišč**, ki jih želimo doseči. Iz teh izpeljemo **nabor pokazateljev**, ki dajejo primerne odgovore na merske cilje in predstavljajo rezultate izvedenih analiz. Podlaga za pokazatelje so rezultati primernega **nabora meritev**, do katerih pridemo z dejanskim merjenjem, združevanjem ali izpeljavo rezultatov merjenja.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Določanje naborov izvedemo z uporabo ustrezne metode. Pri tem sta možna dva pristopa (glej razdelek 2.1.5):

- od zgoraj navzdol oziroma od poslovnih ciljev preko merskih izhodišč in pokazateljev do meritev, kar predstavlja metoda GQM, ter
- od spodaj navzgor oziroma od smiselnih meritev proti merskim izhodiščem, kar predstavlja metoda MGQ.

V našem primeru na podlagi obeh določimo kombinirano metodo, ki predstavlja optimalno kombinacijo obeh. Merska izhodišča določimo na podlagi zahtev ustreznih ključnih področij modela CMM, za kar uporabimo pristop od zgoraj navzdol. Ker naš merski pristop določa, da merjenje izvajajo posamezni izvajalci korakov ali delov korakov, je zaradi enostavnosti primerno izbrati stalen nabor osnovnih meritev, kar je v skladu s korakom od spodaj navzgor. S povezavo merskih izhodišč in meritev določimo še nabor pokazateljev, s čemer se slika merjenja dokončno zaokroži.

d) **Merski repozitorij** predstavlja fizično izvedbo opisanega meta modela, zato bomo ob koncu poglavja omenili še njegove glavne značilnosti.

V nadaljevanju so podrobno predstavljene prve tri ravni in okvirno opisana četrta. Kratko predstavitev vsebine si bralec lahko ogleda tudi v dodatku B.2.

5.2 Merski informacijski model

Merski informacijski model bi lahko predstavili kot mehanizem za opis povezave merskih izhodišč z meritvami in pokazatelji. Model definira jasno strukturo merskih elementov, ki omogočajo natančen opis vseh potrebnih vidikov merjenja (merski plan) in s tem učinkovito izvajanje merskega procesa. Osnova za definicijo modela so merski koncepti in pojmi standarda ISO/IEC 15939. Enako terminologijo uporabljata model CMMI in pristop PSM.

V nadaljevanju je najprej predstavljen opis glavnih delov merskega informacijskega modela, kot so definirani v standardu ISO/IEC 15939 [ISO01, BSi02] in prilagojeno povzeti v našem modelu. Nekateri uporabljeni pojmi so bili s stališča splošne merske teorije kratko že predstavljeni v razdelku 2.1.1.

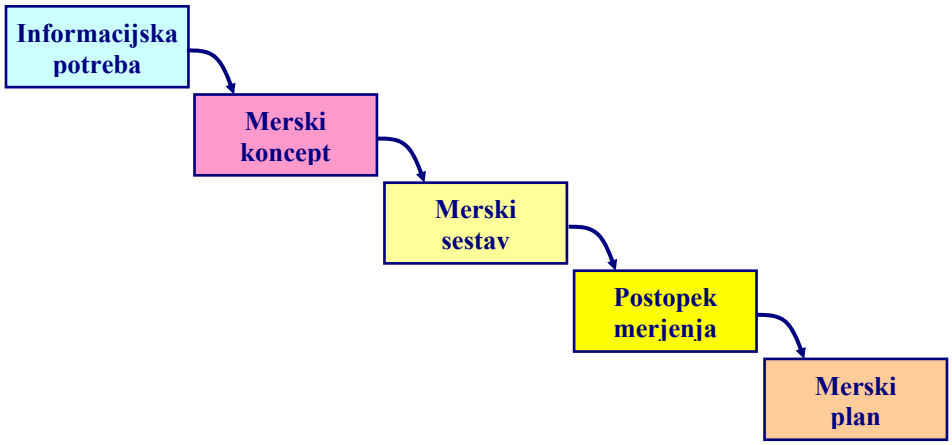
5.2.1 Informacijske potrebe, merska izhodišča in merski koncepti

Slika 5.1 prikazuje, kako postopoma izpeljemo merski plan na podlagi informacijskih potreb organizacije, iz česar so razvidni tudi glavni deli merskega informacijskega modela. Informacijske potrebe izhajajo iz ciljev organizacije ali projekta ter obstoječih ovir (težav, tveganj in pomanjkanja informacij). V razdelku 4.2.2.1 smo opisali, kako na podlagi teh določimo ter izberemo najpomembnejša merska izhodišča.

Vsako mersko izhodišče sodi v enega od štirih razredov meritev:

- potek dela in napredovanje,
- viri in stroški,
- velikost izdelkov ali
- kakovost izdelkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 5.1: Postopna sestava merskega plana na podlagi informacijskih potreb

Meritve v okviru enega razreda se v praksi izvajajo na podobne načine, zato jih predstavimo s podobnimi merskimi koncepti. Merski koncepti so idejni opis predstave o entitetah, ki jih moramo meriti, da bi zadostili zahtevam določenega merskega izhodišča [ISO01].

5.2.2 Merski sestav

Formaliziran opis merskega koncepta je merski sestav, ki je osrednji del merskega informacijskega modela. Pri določanju tega moramo upoštevati informacijske potrebe. V skladu s standardom ISO/IEC 15939 merski sestav natanko določa merjenje izbranih atributov s trinivojsko strukturo, ki jo sestavljajo (glej sliko 5.2):

- osnovne meritve,
- izpeljane meritve in
- pokazatelji.

Naš pristop strukturo dopolnjuje s četrto ravnjo: združenimi meritvami. Kot bo razvidno v nadaljevanju ta dodaten del ne zaplete opisa, temveč pomembno prispeva k jasnosti merjenja.

Vsaka raven meritev je natančneje opredeljena s pravili, modeli in odločitvenimi kriteriji. Taka definicija omogoča natančen opis, ki zmanjšuje odvečnosti ter povečuje točnost zbranih podatkov. Tako opisana struktura omogoča tudi ponovljivost. Ustrezen opis povezav med informacijskimi potrebami in atributi omogoča maksimalni izkoristek osnovnih mer. V nadaljevanju je struktura merskega sestava predstavljena podrobneje.

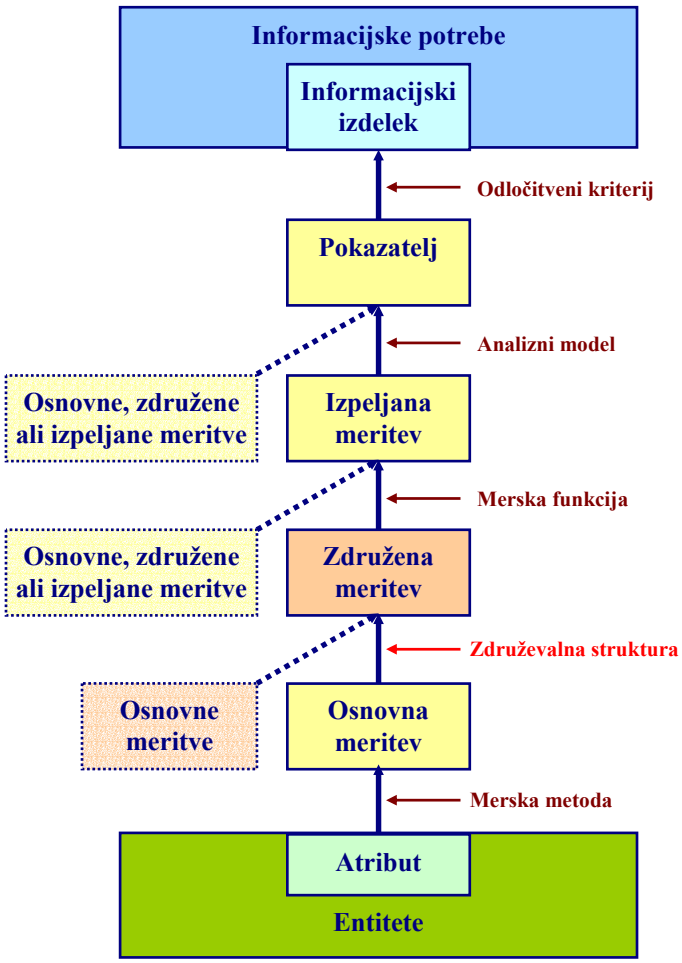
5.2.2.1 Entitete

Naš pristop predvideva merjenje štirih vrst entitet: aktivnosti (procesov), izdelkov, projektov in virov. Med entitete aktivnosti štejemo definirane dele procesa (cikle, faze in korake), medtem ko entitete izdelkov predstavljajo vsi rezultati faz in ciklov. Projekti so opisani s plani. Viri so izvajalci, skupine in orodja.

5.2.2.2 Merljivi atributi

Vsaka od entitet ima več značilnosti, ki so opisane z atributi entitete. Vsi atributi niso primerni za merjenje. Merljivi atributi so tisti, ki jih lahko kakovostno ali količinsko razločujemo med seboj. V merskem sestavu uporabimo tisti del merljivih atributov, ki naslavlja izbrana merska izhodišča. Vsak atribut je lahko del različnih merskih sestavov, ki se nanašajo na eno ali več informacijskih potreb.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 5.2: Struktura merskega sestava

5.2.2.3 Osnovne meritve

Osnovna meritev se nanaša na merjenje izbranega merljivega atributa po določeni merski metodi. Rezultat je izmerjena vrednost, kar prikazuje spodnji matematični prikaz:

$$vOM_i = MM(E{:}A_i)$$

$E{:}A_i$ – (merljiv) atribut A entitete E
i – i-ti primerek entitete E
MM – merska metoda
vOM_i – vrednost osnovne meritve

Osnovna meritev je neodvisna od ostalih mer in se nanaša samo na en merski atribut entitete. Meritvi pripišemo mersko skalo, ki določa obseg dovoljenih vrednosti in tip podatkov. S tem lahko preverjamo kakovost zbranih podatkov. Praktična implementacija osnovnih meritev je zbiranje merskih podatkov, ki ga v našem primeru opravljajo izvajalci vsake aktivnosti.

Merska metoda je opis zaporedja operacij, ki omogoča izračun vrednosti na podlagi vrednosti atributa z upoštevanjem določene merske skale. Običajne operacije so štetje ali merjenje časa. Ločimo dva tipa merskih metod: objektivne in subjektivne (glej razdelek 2.1.1.3). Isto mersko metodo lahko uporablja več atributov, vendar kombinacija enega

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

atributa in ene merske metode vedno določa eno osnovno meritev. Merska metoda je lahko implementirana avtomatsko ali ročno. Opis implementacije merske metode je podan v postopku merjenja.

Merska skala je urejena množica vrednosti ali množica kategorij, ki jih lahko pripišemo atributu. Definira obseg veljavnih vrednosti ter (običajno) enoto merjenja. Enota merjenja je dogovorjena vrednost, ki jo uporabljamo kot merilo. Pripišemo jo objektivnim meritvam, subjektivne je ponavadi nimajo. Glede na razmerja vrednosti in možne operacije poznamo pet tipov merskih skal: nominalne, intervalne, ordinalne, razmernostne in absolutne. Značilnosti posameznih tipov so predstavljene v razdelku 2.1.1.2 (tabela 2.1).

5.2.2.4 Združene meritve

Naš pristop določa, da merljive atribute večjih entitet ne merimo neposredno z osnovno meritvijo, temveč jih določimo z združevanjem več rezultatov osnovnih meritev, za kar uporabimo združevalno strukturo. Matematični prikaz združene meritve je naslednji:

$$vZM_i = ZS(vOM, \ldots)$$

vOM_i – *vrednost osnovne meritve*
ZS – združevalna struktura
vZM_i – *vrednost združene meritve*

Združene meritve bi lahko dobili tudi z neposrednim merjenjem večjih entitet, vendar bi to zahtevalo drugačno izvedbo merjenja. V tem primeru bi jih obravnavali kot osnovne meritve.

Združevalna struktura opisuje zaporedje operacij, s katerimi iz vrednosti več osnovnih meritev določimo vrednost združeni meritvi. Naš pristop predvideva pet vrst združevalnih struktur: po delih izdelka, po funkcijah, po aktivnostih, po vrsti izdelka in po izvajalcih. Združene meritve imajo določeno mersko skalo, ki je (običajno, ne vedno) enaka merskim skalam osnovnih meritev.

5.2.2.5 Izpeljane meritve

Izpeljana meritev združuje informacije o več atributih, zato jo določimo na podlagi dveh ali več osnovnih, združenih ali izpeljanih meritev. Način izpeljave je definiran z mersko funkcijo. Matematična predstava izpeljane meritve je naslednja:

$$vIM_i = MF(vM_j, vM_k, \ldots)$$

vM_i – *vrednost osnovne ali izpeljane meritve*
MF – merska funkcija
vIM_i – *vrednost izpeljane meritve*

Pri tem je pomembno dejstvo, da s kombinacijo značilnosti več atributov pridemo do novih informacij, zato zahtevne manipulacije nad enim atributom ali združevanje rezultatov ne štejemo za izpeljane meritve. Izpeljane meritve se pogosto uporabljajo pri merjenju zunanjih atributov, ki jih ne moremo direktno meriti.

Merska funkcija opisuje algoritem ali izračun, s katerim na podlagi dveh ali več (osnovnih, združenih ali izpeljanih) meritev določimo vrednosti izpeljani meritvi. Merska skala izpeljane meritve je odvisna od skal uporabljenih mer, enako velja za mersko funkcijo (glej tabelo 2.1).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5.2.2.6 Pokazatelji

Pokazatelj je ocena ali vrednost, ki jo določimo na podlagi določenih atributov s pomočjo analiznega modela in z upoštevanjem informacijskih potreb. Matematično si pokazatelje predstavljamo na naslednji način:

$$vP_i=AM(vM_j, vM_k, \ldots)$$

vM_i – vrednost osnovne ali izpeljane meritve
AM – analizni model
vP_i – vrednost pokazatelja

Pokazatelji so tisti del meritev, ki je (običajno grafično) prikazan uporabnikom meritev. Kot taki so podlaga za izvedbo analize in odločanje. Zato je pri prikazovanju pokazateljev pomembno pravilno upoštevanje negotovosti, točnosti in pomena vseh delov pokazateljev.

Analizni model je algoritem ali izračun, ki določa pokazatelje na podlagi dveh ali več (osnovnih, združenih ali izpeljanih) meritev z upoštevanjem odločitvenega kriterija. Analizni model predstavlja logiko razmerij med meritvami in njihovem obnašanju s časovnega vidika. Rezultat analize so ocene ali vrednosti, ki ustrezajo določenim informacijskim potrebam. V našem primeru so rezultati štirih vrst: ocene, analize izvedljivosti, analize performans in napovedi. Kot pri izpeljanih meritvah je tudi tu merska skala odvisna od uporabljenih meritev.

Odločitveni kriterij temelji na razumevanju pričakovanega obnašanja ali na izračunu iz zgodovinskih podatkov. Namen odločitvenega kriterija je olajšati tolmačenje vrednosti pokazatelja. Tipično so to numerične meje, cilji in omejitve za odločanje o ukrepih, ki jih izpeljemo iz zgodovinskih podatkov ali planov z uporabo hevristik ali statističnih metod.

5.2.2.7 Informacijski izdelki

Merski informacijski model sestavlja rezultate merjenja (analiz) v informacijske izdelke. Informacijski izdelki nastanejo ob izvajanju merskega plana in so neposredno povezani z izbranimi informacijskimi potrebami. Sestavlja jih zbirka pokazateljev, pojasnil in priporočil, ki jih merski analitiki predstavijo vodstvu projekta ali organizacije.

5.2.3 Postopek merjenja in merski plan

Merski sestav določa strukturo zbranih merskih podatkov (osnovnih, združenih in izpeljanih meritev ter pokazateljev), medtem ko postopki merjenja določajo načine zbiranja in upravljanja z merskimi podatki. Kot taki so opis izvedbe merskih metod, združevalnih struktur, merskih funkcij in analiznih modelov.

Z uporabo merskega informacijskega modela določimo merski plan. Projektni merski plan je predstavljen, ko obsega zbirko vseh merskih izhodišč, sestavov in postopkov merjenja. Z njim lahko zbiramo vse merske podatke, na podlagi katerih določimo informacijske izdelke.

V našem primeru je glavnina merskih izhodišč določena z naborom stalnih merskih izhodišč, medtem ko se posebna merska izhodišča določijo v fazi planiranja projekta. Tedaj se izvede planiranje meritev, v okviru katerega na podlagi organizacijske predloge definiramo merske entitete, merski sestav in postopke merjenja. V zadnjem koraku planiranja merjenja natanko opišemo merski plan. Med aktivnostmi izvedbe merjenja na podlagi plana zbiramo podatke, jih shranjujemo, analiziramo in na njihovi podlagi določamo informacijske izdelke.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5.3 Meta model - model podatkov o merjenju

Merski plan predstavlja celovit opis merjenja v izbrani organizaciji ali v okviru konkretnega projekta, zato temelji na specifičnih informacijskih potrebah. V našem primeru želimo predstaviti merjenje splošneje; na način, ki bo primeren za množico različnih organizacij z nekaj skupnimi značilnostmi.

Zaradi tega ne bomo predstavljali podrobnosti vseh elementov merskega informacijskega modela, temveč bomo opredelili osrednji del vsebine merjenja. Ta predstavlja jedro, iz katerega lahko majhna organizacija dogradi popoln merski načrt.

Osrednji del merskega informacijskega modela je merski sestav. Merski sestav vsebuje opis vseh podatkov o merjenju in določa razmerja med njimi. Iz teh razmerij je razvidna povezava med merskimi izhodišči, meritvami in analizami, pa tudi postopki merjenja. S tem merski sestav vključuje tudi preostali sestavini merskega plana.

Strukturo merskega sestava lahko predstavimo na različne načine. V našem primeru smo posamezne elemente merskega sestava opisali kot entitete, njihove medsebojne povezave pa z razmerji. Merski sestav je tako predstavljen kot podatkovni model z več diagrami entiteta – razmerje, ki vsebujejo vse podatke o merjenju. Podatki o merjenju so dveh vrst:

- meta podatki oziroma podatki, ki opisujejo merjenje in
- merski podatki, ki predstavljajo izmerjene ali določene (združene ali izpeljane) vrednosti meritev ter rezultate analiz.

Glavni del opisa so meta podatki, zato našo predstavitev merskega sestava z diagrami entiteta - razmerje (v nadaljevanju ER diagrami) kratko imenujemo kar **meta model**.

Opis merjenja z uporabo ER diagramov predlagajo tudi v [Kit01]. Avtorji navajajo, da uporaba meta podatkov in ER diagramov [Mar85] uspešno rešuje težave z definiranjem meritev, opisovanjem vrednosti meritev ter izvajanjem analiz. Njihov model na sliki 5.3[25] obsega 14 entitet, ki so razdeljene na tri dele:

- Splošni del: Splošni merski atributi (recimo velikost ali število napak), enote in obsegi so definirani enotno za vse projekte.
- Razvojni del: Razvojni del določa meta podatke o merjenju, recimo tipe entitet, tipe merjenj, vidike merjenj in podobno.
- Projektni del: Obsega podatke o meritvah projekta, recimo entitete in izmerjene vrednosti glede na mersko skalo in podobno.

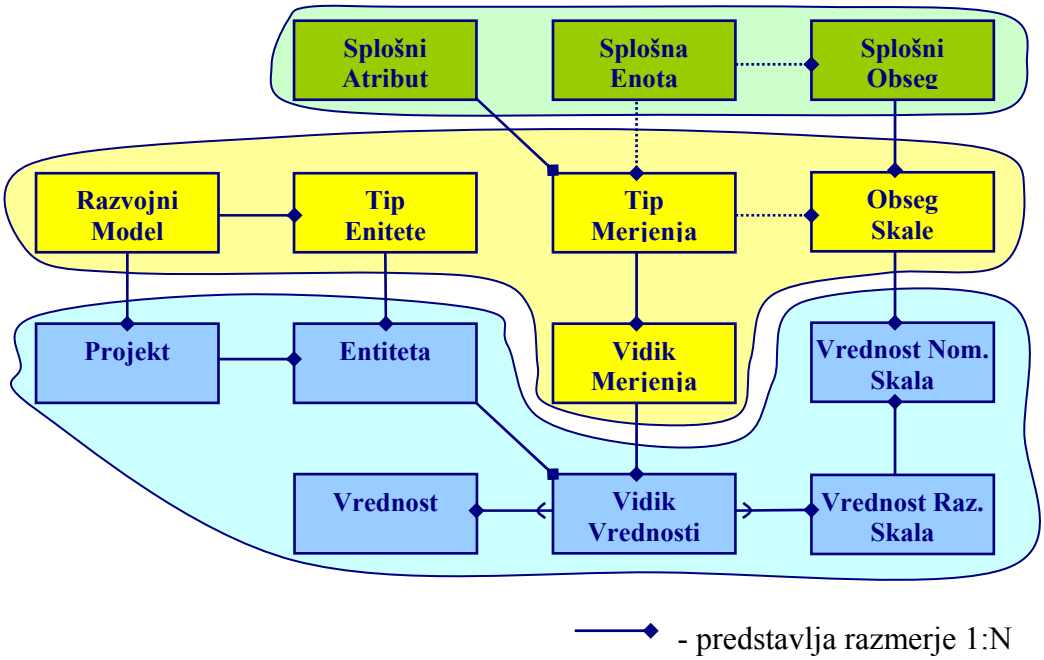
5.3.1 Zahteve, ki vplivajo na meta model

Opis vsebine merjenja mora omogočati, da predstavimo naslednja razmerja med merskimi elementi, ki izhajajo iz zahtev četrte stopnje modela CMM:

- Merjenje je določeno na ravni organizacije, izvaja pa se v okviru projektov. Vsak projekt ga izvaja na predpisan način, vendar ga tudi prilagodi svojim potrebam. Model mora omogočati opis merjenja na organizacijski in projektni ravni.

[25] Zaradi večje preglednosti so razmerja 1:N v ER diagramu označena na poseben način. Enak način prikaza smo uporabili tudi pri vseh naslednjih ER diagramih.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 5.3: Primer: trije deli opisa merskih podatkov po [Kit01]

- Model CMM določa, da merjenje izhaja iz merskih ciljev, glavni namen merjenja pa je nadzirati proces razvoja PO. Iz strukture modela mora biti razvidna povezava med merskimi cilji, meritvami in rezultati, ki izhajajo iz teh.
- Merski informacijski model mora omogočati, da opišemo vse vidike merjenja, ki jih predvideva model CMM. Opis mora zajemati merske podatke o količinskih in kakovostnih značilnostih procesa, projekta in izdelkov.
- Model CMM predvideva shranjevanje podatkov v posebni zbirki merskih podatkov. Del modela mora opisovati shranjevanje merskih podatkov. Model mora omogočati tudi opis zgodovinskih podatkov.

Opis vsebine merjenja v majhni organizaciji za razvoj IS mora omogočati učinkovito izvedbo in avtomatizacijo. Predvidevamo tudi, da bomo hranili dokaj podrobne podatke, ker bo izvedba meritev zaupana posameznim članom projektne skupine.

Določene zahteve izhajajo tudi iz procesa razvoja PO in merskega procesa. Model mora omogočati opis povezave med aktivnostmi, izvajalci in rezultati aktivnosti na eni ter merskimi entitetami in merskimi podatki na drugi. V prejšnjem poglavju opisan merski proces določa nekatere posebnosti, ki tudi vplivajo na strukturo modela:

- Večji del vsebine merjenja je določen s stalnimi merskimi izhodišči in je enak za vse projekte. Drugi del projekt določi na podlagi svojih potreb. Pri izbiri ustreznih pokazateljev za posebna merska izhodišča si lahko pomagamo z merskimi razredi.
- Merjenje izvajajo samo posamezniki, ki z osnovnimi meritvami merijo atribute določenih merskih entitet. Za druge entitete se vrednosti določajo z združevanjem.
- Pomemben del merjenja predstavljajo štiri vrste analiz: ocenjevanje, analiza izvedljivosti, analiza performans in napovedovanje. Rezultati vseh analiz so ustrezno shranjeni.
- Model omogoča, da za vsak merski podatek in merski rezultat določimo, kdaj in kako je bil izmerjen ali določen. Pri izmerjenih vrednostih je znan izvajalec meritev in čas izvedbe, medtem ko je pri določenih (združenih ali izpeljanih) vrednostih mogoče razbrati način, na kateri je bila vrednost združena ali izpeljana.
- Model omogoča hranjenje podatkov o preteklih projektih ter hranjenje podatkov, ki predstavljajo značilnosti vseh preteklih projektov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5.3.2 Opis meta modela

Na podlagi teh zahtev smo zasnovali meta model, ki ga predstavljamo na dveh ravneh:

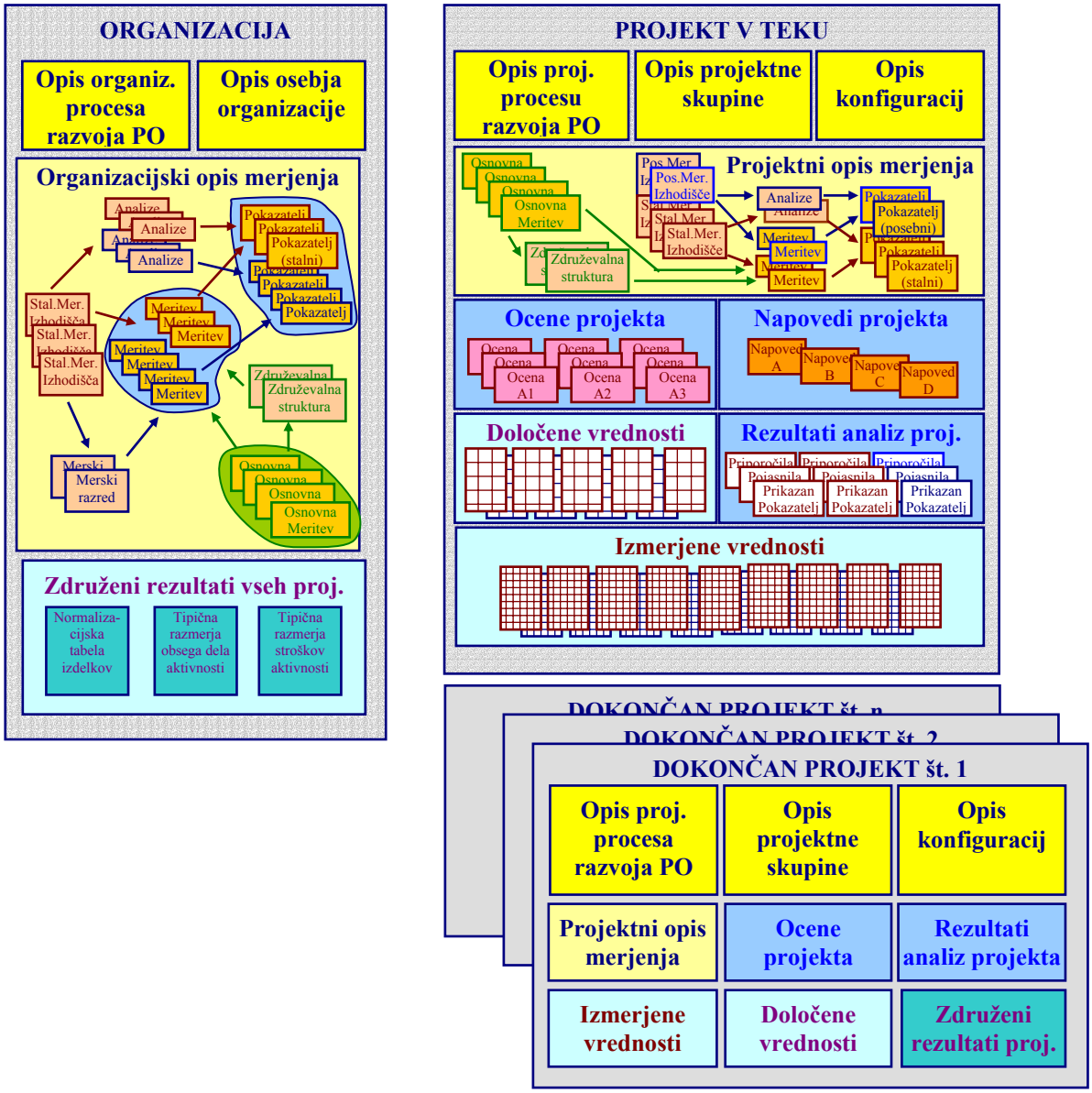
- celovit pogled prikazuje sestavo vseh treh ravni modela in
- podrobni pogledi podrobno opisujejo dele vsake ravni z ustreznimi ER diagrami.

5.3.2.1 Celovit pogled na meta model

Celotna sestava meta modela je predstavljena na sliki 5.4. Model je določen na treh ravneh:

- na organizacijski ravni,
- na ravni projekta v teku in
- na ravni že dokončanih projektov.

Vsem trem ravnem je skupno, da lahko elemente delimo na meta podatke, ki opisujejo izvedbo merjenja, in merske podatke, ki na tak ali drugačen način predstavljajo izmerjene vrednosti. Na sliki 5.4 je mogoče to delitev razbrati iz različnih barv posameznih delov.



Slika 5.4: Celotna sestava meta modela po našem merskem pristopu

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Na organizacijski ravni opisujejo merjenje štirje deli:

- *Organizacijski opis merjenja* celovito definira merjenje na organizacijski ravni.
- *Opis organizacijskega procesa razvoja PO* določa aktivnosti in rezultate procesa.
- *Opis osebja organizacije* predstavlja osebje in skupine v organizaciji.
- *Združeni rezultati vseh projektov* opisujejo značilnosti vseh že dokončanih projektov.

Na ravni projekta v teku sestavljajo meta model naslednji deli:

- *Projektni opis merjenja* natančno definira merjenje na ravni projekta.
- *Opis projektnega procesa razvoja* določa izbran projektni proces.
- *Opis projektne skupine* določa vloge in člane projektne skupine.
- *Opis konfiguracij* predstavlja rezultate aktivnosti med izvajanjem.
- *Izmerjene vrednosti* osnovnih meritev so glavni del merskih podatkov.
- *Določene vrednosti* se določijo iz osnovnih meritev s pomočjo združevanja ali izpeljave.
- *Ocene projekta* so rezultat izvedbe prve vrste analize (ocenjevanja).
- *Napovedi projekta* so rezultat napovedovanja, ki se izvaja neprenehoma.
- *Rezultati analiz projekta* zajemajo preostale informacijske izdelke, ki jih hranimo.

Tretja raven meta modela so **opisi že dokončanih projektov**:

- Za vsak projekt se shranijo vsi podatki o merjenju z izjemo napovedi. Te so namenjene uporabi med razvojem projekta in po zaključku projekta niso več smiselne.
- *Združeni rezultati projekta* se izračunajo ob zaključku projekta in predstavljajo glavne značilnosti celotnega projekta.

Podrobnejši opis vseh delov strukture meta modela je podrobno predstavljen v nadaljevanju.

5.3.2.2 Meta model na ravni organizacije

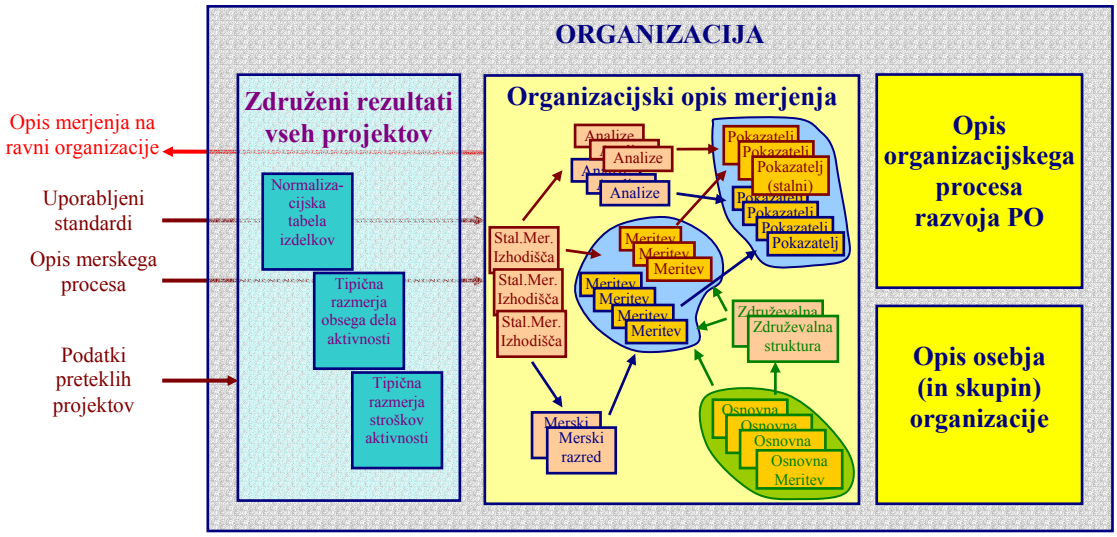
Naš pristop določa, da je merjenje (proces merjenja in vsebina meritev) definirano na organizacijski ravni, posamezni projekti pa ta način povzamejo. Glavnina organizacijske ravni modela so tako podatki o merskih podatkih. Tu so definirani različni nabori, ki jih projekti zgolj povzamejo, recimo nabor stalnih merskih izhodišč, nabor meritev in pokazateljev. Definirana je tudi povezava z organizacijskim procesom razvoja PO. Zadnji del je namenjen združenim podatkom o merjenju preteklih projektov, s čemer izboljšamo učinkovitost izvajanja analiz.

Meta model merskega repozitorija na ravni organizacije je predstavljen na sliki 5.5. V nadaljevanju je sestava vseh štirih delov predstavljena z ločenimi ER diagrami.

Organizacijski opis merjenja

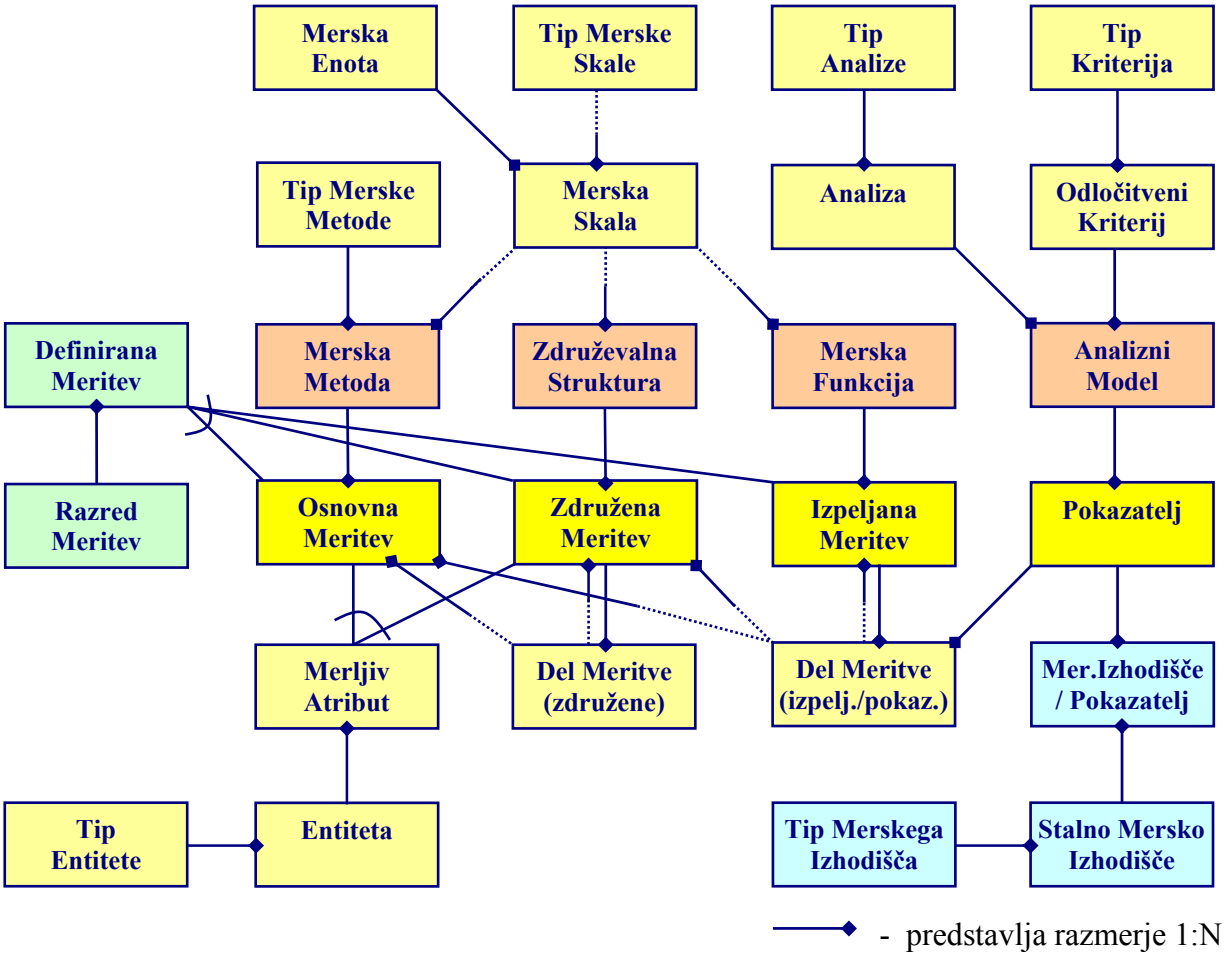
Organizacijski opis merjenja definira vsebino merjenja za vse projekte organizacije. To pomeni, da določa nujni del merjenja: stalni nabor merskih izhodišč, osnovne meritve in pokazatelje, ki jih zahtevajo predpisane analize. Poleg tega opisuje tudi druge možne meritve in pokazatelje, ki jih uporabljajo projekti zaradi specifičnih informacijskih potreb. Slednje so sicer definirane v okviru posameznih projektov, a opisane na organizacijski ravni, da jih lahko uporabijo tudi drugi projekti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 5.5: Struktura meta modela na ravni organizacije

Opis merjenja je usklajen s procesom merjenja na ravni organizacije, uporablja pa tudi sprejete standarde (recimo za določanje kriterijev ob izvajanju analiz). Na sliki 5.6 je predstavljen ER diagram samo za ta del meta modela. Celoten diagram bi dobili, če bi spodnji diagram združili z diagrami preostalih delov, ki so predstavljeni v nadaljevanju.



Slika 5.6: ER diagram za organizacijski opis merjenja

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Iz jedra ER diagrama na sliki 5.6 je razvidna večnivojska hierarhija merskega sestava, ki določa razmerje med merskimi entitetami, merljivimi atributi, osnovnimi meritvami, združenimi meritvami, izpeljanimi meritvami in pokazatelji (glej podpoglavje 5.5).

Entitete obsegajo aktivnosti, izdelke in vire na različnih ravneh podrobnosti. Na organizacijski ravni so določene na podlagi organizacijskega procesa razvoja PO in obsegajo vse možne vrste merskih entitet, recimo vse vrste izdelkov.

Osnovne meritve predstavljajo nabor meritev, ki jih posamezniki stalno merijo. Vsaka osnovna meritev meri en merljiv atribut in vsak merljiv atribut se meri ali določa na en način. Merska metoda določa, kako se to izvede: s štetjem komponent, s preverjanjem števila vrstic, porabo pomnilniškega prostora ali merjenjem dolžine v številu vrstic programske kode. Meritve so povezane z mersko skalo in enoto, kar omogoča preverjanje merskih podatkov.

Tudi združene meritve opisujejo atribute merskih entitet, vendar se te ne merijo, temveč določajo z združevanjem rezultatov več osnovnih ali združenih meritev. Vsi merski atributi so bodisi neposredno merljivi z osnovno meritvijo ali pa se jih (po potrebi) določi z združevanjem.

Izpeljane meritve in pokazatelji so določeni na podlagi dveh ali več (osnovnih, združenih ali izpeljanih) meritev. Tudi izpeljane meritve se določajo po potrebi, kar določajo pokazatelji. Ti so vezani na določeno vrsto analize preko analiznega modela, ki omogoča različne načine prikaza analiziranih podatkov. Opisi merskih postopkov so povezani z merskimi metodami, združevalnimi strukturami, merskimi funkcijami ali analiznimi modeli.

V skladu z določili merskega procesa so definirani razredi meritev. Za vsak razred je definiranih več meritev, ki so lahko osnovne, združene ali izpeljane. Vse definirane meritve predstavljajo celoten nabor meritev, ki je primeren glede na merljive atribute.

Kot rečeno merski proces določa tudi nabor stalnih merskih izhodišč, meritev in pokazateljev, ki jih uporabljajo vsi projekti. Na diagramu so stalna merska izhodišča neposredno povezana samo s pokazatelji. Preko teh posredno določajo stalen nabor meritev. Vsako stalno mersko izhodišče je lahko povezano z več pokazatelji, enako kakor lahko pokazatelj opisuje več merskih izhodišč.

Kratek opis posameznih entitetnih tipov tega dela je predstavljen v tabeli 5.1.

Opis organizacijskega procesa razvoja PO

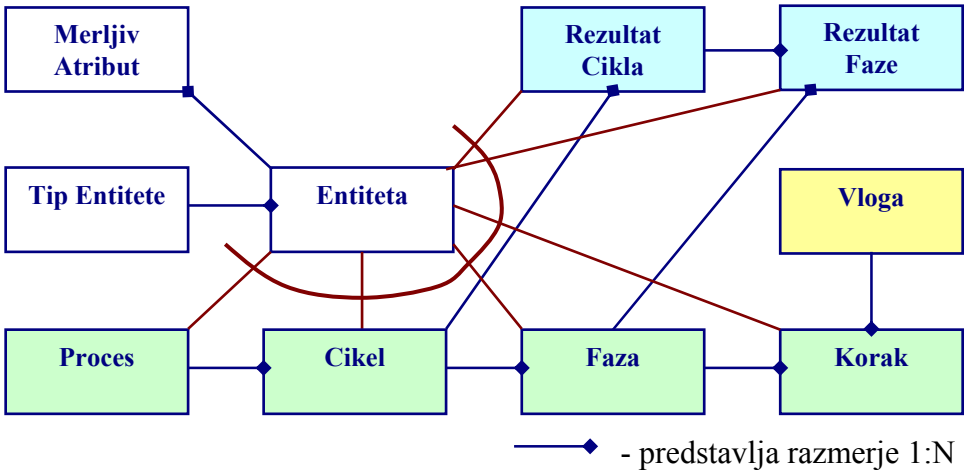
Na podlagi organizacijskega procesa lahko razberemo celotno hierarhijo aktivnosti (cikle, faze in korake), rezultate vseh faz in ciklov ter vloge, ki so zanje zadolžene. Na podlagi teh podatkov določimo glavnino merskih entitet in merljivih atributov. Slika 5.7 prikazuje del meta modela, ki opisuje te podatke.

Aktivnosti so predstavljene na štirih ravneh (proces, cikli, faze, koraki) in obsegajo čisto vse aktivnosti, ki jih predpisuje organizacijski proces razvoja PO. Rezultati so natančno določeni za posamezne cikle in faze, pri čemer rezultati ciklov obsegajo vse rezultate pripadajočih faz. Merska entiteta je bodisi aktivnost, rezultat le-te ali kaj drugega. Vrsto entitete lahko razberemo iz tipa entitete. Za izvedbo vsakega koraka odgovarja določena vloga. Opisi posameznih entitetnih tipov tega dela so razvidni iz tabele 5.2.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Entitetni tip	Opis	Primer vrednosti
Entiteta	Objekti, ki jih merimo (na različnih ravneh podrobnosti)	Modul, faza 3.2, ...
Tip entitete	Aktivnost, izdelek, projekt ali vir	
Merljiv atribut	Značilnost, ki jo lahko kvantitativno ocenimo	Velikost, čas ...
Osnovna meritev	Neposredna meritev merljivega atributa entitete	Velikost modula
Merska metoda	Opis načina, kako izvedemo osnovno meritev	Preštej elemente ...
Tip merske metode	Objektivna ali subjektivna	
Merska skala	Nabor dovoljenih vrednosti rezultatov osnovne meritve	Od 0% do 100%
Tip merske skale	Nominalna, ordinalna, intervalna, razmernostna, ...	
Merska enota	Enota za merjenje	Št. vrstic, minuta ...
Združena meritev	Meritev atributa, ki je določena iz več osnovnih meritev	Čas trajanja cikla
Del meritve (združene)	Posamezni deli združene meritve ali pokazatelja	
Združevalna struktura	Opis načina združevanja osnovnih v združene meritve	Združi korake faze
Izpeljana meritev	Meritev, določena iz več (osnovnih / izpeljanih) meritev	Produktivnost
Del meritve (izp./pok.)	Posamezni deli izpeljane meritve ali pokazatelja	
Merska funkcija	Opis načina izvedbe izpeljane meritve	Velikost/obseg dela
Pokazatelj	Vrednost, določena na podlagi dveh ali več (osnovnih ali izpeljanih) meritev in merskega izhodišča	Ocena in stroški po aktivnostih
Analizni model	Opis načina določanja in prikaza pokazatelja	
Odločitveni kriterij	Kriterij, ki na podlagi merskega izhodišča določa merilo za ustreznost rezultata	Obseg dela - manjši od 110% ocene
Tip kriterija	Omejitev, statistična metoda, hevristika, ...	
Analiza	Nabor postopkov za določanje pokazateljev iz meritev	
Tip analize	Ocenjevanje, izvedljivost, performanse, napovedovanje	
Razred meritev	Poraba časa in napredovanje, velikost izdelkov, viri in stroški, kakovost izdelkov	
Definirana meritev	Nabor definiranih (osnovnih in izpeljanih) meritev	Št. vrstic modula
Stalna merska izhodišča	Nabor merskih izhodišč, ki jih vedno merimo	Število zahtev
Tip merskih izhodišč	Cilj, težava, tveganje ali pomanjkanje informacij	
Mer.izhodišča/Pokazatelj	Pokazatelji za določena merska izhodišča	

Tabela 5.1: Entitete za organizacijski opis merjenja



Slika 5.7: ER diagram za opis organizacijskega procesa razvoja PO

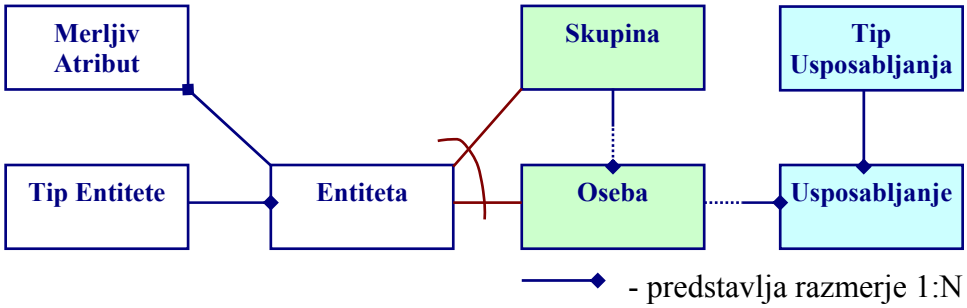
Entitetni tip	Opis	Primer vrednosti
Proces	Opis organizacijskega procesa razvoja PO – en element	
Cikel	Cikli procesa razvoja PO	Izgradnja, testiranje
Faza	Faze, ki pripadajo posameznim ciklom procesa	Planiranje projekta
Korak	Koraki, ki so del posameznih faz procesa	Poročaj vodstvu
Rezultat faze	Vmesni izdelki, ki so dokončani ob zaključku faze	LERD
Rezultat cikla	Izdelki, ki so dokončani ob zaključku cikla	Specifikacija zahtev
Vloga	Vse vloge, ki jih predpisuje proces razvoja PO	Vodja analize

Tabela 5.2: Entitete za opis organizacijskega procesa razvoja PO

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Opis osebja in skupin organizacije

Ta del modela opisuje osebje organizacije ter sestavo skupin na organizacijski ravni v sedanjosti in preteklosti (slika 5.8). Tudi na podlagi teh podatkov določamo entitete in atribute za merjenje.



Slika 5.8: ER diagram za opis osebja in skupin organizacije

Za vsako skupino na ravni organizacije (recimo skupino SQA ali SEPG) so znani člani, ki ji trenutno pripadajo. Sicer so v entitetnem tipu Osebe zbrani podatki o vseh osebah, ki so bile ali so trenutno člani organizacije. Za vsako osebo je mogoče preveriti tudi usposabljanja, ki jih je opravila. Entitetni tipi tega dela opisa so predstavljeni v tabeli 5.3.

Entitetni tip	Opis	Primer vrednosti
Skupina	Opis vseh skupin na ravni organizacije	SQA, SEPG
Oseba	Opis oseb, ki so (bile) zaposlene v organizaciji	Tone Novak...
Usposabljanje	Seznam usposabljanj po posameznih osebah	
Tip usposabljanja	Opis vrste usposabljanja	Tečaj

Tabela 5.3: Entitete za opis osebja in skupin organizacije

Združeni rezultati vseh projektov

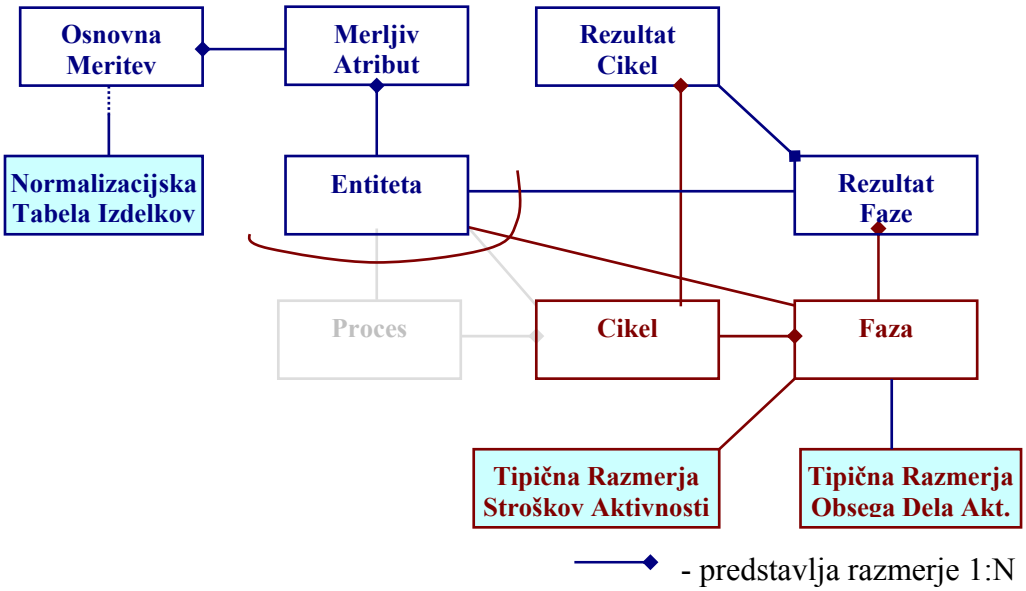
Združeni rezultati vsebujejo tabele z združenimi podatki vseh že dokončanih projektov. Ta skupina podatkov ni nujno potrebna za izvajanje merjenj, vendar je koristna za učinkovitejšo izvedbo analiz (predvsem napovedovanja). Tabele namreč vsebujejo rezultate, ki se pogosto rabijo, a zahtevajo obsežno računanje. Tabele se ažurirajo ob zaključku izvajanja vsakega projekta. Slika 5.9 predstavlja strukturo tega dela modela.

Normalizacijska tabela izdelkov predstavlja tipično razmerje med vrednostjo nekaterih osnovnih meritev izdelkov in velikostjo končnega izdelka s procesnega ali podatkovnega stališča. Tipična razmerja obsega dela in stroškov posameznih aktivnosti so izražena za posamezne faze in posredno cikle procesa. Podrobnejša predstavitev tabel združenih rezultatov je bila opisana v razdelkih 4.3.1.3 in 4.3.4.1. Entitetni tipi tega dela opisa so predstavljeni v tabeli 5.4.

Entitetni tip	Opis
Normalizacijska tabela	Razmerje med vrednostjo nekaterih osnovnih meritev in končno velikostjo s stopnjo odvisnosti (glede na pretekle projekte)
Tipična razmerja obsega dela	V odstotkih izraženo razmerje potrebnega obsega dela za izvedbo posameznih faz in ciklov glede na podatke preteklih projektov
Tipična razmerja stroškov	V odstotkih izraženo razmerje dodatnih stroškov za izvedbo posameznih faz in ciklov glede na podatke preteklih projektov

Tabela 5.4: Entitete za združene rezultate vseh projektov

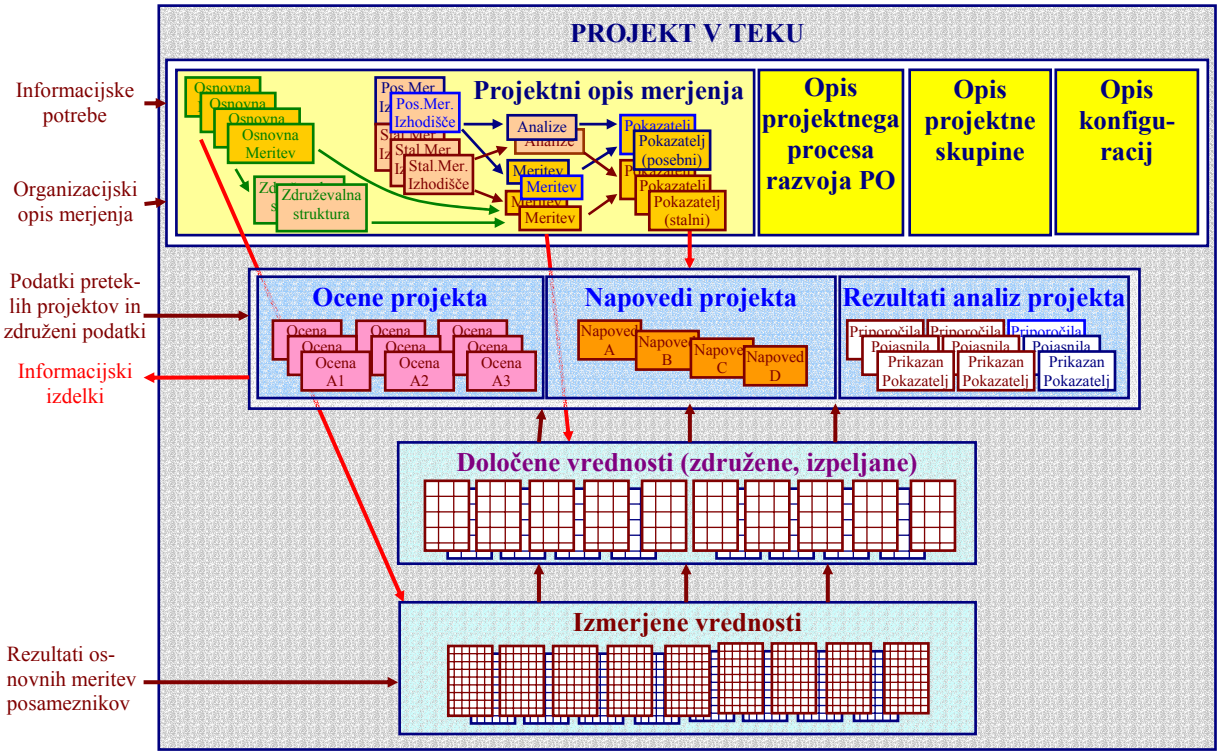
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 5.9: ER diagram za združene rezultate vseh projektov

5.3.2.3 Meta model na ravni projekta v teku

Operativno se merjenje izvaja na ravni projektov. Pri tem posamezni projekti povzamejo način merjenja, ki ga določa organizacija. Večji del merjenja je na ta način predpisan s stalnimi merskimi izhodišči, pokazatelji ter naborom osnovnih meritev, ki jih izvajajo posamezniki. Ta del v skladu z zahtevami CMM predpisuje obvezen del merjenja, ki ga vsi projekti izvajajo na enak način. Drugi del merjenja lahko projekt prilagodi svojim specifičnim zahtevam, vendar ima tudi pri tem možnost uporabiti vnaprej definirane meritve in pokazatelje, ki so določeni na organizacijski ravni.



Slika 5.10: Struktura meta modela na ravni projekta

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

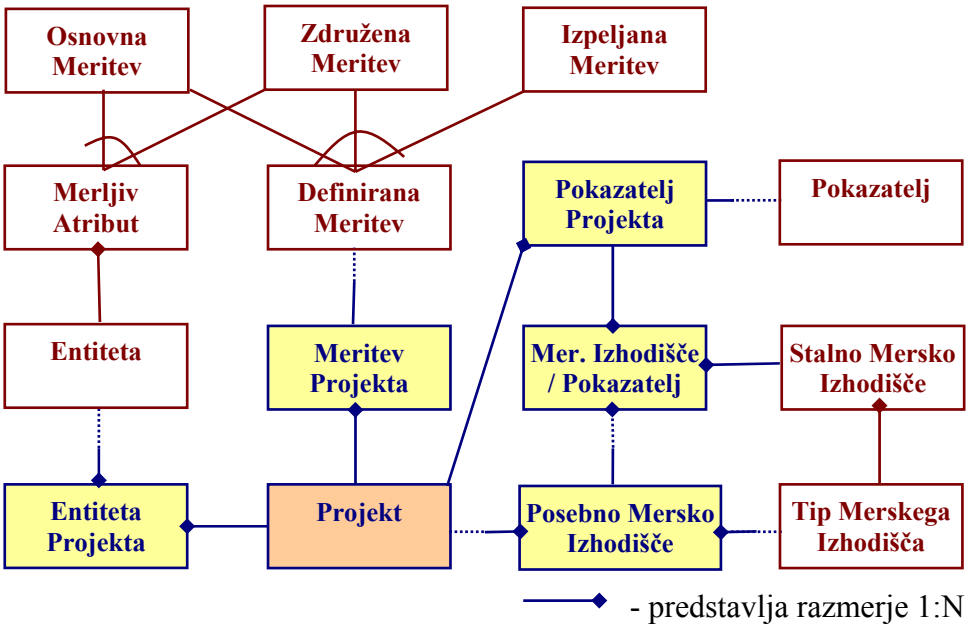
Iz slike 5.10 je razvidna vsebina vseh skupin podatkov o merjenju na projektni ravni. Posamezni deli so natančneje predstavljeni v nadaljevanju.

Projektni opis merjenja

Ta del definira merjenje na ravni projekta. Iz organizacijske ravni projekti prevzamejo način merjenja. Nekatere dele prevzamejo nespremenjene (recimo nabor osnovnih meritev, združevalne strukture, nabor stalnih merskih izhodišč in njim pripadajočih pokazateljev), druge pa uporabijo za oblikovanje svojega opisa. Tak primer je recimo določanje nabora merskih entitet in merljivih atributov projekta. Glavnina merjenja je na ta način natančno določena.

Vsak projekt lahko poleg tega določi še posebna merska izhodišča, ki naslavljajo specifične merske potrebe projekta. Primerne pokazatelje zanje se običajno izbere na podlagi vnaprej definiranega nabora pokazateljev, ki so opisani na organizacijski ravni. Če ti ne zadostujejo, naš proces predvideva definiranje novih (združenih in izpeljanih) meritev in pokazateljev na organizacijski ravni. Slika 5.11 prikazuje diagram tega dela meta modela.

Ker je opis merskih entitet, atributov, meritev in pokazateljev definiran na organizacijski ravni, potrebujemo za opis merjenja na projektni ravni samo nekaj dodatnih entitetnih tipov. Zadoščali bi že entitetna tipa z opisi merskih entitet projekta in posebnih merskih izhodišč, vendar zaradi jasnosti posebej določimo tudi vse meritve in pokazatelje, ki jih projekt uporablja.



Slika 5.11: ER diagram za projektni opis merjenja

Merske entitete projekta določimo na podlagi entitet organizacijske ravni. Za te entitete izvedemo vse osnovne meritve, ki jih vedno izvajamo. Na podlagi specifičnih potreb določimo posebna merska izhodišča, ki skupaj s stalnimi merskimi izhodišči določajo nabor pokazateljev projekta. Ker vsako mersko izhodišče lahko ustreza več pokazateljem in vsak pokazatelj opisuje več izhodišč, predvidimo tudi entitetni tip Mersko izhodišče/Pokazatelj. Na podlagi izbranih pokazateljev določimo, katere združene in izpeljane meritve potrebujemo v okviru projekta, kar skupaj z osnovnimi meritvami obstoječih entitet opišemo v entitetnem tipu Meritev projekta. Vse meritve so povezane z entitetnim tipom Projekt. Tabela 5.5 prikazuje kratke opise posameznih entitetnih tipov.

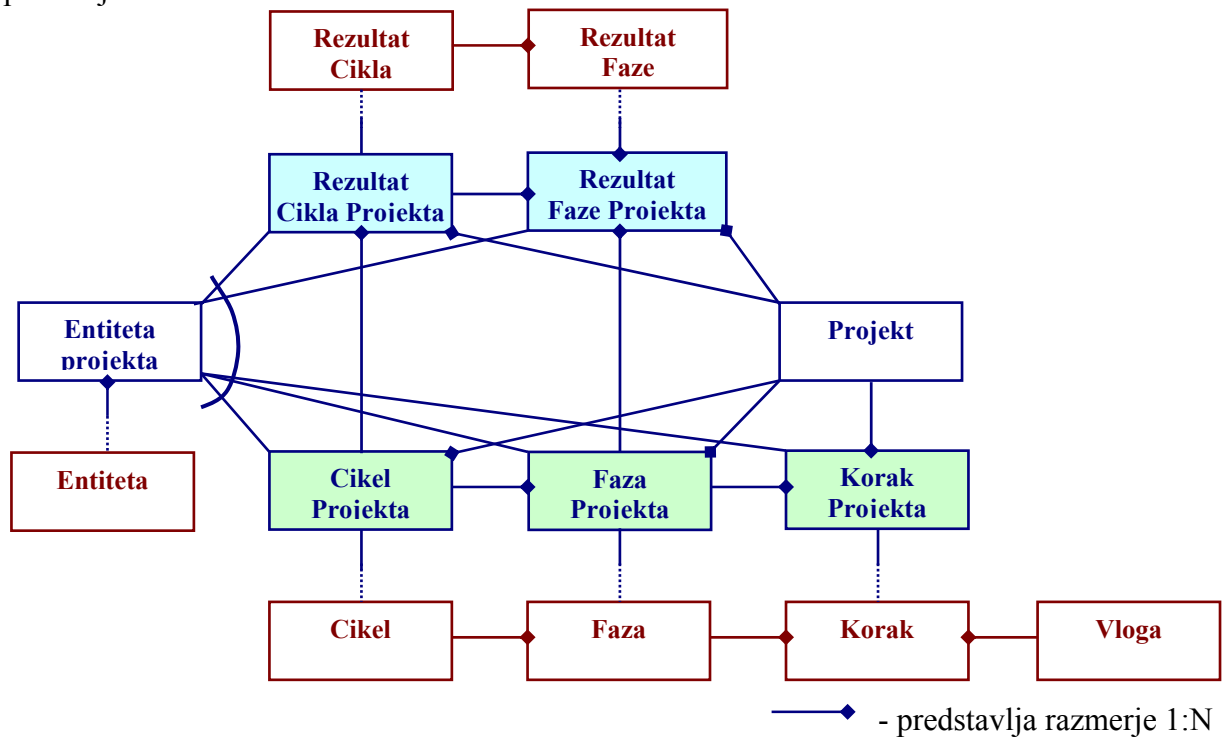
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Entitetni tip	Opis
Projekt	Opis projekta – omogoča združevanje podatkov več projektov
Entiteta projekta	Vsi merski objekti, ki jih lahko merimo v okviru projekta
Meritev projekta	Nabor (osnovnih, združenih in izpeljanih) meritev projekta
Posebno mersko izhodišče	Nabor merskih izhodišč, ki naslavljajo posebne potrebe projekta
Pokazatelj projekta	Nabor pokazateljev, ki jih uporablja projekt
Mersko izhodišče/Pokazatelj	Povezava med pokazatelji in (stalnimi in posebnimi) merskimi izhodišči

Tabela 5.5: Entitete za projektni opis merjenja

Opis projektnega procesa PO

V projektnem opisu procesa razvoja PO je določeno, kateri cikli, faze in koraki se bodo dejansko izvajali ter kateri rezultati bodo nastali ob izvajanju projekta. Slika 5.12 podrobneje prikazuje ta del.



Slika 5.12: ER diagram za opis projektnega procesa razvoja PO

Prilagodljiv proces razvoja PO omogoča, da v okviru projekta izvajamo samo določene aktivnosti celotnega procesa. Projektni proces zato opišemo z naborom izbranih aktivnosti. Aktivnosti na projektni ravni je manj ali kvečjemu enako, kot jih določa organizacijska raven. Ker prilagodljiv proces (lahko) določamo postopoma med izvajanjem (najprej definiramo cikle, nato postopoma faze in korake), moramo uporabiti opis aktivnosti procesa z več merskimi entitetami. Za vsak korak projekta lahko določimo vlogo, ki je odgovorna za njegovo izvedbo.

Za cikle in faze so na organizacijski ravni znani natanko določeni rezultati oziroma izdelki. Z izbiro projektnih aktivnosti določimo tudi te, vendar velja, da lahko obstaja več izdelkov iste vrste. Tipičen primer so recimo moduli: projekt lahko v okviru projekta razvije množico modulov (recimo modul A, modul B) in vsak posebej je merljiva entiteta, medtem ko je modul med merljivimi objekti na organizacijski ravni naveden samo enkrat. To velja za večino rezultatov faz, vendar ne za rezultate ciklov. Organizacijski rezultati ciklov vsebujejo tudi vse projektne rezultate ciklov. Aktivnosti in rezultati določajo (glavni del) nabora merljivih entitet projekta. Tabela 5.6 podrobneje predstavlja posamezne entitetne tipe.

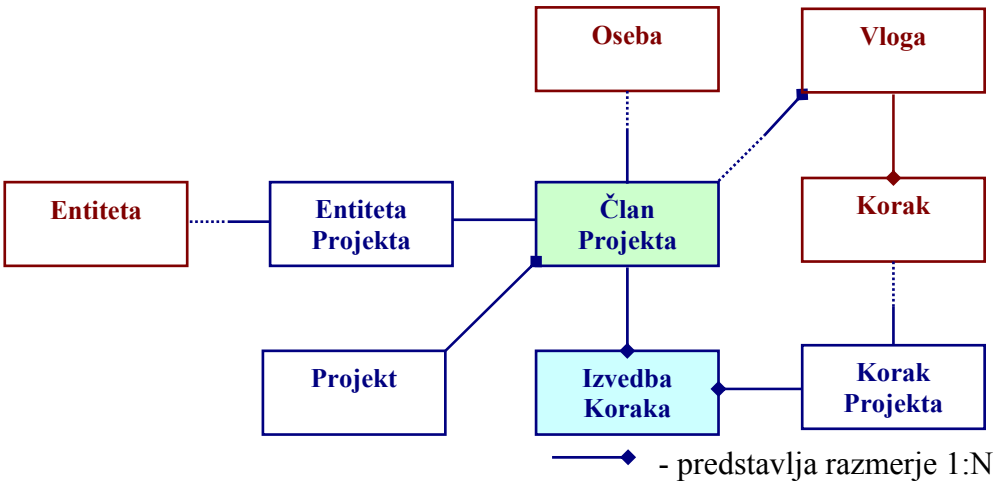
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Entitetni tip	Opis
Cikel projekta	Cikli projektnega procesa razvoja PO
Faza projekta	Faze, ki se izvajajo v okviru projekta
Korak projekta	Koraki, ki izvajajo v okviru projekta
Rezultat koraka projekta	Vmesni izdelki posameznih korakov projekta
Rezultat faze projekta	Izdelki faz, ki so razviti med izvajanjem projekta

Tabela 5.6: Entitete za opis projektnega procesa razvoja PO

Opis projektne skupine

Opis projektne skupine določi člane projektne skupine in razdelitev vlog v okviru projekta. Opis članov je pomemben zaradi tega, ker osnovne meritve izvajajo posamezni člani. V okviru modela moramo zato omogočiti tudi opis manjših aktivnosti od koraka, ki jih izvajajo posamezni izvajalci. Osnova za določitev teh podatkov so projektni plani. Diagram na sliki 5.13 prikazuje entitetne tipe in razmerja, ki pokrivajo to področje.



Slika 5.13: ER diagram za opis projektne skupine

Vsi člani projektne skupine so zavedeni med zaposlenimi organizacije, zato jih v okviru projekta samo naštejemo. Iz tega razmerja lahko razberemo tudi njihovo pripadnost skupinam na ravni organizacije (recimo SQA ali SEPG). Vsak član projektne skupine opravlja lahko eno ali več vlog, pri čemer proces razvoja PO določa, da so tudi pri najmanjših projektih vse vloge razdeljene. Na podlagi vlog lahko določimo, kdo vodi izvajanje posameznih korakov projekta. Dejansko izvajanje projekta je opisano z entitetnim tipom Izvedba koraka. Vsak korak projekta je mogoče izvesti v enem ali več delih, pri čemer posamezne deleže lahko izvajajo različni člani projektne skupine. Člani projektne skupine spadajo med merljive entitete projekta. V tem delu določimo nova entitetna tipa, ki sta predstavljena v tabeli 5.7.

Entitetni tip	Opis
Član projekta	Nabor članov projektne skupine
Izvedba koraka	Opis deleža koraka, ki ga izvaja en izvajalec

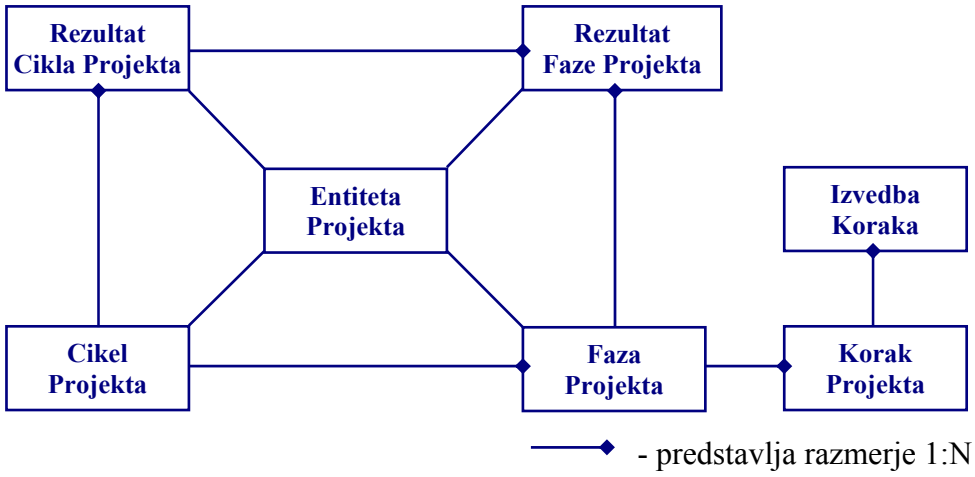
Tabela 5.7: Entiteti za opis projektne skupine

Opis konfiguracij

Upravljanje konfiguracij omogoča, da med izvajanjem projekta razberemo, do kakšne stopnje so posamezni rezultati aktivnosti (izdelki) narejeni. Na podlagi povezave med aktivnostmi in rezultati lahko določimo tudi status projekta do faze natančno.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

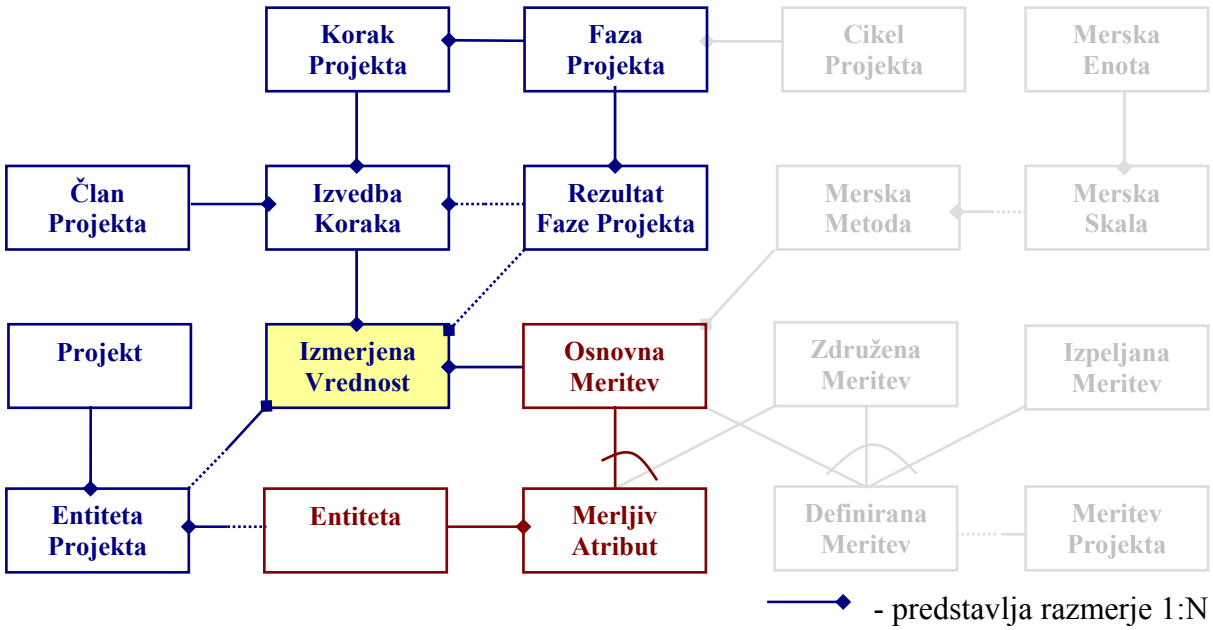
Podatki o konfiguracijah so shranjeni v že predstavljenih entitetnih tipih, ki določajo projektne aktivnosti in rezultate (slika 5.14). V opisu teh obstajajo atributi, v katere shranjujemo status aktivnosti ali rezultatov.



Slika 5.14: ER diagram za opis konfiguracij

Izmerjene vrednosti

Glavni del merskih podatkov so izmerjene vrednosti, ki so izhodišče za izvedbo vseh analiz. Izmerjene vrednosti so rezultati osnovnih meritev, ki jih izvajajo člani projekta ob koncu vsake aktivnosti (koraka ali dela koraka). Pri tem se meri vedno enake atribute. Slika 5.15 prikazuje ER diagram za ta del modela.



Slika 5.15: ER diagram za predstavitev izmerjenih vrednosti

Potek projekta si predstavljamo kot zaporedje korakov. Vsak korak se izvede v enem ali več delih, pri čemer vsak del predstavlja prispevek posameznega člana. Pri tem lahko nastane izdelek ali pa tudi ne; če nastane, je ta izdelek del rezultata celotne faze.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

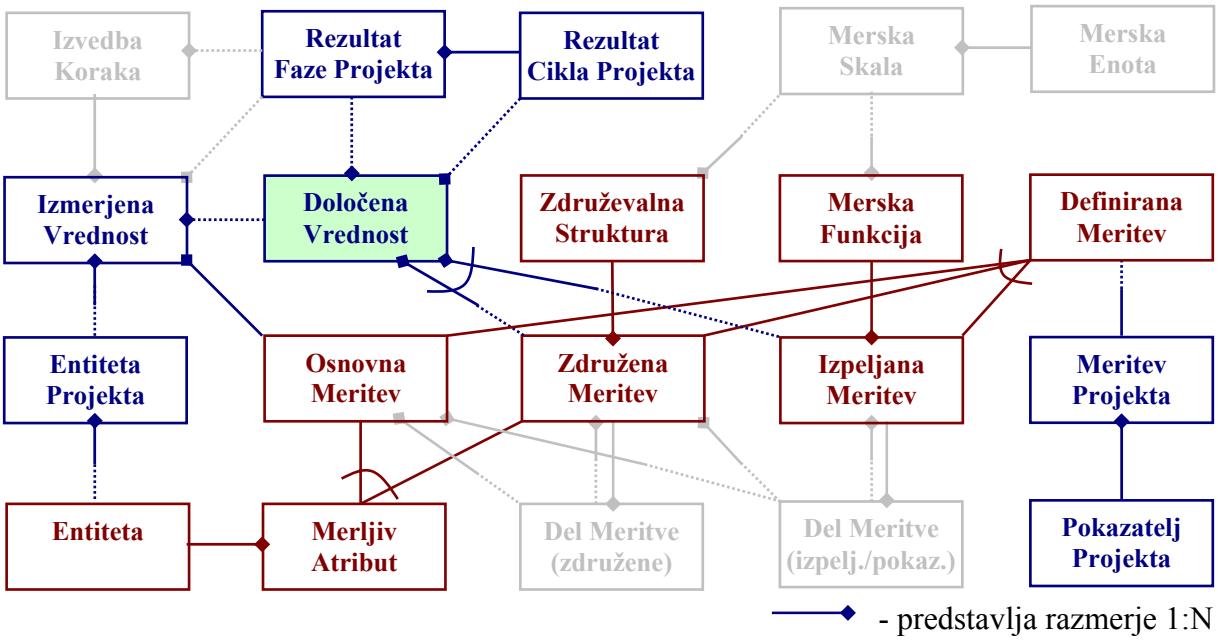
Shranjevanje izmerjenih vrednosti je vezano na posamezne dele koraka, vendar se izmerjene vrednosti shranjujejo posamič v okviru posebnega entitetnega tipa. Osnovne meritve natanko določajo, katere atribute merimo in kako izvedemo vsako meritev. V vsakem primeru izvedemo vsaj merjenje trajanja aktivnosti, vložen obseg dela, obseg stroškov in vse vidike kakovosti (napake). V kolikor je pri aktivnosti nastal rezultat, člani izmerijo tudi velikost rezultata.

Določene vrednosti (združene ali izpeljane)

Vrednosti preostalega dela meritev se ne merijo in so določene na podlagi izmerjenih vrednosti. Določene vrednosti predstavljajo meritve dveh vrst: združene in izpeljane meritve.

Združene meritve merijo atribute entitet, ki jih ne merimo neposredno. Določimo jih tako, da s pomočjo združevalnih struktur (recimo v okviru celotne faze, izdelka ali podobno) sestavimo izmerjene vrednosti več osnovnih ali združenih meritev. Drugi del določenih vrednosti se nanaša na izpeljane meritve, ki kombinirajo dva ali več merskih podatkov (osnovnih, združenih ali izpeljanih vrednosti) v nov rezultat.

Tako združevanje kot izpeljavo izvajamo po potrebi, če izbrani pokazatelji to zahtevajo. Te določimo na podlagi stalnih in posebnih merskih izhodišč, iz pokazateljev pa lahko natanko predvidimo, katera izmed definiranih meritev bo potrebna. Slika 5.16 prikazuje ta del modela.



Slika 5.16: ER diagram za predstavitev določenih vrednosti

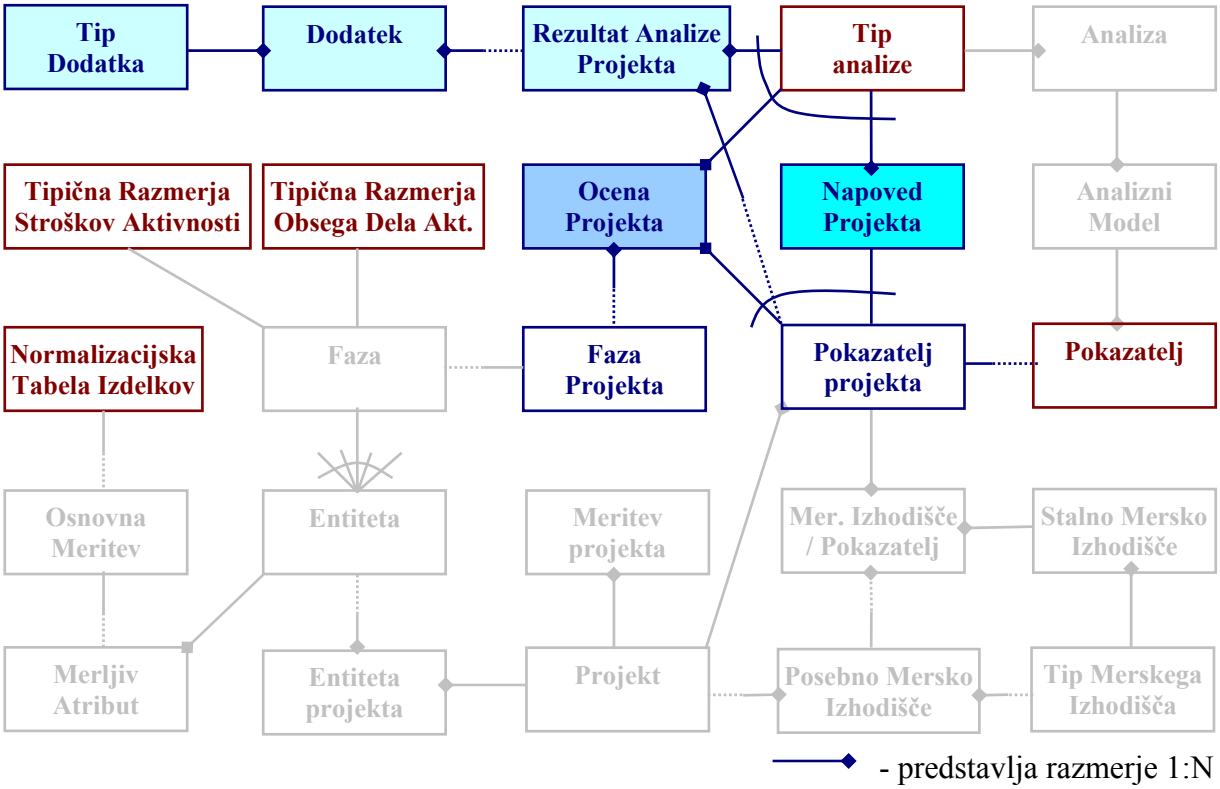
Za vsako združeno ali izpeljano meritev, ki jo projekt uporablja, obstaja natanko predpisan način, kako iz izmerjenih ali prej določnih vrednosti izračunamo novo vrednost, ki se zapiše v poseben entitetni tip. Vsaki meritvi lahko pripada več vrednosti: v primeru združevanja lahko obstaja skupen obseg dela faze A in faze B, enako lahko pri izpeljanih meritvah določimo produktivnost vsakega cikla posebej. Pri določenih vrednostih, ki so nastale z združevanjem, je mogoče razbrati, na podlagi katerih izmerjenih vrednosti so izračunane ter na kateri izdelek faze ali cikla so vezane.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Ocene, napovedi in rezultati analiz

Izmerjene in določene vrednosti so podlaga za izvedbo štirih vrst analiz, rezultat katerih so različni informacijski izdelki[26]. Tudi ti so štirih vrst: ocene, analize izvedljivosti, analize performans in napovedi. V meta modelu so analize opisane s pokazatelji, ki jih povezujejo z merskimi izhodišči in naborom meritev oziroma izmerjenih vrednosti.

Za izvedbo analiz so potrebni podatki o preteklih projektih. Ti so dostopni preko tabel združenih podatkov o preteklih projektih in iz podatkov preteklih projektov. Slika 5.17 prikazuje entitetne tipe, ki predstavljajo informacijske izdelke.



Slika 5.17: ER diagram za predstavitev ocen, napovedi in rezultatov analiz

Ocene projekta so rezultat izvedbe prve vrste analize (ocenjevanja). Izvajamo jo ob vsakem planiranju (v določenih fazah), zato hranimo več različic ocen. Ocene so opisane v okviru stalnih merskih izhodišč, ki določajo ustrezne pokazatelje. V okviru ocen ne shranimo samo vrednosti, temveč tudi meji 90% območja zaupanja.

Napovedi so definirane s stalnim naborom pokazateljev. Za razliko od ocen se napovedi ne hranijo, temveč samo sproti uporabljajo. To pomeni, da za stalen nabor pokazateljev hranimo samo trenutne vrednosti. Enako kot pri ocenah tudi tu shranjujemo območje zaupanja.

Rezultati analiz predstavljajo preostale informacijske izdelke, za rezultate analiz dosegljivosti in performans. Gre za predstavitve pokazateljev z dodatki (pojasnili, predlogi ali čim podobnim), ki so običajno v obliki slike (grafa) ali besedila. Zato jih opišemo s kratkim opisom in naslovom v repozitoriju, kjer so datoteke shranjene. Pokazateljev, ki jih uporabljamo ob sprotnem delu, posebej ne hranimo, ker jih lahko vedno znova določimo na podlagi zgodovinskih podatkov.

[26] Termin informacijski izdelki je uporabljen v skladu s terminologijo pristopa PSM.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Kratek opis entitetnih tipov z izmerjenimi in določenimi vrednostmi ter entitetnih tipov z rezultati analiz je predstavljen v tabeli 5.8.

Entitetni tip	Opis
Izmerjena vrednost	Izmerjene vrednosti osnovnih meritev, ki jih posamezniki izmerijo ob izvedbi vsakega koraka ali dela koraka
Določena vrednost	Rezultati združenih in izpeljanih meritev, izračunane s pomočjo združevalnih struktur ali merskih funkcij – vrednosti, izražene v ustreznih enotah
Ocena projekta	Ocene (vrednost in meji 90% območja zaupanja) za stalen nabor atributov, ki se določajo ob vsakem planiranju
Napoved projekta	Napovedi (vrednost in meji 90% območja zaupanja) za stalen nabor atributov, ki so najboljše v danem trenutku
Rezultat analize projekta	Predstavitev pokazateljev - rezultatov pomembnejših (opis in naslov shranitve) analiz izvedljivosti in analiz performans, določenih s pomočjo analiznih modelov in odločitvenih kriterijev (običajno v obliki)
Dodatek	Priloga, ki pojasnjuje določen rezultat analize – običajno besedilo
Tip dodatka	Vrsta dodatka, recimo pojasnilo, predlog, opis predpostavk ali kaj podobnega

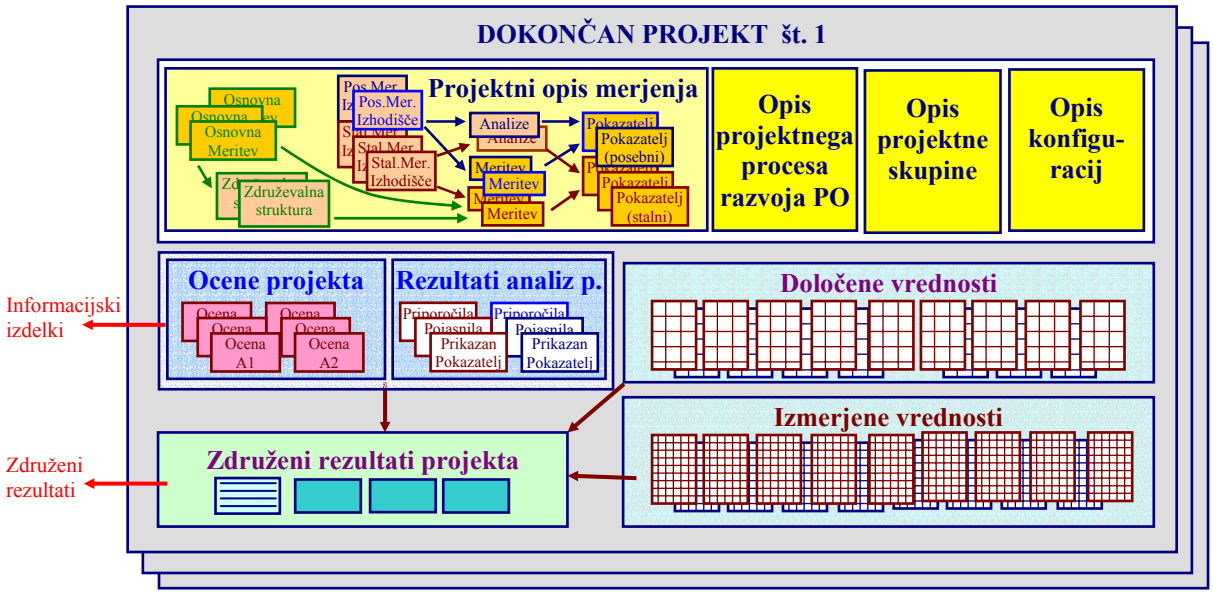
Tabela 5.8: Entitete za opis rezultatov merjenja

5.3.2.4 Meta model, ki predstavlja dokončane projekte

Pomembna značilnost našega pristopa je uporaba zgodovinskih podatkov pri analizah, predvsem za ocenjevanje in napovedovanje. V ta namen za vsak dokončan projekt shranimo vse zbrane meta in merske podatke z izjemo napovedi, ki po zaključku projekta niso več potrebne. Tako lahko po potrebi izvajamo analize naknadno, s čemer lahko primerjamo zgodovinske podatke z merskimi podatki nastajajočih projektov tudi z novih vidikov.

Pokaže se še ena prednost našega pristopa: velike možnosti za dodatne analize že dokončanih projektov. Temelj so natančne izmerjene vrednosti vseh glavnih atributov. Ker je zrnatost majhna, lahko iz teh vrednosti izpeljemo veliko različnih pokazateljev.

Ker se merski podatki preteklih projektov pogosto uporabljajo, je koristno le-te ob zaključku projekta združiti v posebno tabelo, ki omogoča učinkovitejšo izvedbo analiz. Nabor teh podatkov je prilagojen ocenjevanju in napovedovanju, kjer se uporablja linearna regresija. Del rezultatov se shrani na podoben način kot združeni rezultati na organizacijski ravni.



Slika 5.18: Struktura meta modela na ravni dokončanih projektov

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Slika 5.18 prikazuje sestavo meta podatkov in merskih podatkov, kakršna se shrani za vsak dokončan projekt. Ker je glavnina podatkov o merjenju dokončanih projektov enaka kot v razdelku 5.3.2.3, je v nadaljevanju podrobneje predstavljen samo del, ki predstavlja združene rezultate projekta.

Združeni rezultati projekta

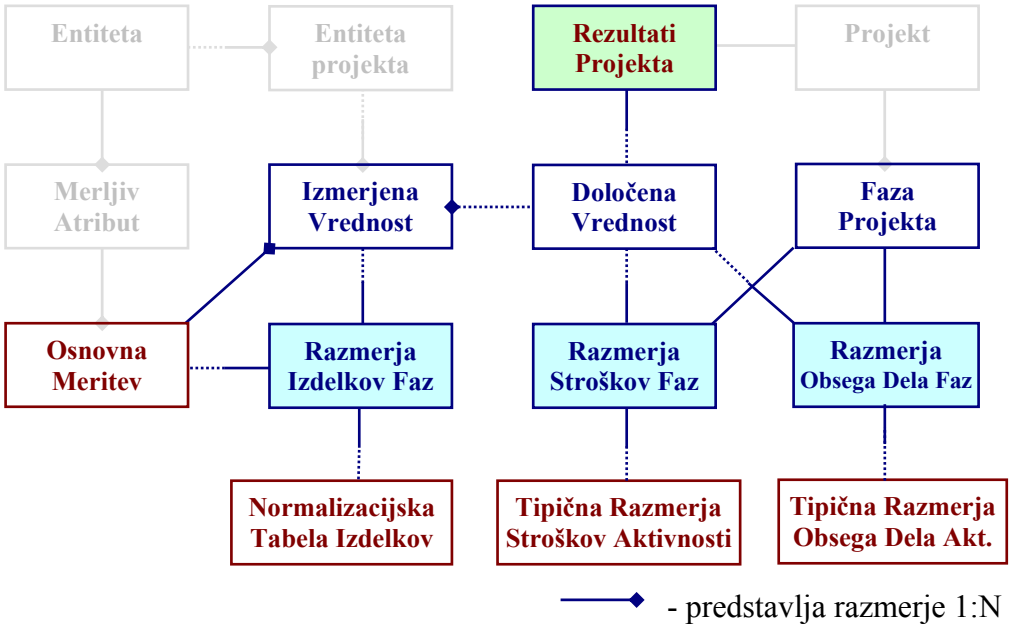
Združeni rezultati predstavljajo povzetek najpomembnejših izmerjenih oziroma določenih vrednosti projekta, ki se uporabljajo za določanje območja zaupanja s pomočjo regresijske enačbe. Pri tem se za izhodišče jemlje razmerje med doseženimi vrednostmi in ocenami, ki so že zbrane v posebni entiteti. Združeni rezultati se nanašajo na celoten projekt in obsegajo naslednje vrednosti:

- končno velikost vseh rezultatov (izdelkov) projekta – za procesni in podatkovni del;
- celoten obseg dela za procesni in podatkovni del;
- celotno dolžino trajanja projekta;
- celotne stroške projekta;
- skupno število odkritih napak;
- skupno število odpravljenih napak;
- skupen obseg dela za odpravljanje napak.

Podatki se določijo ob zaključku projekta. Tedaj se zberejo tudi naslednji podatki, ki se nanašajo na posamezne faze (in cikle):

- razmerje med velikostjo procesnega in podatkovnega dela rezultatov posameznih faz,
- razmerje obsega dela po posameznih fazah in
- razmerje dodatnih stroškov posameznih faz.

Te rezultate upoštevamo pri združenih rezultatih vseh projektov na organizacijski ravni, vendar jih na podoben način shranimo v tri tabele tudi v okviru projekta. Slika 5.19 prikazuje ER diagram, iz katerega so razvidni štiri entitetni tipi, v katerih so shranjeni opisani združeni rezultati projekta.



Slika 5.19: ER diagram za opis združenih rezultatov dokončanega projekta

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Iz slike je razvidno, da se razmerja določajo samo za tiste faze in izdelke, ki jih v okviru projekta izvajamo. Tabela 5.9 prikazuje pomen predstavljenih entitetnih tipov.

Entitetni tip	Opis
Rezultati projekta	Povzetek glavnih rezultatov za celoten projekt
Razmerja izdelkov faz	Razmerje med vrednostjo nekaterih osnovnih meritev in končno velikostjo
Razmerja obsega dela faz	V odstotkih izraženo razmerje potrebnega obsega dela za izvedbo faz in ciklov
Razmerja stroškov faz	V odstotkih izraženo razmerje dodatnih stroškov za izvedbo faz in ciklov

Tabela 5.9: Entitete za opis združenih rezultatov dokončanih projektov

5.4 Določanje vsebine merjenja

Meta model predstavlja razmerja med podatki o merjenju na različnih ravneh, naš cilj pa je določiti vsebino meta modela. Ta del je definiran v okviru obveznega dela merjenja na organizacijski ravni, oziroma v meta modelu v opisu merjenja na organizacijski ravni. Če določimo vsebino entitet tega dela meta modela, ustrezno opišemo merjenje. Pri tem opredelimo naslednje dele:

- nabor stalnih merskih izhodišč,
- nabor merskih entitet,
- nabor merljivih atributov,
- nabor osnovnih meritev,
- nabor združenih meritev,
- nabor izpeljanih meritev in
- nabor pokazateljev.

Posamezni projekti glede na svoje zahteve določajo še posebna merska izhodišča. Za lažjo povezavo teh s pokazatelji in meritvami je treba izhajati iz razredov meritev, ki so bili predstavljeni v razdelku 4.2.2.1. Naš pristop določa štiri razrede meritev:

- porabo časa in napredovanje,
- vire in stroške,
- velikost izdelka in
- kakovost izdelka.

Predstavljeni meta model vse dele ustrezno povezuje v celoto. Z izbiro primerne metode za določanje nabora meritev je mogoče poleg meritev učinkovito izpeljati tudi ustrezne nabore merskih izhodišč, pokazateljev in meritev. V našem primeru morajo ti upoštevati merski proces in opisane povezave med posameznimi deli meta modela. Poleg tega je pomembno, da z izbrano metodo na definiran, a čim učinkovitejši način opravimo to nalogo.

V razdelku 2.1.5 sta bili predstavljeni dve metodi, ki se običajno uporabljata za določanje nabora meritev: metoda Goal Question Metric [Alb83] in metoda Metric Question Goal [Het93]. V nadaljevanju predstavljamo, kako bi ju uporabili v našem primeru.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5.4.1 Izpeljava nabora meritev z uporabo metode GQM

Pri določanju nabora ustreznih meritev je eno najpomembnejših načel utemeljiti le-te na dejanskih potrebah in ciljih organizacije [Alb83, Fen97, Roz00, Ali02]. Brez tega obstaja verjetnost, da bomo določene potrebne vidike merjenja izpustili oziroma merili določene atribute, ki nikogar ne zanimajo. To je tudi formalna zahteva, ki izhaja iz modela CMM.

Na tem načelu temelji metoda GQM, ki danes velja za standarden način pri določanju nabora meritev (glej razdelek 2.1.5.1). Metoda implementira pristop od zgoraj navzdol (ang. top down). V skladu z definicijo metode GQM najprej določimo poslovne cilje organizacije, na podlagi teh merske cilje[27], vprašanja, pokazatelje in na koncu še meritve, s pomočjo katerih lahko presodimo, v kolikšni meri merske cilje dejansko dosegamo.

Tudi Fenton [Fen97] v svojem pristopu predlaga uporabo te metode za določanje smiselnega nabora meritev, vendar opozarja, da ni nujno, da so vse tako določene meritve tudi zares uporabne v izbrani organizaciji za razvoj PO. Dejanska uporaba določene meritve v organizaciji je odvisna od zrelosti organizacije (v skladu z modelom CMM) oziroma tega, kateri elementi (deli procesa, izdelki, viri) so v procesu razvoja PO organizacije zares sledljivi. Tako si lahko tudi organizacija na 2. stopnji modela CMM zastavi za cilj spremljanje kakovosti izvajanja usposabljanja, vendar njen proces (običajno) ni definiran do te stopnje, da bi obvladoval tudi to aktivnost oziroma bi bilo mogoče usposabljanje mersko spremljati. Poleg smiselnosti meritev se je zato treba zavedati tudi njihove izvedljivosti v organizaciji.

Če bi uporabili Fentonov pristop v našem primeru, bi najprej uporabili metodo GQM za določanje vseh naborov. Ker v našem primeru ne moremo izhajati iz poslovnih ciljev resnične organizacije, moramo te določiti glede na dano nalogo. Izhodišči za cilje sta tako dve:

- organizacija mora ustrezati zahtevam 4. stopnje modela CMM in
- organizacija je namenjena za razvoj IS.

Glede na prvo izhodišče lahko določimo poslovne cilje na podlagi zahtev posameznih ključnih področij druge, tretje in četrte stopnje modela CMM. Kot je predstavljeno v razdelku 2.2.6.2 je na vsakem ključnem področju definiranih nekaj ciljev, ki predstavljajo namen ključnega področja, in te lahko neposredno uporabimo kot izhodišče metode GQM. Pri določanju procesa razvoja PO v majhni organizaciji smo sicer izhajali iz prilagojenega modela CMM, vendar so cilji ključnih področij sodili v obvezen del modela in potemtakem veljajo tudi v prilagojenem modelu CMM.

Organizacija za razvoj IS ima specifičen tehnološki proces. Ta določa nekatere posebne poslovne cilje, ki bodo tudi vplivali na nabor smiselnih meritev. Predvsem velja pri tem izpostaviti dva vidika izdelkov (podatkovni in procesni).

Na podlagi poslovnih ciljev določimo merske cilje (oziroma merska izhodišča), ki so natančno definirana. Iz teh po postopku določimo vprašanja, pokazatelje in slednjič nabor meritev. Tak postopek je zahteven, vendar določen. Končni rezultat so dokaj obsežni nabori poslovnih ciljev, merskih ciljev (oziroma stalnih merskih izhodišč), pokazateljev in meritev, pa tudi nabor entitet in ustreznih merljivih atributov.

[27] Metoda GQM uporablja nekoliko drugačno terminologijo od naše, ki je utemeljena na standardu ISO/IEC 15939 in pristopu PSM. V našem primeru termin merski cilj približno ustreza merskemu izhodišču.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Drugi del Fentonovega pristopa zahteva ugotavljanje izvedljivosti izbranih meritev. S stališča zrelosti organizacije s tem ni težav, saj predstavljeni proces razvoja PO ustreza zahtevam 4. stopnje prilagojenega modela CMM.

Vendar pri določanju nabora meritev z metodo GQM nismo upoštevali majhnosti organizacije. Ta poleg poslovnih določa še splošen cilj: nabori merskih izhodišč, pokazateljev in meritev naj bodo maloštevilni, meritve enostavne in uporabne. Zahteva bi morala vplivati na obseg naborov, vendar tega cilja v okviru metode GQM moremo neposredno implementirati.

Vir težav je v obsežnosti modela CMM, ki naslavlja veliko ključnih področij, merskih ciljev in posredno tudi meritev. Svoje prispevajo tudi posebnosti organizacije za razvoj IS, ki definirajo specifične izdelke ali dele procesa, kar ima za posledico uporabo dodatnih meritev. Nekatere bodo sicer nadomestile meritve, ki smo jih določili z uporabo modela CMM, vendar se bo število meritev še vedno povečalo. S tem postane še večji že tako velik nabor meritev, ki so nastale ne podlagi modela CMM.

Glavna težava pa izhaja iz same metode GQM. Čeprav je ta za določanje nabora meritev priporočljiva, je njena slabost ravno veliko število predvidenih meritev [Het93]. Do tega pride zaradi večnivojskega drevesa izpeljave, ki ga tvorimo pri postopnem določanju meritev.

5.4.2 Izpeljava nabora meritev z metodo MQG

S stališča majhnosti organizacije je zato primerneje izhajati iz omejenih zmožnosti organizacije in uporabiti pristop od spodaj navzgor. V skladu z metodo MQG [Het93] (glej razdelek 2.1.5.2) je treba najprej določiti, katere so primerne merske entitete (izdelki, viri) in kaj vse lahko učinkovito merimo pri njih. V našem primeru bi tako na podlagi znanega procesa razvoja PO določili, kaj je mogoče na enostaven način meriti ali določiti. Ta nabor meritev bi preverili z zahtevami 4. stopnje modela CMM in organizacije za razvoj IS.

Vendar metoda MQG poslovnih ciljev organizacije ne upošteva kot izhodišče, temveč zgolj ugotavlja dejansko stanje z merjenjem vsega, kar je mogoče meriti. S tem odstopamo od osnovne ideje našega merskega pristopa: utemeljitve merjenja na informacijskih potrebah. Poleg tega v našem primeru ne želimo zares izvajati vseh možnih meritev, temveč samo manjši stalen nabor osnovnih meritev. Če bi izvajali vse smiselne meritve, bi določene atribute merili brez potrebe ali večkrat.

Tudi če bi nabor meritev po metodi MQG ustrezal, bi še vedno ostale težave z določanjem pokazateljev in merskih izhodišč. Ker metoda MQG ne vsebuje mehanizma za izpeljavo od zgoraj navzdol, bi morali nabore teh določiti na neformalen način, z ugibanjem. To za nas ne zadošča. Zaradi tega kljub obetavnemu začetku metoda MQG za nas ni primerna.

5.4.3 Izpeljava nabora meritev s kombinirano metodo

Čeprav se nismo odločili za uporabo nobene od opisanih metod je očitno, da imata obe metodi določene dobre lastnosti, ki jih je smiselno uporabiti. Pri metodi GQM je to utemeljitev merjenja na poslovnih ciljih organizacije ter definiran postopek za določanje naborov. Prednost metode MQG je realna izbira nabora entitet in učinkovitih meritev, ki je podlaga za vse ostalo. Zato predvidevamo, da je z združevanjem dobrih lastnosti mogoče zasnovati ti. **kombinirano metodo**, ki bi bila primerna za uporabo v našem primeru.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pri določanju kombinirane metode smo za izhodišče izbrali značilnost metode MQG: določitev nabora meritev, ki jih je glede na merske entitete smiselno izvajati. Ob koncu prejšnjega razdelka smo omenili, da metoda MQG predvideva uporabo kar vseh smiselnih meritev. Zato je naša prva naloga nabor meritev omejiti oziroma primerneje izbrati.

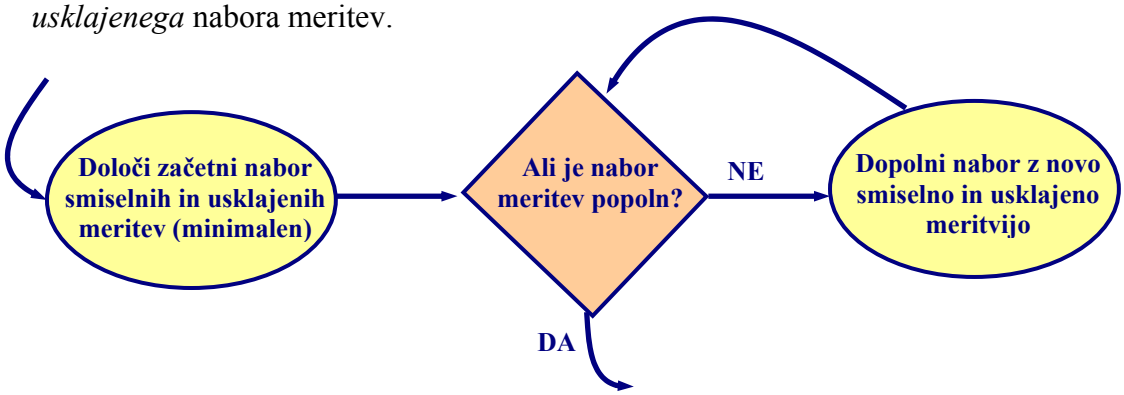
Določiti želimo (približno) minimalen nabor (osnovnih, združenih in izpeljanih) meritev, ki bo zadoščal zahtevam 4. stopnje prilagojenega modela CMM in organizacije za razvoj IS. V bistvu želimo sestaviti *smiselen*, *popoln*, *zadosten* in *usklajen* nabor meritev, kar so standardne zahteve za dober nabor meritev [Men00]:

- ***Smiselen*** (ang. sound) nabor meritev je tisti, ki je veljaven v okolju, kjer se uporablja.
- ***Popoln*** (ang. complete) nabor meritev meri vse, kar glede na zahteve mora meriti.
- ***Zadosten*** (ang. lean) nabor meritev meri samo tisto, kar je nujno in nič več.
- ***Usklajen*** (ang. consistent) nabor meritev ustreza ciljem organizacije, ki ga uporablja.

Z določanjem *smiselnih* in *usklajenih* posameznih meritev ni težav, ker lahko te določimo z uporabo dela metode MQG na podlagi izdelkov našega procesa razvoja PO. Težje je pri določanju *zadostnega* nabora, saj tega ne moremo določiti naenkrat, ampak postopoma, z več dopolnitvami na začetku izbranega minimalnega nabora *smiselnih* in *usklajenih* meritev, dokler ne zadostimo pogoju *popolnosti*.

Glavna težava je ravno s preverjanjem *popolnosti* nabora meritev. Pri tem preverjamo, če izbrani nabor meritev zagotavlja doseganje vseh ciljev in v kolikor ga ne, nabor dopolnimo z novo *smiselno* in *usklajeno* meritvijo in preverjanje ponovimo. Zahtevno in nedorečeno je zlasti preverjanje razmerja med posamezno meritvijo in cilji, ki kot taki trenutno še niso natanko definirani oziroma nam predstavljajo splošne cilje ključnih področij modela CMM.

Na sliki 5.20 je prikazan postopek določanja *smiselnega*, *popolnega*, *zadostnega* in *usklajenega* nabora meritev.

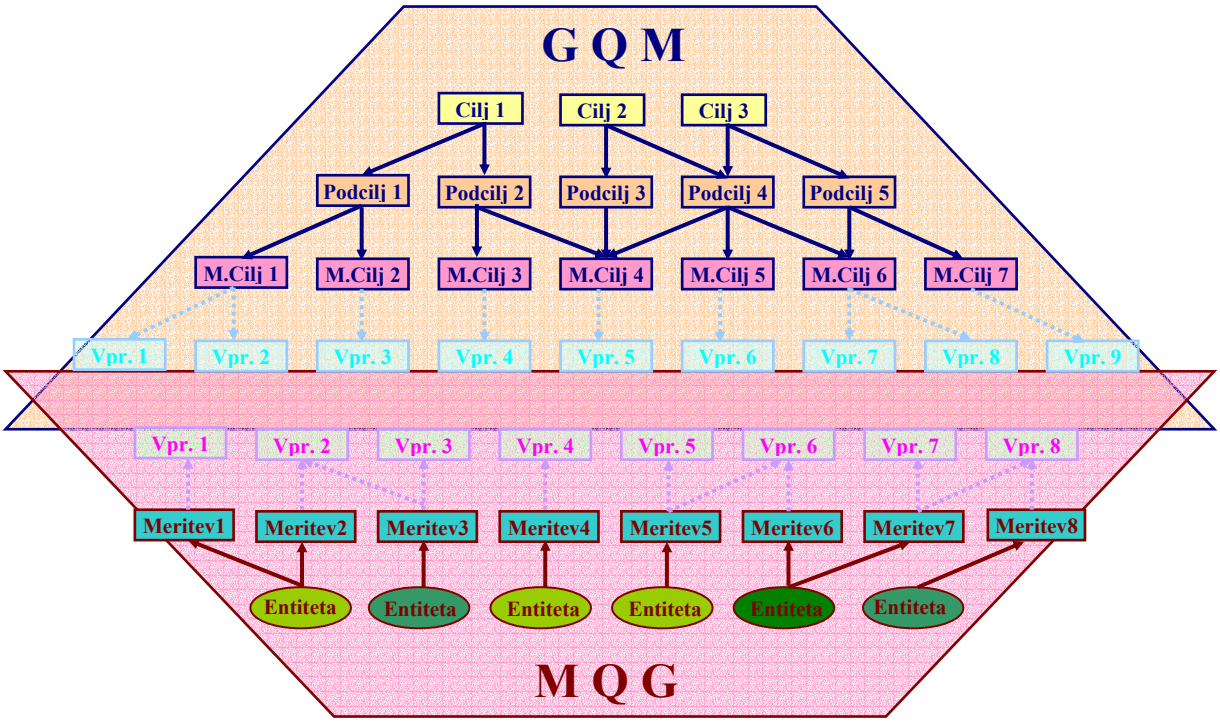


Slika 5.20: Postopek določanja primernega nabora meritev

Preverjanje razmerja med meritvami in cilji je veliko lažje, če cilje natančneje opredelimo. V ta namen je primerno izvesti nekaj korakov metode GQM in iz splošnih ciljev modela CMM določiti vsaj merske cilje, mogoče pa tudi vprašanja in pokazatelje. Na ta način se razkorak zmanjša in razmerje postane bolj očitno, s tem pa tudi naš postopek bolj določen.

Čeprav problem preverjanja razmerja med meritvami in cilji s tem še ni dorečen, pa lahko pritrdimo začetni predpostavki. Najprimernejši pristop za določanje nabora meritev združuje tako dobre značilnosti pristopa od spodaj navzgor (določanje meritev) kot dobre lastnosti od zgoraj navzdol (določanje merskih ciljev). Slika 5.21 kaže idejo kombiniranega pristopa.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 5.21: Osnovna ideja kombiniranega pristopa

Vendar moramo določanje nabora primernih meritev z uporabo obeh metod natančneje določiti. Določiti moramo optimalni meji, do kod vsako od metod izvajati in na kakšen način preverjati, če določena meritev zares ustreza izbranemu cilju.

Določanje nabora meritev s kombinirano metodo moramo tudi prilagoditi našemu pristopu. Naš pristop recimo ne predvideva hierarhije poslovni cilji – podcilji – merski cilji, temveč namesto tega določa samo merska izhodišča, ki so v skladu s pojmovanjem GQM nekoliko splošneje določeni merski cilji. Po drugi strani naš pristop določa več vrst meritev: osnovne, združene in izpeljane meritve, metodi GQM in MQG pa poznata samo meritve. Pri določanju kombinirane metode je treba vse dvoumnosti odpraviti. Tabela 5.10 prikazuje termine, ki jih bomo uporabljali v okviru kombinirane metode in njihovo razmerje s terminologijo GQM in MQG.

Kombinirana metoda	GQM	MQG	Opis
Cilj	Poslovni cilj	Cilj	Splošen cilj organizacije (recimo cilj KPA CMM)
-	Podcilj	-	Natančneje opredeljen cilj
Mersko izhodišče	Merski cilj	-	**Komb.metoda:** formalno opisano mersko izhodišče; **GQM:** splošnejši merski cilj, ki je formalno opisan
-	Vprašanje	Vprašanje	Odgovor na vprašanje pojasnjuje merski cilj
Pokazatelj	Pokazatelj	-	Ustrezno prikazane vrednosti več meritev, ki pojasnjujejo merski cilj
Izpeljana meritev	Meritev	-	Meritev več atributov entitete, ki jo dobimo z izpeljavo drugje izmerjenih vrednosti
Združena meritev	Meritev	Meritev	Meritev enega atributa entitete, ki jo dobimo z združevanjem drugje izmerjenih vrednosti
Osnovna meritev	Meritev	Meritev	Meritev enega atributa entitete, ki jo izvede posameznik
Entiteta	Entiteta	Entiteta	Merska entiteta (izdelki, viri, aktivnosti)
Atribut	Atribut	Atribut	Merljiv atribut

Tabela 5.10: Uporaba terminologije v različnih metodah

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Natančneje moramo opredeliti tudi celoten postopek določanja *smiselnega*, *popolnega*, *zadostnega* in *usklajenega* nabora meritev, ki ustrezajo našim zahtevam. Predvsem je treba doreči način ugotavljanja *popolnosti* ter kako poskrbimo za *zadostnost* nabora meritev.

Preden definiramo posamezne dele postopka si izberemo naslednje smernice:

- Načela odločanja naj bodo jasna:
  - nedvoumno definirani cilji in merska izhodišča;
  - nedvoumna odločitev, če je meritev povezana z merskim izhodiščem;
  - nedvoumna odločitev, če je mersko izhodišče popolno doseženo.
- Postopek minimizira zahtevnost:
  - postopek je enostaven in razumljiv;
  - postopek je relativno kratek;
  - izogibamo se zahtevnim korakom;
  - posplošujemo, kjer je mogoče.
- Postopek minimizira obseg:
  - težimo k majhnemu številu merskih izhodišč;
  - težimo k majhnemu številu osnovnih in združenih meritev.

Ko bomo z upoštevanjem smernic določili posamezne dele, moramo vse kamenčke sestaviti v mozaik, ki bo obsegal celovito metodo za določanje nabora primernih meritev. V okviru tega bomo vse dele povezali v smiselno zaporedje.

5.4.3.1 Uporaba metode GQM

S strani metode GQM izvajamo korake od ciljev do merskih izhodišč. To je nujen del – v kolikor bi določali še vprašanja, bi postopek postal prezahteven glede na prednosti, ki jih to prinaša. Pri vsakem merskem izhodišču bomo določili entiteto, namen, vidik in okolje.

Za poenostavitev postopka bomo za izhodišče vzeli prilagojen model CMM. Cilji ključnih področij sicer ostajajo isti, vendar je implementacija teh drugačna in to lahko olajša določanje merskih izhodišč. Poleg tega se pri ciljih ne bomo spuščali v podrobnosti, temveč bomo v skladu z določili prilagojenega modela CMM uporabljali smiselno presojo. Skušali bomo definirati majhno število merskih izhodišč, tako da bodo ta po pojmovanju GQM splošnejša od merskih ciljev, vendar dovolj natanko opisana, da bomo lahko preverili usklajenost.

Poleg tega bomo za cilje modela CMM, ki se ponavljajo v vsakem ključnem področju (recimo spremljanje statusa aktivnosti področja) določili merska izhodišča samo enkrat. Zdi se nam tudi smiselno, da ohranjamo cilje posameznih ključnih področij skupaj, enako tudi cilje posameznih stopenj. Na tak način bo mogoče določati tudi primerne meritve po posameznih področjih oziroma stopnjah.

5.4.3.2 Uporaba metode MQG

Po metodi MQG bomo izvedli en sam korak – določili meritve izbranim atributom izdelkov, virov in aktivnosti. Pri tem ne bomo določili vseh smiselnih meritev in v prvem koraku tudi ne vseh potrebnih meritev. Na začetku bomo določili samo osnovne in nekaj združenih meritev za nekaj glavnih atributov entitet. To so tiste meritve, ki jih gotovo moramo izvajati.

Vse izdelke procesa za razvoj PO, vire in aktivnosti bomo najprej jasno definirali kot merske entitete na podlagi procesa razvoja PO. V razdelku 4.2.2.1 smo definirali razrede meritev, ki

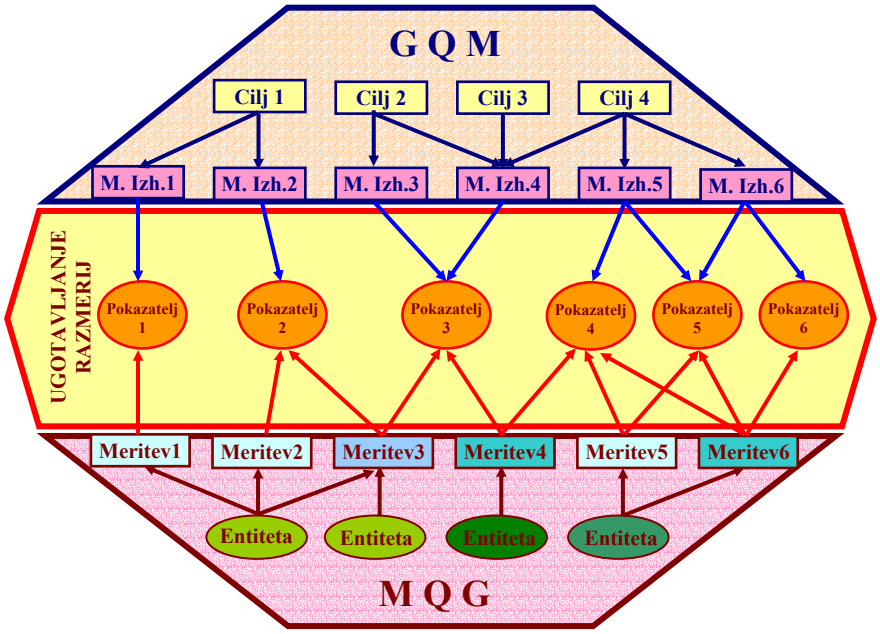
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

določajo paleto glavnih značilnosti, ki jih merimo. To so tiste značilnosti, ki jih gotovo moramo meriti v vsaki entiteti, če je to mogoče. Na podlagi teh lahko vsaki entiteti pripišemo nabor glavnih atributov, iz teh pa majhen nabor osnovnih in združenih meritev, ki jih gotovo moramo zbirati. Za vsak atribut bomo predvideli samo eno meritev. Vse meritve bomo kratko predstavili z glavnimi značilnostmi: entiteto, atributom in enoto. Ob tem bomo določili še to, ali gre za osnovno meritev, ki jo bodo posamezniki zares izvajali, ali bo meritev združena in se bo določala na podlagi drugih izmerjenih vrednosti.

Praktično gledano bomo z uporabo metode MQG v začetni nabor uvrstili samo (vse) osnovne in nekatere (očitno potrebne) združene meritve ter nobene izpeljane meritve. Te bomo dodajali postopoma ob preverjanju popolnosti nabora, ko bo to potrebno. Pri tem je bistveno, da se nabor osnovnih meritev med preverjanjem popolnosti ne bo širil, saj predvideva merjenje vsega, kar je z osnovnimi meritvami smiselno meriti. Dodajale se bodo samo združene in izpeljane meritve. Na ta način želimo ohraniti čim manjši nabor meritev. Pri izbiri osnovnih meritev upoštevamo tudi izkušnje drugih (glej razdelek 2.1.6.2), recimo predlagane nabore meritev [Car92, SEL95].

5.4.3.3 Ugotavljanje razmerja med meritvami in merskimi izhodišči

Ugotavljanje razmerja je tisti del postopka, ki definira praznino med izbranimi meritvami in izpeljanimi merskimi izhodišči. Natančneje gledano definira način preverjanja, če izbrana meritev izpolnjuje zahteve opisanega merskega izhodišča. Slika 5.22 prikazuje postopek ugotavljanja razmerja med meritvami in merskimi izhodišči.



Slika 5.22: Ugotavljanje razmerja med meritvami in merskimi izhodišči

Ugotavljanje obsega tri korake:

- primerjavo entitet merjenja,
- primerjavo razredov meritev in
- določitev povezave (pokazatelja).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Primerjava entitet merjenja omogoča primerjavo meritve in merskega izhodišča na formaliziran način. Entiteta merjenja je tako pri meritvah kot pri merskih izhodiščih vedno določena: pri meritvah natančno, pri merskih izhodiščih pa običajno splošneje. Ujemanje obeh entitet je potreben pogoj za to, da meritev ustreza merskemu cilju. Ujemanje obstaja, ko sta entiteti enaki ali (pogosteje) je entiteta meritve podmnožica entitete, ki ga določa merski cilj.

Primerjava razredov meritev tudi omogoča direktno primerjavo merskega izhodišča in meritve. Vsako mersko izhodišče lahko razvrstimo v enega izmed štirih razredov, na podlagi katerih so določene tudi osnovne in večina združenih meritev. Če direktna povezava obstaja, se morata razreda meritve in merskega izhodišča ujemati. Recimo: mersko izhodišče, ki zahteva spremljanje stroškov razvoja, se nanaša na razred viri in stroški. Ena ključnih meritev za to bo gotovo merjenje dodatnih stroškov posameznih aktivnosti, ki očitno sodi v isti razred.

V kolikor to ni res, bo potrebno dodati naboru meritev novo izpeljano meritev, za katero bo primerjava uspešna. Eno izmed merskih izhodišč recimo zahteva spremljanje produktivnosti posameznih aktivnosti. Mersko izhodišče sodi v razred viri in stroški. V ta razred sodi tudi izpeljana meritev za produktivnost, ki se določa iz razmerja med velikostjo rezultata (izdelka) in porabljenim časom. Meritev velikosti sodi očitno v razred velikost izdelka, meritev porabe časa pa v razred poraba časa in napredovanje.

Ker so razredi meritev samo štirje, je žal primerjava dokaj ohlapna. Zato primerjavo razredov meritev uporabljamo v povezavi s tretjim korakom: določitvijo povezave.

Določitev povezave je način, s katerim nedvoumno pokažemo razmerje med meritvijo in merskim izhodiščem. Če povezava obstaja, mora med meritvijo in merskim izhodiščem obstajati (vsaj en) pokazatelj. V kolikor se pokaže, da pokazatelja ne moremo določiti, je treba nabor meritev dopolniti z novo združeno ali izpeljano meritvijo. Tako pokazatelji kot združene ali izpeljane meritve so določene na podlagi več drugih meritev, vendar pokazatelji upoštevajo okoliščine in zahteve merskega izhodišča. S tega stališča so združene in izpeljane meritve vmesni korak med osnovnimi meritvami in pokazateljem, čeprav so vse tri vrste enakovredno navedene v naboru meritev.

Pokazatelje določimo na podlagi smiselne presoje razmerja. Smiselna presoja ugotovi pomen razmerja na podlagi celovitega poznavanja zahtev izbranega merskega izhodišča, značilnosti meritve in okolja.

Pokazatelje formalno zapišemo. Če so pokazatelji primerni, morajo ustrezati zahtevam merskega izhodišča, vsebovati pa morajo tudi izbrano meritev. Pokazatelj običajno vključuje tudi druge meritve. Med merskim izhodiščem in meritvijo lahko določimo tudi več pokazateljev, vendar za dokaz razmerja zadošča eden.

Pri ugotavljanju razmerja med meritvami in merskimi izhodišči najprej preverimo merske entitete. V kolikor se te ne ujemajo, potem razmerja med njima ne more biti. V nasprotnem primeru določimo razrede meritev, ki tudi predstavljajo zgolj potreben pogoj. Slednjič presodimo še pomen razmerja med meritvijo in merskim izhodiščem. V kolikor ustrezno razmerje obstaja, lahko na njegovi podlagi določimo ustrezen pokazatelj.

Za odločitev o tem, če med pokazateljem in meritvijo obstaja neposredna povezava ali je vmes še katera združena ali izpeljana meritev, uporabimo določanje razredov meritev. Ker so razredi ohlapno merilo, moramo uporabiti še smiselno presojo. V smislu zmanjševanja

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

velikosti nabora meritev bi na ta način lahko določene izpeljane ali združene meritve tudi izpustili, kar bi imelo za posledico zahtevnejše pokazatelje. Bolje je, če vmesne meritve določimo in s tem naredimo opis merjenja razumljivejši.

5.4.3.4 Preverjanje popolnosti

Preverjanje popolnosti definira, če izbran nabor meritev **izpolnjuje** vsa merska izhodišča. V ta namen se v postopku preveri vsako mersko izhodišče posebej in s pomočjo ugotavljanja razmerij preveri, če je to izpolnjeno. V kolikor so izpolnjena vsa merska izhodišča, je nabor popoln. V nasprotnem primeru moramo nabor meritev dopolniti z novo združeno ali izpeljano meritvijo in preverjanje popolnosti ponoviti.

Pri tem bi opozorili na dve stvari. Kot prvo moramo pri preverjanju izpolnjenosti merskega izhodišča natančno definirati, kaj to pomeni. Mersko izhodišče je izpolnjeno takrat, ko:

- obstaja ena ali več meritev, ki izpolnjuje(jo) mersko izhodišče, in velja, da sta entiteta te(h) meritve in entiteta merskega izhodišča enaki ali
- obstaja ena ali več meritev, ki izpolnjuje(jo) mersko izhodišče in velja, da je unija vseh entitet te(h) meritev enaka entiteti merskega izhodišča.

Preprosteje rečeno to pomeni, da mora biti vsako mersko izhodišče doseženo vsaj z eno meritvijo za vsako entiteto, ki jo naslavlja.

Kot drugo moramo opozoriti na težavo, ki izhaja iz ponavljajočega postopka preverjanja popolnosti. Vsako preverjanje popolnosti obsega množico ugotavljanj razmerij med merskimi izhodišči in meritvami. Ko neko razmerje prvič potrdimo, zanj določimo pokazatelja in v kolikor se meritev ali mersko izhodišče ne spremeni, je ob naslednjih ponovitvah dovolj, da samo ugotovimo obstoj ustreznega pokazatelja.

5.4.3.5 Preverjanje zadostnosti

Pri preverjanju zadostnosti skušamo določiti takšen nabor meritev, ki bo izpolnjeval vsa merska izhodišča, vendar ne bo obsegal odvečnih meritev. V ta namen na začetku določimo minimalen nabor meritev, za katere vemo, da jih gotovo moramo meriti.

Izhajamo iz tega, da bo pri vsaki entiteti treba meriti določene značilnosti (vsaj na 4. stopnji modela CMM). Ker predvidevamo izvajanje osnovnih meritev v okviru vsake aktivnosti posameznika, bo začetni minimalni nabor obsegal vse osnovne meritve in tiste združene meritve, ki merijo glavne značilnosti večjih entitet. Tak nabor zagotovo ne bo popoln, saj mu manjkajo vsaj še nekatere izpeljane meritve. S temi in združenimi meritvami bomo po potrebi postopoma dopolnjevali nabor.

Določeno meritev bomo dodali samo v primeru, če ta dosega mersko izhodišče, ki pred tem ni bilo doseženo. Pri tem moramo biti pozorni na to, da lahko na novo dodana meritev tudi nadomesti katero izmed meritev v naboru. Meritve, ki se nanašajo na iste entitete, med seboj namreč niso neodvisne. Postopek dodajanja novih meritev bo bolj učinkovit, če bomo pri izbiri upoštevali izkušnje drugih in smiselno presojo na podlagi merskih ciljev.

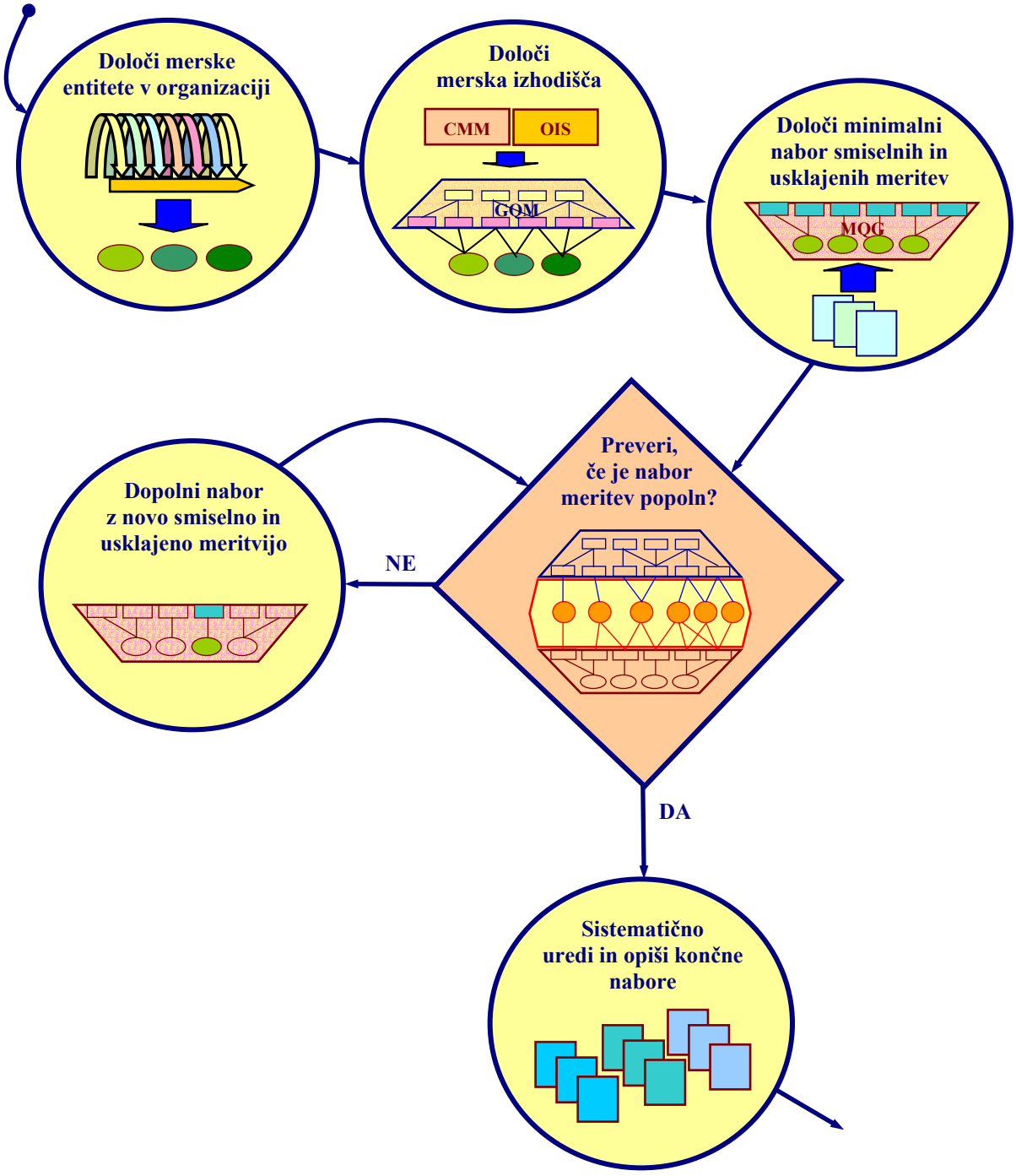
Za cilj smo si zadali določiti zadosten nabor meritev, ker želimo zaradi majhnosti organizacije čim enostavnejši merski proces. Vendar je zadostnih naborov meritev več: nekateri obsegajo zahtevnejše meritve od drugih. Včasih je z vidika uporabnosti tudi primerno dodati meritev, ki sicer ni nujno potrebna, vendar bolje predstavi določene značilnosti entitete. S tega stališča pri

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

zahtevi zadostnosti ne bomo vztrajali za vsako ceno, temveč bomo s praktičnega stališča skušali doseči približno zadostnost.

5.4.3.6 Celoten postopek določanja nabora meritev

Vse opisane dele združimo v celotno kombinirano metodo, ki jo prikazuje tudi slika 5.23.



Slika 5.23: Celotna struktura kombinirane metode

Kombinirana metoda tako obsega naslednje dele:

- Določanje merskih entitet (aktivnosti, virov, izdelkov) in njihovih glavnih atributov z uporabo načel metode MQG: na podlagi procesa razvoja PO;

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Določanje merskih izhodišč z uporabo dela metode GQM: na podlagi ciljev modela CMM in organizacije za razvoj IS;
- Določanje minimalnega nabora smiselnih in usklajenih (osnovnih in nekaterih združenih) meritev z uporabo dela metode MQG in z upoštevanjem tujih izkušenj;
- Preverjanje popolnosti nabora meritev z uporabo našega pristopa za ugotavljanje razmerij med merskimi izhodišči in meritvami za vsako mersko izhodišče posebej;
- **Dokler** nabor meritev ni popoln **izvajaj**:
  - dopolnjevanje nabora z novimi smiselnimi in usklajenimi (združenimi ali izpeljanimi) meritvami **ter**
  - preverjanje popolnosti nabora meritev.
- Sistematično urejanje in opisovanje končnih naborov:
  - nabor entitet in njihovih glavnih atributov,
  - nabor (stalnih) merskih izhodišč,
  - nabor pokazateljev in
  - nabor osnovnih, združenih in izpeljanih meritev.

Ob koncu želimo opozoriti še na termin »metoda za določanje meritev«. Ta ne odraža najbolje značilnosti naše metode. Uporabljamo ga predvsem zaradi metode GQM, katere končni rezultat je zares nabor meritev, čeprav spotoma določimo tudi druge nabore. S tega stališča je končni rezultat kombinirane metode nabor ustreznih pokazateljev, drugi nabori pa so vmesni rezultati. Poleg pokazateljev v iterativnem delu določimo samo združene in izpeljane meritve, medtem ko so ostali nabori fiksni. Stalen nabor osnovnih meritev omogoča dodatno funkcionalnost, ki je pomembna prednost našega merskega pristopa.

V naslednjem podpoglavju bomo prikazali rezultate uporabe kombinirane metode.

5.5 Vsebina merjenja

Z uporabo metode določimo vse pomembne dele vsebine merjenja na organizacijski ravni. To je tudi končni rezultat disertacije, zato smo ta del izpostavili v posebnem podpoglavju.

Opis merjenja za majhne organizacije za razvoj IS določajo nabori meta podatkov, ki so predstavljeni v podpoglavju 5.3. Mi si jih bomo ogledali v štirih delih:

- nabor stalnih merskih izhodišč,
- nabor entitet in atributov,
- nabor osnovnih in dela združenih meritev in
- nabor preostanka združenih meritev, izpeljanih meritev in pokazateljev.

Vse nabore bomo vsebinsko predstavili ter kratko prikazali uporabo tistega dela kombinirane metode, ki do takšnega rezultata privede. Celotni rezultati so zbrani v dodatku B.2.

5.5.1 Nabor stalnih merskih izhodišč

Kombinirana metoda določa, da nabor merskih izhodišč določimo na podlagi:

- omejitev in zahtev majhnih organizacij za razvoj IS in
- ciljev vseh ključnih področij četrte stopnje modela CMM.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Omejitve in zahteve zaradi majhnosti organizacij in posebnosti razvoja IS se že odražajo v več elementih, ki vplivajo na izbiro nabora merskih izhodišč:

- v prilagodljivem procesu razvoja PO,
- v izbranem merskem procesu, ki je del procesa razvoja PO,
- v meta modelu in
- v kombinirani metodi za določanje vseh naborov.

Uporaba teh elementov že sama po sebi vpliva na izbiro nabora, ki ustrezno upošteva značilnosti tovrstnih organizacij. Z uporabo prilagodljivega procesa razvoja PO mersko spremljanje aktivnosti dejansko pomeni merjenje ciklov, faz in korakov prilagodljivega procesa, medtem ko merjenje značilnosti izdelkov pomeni merjenje rezultatov tega procesa, recimo tudi podatkov v PB. Merski proces predvideva izvajanje merjenja ob vsaki aktivnosti posameznika, združevanje rezultatov ter majhnim organizacijam prilagojene analize, kar ustrezno podpira tudi meta model. Uporaba kombinirane metode predvideva majhen nabor splošnejših merskih izhodišč, kar ustreza zahtevam majhne organizacije.

V tem delu težko govorimo o specifičnih merskih izhodiščih, temveč bolj o splošnih ciljih, ki jih moramo upoštevati pri določanju vseh naborov, ne zgolj nabora merskih izhodišč. Na ravni nabora merskih izhodišč se namreč podrobnosti v glavnem še ne vidi; te bodo prišle bolj do izraza ob določanju posameznih meritev.

Zato bomo na podlagi zahtev in omejitev majhnih organizacij za razvoj IS določili splošne cilje, ki jih bomo najprej posredno upoštevali pri določanju nabora merskih izhodišč na podlagi zahtev ključnih področij modela CMM.

V razdelku 2.3.3 smo določili tri posebnosti organizacije za razvoj IS: specifičen tehnološki proces, specifične vmesne rezultate razvoja in velik pomen avtomatizacije. Vse so povezane s tehnološko izvedbo, zato tudi vplivajo predvsem na merjenje tega dela. Na izbiro merskih izhodišč vplivajo naslednje značilnosti:

a) Drugačna je sestava tehnološkega procesa. Tudi če odmislimo prilagodljiv proces razvoja PO s cikličnim pristopom, so pri razvoju IS drugačne aktivnosti, recimo faze. Tudi pomen oziroma razmerje faz ni enako, saj recimo velik pomen pripisujemo raziskavi zahtev, medtem ko je izvedba načrtovanja v veliki meri predpisana.

   Cilj: Upoštevati je treba drugačno strukturo procesa razvoja PO in pri merjenjih ustrezno podpreti posamezne dele razvoja.

b) Proces razvoja PO obsega najprej dejanski razvoj, nato pa še uvajanje končnih izdelkov v uporabo. Enako kot mersko spremljamo aktivnosti razvoja PO, moramo spremljati tudi uporabo le-tega. Pri tem bodo v drugem delu pomembnejša merjenja raznih vidikov kakovosti končnega izdelka in ne več procesne meritve.

   Cilj: Na ustrezen način moramo meriti tako aktivnosti razvoja kot aktivnosti uvajanja rezultatov v uporabo.

c) Tudi pri izdelkih ločimo dve stopnji: najprej jih razvijemo (PB ali IS), nato še pripravimo za uporabo (PB napolnimo ali IS nastavimo parametre). Meritve morajo obsegati oba dela.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Cilj: Razlikovati moramo med merjenjem samih izdelkov in merjenjem dela izdelkov, ki slednje pripravi za namestitev.

d) Izdelki so sestavljeni iz podatkovnega in procesnega dela. Oba dela moramo ločeno meriti, vendar po drugi strani tudi upoštevati kot celoto.

   Cilj: Merjenje mora omogočati spremljanje procesnega in podatkovnega vidika izdelkov.

e) Določene aktivnosti so posebej pomembne (preverjanja z naročnikom, uporabniško testiranje, ipd.), kar naj se odraža tudi pri merjenju. Tem aktivnostim določimo poseben pomen in jih posebej spremljamo in preverjamo. Tako jih recimo določimo za mejnike (in spremljamo doseganje le-teh) ali posebej merimo zahteve, ki izhajajo iz takšnih srečanj.

   Cilj: Za pomembne aktivnosti predvidimo posebna merjenja.

f) Med razvojem nastanejo vmesni izdelki, ki so med seboj smiselno povezani. Pri teh vmesnih izdelkih lahko predvidimo merjenja enakih značilnosti na različne načine. Na podlagi baze merskih podatkov kvantitativno določimo razmerje med izmerjenimi vrednostmi, kar nam omogoča recimo učinkovito napovedovanje.

   Cilj: Merjenja morajo odražati povezave med določenimi vmesnimi izdelki in te povezave uporabljati pri analizah.

g) Pri razvoju PO so pomembna programska orodja, ki podpirajo določene vrste vmesnih izdelkov. Prilagodljiv proces razvoja PO predvideva tovrstne vmesne izdelke. Ker orodja uporabljamo tudi za merjenje, moramo merjenje smiselno prilagoditi zahtevam teh orodij.

   Cilj: Merjenja naj upoštevajo standardne pristope, ki jih uporabljajo programska orodja.

Naj k temu dodamo še zadnji splošni cilj, ki izhaja iz zahtev majhne organizacije:

Cilj: Merjenje naj bo prilagojeno zahtevam in zmožnostim majhne organizacije.

Vsi splošni cilji so prikazani v dodatku B.2.3.1. Upoštevamo jih pri določanju merskih izhodišč **na podlagi zahtev četrte stopnje modela CMM**, ki je drugo izhodišče za določanje nabora merskih izhodišč. V nadaljevanju bomo kratko predstavili značilnosti merjenja na vsaki stopnji modela CMM, ki se odraža tudi v naboru merskih izhodišč. To je potrebno zaradi tega, ker pri določanju merskih izhodišč poleg ciljev smiselno presojamo tudi ostale okoliščine. Kot primer bomo v skladu s kombinirano metodo izpeljali merska izhodišča za ključno področje Upravljanje z zahtevami, merska izhodišča ostalih področij pa okvirno opisali. Nabor merskih izhodišč je v celoti predstavljen v dodatku B.2.3.

Površen pregled vsebine ključnih področij posameznih stopenj modela CMM daje napačen vtis, da je merjenje zares uvedeno šele na četrti stopnji zrelosti modela CMM [Wie96]. Dejansko je ravno obratno – merjenje je sestavni del vseh stopenj zrelosti, le da je na različnih stopnjah uvedeno z različnimi nameni: na drugi stopnji recimo zgolj za spremljanje, na tretji za primerjavo, na četrti za (statistični) nadzor in na peti stopnji za izboljšavo [Fen97, Pfl90]. V skladu s tem so določena tudi merska izhodišča: na drugi stopnji so namenjena spremljanju, na tretji spremljanju in primerjavi ter na četrtem spremljanju in nadzorovanju. Merjenje v okviru modela CMM si lahko bralec ogleda v [Ray99, Lay00, Wel01].

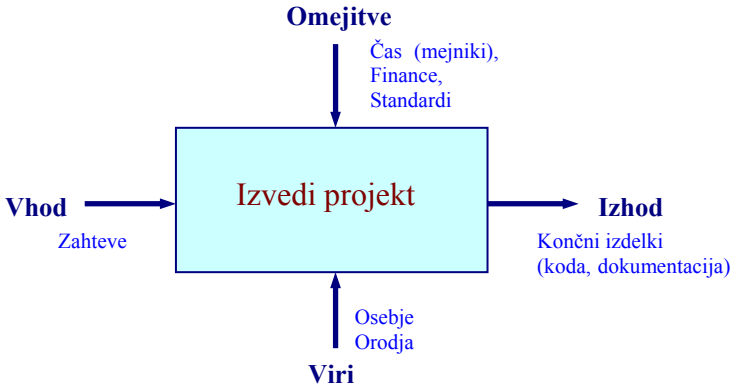
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Da vsako ključno področje obsega nekaj merjenj kaže skupina aktivnosti Merjenje in analiza (razdelek 2.2.6.2), ki za vsako ključno področje določa vsaj merjenje statusa izvajanja aktivnosti.

Določanje nabora merskih izhodišč poteka po posameznih stopnjah in ključnih področjih modela CMM. Pri tem začnemo z drugo stopnjo, saj je proces na prvi stopnji modela CMM nedefiniran. Na tej stopnji uvajanje meritev ni smiselno, temveč je primerno najprej definirati proces na ustrezen način.

5.5.1.1 Druga stopnja modela CMM: Ponovljiv proces

Procesu razvoja PO na drugi stopnji zrelosti lahko določimo samo vhode (zahteve), izhode (končne izdelke), omejitve (recimo čas razvoja, finance in standarde), vire (razvijalce in orodja) in glavne mejnike, ostalih podrobnosti pa iz opisa procesa ne moremo razbrati (slika 5.24). S tega stališča proces omogoča ponoviti enak izhod, če ga izvajamo z enakimi vhodi, omejitvami in viri. Ključna področja druge stopnje so namenjena predvsem uvedbi ustreznega vodenja projektov, zato so tudi merska izhodišča predvsem s tega področja. Obsegajo spremljanje zahtev, končnih izdelkov (velikosti, kakovosti), virov (količine), stroškov in poteka dela.



Slika 5.24: Merjenje na drugi stopnji modela CMM

Druga stopnja modela CMM omogoča spremljanje dejanskih vrednosti v primerjavi s planiranimi. Zato na tej stopnji lahko samo ugotavljamo doseganje zastavljenih ciljev z vidika izvajanja projekta, ne moremo pa izvajati primerjav ali celo presojati o vplivu določenih izmerjenih vrednosti na ostale atribute.

Upravljanje z zahtevami

Namen ključnega področja upravljanje z zahtevami je popolno in enolično razumevanje naročnikovih zahtev s strani razvijalcev projekta. Dogovor z naročnikom je podlaga za planiranje in vodenje projekta. Ker se zahteve lahko spreminjajo, je treba te spremembe upoštevati pri razvoju. Za Upravljanje z zahtevami sta določena dva cilja:

Cilj 1: *Zahteve sistema glede PO so pod nadzorom in predstavljajo izhodišče tako za tehnično realizacijo kot za vodenje projekta.*
Cilj 2: *Plani razvoja, izdelki in aktivnosti so konsistentni z zahtevami.*

Prvi cilj zahteva spremljanje statusa zahtev in spremljanje usklajenosti zahtev s končnimi izdelki (tehnična realizacija) in plani (vodenje projekta). Podobno zahteva tudi drugi cilj. Na podlagi vsakega cilja lahko izpeljemo vsaj eno mersko izhodišče, pri čemer moramo upoštevati tudi okoliščine. Lažje je, če v skladu z metodo GQM cilja razdelamo na podcilje. Primerni podcilji so:

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Spremljaj status posameznih zahtev (veljavne, spremenjene);
- Spremljaj spremembe zahtev med razvojem;
- Preverjaj usklajenost zahtev z ostalimi razvojnimi dokumenti (specifikacijami, načrti);
- Preverjaj usklajenost zahtev s projektnimi plani.

Raven podrobnosti podciljev približno ustreza ravni merskih izhodišč, poleg tega pa je podciljem lažje določiti merske entitete.

Spremljanje statusa zahtev ustreza prvemu merskemu izhodišču, ki se očitno nanaša na seznam zahtev. Ker na drugi stopnji še ne moremo preverjati usklajenosti z različnimi dokumenti, bomo usklajenost preverjali kar na podlagi statusa in sprememb zahtev. S tem zadnja dva podcilja združimo z drugim podciljem v drugo mersko izhodišče. To se ujema tudi z zahtevami metode po majhnem številu splošnejših merskih izhodišč. Poleg tega vsako ključno področje zahteva tudi spremljanje statusa izvajanja aktivnosti, kar določa tretje mersko izhodišče. V našem primeru spremljamo izvajanje aktivnosti za upravljanje z zahtevami, ki so sestavni del faze 0.4.

Merska izhodišča so predstavljena z entiteto, namenom, vidikom in okoljem, kot pri primerih na sliki 5.25. V dodatku B.2.3 so merska izhodišča predstavljena bolj strnjeno, s kratkim opisom namena in mersko entiteto.

```
2.1 Upravljanje z zahtevami

Mersko izhodišče 1
Entiteta:   seznam zahtev
Namen:      spremljaj status zahtev (število zahtev, število realiziranih zahtev ipd.)
Vidik:      vidik vodje projekta
Okolje:     organizacija za razvoj inf. sistemov, 2. stopnja CMM, orodje Oracle, projekt X.

Mersko izhodišče 2
Entiteta:   seznam zahtev
Namen:      spremljaj spremembe zahtev (predvsem z vidika realizacije)
Vidik:      vidik vodje projekta
Okolje:     organizacija za razvoj inf. sistemov, 2. stopnja CMM, orodje Oracle, projekt X.

Mersko izhodišče 3
Entiteta:   aktivnosti upravljanja z zahtevami
Namen:      spremljaj status aktivnosti upravljanja z zahtevami
Vidik:      vidik vodje projekta
Okolje:     organizacija za razvoj inf. sistemov, 2. stopnja CMM, orodje Oracle, projekt X.
```

Slika 5.25: Primer merskih izhodišč za ključno področje modela CMM

Planiranje projektov

Namen planiranja projektov je vzpostavitev razumnih planov za tehnično realizacijo razvoja PO in vodenje projekta. Merska izhodišča za planiranje so zato povezana z ocenjevanjem opravil in potrebnega obsega dela projekta ter s spremljanjem aktivnosti planiranja.

Zasledovanje in nadzor nad projekti

Namen zasledovanja in nadzora nad projekti je zagotovitev vpogleda v dejanski potek dela na projektu. To je podlaga za ukrepanje vodstva v primeru večjih odstopanj od plana. To ključno področje je na drugi stopnji modela CMM najbolj neposredno povezano z merjenjem. Merska izhodišča določajo spremljanje značilnosti projekta (poteka dela in težav), končnih izdelkov (velikosti) in virov (obsega dela). Enako kot v drugih primerih moramo spremljati tudi status izvajanja aktivnosti tega ključnega področja.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Urejanje razmerij s poddobavitelji

Namen urejanja razmerij s poddobavitelji je izbor ustreznega dobavitelja in učinkovit nadzor nad njegovim delom. Za majhno organizacijo sodelovanje s poddobavitelji ni smiselno, zato naš proces ne pokriva tega področja.

Zagotavljanje kakovosti PO

Namen zagotavljanja kakovosti PO je omogočiti primerno stopnjo vpogleda v proces razvoja PO in razvijajoče izdelke. Druga stopnja modela CMM še ne zagotavlja vpogleda v posamezne izdelke, zato lahko zagotavljanje kakovosti spremljamo samo z vidika izvajanja ustreznih aktivnosti in težav, ki so pri tem odkrite. Merska izhodišča so zato vezana na preverjanja in preglede, ki jih izvajajo člani skupine SQA.

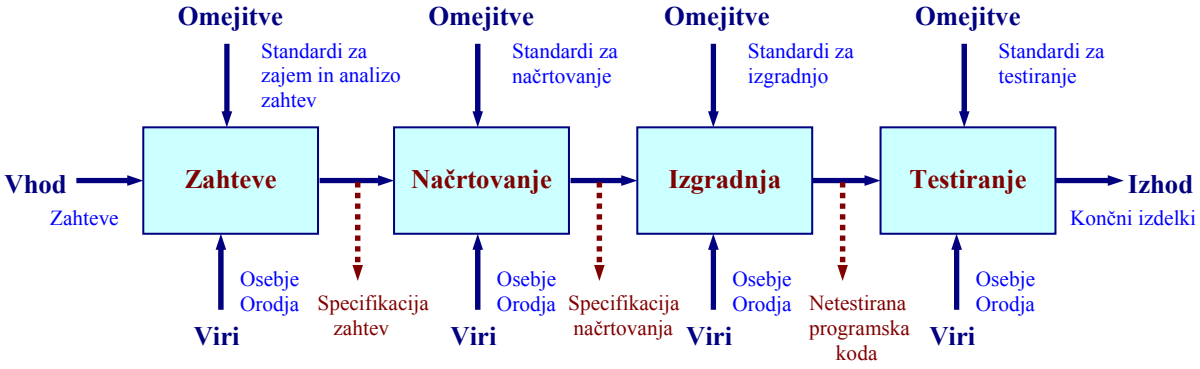
Upravljanje s konfiguracijo

Namen upravljanja s konfiguracijo je vzdrževanje integritete izdelkov v vseh delih življenjskega cikla. Na tem področju merska izhodišča predvidevajo spremljanje aktivnosti, števila sprememb in odstopanj, ki so se zgodila med izvajanjem projekta. Podobno kot pri vseh drugih ključnih področjih moramo tudi tu spremljati status vseh aktivnosti ključnega področja.

5.5.1.2 Tretja stopnja modela CMM: Definiran proces

Projektni proces razvoja PO na tretji stopnji po modelu CMM je natančno določen na podlagi organizacijskega procesa, kar odpira nove možnosti za merjenje. Če je bil na drugi stopnji proces še črna škatla, lahko sedaj razberemo posamezne dele procesa, njihove vhode, omejitve in izhode. Zato na tej stopnji merimo predvsem tehnološki vidik razvoja. Ker vhode in izhode običajno predstavljajo izdelki, na tretji stopnji merimo predvsem posamezne izdelke.

Izdelke ne merimo samo z vidika velikosti v ožjem smislu (število vrstic kode, število strani), temveč lahko merimo tudi velikost v širšem smislu (kompleksnost, strukturiranost). Podobno lahko opredelimo kakovost posameznih izdelkov, običajno s pomočjo števila odkritih napak v izdelku. Slika 5.26 prikazuje primer procesa s štirimi fazami – običajno jih bo seveda več.



Slika 5.26: Merjenje na tretji stopnji modela CMM

Na tej stopnji je določen enoten pristop merjenja za vse projekte. Ker proces dovolj natančno definira posamezne aktivnosti, zmoremo na podlagi zgodnjih rezultatov napovedovati značilnosti poznejših izdelkov. Recimo: slaba kakovost zahtev se odraža v slabem načrtu, zato je smiselno zahteve dodatno preveriti pred načrtovanjem. Čeprav na tej stopnji razvoj PO še ni voden s pomočjo merskih podatkov, nam definiran proces omogoča pravočasno odkrivanje večjih odklonov in ukrepanje. Tako lahko nadziramo kakovost in izboljšamo produktivnost.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Osredotočenje na proces, ki ga uporablja organizacija

Namen tega ključnega področja je vzpostavitev odgovornosti organizacije za proces razvoja PO, s čemer se omogoči izboljševanje zmožnosti organizacije. To ključno področje tretje stopnje modela CMM ni neposredno povezano z izdelki, zato so merska izhodišča povezana samo s spremljanjem aktivnosti za koordinacijo in izboljšanje procesa razvoja PO. Na tej stopnji lahko spremljamo tudi usklajenost z mejniki in stroške.

Definicija procesa razvoja PO organizacije

Namen definicije procesa organizacije je razvoj in vzdrževanje programske lastnine – množice programja, predlog standardov in dokumentacije organizacije, ki predstavlja podlago za dolgoročne izboljšave organizacije (glej razdelek 2.1.3.3). Tudi na tem področju lahko spremljamo predvsem status aktivnosti za razvoj in vzdrževanje enotnega procesa razvoja PO. Poleg tega tudi tu merska izhodišča določajo spremljanje obsega dela, usklajenosti z mejniki in stroškov.

Program izobraževanja

Namen programa izobraževanja je razviti spretnosti in znanja pri članih organizacije, da bi lahko učinkovito opravljali svojo vlogo v njej. Učinek izobraževanja se zares lahko oceni samo posredno (z doseženimi rezultati pri delu), zato je največ kar lahko neposredno merimo ustrezno izvajanje teh aktivnosti. Temu primerna so tudi merska izhodišča. Nadzor nad kakovostjo izobraževanja lahko ocenimo preko anket udeležencev izobraževanj.

Integrirano vodenje projektov razvoja PO

Namen tega ključnega področja je integracija upravljalskih in tehničnih aktivnosti v skladen proces, ki ustreza standardnemu procesu razvoja PO v organizaciji. Integrirano vodenje projektov razvoja PO je na tretji stopnji modela CMM najbolj mersko usmerjeno ključno področje. Predpisuje uvedbo centraliziranega merjenja za vse projekte in v ta namen predvidi skupno bazo merskih podatkov na ravni organizacije. Namen baze ni zgolj zbiranje usklajenih merskih podatkov o različnih projektih, temveč predvsem njihova uporaba za napovedovanje oziroma primerjavo s podatki konkretnih projektov.

Merska izhodišča določajo mersko spremljanje glavnih atributov vseh entitet. Tako določajo spremljanje velikosti izdelkov, obsega dela, porabe časa, virov, stroškov, napak in nenazadnje tudi statusa vseh aktivnosti s tega področja.

Tehnologija izdelkov PO

Namen tehnologije izdelkov PO je konsistentno izvajanje dobro definiranega procesa razvoja PO, ki predpisuje vse tehnične aktivnosti in učinkovito proizvaja pravilne izdelke PO. Dobro definiran tehnološki proces zahteva ustrezno mersko spremljanje izdelkov, ki pri tem nastanejo. Pri tem merimo funkcionalnost in kakovost izdelkov. Prvo spremljamo z vidika izvedbe posameznih zahtev, medtem ko kakovost merimo s številom napak. Na tem področju lahko določimo tudi merska izhodišča, ki so povezana s specifičnim (tehnološkim) procesom razvoja PO oziroma splošnimi cilji, ki odražajo značilnosti majhnih organizacij za razvoj IS.

Koordinacija med različnimi skupinami

Namen koordinacije med različnimi skupinami je skrb za koordinirano in nadzorovano sodelovanje razvojne skupine z drugimi tehničnimi skupinami, s čemer se doseže učinkovitejše projektno delo. V našem primeru ne predvidevamo razvojnih skupin znotraj projektne skupine, zato so merska izhodišča vezana na spremljanje aktivnosti, kjer sodelujejo člani skupin SQA in SQPG. Aktivnosti nadziramo preko stroškov, ki so zanje namenjeni.

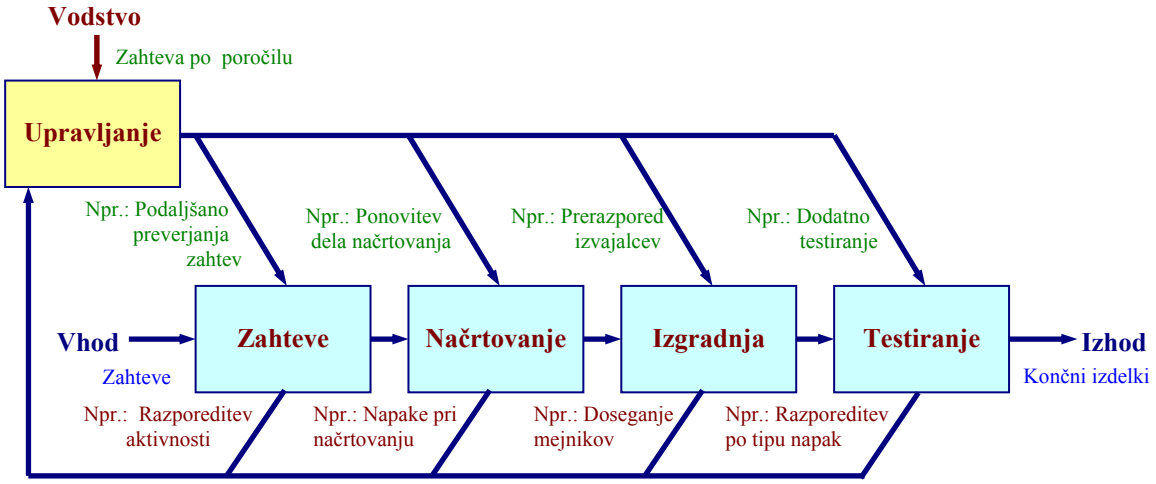
© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pregledi med kolegi

Osnovni namen pregledov med kolegi je zgodnje odkrivanje in odstranjevanje napak. Kot stranski učinek se razvije boljše razumevanje rezultatov dela in pomanjkljivosti v razvoju, ki se jih da odpraviti. Ključno področje pregledi določa eno mersko izhodišče za merjenje statusa tistih aktivnosti, ki v našem primeru nadomeščajo preglede med kolegi.

5.5.1.3 Četrta stopnja modela CMM: Upravljan proces

Četrta stopnja modela CMM nadgradi proces do te mere, da so razvidne vse podrobnosti razvoja. To nam omogoča meriti značilnosti procesa in le-te uporabljati za učinkovit nadzor procesa. Če je bilo na tretji stopnji mogoče odkrivati odklone in po potrebi ukrepati, je na četrti stopnji to osnovni način dela. Vse aktivnosti mersko spremljamo, statistično analiziramo s pomočjo baze merskih podatkov in se na podlagi ugotovitev odločamo, kako naprej. Pri tem ne spreminjamo osnovnih aktivnosti, temveč prerazporejamo vire, da bi s tem dosegli večjo učinkovitost in kakovost. Slika 5.27 prikazuje merjenje na četrti stopnji modela CMM.



Slika 5.27: Merjenje na četrti stopnji modela CMM

Upravljan proces meri vse značilnosti procesa in medsebojna razmerja vseh aktivnosti. Zaradi tega ima tudi baza merskih podatkov bogatejši nabor in omogoča temeljitejše odločanje na podlagi zahtevnejših, a bolj učinkovitih metod. Na ta način lahko proces nenehno usmerjamo in v vsakem trenutku statistično določimo, kakšen naj bo rezultat na koncu.

Kvantitativno vodenje procesov

Namen tega ključnega področja je kvantitativno spremljanje rezultatov procesa razvoja PO. Pri tem je poudarek na identifikaciji odstopanj od stabilnega procesa, ugotavljanju vzrokov zanje in njihovemu odpravljanju.

Osnova za mersko spremljanje je merski plan, ki temelji na organizacijskem procesu razvoja PO. Proces mersko spremljamo ter nadziramo s pomočjo statistične analize podatkov o aktivnostih procesa in izdelkih. Pri analizi se opremo na bazo merskih podatkov preteklih projektov. Razmerje med kakovostjo izdelka, produktivnostjo procesa in časom izvajanja projekta je razvidno iz izmerjenih podatkov.

Merjenje na tej stopnji obsega celovito kvantitativno predstavitev značilnosti procesa in temu primerna so tudi merska izhodišča. Ta so določena z vidika vodstva projekta ali organizacije in določajo višjenivojski pogled. Iz ciljev tega ključnega področja ne moremo določiti kaj meriti, temveč samo, da mora biti merjenje omogočeno. Merjenje je tako odvisno od potreb

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

organizacije. Naša merska izhodišča predlagajo merjenje nekaj ključnih zunanjih atributov: produktivnosti, ponovne uporabljivosti in učinkovitosti izvedbe. Enako kot pri drugih ključnih področjih lahko tudi tu spremljamo status aktivnosti tega področja.

Upravljanje za kakovost

Namen upravljanja za kakovost je razvoj kvantitativnega razumevanja kakovosti izdelkov PO in doseganje specifičnih ciljev, ki so povezani s kakovostjo.

Doslej smo spremljali kakovost na različne načine, a vedno posredno, recimo preko izvajanja aktivnosti preverjanj. V okviru tega ključnega področja določimo kakovost kvantitativno, kar pomeni da opredelimo standardne načine preverjanja, kakovost planiramo, spremljamo in preverjamo. Kakovost razumemo predvsem v smislu odsotnosti napak, zato tudi izpeljane meritve (recimo zanesljivost) utemeljimo na merjenju napak. Mersko izhodišče je splošno in določa spremljanje kakovosti v okviru vseh aktivnosti.

S pomočjo ciljev ključnih področij modela CMM in kombinirane metode smo določili nabor merskih izhodišč, ki jih naslavljajo prve štiri stopnje modela CMM. Kot je razvidno iz dodatka B.2.3 na tej stopnji še ni zaznati veliko posebnosti majhnih organizacij, bo pa tega občutno več pri naboru meritev in pokazateljev.

5.5.2 Nabor entitet in atributov

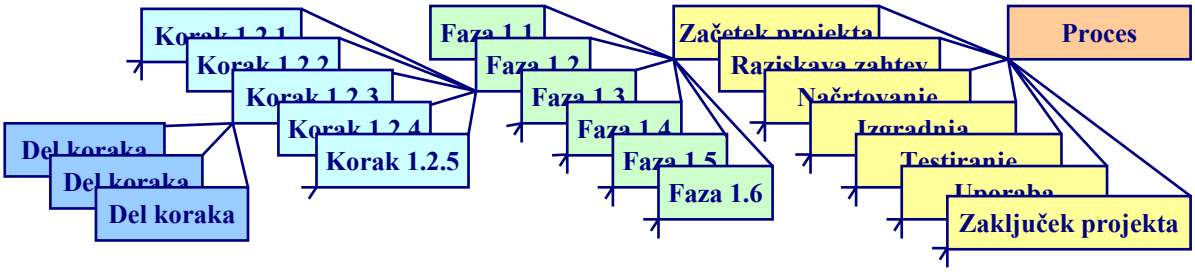
5.5.2.1 Nabor merskih entitet

Prvi del kombinirane metode predvideva določitev merskih entitet. Te pravzaprav določimo z uporabo metode MQG (glej razdelek 2.1.5.2) na podlagi opisanega procesa razvoja PO. Ločimo tri vrste entitet:

- aktivnosti procesa razvoja PO,
- izdelke in
- vire.

Aktivnosti

Aktivnosti nam predstavljajo dele procesa razvoja PO na različnih ravneh: cikle, faze, korake in za potrebe merjenja še dele korakov, ki jih izvedejo posamezni člani projektne skupine. V skladu z opisom procesa so aktivnosti sestavljene v hierarhijo (slika 5.28). Prilagodljiv proces razvoja PO določa največ 7 ciklov, 36 faz in 317 korakov (glej razdelek 3.4.4), ki jih z dinamičnim prilagajanjem izberemo v ustrezen nabor aktivnosti procesa. Zato se natančen nabor entitet aktivnosti določa med izvajanjem projekta: cikli so določeni ob začetnem planiranju, faze in koraki ob planiranjih na začetkih ciklov, deli korakov pa ob samem izvajanju posameznih korakov.



Slika 5.28: Aktivnosti v prilagodljivem procesu razvoja PO

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Celoten nabor entitet aktivnosti je razviden iz opisa prilagodljivega procesa razvoja PO v dodatku A.2.

Izdelki

Vse izdelke lahko razberemo iz rezultatov posameznih faz in ciklov prilagodljivega procesa razvoja PO. Med izvajanjem projekta nastanejo tri vrste izdelkov:

- izdelki, ki opisujejo končne rezultate, in končni rezultati (procesni in podatkovni del),
- izdelki, ki so povezani z opisom izvajanja projekta in
- izdelki, ki opisujejo procesiranje.

Izdelki v okviru vsake vrste so natanko predstavljeni v razdelku 3.4.5. Tabela 5.11 zaradi boljše predstave povzema samo rezultate posameznih ciklov, medtem ko so rezultati faz in ciklov predstavljeni v dodatku A.2.

Cikel	Rezultati - Procesni in podatkovni del	Opis projekta	Opis procesiranja
1	Inicializacijski dokument Strateški dokument	Projektna knjiga Globalni plan	Merski podatki Predlogi za izboljšave
2	Standardi analize Specifikacija zahtev	Plan raziskave zahtev Projektna knjiga (dop.)	Merski podatki Predlogi za izboljšave
3	Standardi načrtovanja Specifikacija načrtovanja	Plan načrtovanja Projektna knjiga (dop.)	Merski podatki Predlogi za izboljšave
4	Standardi izgradnje IS (po delih) Pomožni deli IS PB Pomožni deli PB Testni podatki Dokumentacija	Plan izgradnje Projektna knjiga (dop.)	Merski podatki Poročila o testiranju Predlogi za izboljšave
5	Standardi testiranja IS Pomožni deli IS PB Pomožni deli PB Testni podatki Sistemska dokumentacija Uporabniška dokumentacija	Plan testiranja Projektna knjiga (dop.)	Merski podatki Poročila o testiranju Predlogi za izboljšave
6	Standardi namestitve Standardi vzdrževanja IS Pomožni deli IS PB Pomožni deli PB Pravi podatki Sistemska dokumentacija Uporabniška dokumentacija	Plan namestitve Plan vzdrževanja Dokument o namestitvi Dokument o vzdrževanju Projektna knjiga (dop.)	Merski podatki Podatki o delovanju sistema Predlogi za izboljšave
7	Arhiviran IS Arhivirana Knjiga o razvoju	Končno poročilo Arhivirana Projektna knjiga	Končno poročilo (analize) Predlogi za izboljšave

Tabela 5.11: Rezultati ciklov prilagodljivega procesa razvoja PO

Rezultate določimo na podlagi izbranih aktivnosti projektnega procesa. Vsak rezultat aktivnosti določa eno ali več merljivih entitet (recimo več modulov). Merljive entitete izdelkov so razvidne iz dodatka B.2.4.1, kjer so predstavljene meritve velikosti izdelkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Med merskimi entitetami so izdelki na različnih ravneh. Nekateri izdelki so elementarni (jih ni mogoče več deliti naprej), medtem ko so drugi sestavljeni iz več drugih izdelkov. Izdelki, ki sestavljajo nek izdelek, so lahko iste vrste ali pa različni. Delitev se lahko nadaljuje v več stopnjah. Recimo: končni izdelek procesnega dela je sestavljen iz več menujev, zaslonov in modulov. Posamezni moduli so sestavljeni iz več podmodulov.

Po značilnostih bi izdelke lahko združevali v več skupin:

- **Dokumenti**: besedila (recimo poročila) ali združeni izdelki, ki nastanejo na podlagi več besedil ali različnih izdelkov. Lahko vsebujejo tudi slike (recimo specifikacija zahtev, ki vsebuje tudi ER diagram). Dokumenti so nekateri rezultati, praktično vsi opisi projekta in del opisov procesiranja.
- **Diagrami**: opis procesnega ali podatkovnega dela rezultatov, ki je prikazan s strukturo definiranih elementov.
- **Matrike**: razmerja med različnimi vrstami elementov. Omogočajo prikaz rezultatov.
- **Programska koda**: opis različnih delov procesiranja, recimo modulov, zaslonov, menujev. Predstavlja del končnega izdelka.
- **Podatkovna baza**: opis posameznih elementov PB, recimo tabel, indeksov, pogledov in podobno. Predstavlja del končnega izdelka.
- **Podatki**: konkretna vsebina PB**,** podatki.
- **Merski podatki**: izmerjene vrednosti osnovnih meritev, določene vrednosti združenih in izpeljanih meritev ter pokazatelji. Merski podatki so del opisa procesiranja.

Viri

Med entitete virov sodi osebje in finančna sredstva. Pri osebju so entitete skupine (projektna skupina ter skupini SQA in SEPG na organizacijski ravni) in posamezni člani projektne skupine. Na posameznike lahko gledamo tudi z vidika vlog, ki jih imajo pri projektu. Finančna sredstva predstavljajo omejitev za stroške projekta.

5.5.2.2 Nabor merskih atributov

Naš pristop določa stalen nabor osnovnih meritev, ki se izvajajo ob vsaki aktivnosti posameznika. Tedaj merimo glavne značilnosti entitet aktivnosti, izdelkov teh aktivnosti in porabljenih virov, ki so določene z merljivimi atributi. Atributi so izbrani v skladu z izbranimi razredi meritev:

- porabo časa in napredovanjem,
- velikostjo izdelkov,
- viri in stroški ter
- kakovostjo izdelkov.

Poraba časa in napredovanje se navezuje na značilnosti aktivnosti. Merljivi atributi so datum in čas začetka ter zaključka posameznih aktivnosti.

Velikost je značilnost izdelkov, ki se izraža na različne načine. Za opisane skupine izdelkov lahko določimo tipične atribute, ki predstavljajo velikost:

- **Dokumenti**: dolžina dokumenta;
- **Diagrami**: število različnih vrst sestavnih delov;
- **Matrike**: število elementov;
- **Programska koda**: dolžina programske kode;
- **PB**: število posameznih elementov PB;

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- **Vsebina PB**: poraba prostora v pomnilniku;
- **Merski podatki**: število izmerjenih vrednosti.

Pri **virih in stroških** običajno merimo porabo. Osebje ali skupine merimo s štetjem udeležencev v okviru določene aktivnosti. V drugem primeru spremljamo višino stroškov, ki jo izražamo v primernih enotah.

Kakovost izdelkov se izraža preko števila odkritih in odpravljenih napak v okviru posameznih aktivnosti. Merljivi atributi so torej število odkritih napak, število odpravljenih napak in čas, ki je bil porabljen za odpravo napak v vsaki aktivnosti.

5.5.3 Nabor osnovnih in dela združenih meritev

Nabor osnovnih meritev določimo na podlagi razredov meritev oziroma izbranih atributov entitet. Poleg tega sproti določimo še tisti del združenih meritev, ki so ravno tako nepogrešljiv del merjenja vsakega projekta. Gre za združene meritve entitet iz definiranega nabora entitet, ki jih določimo z združevanjem na podlagi aktivnosti ali rezultata. Primera sta recimo merjenje obsega dela za celoten cikel ali število odkritih napak v celotnem IS.

Meritve si bomo ogledali po vrstnem redu razredov meritev. Predstavljene bodo z vsebinskega stališča. V okviru vsakega merskega izhodišča razlikujemo med osnovnimi in združenimi meritvami. Celoten nabor osnovnih meritev se nahaja v dodatku B.2.4.1, združene meritve pa v dodatku B.2.4.2.

5.5.3.1 Poraba časa in napredovanje

Merjenje **porabe časa** meri trajanje posameznih aktivnosti v okviru izvajanja projekta. Kot rečeno so aktivnosti natančno določene s trinivojsko hierarhijo: cikel, faza, korak in del koraka. Osnovna meritev porabe časa meri porabo časa za dele korakov ali korake, ki jih izvajajo posamezni člani projekta. S pomočjo združevalne strukture po aktivnostih na podlagi osnovnih meritev združujemo rezultate v natančne vrednosti za ostale ravni aktivnosti (faze, cikle ter včasih korake).

Porabo časa merimo v minutah. Minute omogočajo zelo natančno meriti čas, kar je potrebno pri aktivnostih posameznikov. Za večje aktivnosti po potrebi pretvorimo porabo časa v ure ali dneve, pri čemer upoštevamo, da ima vsak delovni dan 8 delovnih ur, torej 480 minut. Tabela 5.12 prikazuje vse osnove in del združenih meritev porabe časa.

Razred meritev	Entiteta	Atribut	Meritev	Tip	Enota	Skala
Poraba časa	Del koraka (Korak)	Datum in čas začetka Datum in čas zaključka	Dolžina trajanja	O	minuta (ura, dan)	0 - ...
	Cikel, Faza (Korak)	Datum in čas začetka Datum in čas zaključka	Dolžina trajanja	ZA	minuta (ura, dan)	0 - ...

Tabela 5.12: Meritve porabe časa

Napredovanje podobno kot poraba časa meri časovni vidik izvajanja projekta, vendar ga meri na podlagi doseganja mejnikov. Naš proces razvoja natanko definira rezultate vsake faze oziroma cikla, ki nam predstavljajo merljive mejnike (tabela 5.13).

Razred meritev	Entiteta	Atribut	Meritev	Tip	Enota	Skala
Napredovanje	Vsi izdelki cikla ali faze (mejniki)	Zaključen izdelek (mejnik)	Zadnji dosežen mejnik	O	1	mejniki

Tabela 5.13: Meritve napredovanja

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Ko je aktivnosti konec in član izvede merjenje, z meritvami opiše tudi rezultat aktivnosti. Kot bomo videli v nadaljevanju temu ustrezno izmeri velikost. V zadnjih korakih vseh faz oziroma ciklov mora vodja koraka preveriti vse rezultate in jih označiti za dokončane. To omogoča, da iz merskih podatkov razberemo doseganje posameznih mejnikov. Za merjenje napovedovanja moramo torej za vse izdelke faz posebej označiti, ko jih zaključimo.

5.5.3.2 Velikost izdelkov

Merske metode za merjenje velikosti

Merjenje velikosti meri osnovno notranjo značilnost izdelkov, ki jih razvijamo. Pri tem ne merimo samo končne izdelke, temveč izdelke vseh aktivnosti, kar nam omogoča izvajanje analiz. Način merjenja velikosti je odvisen od skupine, v katero izdelek sodi. Pri tem naš pristop določa štiri različne merske metode:

- štetje sestavnih delov entitete,
- štetje števila vrstic dokumentov,
- štetje števila vrstic programske kode in
- velikost porabe prostora.

Štetje sestavnih delov je osnovna metoda, ki jo največ uporabljamo. Izmed predstavljenih skupin izdelkov z njo merimo diagrame, matrike, PB in merske podatke. V okviru izbrane merske entitete običajno štejemo eno vrsto elementov, lahko pa štejemo tudi več različnih vrst elementov bodisi skupaj ali ločeno [DeM86].

Štetje števila vrstic je namenjeno merjenju dolžine različnih dokumentov, ki se pojavljajo na različnih ravneh. Čeprav je zaradi natančnosti za osnovno enoto izbrana vrstica, pogosto za enoto uporabimo tudi stran. Pri tem definiramo stalno razmerje, po katerem ena stran ustreza 50 vrsticam. Na ta način lahko merimo velik razpon izdelkov, od recimo opisa ene zahteve do celotne projektne knjige. Proces določa, da ob zaključkih ciklov sestavljamo vmesne rezultate faz v enoten izdelek (primer je recimo specifikacija zahtev) in te izdelke predstavimo vodstvu. Ob tem jih oblikujemo v tako obliko, da lahko izrazimo njihovo velikost v številu strani.

Naš pristop določa, da merjenje velikosti oziroma dolžine programske kode izvajamo s **štetjem števila vrstic programske kode** (ang. Lines Of Code ali LOC). Tak pristop smo izbrali predvsem zaradi enostavnosti, čeprav tovrstno merjenje ne opisuje vseh treh vidikov velikosti (dolžino, funkcionalnost in kompleksnost) [Fen97, Rož01] (glej razdelek 4.3.1.3).

Številni avtorji navajajo, da je LOC najprimernejši za merjenje zaradi svoje razumljivosti in enostavnosti [Fen97, Par92]. Pri tem je nujno zelo natančno definirati izvajanje štetja, kar v našem primeru opišemo s posebnim standardom. V njem določimo, kako štejemo razne vrstice, komentarje, deklaracije in podobno.

Štetje vrstic je namenjeno predvsem merjenju končnih izdelkov. Ker ga želimo uporabiti tudi za splošno merilo procesnega dela (recimo pri napovedovanju), si pri tem pomagamo z normalizacijo. Pri tem določimo razmerje med velikostmi posameznih izdelkov na podlagi podatkov preteklih projektov. Podoben pristop s prilagajanjem izmerjenih vrednosti na podlagi zgodovinskih podatkov opisuje tudi metoda EarlyE [Hor98], ki je sicer namenjena merjenju funkcionalne komponente velikosti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Merjenje porabe prostora je zadnja metoda merjenja velikosti. Primerna je za merjenje količine podatkov, ki jih napolnimo v PB, ali za merjenje arhiviranih delov.

Sistematičnost meritev velikosti

Za vsak merljiv izdelek projekta z uporabo ene (ali več) od opisanih metod določimo osnovno (ali združeno) meritev atributa, ki predstavlja velikost. Kot je razvidno iz opisa nabora entitet (glej razdelek 5.5.2.1) je izdelkov kar precej, kar kaže na zapletenost izbire primerne osnovne meritve. Zato smo osnovne meritve zastavili na sistematičnih načelih, ki izvajalcem merjenj olajšajo izbiro primernega pristopa.

a) Elementarni in sestavljeni izdelki

Pri izbiri nam pomaga dejstvo, da so nekateri izdelki sestavljeni, drugi elementarni. Čeprav se elementarni izdelki pojavljajo na različnih stopnjah, jih vedno merimo z (eno) najprimernejšo metodo. Sestavljene izdelke merimo tako, da izmerimo celotno velikost izdelka, preštejemo število sestavnih delov ter izmerimo vsak del posebej, če je to smiselno.

b) Skupine izdelkov in merske metode

Za posamezne skupine izdelkov lahko presodimo, katera merska metoda je zanje primerna (diagrame recimo merimo s štetjem sestavnih delov). Izbira primerne metode je razvidna iz predstavitve štirih merskih metod.

c) Merjenje velikosti za vrste izdelkov

Vendar potrebujemo natančnejši pristop, zato sistematično določimo merjenja za vsako vrsto izdelkov. Kot smo že omenili ločimo tri vrste izdelkov:

- izdelke, ki opisujejo končne rezultate, in končni rezultati,
- izdelke, ki so povezani z opisom izvajanja projekta in
- izdelke, ki opisujejo procesiranje.

Izdelki, ki opisujejo končne rezultate, in končni rezultati so z vidika merjenja velikosti ključnega pomena, ker omogočajo spremljanje razvoja in analize. V našem pristopu merimo oba vidika velikosti: procesni in podatkovni. V obeh primerih poteka razvoj izdelka v treh stopnjah: zahteve, opis končnih izdelkov ter končni izdelek.

- Zahteve merimo s količino zahtev in dolžino opisa posameznih zahtev izraženo v številu vrstic.
- Za opis končnih izdelkov uporabljamo diagrame in matrike. Kot rečeno diagrame tipično merimo s štetjem različnih vrst komponent in skupno dolžino opisa vseh komponent. Naš pristop predvideva uporabo treh vrst diagramov:
  - Diagrame entiteta razmerje (ERD) merimo s štetjem števila entitet, razmerij in atributov ter dolžino opisa vseh komponent. Primera uporabe sta visokonivojski ERD in logični ERD.
  - Diagrame podatkovnih tokov (DFD) merimo s štetjem števila elementarnih procesov, podatkovnih skladišč, zunanjih entitet, podatkovnih tokov in skupno dolžino opisa komponent. Uporabljamo jih za visokonivojski DFD in logični DFD.
  - Diagrame sestave merimo s številom ravni, številom elementov (listov) diagrama in skupno dolžino opisa elementov. Te diagrame uporabljamo za opis funkcijske dekompozicije (FD), konceptualnega načrta, navigacije ter strukture menujev.

  Matrike merimo s štetjem števila elementov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Končne izdelke merimo kot sestavljene izdelke na več stopnjah. Celotno velikost procesnega dela izražamo v številu vrstic programske kode, medtem ko podatkovni del štejemo s skupnim številom vseh komponent opisa PB.

  Končne izdelke procesnega dela merimo predvsem z dolžino programske kode, ki jim ustreza. Celotne izdelke razdelimo na module, pri čemer moduli predstavljajo dele programa (recimo dinamične strani), poročila, obrazce, podprograme, funkcije ali kaj podobnega. Poleg tega merimo uporabniške vmesnike s številom zaslonov in elementov na njih ter menujsko strukturo in navigacijo z ustreznimi diagrami.

  PB merimo s številom različnih vrst elementov PB, recimo s številom tabel, atributov, pogledov in podobno. Z velikostjo porabljenega prostora merimo tudi vsebino PB.

  Pomemben del končnega izdelka je tudi dokumentacija. Velikost dokumentacije različnih vrst (sistemska, uporabniška, učna) merimo v številu strani.

Izdelki, ki so povezani z opisom izvajanja projekta, so druga vrsta izdelkov, ki jih merimo. Smisel merjenja teh izdelkov je predvsem spremljanje obsega dela za njihovo izdelavo. Vsi izdelki iz tega dela so v našem primeru združeni v projektno knjigo. Tipični deli so opis projekta, različni plani in poročila. Velikost vseh merimo s številom vrstic ali številom strani.

Izdelki, ki opisujejo procesiranje so predvsem merski podatki, poleg tega pa še poročila o izvedbi testiranja, predlogi za izboljšave, zahtevki za spremembe zahtev ali konfiguracije in podobno. Tudi velikost teh podatkov merimo zaradi spremljanja obsega dela.

Merski podatki so dveh vrst. Zbrane rezultate (osnovnih) meritev merimo z enostavnim štetjem, medtem ko informacijske izdelke preštejemo in nato še vsakega posebej zmerimo. Velikost informacijskih izdelkov izrazimo v številu vrstic, pri čemer upoštevamo, da ena stran ustreza petdesetim vrsticam.

Podobno ravnamo tudi pri merjenju velikosti poročil o testiranju: štejemo njihovo količino ter velikost vsakega v številu strani. Predloge in zahtevke samo preštejemo.

Merjenje celotne velikosti in merjenje spremembe velikosti izdelka

Nekaj izdelkov, ki jih določa prilagodljiv proces razvoja PO, se razvija postopoma. V okviru določene faze se recimo oblikuje zasnova izdelka, ki se v naslednji fazi dopolni in šele ob zaključku cikla dokončno oblikuje. Lep primer je recimo projektna knjiga, ki se dopolnjuje celo ves čas izvajanja projekta: zasnuje se na začetku projekta, skozi vse cikle dopolnjuje in ob zaključku projekta s končnim poročilom dokončno oblikuje.

To je posebej res, če se spustimo na raven korakov ali delov korakov, kjer rezultatov naš proces sicer formalno ne predpisuje. Z osnovnimi meritvami merimo nastale dele rezultatov faz, ki jih postopoma (ob zaključku faze) sestavimo v celoten izdelek.

Merjenje velikosti je treba potemtakem razumeti na dva načina:

- Z vidika rezultatov posameznih faz je vsak izdelek celota, zato ga tako tudi merimo.
- Z vidika rezultatov aktivnosti posameznikov ni prav, da ob vsakem merjenju istega razvijajočega izdelka merimo njegovo celotno velikost. Smiselno je meriti zgolj tisti del velikosti, ki je bila dodana v okviru izpeljane aktivnosti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Naš pristop določa, da merimo oboje in za oboje uporabimo osnovne meritve. Merjenje celotne velikosti izdelka se izmeri ob vsakem zaključku faze. Za izvedbo tega merjenja je odgovoren vodja zadnjega koraka v fazi. Če se izdelek v naslednji fazi nadgradi, bo celotna velikost zopet izmerjena ob zaključku naslednje faze.

Med izvajanjem aktivnosti posameznikov merimo velikost natančneje. Merjenje velikosti ovrednotimo z naslednjimi vrednostmi:

- V velikost izdelka pred začetkom aktivnosti (v ustreznih enotah);
- V_d velikost, ki smo jo med izvajanjem aktivnosti dodali (v ustreznih enotah);
- V_p velikost, ki smo jo med izvajanjem aktivnosti dodali, vendar je bil ta del izdelka pred tem že uporabljen nekje drugje (v ustreznih enotah);
- V_s velikost, ki smo jo med izvajanjem aktivnosti spremenili (v ustreznih enotah);
- V_b velikost, ki smo jo med izvajanjem velikosti izbrisali (v ustreznih enotah).

Ob zaključku aktivnosti lahko izrazimo novo celotno velikost izdelka na naslednji način:

$$V_n = V + V_d + V_p - V_b$$

Prispevek izvedbe aktivnosti je v tem primeru:

$$V_a = V_d + V_s$$

Naš pristop predvideva, da posamezniki za vsako aktivnost izmerijo oboje:

- celotno velikost izdelka po zaključeni aktivnosti (V_n) in
- spremembo velikosti izdelka, ki so jo prispevali sami z izvedbo aktivnosti (V_a).

Pristop z merjenjem celotne in spremenjene velikosti sicer zaplete merjenje, vendar nam omogoči bistveno večjo natančnost meritev. Ne smemo pozabiti, da na podlagi meritev velikosti ocenjujemo veliko značilnosti.

Če posamezni člani uporabljajo ustrezne osebne procese razvoja PO, tak način merjenja dejansko uporabljajo za svoje potrebe in jim to ne predstavlja dodatnega dela. Nasploh velja, da večino izdelkov razvijemo naenkrat in tedaj merjenje ni težavno. Pri številnih izdelkih (recimo knjigi projekta) je celoten prispevek aktivnosti kar enak dodanemu delu. Posamezniki lahko pri merjenju posamezne dele ocenijo tudi na podlagi preteklih izkušenj. Glede na to, da so merjenja na ravni aktivnosti posameznikov zelo podrobna, napaka pri tem ne bo prevelika.

Osnovne in združene meritve velikosti izdelka

Opis merjenja velikosti, ki smo ga predstavili, se nanaša tako na osnovne kot na združene meritve velikosti izdelkov. Posamezniki uporabljajo osnovne meritve za merjenje izdelkov, ki jih pri izvedbi aktivnosti (koraka ali dela koraka) ustvarijo. Pogosto to niso čisto vsi nastali izdelki, temveč rezultati posameznih faz ali njihovi sestavni deli. Izmerjene vrednosti so podlaga za določanje vrednosti združenih meritev.

Združene meritve se navezujejo na večje entitete, predvsem na rezultate cikla (recimo razne specifikacije), saj glavnino rezultatov faz merimo kot celotne izdelke z osnovnimi meritvami. Ne glede na to je del združenih meritev nujen.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Razred meritev	Vrsta Entitete	Atribut	Osnovna meritev	Tip	Enota
Velikost – rezultati	**Dokument, predlog, opis, poročilo**	Skupna dolžina	Število vrstic ali strani	O	vrstica ali stran
	Seznam zahtev, standard (raziskave zahtev, načrtovanja, kodiranja, testiranja)	Količina delov Dolžina vsakega dela Skupna dolžina	Število delov Število vrstic Število vrstic ali strani	O O O	del vrstica vrstica ali stran
	Diagram (ERD, DFD)	Količina vsake vrste elem. Dolžina celotnega opisa	Število vsake vrste elem. Število vrstic ali strani	O O	element vrstica ali stran
	Diagram sestave (FD, koncept.načrt,struktura menuja,diagram navigacije)	Količina ravni Količina elementov Dolžina celotnega opisa	Število ravni Število elementov Število vrstic ali strani	O O O	raven element vrstica ali stran
	Matrika	Količina elementov	Število elementov	O	element
	Prototip	Količina delov Dolžina pr.kode za vsak del Dolžina pr.kode skupaj	Število delov Število vrstic kode Število vrstic kode	O O O	del vrstica kode vrstica kode
	Fizični načrt (procesiranja in PB), **del IS**	Količina vsake vrste Elementov	Število vsake vrste elementov	O	element
	Modul (deli programa, poročila, obrazci), **sprožilec, bazna procedura**	Dolžina prog. kode	Število vrstic kode	O	vrstica kode
	Zaslon	Količina elementov	Število elementov	O	element
	PB	Količina vsake vrste Elementov	Število vsake vrste elementov	O	element
	Podatki v PB (testni, pravi), **arhivirani deli**	Obseg porabljenega Prostora	Poraba pomnilnika	O	MB
	Dokumentacija (sistemska uporabniška, učna)	Količina delov Dolžina vsakega dela Skupna dolžina	Število delov Število vrstic ali strani Število vrstic ali strani	O O O	del vrstica ali stran vrstica ali stran
Velikost – projekt	**Projektna knjiga** (opis, plani, poročila)	Dolžina vsakega dela Skupna dolžina	Število vrstic ali strani Število vrstic ali strani	O O	vrstica ali stran vrstica ali stran
Velikost – procesiranje	**Merski podatki** (izmerjene vrednosti, informacijski izdelki)	Količina podatkov Količina inf. izdelkov Dolžina inf. izdelkov	Število merjenih vrednosti Število inf. izdelkov Število vrstic ali strani	O O	meritev inf. izdelkov vrstica ali stran
	Poročilo o testiranju	Dolžina poročila	Število vrstic ali strani	O	vrstica ali stran
	Predlogi, zahtevki	Količina predlogov ali Zahtevkov	Število predlogov ali zahtevkov	O	predlogov ali zahtevkov

Tabela 5.14: Osnovne meritve velikosti izdelkov

Tabela 5.14 predstavlja osnovne meritve velikosti za vse **vrste izdelkov**, ki jih določa naš nabor entitet. V dodatku B.2.4.1 je predstavljena podrobnejša tabela z vsemi izdelki faz, iz katere so razvidni tudi elementarni in sestavljeni izdelki. Obe tabeli opisujeta samo merjenje celotnih izdelkov, ker se merjenje sprememb velikosti meri na enak način. Izvajamo ga samo tedaj, ko izdelek razvijamo postopoma.

Tabela 5.15 prikazuje združene meritve velikosti. Velikost za večje entitete merimo običajno z enim samim atributom. Običajen način združevanja je po rezultatih.

Razred meritev	Vrsta Entitete	Atribut	Združena meritev	Tip	Enota
Velikost – rezultati	**Specifikacija** (zahtev, načrtovanja)	Skupna dolžina	Število strani	ZR	stran
	IS	Skupna dolžina	Število vrstic prog. kode	ZR	vrstica kode
	PB	Skupna količina elementov	Število vseh elementov	ZR	element
	Dokumentacija	Skupna dolžina	Število strani	ZR	stran
Velikost – projekt	**Projektna knjiga**	Skupna dolžina	Število strani	ZR	stran
Velikost – procesiranje	**Merski podatki**	Skupna količina podatkov Skupna dolžina inf. izdelkov	Število merjenih vrednosti Število strani	ZA ZR	meritev stran

Tabela 5.15: Združene meritve velikosti izdelkov

5.5.3.3 Viri in stroški

Vire merimo v našem merskem procesu predvsem z obsegom dela, ki je potreben za izvedbo posameznih aktivnosti. Obseg dela je pomembna značilnost aktivnosti, s pomočjo katere napovedujemo druge značilnosti. Poleg tega po potrebi merimo tudi udeležbo posameznih članov in skupin pri aktivnostih, recimo korakih, fazah in ciklih.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Posamezniki ob vsaki izvedbi aktivnosti (koraku ali delu koraka) z osnovno meritvijo izmerijo tudi obseg dela. Ker se vsaka aktivnost nanaša na enega samega izvajalca, se ta izraža kar v številu delovnih minut (ali ur), ki jih je potreboval za izvedbo aktivnosti. Minute uporabimo zaradi natančnosti, saj posamezniki lahko merijo obseg dela tudi za zelo kratke aktivnosti. Po potrebi lahko uporabimo večje enote: ure ali dneve. Pri tem upoštevamo, da ima delovni dan 8 delovnih ur. Enota merjenja je človek-minuta, človek-ura ali človek-dan.

Pri določanju obsega dela moramo razlikovati med trajanjem aktivnosti in potrebnim časom za izvedbo: trajanje je določeno s časom (datumom, uro in minuto) začetka in konca aktivnosti ter služi za merjenje porabe časa. Potreben čas za izvedbo meri čas dejanskega izvajanja in ne šteje prekinitev, zato ga lahko uporabimo za merjenje obsega dela.

Za vse večje aktivnosti (cikle, faze in tiste korake, ki se izvajajo v več delih) določamo obseg dela z združenimi meritvami. Pri tem seveda združujemo izmerjene vrednosti po aktivnostih. Po potrebi lahko združujemo obseg dela tudi po rezultatih ali izvajalcih, vendar to ne sodi med obvezne združene meritve, ki jih predstavljamo v tem delu.

Razred meritev	Entiteta	Atribut	Meritev	Tip	Enota	Skala
Viri	Del koraka (Korak)	Dejanski čas za izvedbo	Obseg dela aktivnosti	O	človek-minuta (človek-ura, človek-dan)	0 - ...
	Cikel, Faza (Korak)	Potreben obseg dela za izvedbo	Obseg dela aktivnosti	ZA	človek-minuta (človek-ura, človek-dan)	0 - ...

Tabela 5.16: Meritve virov

V tabeli 5.16 so predstavljene osnovne meritve virov, zraven pa so tudi združene meritve, s katerimi določamo udeležence (večjih) aktivnosti. Pri tem preštejemo vse udeležence, ki oddajo izmerjene vrednosti za določeno meritev. Na podoben način lahko določimo tudi sodelovanje članov posameznih skupin pri koraku. Obe meritvi sodita med tiste, ki jih izvajamo zgolj po potrebi, zato jih v tabeli 5.16 ni.

Pomembna značilnost vsake aktivnosti so tudi **stroški**. V razdelku 4.3.1.3 smo opisali, da je del stroškov odvisen od obsega dela, medtem ko se dodatni stroški določajo po drugačnih zakonitostih. Ti predstavljajo recimo stroške opreme, potne stroške, stroške inventarja in podobno. Za naše potrebe ni pomembno ločevanje različnih vrst, temveč zgolj njihova vsota.

Stroškov, ki so odvisni od obsega dela, posebej ne merimo in jih (po potrebi) določimo na podlagi izmerjenega obsega dela. Moramo pa meriti dodatne stroške. Tako posamezniki ob vsaki izvedbi aktivnosti izmerijo tudi svoje stroške. V organizaciji obstaja predpis, ki natanko opisuje, kaj sodi med stroške ter kakšni so normativi za izračun stroškov (recimo kilometrina). Vodje korakov v sodelovanju z vodjo projekta so zadolženi beleženje večjih stroškov (recimo plačilo zunanjemu sodelavcu, ki je sodeloval pri koraku, skupni stroški potovanja in podobno). Tudi pri tem je uporaba predpisa nujna, da se prepreči pomanjkljivo ali večkratno beleženje stroškov. Za stroške na ravni celotnega projekta (recimo nakup strojne ali programske opreme) in tiste, ki ne sodijo v nobeno aktivnost (recimo večerja z vodstvom organizacije), je zadolžen vodja projekta. Meritev vnese na enak način, kot vnašajo stroške posamezni izvajalci aktivnosti.

Za vse večje aktivnosti (cikle, faze in nekatere korake) se skupni stroški določajo z združevanjem, pri čemer uporabimo združevalno strukturo po aktivnostih. Vse stroške merimo v denarnih enotah, v našem primeru v slovenskih tolarjih (SIT). Tabela 5.17 prikazuje vse nujne meritve stroškov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Razred meritev	Entiteta	Atribut	Meritev	Tip	Enota	Skala
Stroški	Del koraka (Korak)	Dodatni stroški	Znesek	O	SIT	0 - ...
	Cikel, Faza (Korak)	Dodatni stroški	Znesek	ZA	SIT	0 - ...

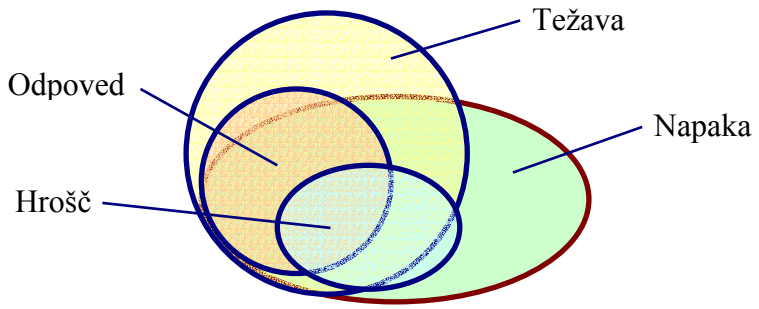
Tabela 5.17: Meritve stroškov

5.5.3.4 Kakovost izdelkov

Ustrezno mersko spremljanje kakovosti izdelkov je izredno pomembno za nadzor procesa razvoja PO. De Marco celo navaja, da je ustrezno spremljanje kakovosti edina zares nepogrešljiva meritev vsakega razvoja PO [DeM95]. Kot smo opisali v razdelku 4.3.1.3 lahko kakovost PO razumemo kot množico zaželjenih lastnosti - atributov [McC77, Boe78]. Žal je večina teh atributov kakovosti zunanjih in jih ne moremo direktno meriti.

Drugi pristopi temeljijo na pojmovanju kakovosti z vidika napak: merjenje kakovosti se presoja na podlagi števila in vrste napak izdelkov. Tak pristop uporabimo tudi mi. Ker s tem zajamemo samo nekaj lastnosti iz McCallovega nabora (recimo točnost, natančnost, usklajenost, popolnost) [McC77], moramo biti pri merjenju čimbolj natančni.

Pri tem je pomembno pojem napake ustrezno definirati, še posebej spričo drugih pojmov s podobnim pomenom. Težave PO (ang. software problems) predstavljajo kakršnokoli težavo pri razvoju ali uporabi PO, do katere lahko pride iz različnih vzrokov. Eden od vzrokov so napake (ang. faults) v PO. Povzročajo jih človeške pomote (ang. human error) in jih zato lahko odpravimo. Za posledico imajo lahko odpovedi (ang. failures) oziroma nepravilno delovanje. V kolikor so napake v programski kodi, govorimo o hroščih (ang. bugs), anomalije (ang. anomalies) pa so napake, ki ne povzročajo težav[28]. V našem primeru kot napako štejemo vsako netočnost ali nepopolnost v izdelku PO, ki se pojavi pri izvedbi procesa razvoja PO. Slika 5.29 prikazuje pomen posameznih pojmov (razmerja velikosti niso ustrezna).



Slika 5.29: Pomen pojmov za opis kakovosti izdelkov

Za določanje kakovosti s pomočjo napak se uporabljata dva pristopa [Fen97, DoD02]:

- določanje kakovosti z ugotavljanjem povprečnega časa do naslednje napake ter
- določanje kakovosti z ugotavljanjem števila preostalih napak v izdelkih, kar določimo iz števila odkritih in odpravljenih napak v posameznih fazah razvoja.

V našem primeru uporabljamo drugi pristop. V ta namen mora vsak posameznik pri izvedbi vsake aktivnosti mersko spremljati dve stvari: **odkrite napake in odpravljene napake**.

[28] Definicijo smo povzeli po [Fen97] in je usklajena s standardom IEEE 729 [IEEE]. Obstajajo tudi druge, ki se nekoliko razlikujejo (recimo [SEL95]).

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Odkrite napake

Opisan proces razvoja PO in še posebej merski proces implicitno vsebuje zahtevo, da vsi člani razvojne skupine po svojih najboljših močeh izvajajo vse predpisane aktivnosti. To med drugim pomeni, da se poleg formalnega testiranja končnega izdelka pri izvedbi vsakega koraka (ali dela koraka) predvidijo podaktivnosti za preverjanje pravilnosti razvitega (dela) izdelka (recimo ponovni pregled), s čemer izvajalci odkrijejo eventualne napake. Tudi nasploh med izvajanjem vsake aktivnosti so razvijalci pozorni na napake, ki se pojavljajo pri izdelkih, ki jih samo uporabljajo (recimo zahteve, uporabljeni moduli in podobno).

Naš proces razvoja PO predvideva, da je treba vsako odkrito napako primerno zabeležiti in jo opisati. Če gre za drobno aktivnost posameznika, se beleženje napak začne tedaj, ko je zaključena izdelava in nastopi podaktivnost preverjanja pravilnosti razvitega izdelka. V tem primeru se napake beležijo v poročilo o odkritih napakah posameznika. V primeru korakov, ki so v celoti posvečeni odkrivanju napak (recimo ponovni pregledi, preverjanje usklajenosti ali testiranje), je cilj aktivnosti odkrivanje napak, zato so rezultati ustrezna poročila (recimo poročilo o testiranju).

Pri odkritih napakah je ključno merilo število napak, ki jih odkrije izvajalec vsake aktivnosti. Poleg tega sta za mersko spremljanje pomembna še dva podatka o vsaki napaki:

- izdelek (ali del izdelka), pri katerem je bila napaka odkrita in
- razred napake, ki predstavlja vrsto oziroma resnost napake.

Ker so vsi izdelki rezultati določene faze, lahko iz tega podatka presodimo, kdaj (v kateri fazi) je bila napaka dejansko vnešena. To je za naš pristop zelo pomemben podatek. Na podlagi teh podatkov za posamezne faze določimo, koliko napak je bilo tedaj vnešeno.

Razred napake nam z merskega stališča omogoča, da objektivneje presojamo o kakovosti posameznega izdelka. Obstaja več standardnih razvrstitev napak. Humphrey v okviru modela PSP [Hum95] recimo povzema interni standard IBM[29] (tabela 5.18).

Razred napake	Opis razreda napake
10	Dokumentacija
20	Sintaksa
30	Izgradnja ali povezljivost
40	Namen
50	Uporabniški vmesnik
60	Preverjanje mej
70	Podatki
80	Funkcije
90	Sistem
100	Okolje

Tabela 5.18: Razredi napak po standardu IBM [Hum95]

V našem primeru smo se raje odločili za enostavnejši opis resnosti (ang. severity) napake (tabela 5.19). Z osnovno meritvijo posamezniki določijo število napak za vsak razred posebej.

[29] O problematiki odpravljanja napak v podjetju IBM si bralec lahko prebere tudi v [Ada84].

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

	Razred napake	Opis
K	Kritična napaka	Napaka, ki povzroči odpoved ali zelo nepravilno delovanje
R	Resna napaka	Napaka, zaradi katere izdelek vsebinsko ne deluje pravilno
S	Sintaksna napaka	Napaka v sintaksi, ki ne povzroči odpovedi in ni vsebinska (recimo meje)
L	Lepotna napaka	Napaka uporabniškega vmesnika ali dokumentacije brez nepravilnega delovanja

Tabela 5.19: Razredi napak po našem pristopu

Odpravljene napake

Med izvajanjem aktivnosti posamezniki tudi odpravljajo napake. Naš proces zahteva, da odpravijo vse odkrite napake, ki so posledica izvajanja njihove aktivnosti, o ostalih pa poročajo. To ne velja za korake, ki so posebej namenjeni zagotavljanju kakovosti (preverjanje usklajenosti, testiranje in podobno). V okviru teh aktivnosti je treba odpraviti vse odkrite napake.

V skladu s tem je glavna osnovna meritev število odpravljenih napak v okviru posamezne aktivnosti. Podobno kot v primeru odkritih napak to število natančneje razdelimo po aktivnostih vnosa in razredih. Drugi pomemben podatek je celoten obseg dela, ki je bil za odpravo napak potreben. Na podlagi podatkov o posameznih odpravljenih napakah lahko presodimo tudi o tem, kakšne vrste napak smo odpravili, obsega dela po aktivnostih in fazah pa ne opredeljujemo natančneje. To je podlaga za različne izpeljane meritve, ki natančneje napovedujejo kakovost končnega izdelka.

Tabela 5.20 prikazuje osnovne meritve, ki se nanašajo na kakovost izdelkov.

Razred meritev	Entiteta	Atribut	Meritev	Tip	Enota	Skala
Kakovost izdelkov	Del koraka (Korak)	Število odkritih napak	Število vseh odkritih napak	O	napaka	0 - ...
	Del koraka (Korak)	Število odkritih napak po fazah vnosa	Število odkritih napak po fazah vnosa	O	napaka	0 - ...
	Del koraka (Korak)	Število odkritih napak po razredih napak	Število odkritih napak po razredih napak	O	napaka	0 - ...
	Del koraka (Korak)	Število odpravljenih napak	Število vseh odpravljenih napak	O	napaka	0 - ...
	Del koraka (Korak)	Število odpravljenih napak po fazah vnosa	Število odpravljenih napak po fazah vnosa	O	napaka	0 - ...
	Del koraka (Korak)	Število odpravljenih napak po razredih napak	Število odpravljenih napak po razredih napak	O	napaka	0 - ...
	Del koraka (Korak)	Potreben čas za odpravo napak	Skupen obseg dela odpravljanja napak	O	napaka	0 - ...

Tabela 5.20: Osnovne meritve kakovosti izdelkov

Na podlagi vseh izmerjenih vrednosti z združenimi meritvami določimo združene vrednosti. Pri tem uporabimo naslednje združevalne strukture:

- po aktivnostih: sestavimo vse vrste napak za posamezne faze in cikle;
- po rezultatih: sestavimo vse vrste napak za rezultate posameznih ciklov ali faz;
- po vidikih: sestavimo vse vrste napak za podatkovni oziroma procesni vidik;
- po izvajalcih: sestavimo vse vrste napak glede na izvajalce;
- po funkcijah: sestavljamo samo po potrebi.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Tabela 5.21 prikazuje združene meritve za kakovost izdelkov. Celovit pregled vseh osnovnih in združenih meritev je v dodatkih B.2.4.1 in B.2.4.2.

Razred meritev	Entiteta	Atribut	Meritev	Tip	Enota
Kakovost izdelkov	Faza ali cikel	Število odkritih napak aktivnosti	Število odkritih napak aktivnosti	ZA	napaka
	Izdelek faze ali cikla	Število odkritih napak izdelka	Število odkritih napak izdelka	ZR	napaka
	Izdelki faze ali cikla	Število odkritih napak procesnih ali podatkovnih izdelkov	Število odkritih napak v (procesnem ali podatkovnem) izdelku	ZV	napaka
	Faza ali cikel	Število odkritih napak izbranega razvijalca	Število odkritih napak izbranega razvijalca	ZI	napaka
	Faza ali cikel	Število odpravljenih napak aktivnosti	Število odpravljenih napak aktivnosti	ZA	napaka
	Izdelek faze ali cikla	Število odpravljenih napak izdelka	Število odpravljenih napak izdelka	ZR	napaka
	Izdelki faze ali cikla	Število odpravljenih napak procesnih ali podatkovnih izdelkov	Število odpravljenih napak v (procesnem ali podatkovnem) izdelku	ZV	napaka
	Faza ali cikel	Število odpravljenih napak izbranega razvijalca	Število odpravljenih napak izbranega razvijalca	ZI	napaka
	Faza ali cikel	Potreben čas za odpravo napak aktivnosti	Skupen obseg dela odpravljanja napak aktivnosti	ZA	napaka
	Izdelek faze ali cikla	Potreben čas za odpravo napak izdelka	Skupen obseg dela odpravljanja napak izdelka	ZR	napaka
	Izdelki faze ali cikla	Potreben čas za odpravo napak procesnih ali podatkovnih izdelkov	Skupen obseg dela odpravljanja napak v (procesnem ali podatkovnem) izdelku	ZV	napaka
	Faza ali cikel	Potreben čas za odpravo napak izbranega razvijalca	Skupen obseg dela odpravljanja napak izbranega razvijalca	ZI	napaka

Tabela 5.21: Združene meritve kakovosti izdelkov

5.5.4 Nabor preostanka združenih meritev, izpeljanih meritev in pokazateljev

Z uporabo kombinirane metode določimo še preostale nabore, ki manjkajo do popolne predstavitve vsebine merjenja: nabor preostalih združenih meritev, nabor izpeljanih meritev in nabor pokazateljev. Kot je prikazano v razdelku 5.4.3.3 ti nabori nastanejo ob usklajevanju merskih izhodišč s posameznimi meritvami. Z iterativnim postopkom preverjanja popolnosti in zadostnosti meritev na koncu zagotovimo, da podpirajo vsa stalna merska izhodišča. Pri tem upoštevamo tudi splošna merska izhodišča, ki so bila določena na podlagi zahtev posebnosti organizacije za razvoj IS.

Pomembno je, da težimo k majhnosti vseh naborov. K temu nas sicer sili že sama metoda, vendar moramo to upoštevati tudi ob posameznih presojah usklajenosti merskih izhodišč s posameznimi meritvami. Številni predstavljeni deli našega merskega pristopa, ki so bili dokaj podrobno in obsežno zastavljeni, namreč ne vplivajo neposredno na obseg izvajanja merjenja ob procesu razvoja, temveč predvsem le-tega natančno definirajo. Recimo: podrobna opredelitev vseh izdelkov faz procesa razvoja PO in natanko predpisani načini merjenja velikosti teh služijo samo temu, da bi razvijalec natanko vedel, kateri izdelek naj ob izvedbi aktivnosti (koraka ali dela koraka) meri in na kakšen način naj izrazi njegovo velikost. Praktično gledano pa posameznik opravi samo dve meritvi (celotne in spremenjene) velikosti, pa še to le v primeru, če s svojo aktivnostjo razvije izdelek, ki ga je treba meriti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Na obsežnost merjenja z vsebinskega vidika seveda najbolj vpliva nabor osnovnih meritev, ki smo ga omejili, kolikor je bilo mogoče. Drugi vzrok obsežnosti je število pokazateljev in posledično vseh meritev, ki se ne merijo direktno. Ta sicer ne vpliva na obseg dela vseh razvijalcev projektne skupine, temveč predvsem na delo vodje procesa in merjenja. Po drugi strani pokazatelji določajo možnost vpogleda v proces razvoja PO, zato je s tega stališča njihova množičnost in raznovrstnost zaželjena. Ta razmislek smo upoštevali pri določanju ocen in napovedi (razdelka 4.3.1 in 4.3.4). Vendar je preobilica nesistematično urejenih pokazateljev moteča. Še posebej je to res v majhni organizaciji, kjer je v vlogi vodje procesa in merjenja eden izmed članov majhne projektne skupine, ki poleg te vloge še aktivno sodeluje pri razvoju in opravlja druge vloge. S tega stališča je pomembno, da je število pokazateljev smiselno omejeno, vendar ustrezno izbrano. To pa pravzaprav zagotavlja dosledna uporaba kombinirane metode.

V nadaljevanju so prikazani trije primeri uporabe kombinirane metode za določanje pokazatelja. V prvem primeru je pokazatelj določen na podlagi meritve, ki že obstaja v naboru osnovnih meritev. V drugem primeru je mersko izhodišče izbrano tako, da prikaže tudi definiranje nove izpeljane meritve. Tretji primer poleg nove izpeljane meritve prikaže še izpeljavo nove združene meritve ter določanje zahtevnejšega pokazatelja za kakovost.

5.5.4.1 Trije primeri izpeljave pokazateljev in (združenih ali izpeljanih) meritev

Formalno gledano smo z uporabo kombinirane metode doslej opravili prve tri korake:

- določili smo entitete (razdelek 5.4.3.1),
- določili smo stalna merska izhodišča (razdelek 5.4.3.2) in
- določili smo minimalni nabor meritev, ki v našem primeru obsega vse osnovne in del združenih meritev (razdelek 5.4.3.3).

Naslednji korak je preverjanje popolnosti, ki zahteva, da je vsako mersko izhodišče izpolnjeno, torej v razmerju z vsaj eno meritvijo. Razmerje je formalno opisano s pokazateljem. V kolikor določenega merskega izhodišča ne moremo izpolniti z obstoječimi meritvami, definiramo nove meritve.

Obseg dela aktivnosti

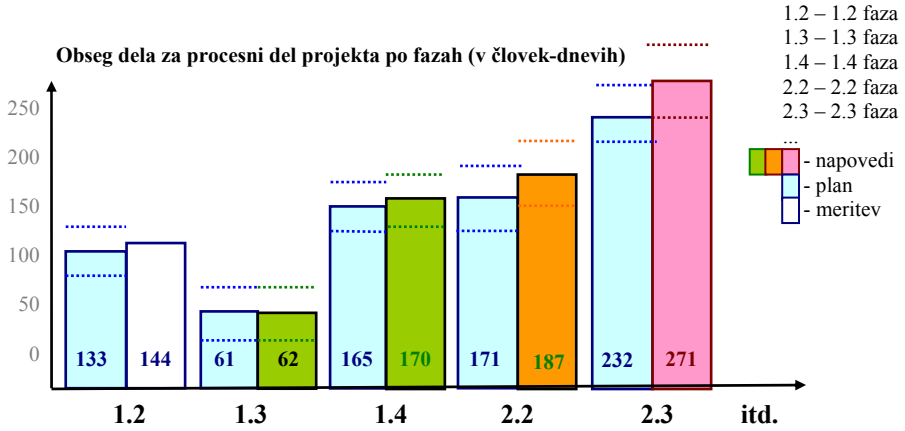
Za izhodišče vzemimo mersko izhodišče, ki izhaja iz zahtev ključnega področja »Integrirano vodenje projektov razvoja PO« na tretji stopnji modela CMM: Spremljaj in napoveduj potreben obseg dela za posamezne aktivnosti. Entiteta merskega izhodišča je določena splošno in obsega vse planirane aktivnosti.

Očitno je mersko izhodišče povezano z meritvami obsega dela. V naboru meritev že obstaja osnovna meritev obsega dela vsake aktivnosti in več združenih meritev, ki določajo obseg dela za posamezne faze in cikle ločeno za procesni in podatkovni vidik.

V skladu z metodo izvedemo primerjavo usklajenosti v treh delih. Najprej preverimo entitete meritev in merskega izhodišča. S tega stališča tako osnovna kot združene meritve ustrezajo, saj se nanašajo na aktivnosti procesa razvoja PO. Drugi del je preverjanje razreda meritev. Izbrano mersko izhodišče sodi v razred viri in stroški, enako kakor vse meritve. V tretjem delu na podlagi smiselne presoje določimo pokazatelj. Pri tem je pomembno smiselno upoštevati zahteve in okoliščine.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

V našem primeru to pomeni, da se omejimo na primerno raven aktivnosti – recimo na raven posameznih faz. Ključna meritev je zato združena meritev obsega dela po fazah, ki jo bomo v okviru pokazatelja prikazali grafično v obliki stolpičnega diagrama. Upoštevati moramo tudi značilnosti razvoja IS, zato ločujemo med obsegom dela procesnega in podatkovnega dela. Poleg tega mora pokazatelj omogočati tudi napovedovanje, kar pomeni, da že znane vrednosti obsega dela primerjamo s planiranimi. Za planirane, a še neizpeljane faze procesa tudi napovemo obseg dela na podlagi podatkov preteklih projektov. Podatke preteklih projektov upoštevamo z upoštevanjem tabele tipičnih razmerij obsega dela, za vrednosti pa določimo tudi območje 90% zaupanja. Slika 5.30 prikazuje primer opisanega pokazatelja za procesni del. Enak bi obstajal za podatkovni del. Opisan pokazatelj je dejansko napoved obsega dela po fazah, kot je bil predstavljen v razdelku 5.5.3.3.



Slika 5.30: Primer pokazatelja za obseg dela aktivnosti

Produktivnost aktivnosti

Prvo mersko izhodišče iz zahtev ključnega področja »Kvantitativno vodenje procesov« na 4. stopnji modela CMM zahteva spremljanje in nadzorovanje produktivnosti vseh aktivnosti. Podobno kot v prejšnjem primeru so entitete merskega izhodišča vse aktivnosti procesa.

Produktivnost je definirana kot razmerje med velikostjo rezultata posameznih aktivnosti in obsegom dela, ki je bil potreben za izvedbo aktivnosti. V naboru meritev obstajajo tako meritve velikosti vseh rezultatov faz kot obsega dela, vendar ni meritve, ki bi oboje povezovala. Zato je primerno, da nabor meritev dopolnimo z novo izpeljano meritvijo za produktivnost.

$$produktivnost = \frac{velikost\ rezultata\ aktivnosti}{obseg\ dela\ aktivnosti}$$

Enota izpeljane meritve je odvisna od enote za velikost rezultata in obseg dela. Ker je izpeljana meritev definirana zgolj kot razmerje, jo lahko uporabimo na več ravneh:

- za podrobno merjenje produktivnosti razvijalcev v okviru aktivnosti (koraka ali dela koraka), če za izhodišče vzamemo osnovne meritve spremembe velikosti in obsega dela;
- za merjenje produktivnosti faz ali ciklov, če uporabimo združene meritve velikosti in obsega dela po aktivnostih;

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- za merjenje produktivnosti procesnega ali podatkovnega vidika, če uporabimo združene meritve velikosti in obsega dela po vidiku;
- za merjenje produktivnosti izdelave izbranega izdelka, če uporabimo združene meritve po rezultatih;
- za merjenje celotne produktivnosti posameznikov, če uporabimo združene meritve velikosti in obsega dela po razvijalcih.

Izbira primerne ravni je odvisna od zahtev posameznega merskega izhodišča. V našem primeru so smiselne entitete faze in rezultati posameznih faz, zato je primerno uporabiti v združene meritve velikosti za rezultate faz in združene meritve obsega dela po fazah, ki že obstajajo v minimalnem naboru meritev.

Prikaz produktivnosti po fazah je pokazatelj, ki predstavlja razmerje med temi meritvami in merskim izhodišči. Po potrebi ga lahko ločimo še na procesni in podatkovni del. Entitete smo tokrat uporabili za izhodišče, zato se zagotovo ujemajo. Drugi del preverjanja usklajenosti meritve in merskega izhodišča nam pokaže, da je bilo treba izpeljano meritev minimalnemu naboru meritev zares dodati. Če namreč preverjamo razrede primernih obstoječih meritev in merskega izhodišča, ugotovimo neusklajenost. Tudi v primeru tega pokazatelja produktivnosti moramo napovedovati, pri čemer si pomagamo s povprečjem produktivnosti preteklih projektov. Za prikaz pokazatelja bi lahko uporabili podobno obliko kot v predhodnem primeru, le da primerjalno ne bi navajali planiranih vrednosti, temveč povprečne produktivnosti posameznih faz predhodnih projektov.

Kakovost na podlagi napak

Tudi v tretjem primeru smo za izhodišče izbrali mersko izhodišče, ki izhaja iz zahtev četrte ravni modela CMM. V okviru ključnega področja »Upravljanje za kakovost« obstaja mersko izhodišče, ki zahteva spremljanje in nadzorovanje kakovost s pomočjo razporeditve napak. Entitete merskega izhodišča so vse aktivnosti.

Primerne meritve za izpolnjevanje tega merskega izhodišča so očitno meritve odkritih (vnešenih) in odkritih (odpravljenih) napak pri izvedbi posameznih aktivnosti. Minimalni nabor obsega poleg osnovnih tudi združene meritve za faze in cikle, ki so razdeljene še na procesni in podatkovni vidik.

Če primerjamo meritve z merskim izhodiščem po kriteriju entitet in merskih razredov, ugotovimo v obeh primerih ujemanje. Za določanje pokazatelja je odločilen tretji del, ki ni tako formalen. Ker je mersko izhodišče dokaj splošno, dopušča več rešitev. Zato pri določanju pokazatelja upoštevamo zahteve splošnih merskih ciljev (enostavnost) in definiramo tak pokazatelj, ki bo na enostaven in nazoren način prikazal kakovost.

Primeren pokazatelj je prikaz rezultatov izpeljane meritve za **izkoristek odpravljanja napak** (ang. yield) po aktivnostih. Ta predstavlja razmerje med vsemi odpravljenimi napakami aktivnosti in napakami, ki so nastale ali so bile prinesene v izbrano aktivnost [Hum95].

$$izkoristek\ odpravljanja\ napak = \frac{100 * število\ odpravljenih\ napak}{število\ obstoječih\ napak}$$

Meritev izražamo v odstotkih in jo podobno kot produktivnost lahko uporabljamo na različnih ravneh. Pri tem uporabimo aktivnosti, v okviru katerih merimo napake: recimo izkoristek odpravljanja napak določenega koraka, cikla, izdelka in podobno.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Celotnega števila napak v aktivnosti ni mogoče natančno določiti, saj nekaterih napak nikoli ne odkrijemo. Za odkrite napake velja, da jih glede na trenutek vnosa odkrivamo postopoma ob izvajanju vseh nadaljnjih aktivnosti do konca razvoja. Ravno iz poznejšega odkrivanja napak lahko presojamo, koliko napak smo odkrili sproti in koliko jih ostaja za naslednje aktivnosti. To nam omogoča sklepati o tem, koliko napak bo na koncu ostalo v izdelku.

Primer 1:
Denimo, da določen izdelek razvijemo samo v treh fazah. Tabela 5.22 prikazuje število odkritih in odpravljenih napak po posameznih fazah in pripadajoče izkoristke odpravljanja napak ob predpostavki, da vse odkrite napake sproti odpravljamo.

	Napake 1. faze	Napake 2. faze	Napake 3. faze	Izkoristek po izvedbi faze
Ob izvedbi 1. faze	4 vnešene 4 odpravljene			1. faza: 4/4 = 100%
Ob izvedbi 2. faze	2 odkriti 2 odpravljeni	3 vnešene 3 odpravljene		1. faza: 4/6 = 67% 2. faza: 5/5 = 100%
Ob izvedbi 3. faze	1 odkrita 1 odpravljena	2 odkriti 2 odpravljeni	3 vnešene 3 odpravljene	1. faza: 4/7 = 57% 2. faza: 5/8 = 62% 3. faza: 6/6 = 100%
Skupaj	7/7 = 100%	5/5 = 100%	3/3 = 100%	

Tabela 5.22: Prvi primer določanja izkoristka odpravljanja napak

Po izvedbi prve faze izgledajo stvari idealno: izkoristek je 100%, saj smo odkrili in odpravili vse znane napake. Vendar ta rezultat ne pove veliko; pomemben je rezultat ob koncu razvoja. Ta pokaže, da smo ob izvedbi prve faze dejansko odkrili samo 57% vnešenih napak, ob izvedbi druge faze pa 62% napak. Če upoštevamo še neodkrite napake, ki skoraj zagotovo obstajajo, je rezultat še slabši. Tak rezultat kaže, da preverjanje kakovosti ob koncu prve in druge faze ni dovolj učinkovito. Čeprav smo se trudili sproti odpravljati odkrite napake, lahko z upoštevanjem prikazanih razmerij pričakujemo še približno tri neodkrite napake, kar je precejšen odstotek glede na skupno število odkritih napak.

Primer 2:
Tabela 5.23 prikazuje pozitiven vpliv natančnega preverjanja kakovosti na višjo kakovost končnega izdelka. Izkoristek je bistveno boljši, vendar moramo biti pozorni tudi na število vnešenih napak: če je vnešenih ogromno napak, bo v končnem izdelku teh ostalo še veliko tudi po zelo temeljitem preverjanju. V našem primeru lahko ne glede na višje izkoristke pričakujemo še štiri ali pet neodkritih napak. To pomeni, da kakovosti ne moremo zagotoviti samo z dobrim preverjanjem le-te, temveč jo je treba vgraditi s kakovostno izvedbo vseh aktivnosti.

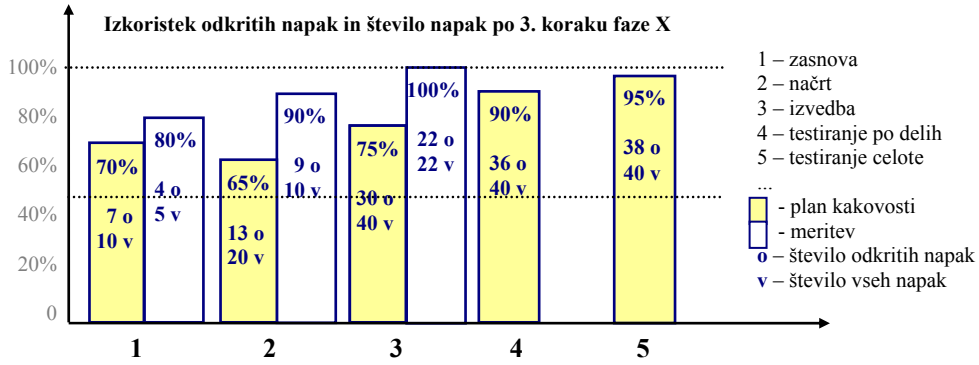
	Napake 1. faze	Napake 2. faze	Napake 3. faze	Izkoristek po izvedbi faze
Ob izvedbi 1. faze	10 vnešenih 10 odpravljenih			1. faza: 10/10 = 100%
Ob izvedbi 2.faze	2 odkriti 2 odpravljeni	13 vnešenih 13 odpravljenih		1. faza: 10/12 = 83% 2. faza: 15/15 = 100%
Ob izvedbi 3.faze	1 odkrita 1 odpravljena	3 odkrite 3 odpravljene	11 vnešenih 11 odpravljenih	1. faza: 10/13 = 77% 2. faza: 15/19 = 79% 3. faza: 16/16 = 100%
Skupaj	13/13 = 100%	16/16 = 100%	11/11 = 100%	

Tabela 5.23: Drugi primer določanja izkoristka odpravljanja napak

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Praktično to pomeni, da mora pokazatelj kakovosti poleg izkoristkov prikazati tudi obseg odkritih napak. Podatki o napakah morajo biti natančni, zato izhajamo iz osnovnih meritev (števila odkritih in odpravljenih napak vsake aktivnosti ter aktivnost, v kateri so bile te napake vnešene), ki jih z novo definirano združeno meritvijo ustrezno seštejemo. To združeno meritev moramo dodati v nabor meritev, omogoča pa združevanje napak glede na aktivnosti (faze), kjer so bile vnešene.

Za utemeljeno sklepanje o kakovosti potrebujemo čim več podatkov, ki jih zberemo šele proti koncu ciklov ali celotnega razvoja. Ob začetku razvoja so vrednosti izkoristkov lahko zelo zavajajoče, kar je mogoče razbrati iz razmerja med pričakovanim in dejansko odkritim številom napak. Zato je v okviru pokazatelja potrebno prikazati tudi planirane vrednosti iz plana kakovosti[30], ki so določene na podlagi povprečja meritev kakovosti preteklih projektov. Primerna raven prikaza tega pokazatelja so običajno koraki v okviru izbrane faze. Slika 5.31 prikazuje primer grafičnega prikaza pokazatelja kakovosti na podlagi merjenja napak za hipotetično fazo X, v kateri zasnujemo, izdelamo in testiramo določen izdelek.



Slika 5.31: Primer pokazatelja za kakovost izdelkov

5.5.4.2 Preostali del nabora pokazateljev in meritev

Na tak formalen način določimo tudi vse ostale pokazatelje, pri tem pa izpeljemo še ostale meritve. Podrobno opisovanje posameznih izpeljav vseh meritev in pokazateljev je preobsežno. Pokazatelji bodo zato samo kratko predstavljeni z drugega vidika v naslednjem razdelku. Celotni nabori pokazateljev in meritev, ki predstavljajo zadnji del vsebine našega merskega pristopa, so prikazani v dodatku B.2.5.

Naj pri tem opozorimo, da moramo na predstavljene nabore gledati celovito. V naboru so nekatera merska izhodišča povezana s pokazatelji, ki mogoče niso najustreznejši, vendar zadoščajo in omogočajo izpolnjevanje več merskih izhodišč hkrati. Pomemben splošni merski cilj je namreč tudi minimizirati nabor. S takim pristopom ne okrnemo funkcionalnosti merskega pristopa, saj posebna merska izhodišča omogočajo vse nabore ustrezno dopolniti, če je to v okviru projekta potrebno. Novo definirani pokazatelji in meritve se opišejo na organizacijski ravni. S tem postanejo dosegljivi tudi ostalim projektom, čeprav niso del stalnega nabora pokazateljev ali meritev.

5.5.4.3 Nabor pokazateljev z vidika analiz

Pokazatelje lahko prikažemo tudi z drugega vidika, ki nazorno umesti le-te v celovit merski pristop. V podpoglavju 4.3 smo predstavili štiri vrste analiz našega merskega procesa:

[30] Plan kakovosti vsebuje pričakovane obsege odkritih in odpravljenih napak po aktivnostih, iz česar lahko določimo tudi pričakovane izkoristke.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- ocenjevanje,
- analizo izvedljivosti,
- analizo performans in
- napovedovanje.

V razdelkih 4.3.1 in 4.3.4 smo natančno definirali dve izmed štirih analiz (ocenjevanje in napovedovanje). To smo storili predvsem na podlagi izbranih razredov meritev, izkušenj in zahtev procesa razvoja PO. Vendar obstajajo tudi formalni razlogi, ki zahtevajo tak nabor ocen. Zapisani so v stalnih merskih izhodiščih, ki določajo še druge analize merskega procesa.

Rezultati izvedbe vsake analize so pokazatelji, ki so različnih oblik. V primeru ocenjevanja so to vrednosti, izražene v ustreznih enotah, medtem ko pokazatelje drugih vrst analiz običajno prikazujemo grafično. To je povezano z namenom posameznih vrst analiz: ocene so namenjene predvsem planiranju, drugi pokazatelji pa služijo za nazoren prikaz stanja ali razmerij. Ko vodja procesa in merjenja pripravi predstavitev rezultatov analize vodstvu, pokazatelje z dodatnimi pojasnili in raziskavami dopolni v informacijske izdelke.

Naš merski pristop temelji predvsem na dveh vrstah analiz: ocenjevanju in napovedovanju. Zaradi tega smo v opisu merskega procesa ocene in napovedi natančno vsebinsko definirali in opisali postopek njihovega določanja, medtem ko smo za ostali dve vrsti analiz prikazali samo način grafične predstavitve. Kot sledi iz podpoglavja 4.3 sta ocenjevanje in napovedovanje zastavljeni tako, da pokrivata vse razrede merjenja. Temeljita predvsem na osnovnih in združenih meritvah. V našem pristopu pokrivata glavnino stalnih merskih izhodišč.

Pokazatelji analize izvedljivosti in performans so zato v okviru stalnega nabora redkeje zastopani, vendar naš pristop omogoča, da jih lahko uporabimo v okviru posebnih merskih izhodišč. V nadaljevanju bomo z vidika analiz kratko predstavili nabore pokazateljev, z njimi pa tudi združene in izpeljane meritve, ki izhajajo iz stalnega nabora merskih izhodišč.

Ocene

Ocenjevanje je zelo pomemben del natančnega planiranja, ki pomembno vpliva na učinkovitost razvoja. Temu je posvečena posebna pozornost v prilagodljivem procesu razvoja PO. Temeljni namen procesa je sicer omogočiti učinkovito delo majhni organizaciji s prilagajanjem različnim zahtevam, vendar je struktura procesa zastavljena tako, da temelji na natančnem planiranju. Učinkovito ocenjevanje je prvi korak uspešnega merjenja.

V skladu s tem so definirani tudi pokazatelji analiz ocenjevanja. Ti so podrobno opisani v podpoglavju 4.3.1 in povzeti v dodatku B.2.5.1, zato jih ne bomo posebej naštevali. Čeprav so ocene predstavljene samo kot vrednosti in zapisane v ustreznih planih, predstavljajo pomembno podlago za vse druge pokazatelje. Napovedi so recimo izpeljane ravno na podlagi ocen in rezultatov preteklih projektov, medtem ko so tako pokazatelji izvedljivosti kot performans običajno prikazani primerjalno s plani.

Pokazatelji izvedljivosti

Izvajanje analiz izvedljivosti se izvaja po aktivnostih planiranja. Z njim presojamo, če so plani realistični. V razdelku 4.3.2 smo prikazali oblike grafičnih pokazateljev, ki jih uporabljamo v ta namen. Vendar skuša naš merski pristop vodji procesa in merjenja čim bolj poenostaviti in olajšati delo, zato omogoča, da vse obvezne analize izvedljivosti izvede zgolj s pomočjo pokazateljev napovedi. V okviru stalnega nabora pokazateljev zato ni predvidenih posebnih pokazateljev izvedljivosti.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

To seveda ne pomeni, da v našem merskem pristopu posebni pokazatelji izvedljivosti niso potrebni, temveč samo to, da niso nujni. V okviru posebnih merskih izhodišč lahko projekt predvidi posebne zahteve za preverjanje izvedljivosti. Tipični primeri pokazateljev omogočajo primerjavo različnih scenarijev, recimo:

- projekcije stroškov po aktivnostih glede na različno število razvijalcev ali
- pokazatelji porabe časa glede na različne obsega potrebnega obsega dela.

Pokazatelji performans

Tudi za pokazatelje performans velja podobno. Naš merski pristop temelji na uporabi avtomatiziranih analiz, ki omogočajo celovit vpogled in hitro ukrepanje. S tega stališča tudi pokazatelje performans v pretežno nadomeščajo napovedi, seveda predvsem v okviru stalnega nabora pokazateljev.

Vendar brez pokazateljev performans ne gre. Čeprav napovedi pokrivajo vse razrede meritev, omogočajo predvsem globalen pogled, ki seže do ravni posameznih faz. V kolikor želimo natančnejši vpogled, moramo izvesti analizo performans. Ta je za vodjo procesa in merjenja sicer časovno potratna, ker se izvaja na ukaz, vendar omogoča natančno raziskovanje z natančnim definiranjem zahtev.

Za izpolnjevanje stalnega nabora merskih izhodišč zadošča, če določimo naslednje pokazatelje performans:

- status merskih entitet (aktivnosti, izdelkov in virov),
- produktivnost po aktivnostih,
- izkoristek odpravljanja napak in
- strošek kakovosti.

a) Status merskih entitet (izdelkov, aktivnosti, virov)

Osnovni pokazatelj performans je pregled statusa merske entitete, ki omogoča podroben pregled obstoječih podatkov o izbrani merski entiteti. Gre za splošen pokazatelj, ki se lahko uporablja nad različnimi vrstami entitet in na različnih ravneh. Pokazatelj ni vezan na specifično meritev, temveč zgolj zbere in prikaže vse znane podatke o izbrani merski entiteti. Te zbere iz planov, opisov projekta in zbranih merskih rezultatov. Pokazatelj prikazuje status vseh treh vrst merskih entitet:

- aktivnosti,
- izdelkov in
- virov.

Če je merska entiteta **aktivnost**, pokazatelj prikaže vse podatke o tej aktivnosti. Če gre za korak ali celo del koraka, bodo prikazane osnovne značilnosti (oznaka, izvajalec, planiran čas izvedbe) in rezultati osnovnih meritev (obseg dela, čas začetka in konca, velikost rezultata, število odkritih napak in podobno), v kolikor obstajajo. Če gre za večjo aktivnost (recimo fazo ali cikel), bodo prikazane vse podrobnosti njenih sestavnih delov in tiste združene meritve, ki so definirane; novih združenih meritev v ta namen ne predvidimo. Na podlagi statusa lahko ugotovimo, katere aktivnosti so zaključene in katere se trenutno izvajajo.

V kolikor je merska entiteta **izdelek** (rezultat faze ali cikla), pokazatelj pokaže podatke o njem. V tem primeru prikaže osnovne značilnosti (naziv, sestavni del česa je, razvijalci), z

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

izdelkom povezane aktivnosti ter vse meritve, iz katerih je razvidno, če je izdelek končan in kako je nastajal (spremembe velikosti). Če gre za izdelke, ki vsebujejo več drugih izdelkov, predstavimo vsako izmed teh in tiste združene meritve, ki so v naboru meritev.

Preverjamo lahko tudi **vire**. Za člane projekta lahko določimo vloge in njihovo sodelovanje pri projektu: aktivnosti, pri katerih so ali bodo sodelovali, izdelke in izmerjene rezultate. Na podoben način lahko opišemo tudi stroške.

b) Produktivnost po aktivnostih
Spremljanje produktivnosti je potrebno zaradi ugotavljanja težav pri določeni aktivnosti, omogoča pa tudi napovedovanje aktivnosti. Pokazatelj je bil predstavljen v okviru razdelka 5.5.4.1.

c) Izkoristek odpravljanja napak
Izkoristek odpravljanja napak je eden izmed dveh ključnih pokazateljev kakovosti. Tudi ta pokazatelj je bil predstavljen v okviru razdelka 5.5.4.1.

d) Strošek kakovosti
Drugi pomemben pokazatelj kakovosti je strošek kakovosti, ki ga določamo iz razmerja med obsegom dela aktivnosti razvoja, aktivnosti zagotavljanja kakovosti in aktivnosti odpravljanja napak. V ta namen moramo definirati nove združene meritve na podlagi združevalne strukture po funkcionalnosti. Z njimi v okviru posameznih faz ali ciklov zberemo obseg dela aktivnosti razvoja (tipično za načrtovanje in izvedbo), aktivnosti zagotavljanja kakovosti (ponovne preglede, zunanja preverjanja in različna testiranja) ter obseg dela odpravljanja napak, ki jih zberemo iz osebnih meritev posameznikov v ustreznih aktivnostih.

Nato določimo tri izpeljane meritve. Prva določa razmerje med obsegom dela zagotavljanja kakovosti in obsegom dela razvoja, druga razmerje med obsegom dela odpravljanja napak in obsegom dela razvoja. Tretja izpeljana meritev je razmerje med obsegom dela odpravljanja napak in obsegom dela zagotavljanja kakovosti.

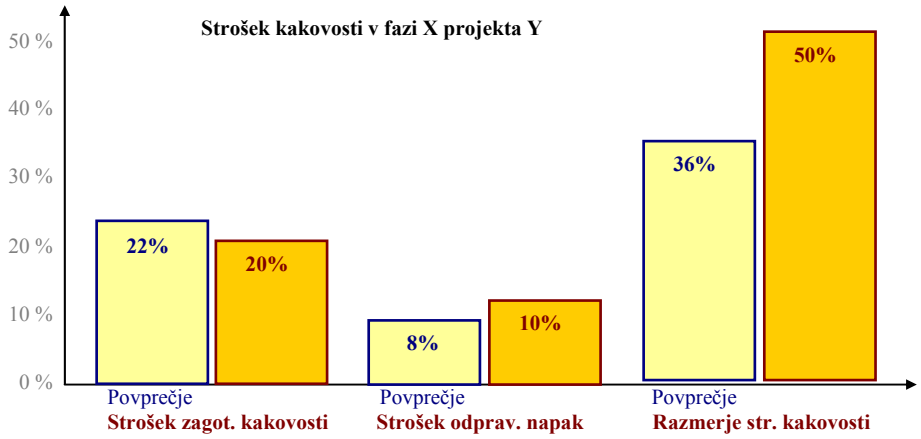
$$\textit{strošek zagotavljanja kakovosti} = \frac{\textit{100 * obseg dela aktivnosti zagotavljanja kakovosti}}{\textit{obseg dela aktivnosti razvoja}}$$

$$\textit{strošek odpravljanja napak} = \frac{\textit{100 * obseg dela odpravljanja napak}}{\textit{obseg dela aktivnosti razvoja}}$$

$$\textit{razmerje stroškov kakovosti} = \frac{\textit{100 * obseg dela odpravljanja napak}}{\textit{obseg dela aktivnosti zagotavljanja kakovosti}}$$

Vse tri izpeljane meritve prikažemo v okviru istega pokazatelja, izražamo pa jih v odstotkih (slika 5.32). Za primerjavo prikažemo tudi povprečne vrednosti preteklih projektov ter določimo meje, v katerih naj bi se vrednosti predvidoma gibale.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Slika 5.32: Primer pokazatelja za stroške kakovosti

Iz pokazatelja je razvidno, če preverjanju kakovosti posvečamo dovolj pozornosti ter kolikšen je učinek tega. V kolikor je razmerje stroškov kakovosti visoko, je treba zagotavljanju kakovosti posvetiti več časa. Če je to razmerje ugodno, a je strošek zagotavljanja kakovosti zelo nizek, lahko to pomeni, da napak sploh ne odkrivamo, ker mogoče ne izvajamo ustreznega testiranja. Vse tri vrednosti si je treba razlagati kot celoto in v primerjavi z vrednostmi preteklih projektov.

Analize performans so predstavljene v dodatku B.2.5.3.

Napovedi

Napovedovanje v našem merskem pristopu zaseda osrednjo vlogo, ki omogoča hiter in učinkovit vpogled. Razumemo ga predvsem kot nadgradnjo ocenjevanja, ki sproti projecira glavne kazalce razvoja projekta glede na trenutno dosežene vrednosti. Nabor pokazateljev je tak, da daje celovito sliko stanja projekta. S tega stališča v veliki meri nadomešča analize performans, ki so bolj usmerjene in natančne, a tudi bolj časovno potratne. Glede na stalna merska izhodišča napovedovanje v celoti nadomešča tudi analize izvedljivosti.

V napovedih se dejansko skrivajo izkušnje preteklih projektov. Čeprav so pokazatelji napovedi predstavljeni na enostaven način, so ravno tipična razmerja izmerjenih vrednosti preteklih projektov tista, ki določajo predvidene vrednosti. Tipična primera napovedi sta recimo:

- pokazatelj razmerij med velikostjo in trajanjem preteklih projektov ali
- pokazatelj razmerij med vstavljenimi in odkritimi napakami po aktivnostih v preteklih projektih.

Pomembno je, da za vse pokazatelje napovedi zadoščajo osnovne in združene meritve, ki se že nahajajo v minimalnem naboru meritev. O napovedih je bilo veliko povedanega v razdelku 4.3.4, v skrajšani obliki pa tudi v dodatku B.2.5.4. Zato jih ne bomo posebej predstavljali.

5.5.4.5 Preverjanje ustreznosti določenih naborov

S tem smo zaokrožili prikaz vsebine našega merskega pristopa. Opisani pristop popolno in natančno določa merjenje v majhni organizaciji, ker smo ga določili na podlagi jasnih izhodišč in z dosledno uporabo ustreznega postopka. To dokazujejo tudi rezultati primerjave, ki so prikazani v dodatku B.2.6.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

V dodatku smo določili povezavo med stalnimi merskimi izhodišči in predstavljenimi pokazatelji. Dodatek B.2.6.1 prikazuje povezavo z vidika merskih izhodišč. Iz tega lahko razberemo, da je vsako stalno mersko izhodišče ustrezno pokrito s vsaj enim pokazateljem. V dodatku B.2.6.2 je povezava predstavljena obratno: za vsak pokazatelj so prikazana merska izhodišča, ki ga ta pokriva.

Čeprav je vsebina našega merskega pristopa dokaj podrobno predstavljena, se v njej zrcalijo jasni merski koncepti, ki izhajajo iz naših začetnih zahtev (glej razdelek 4.1.2), prvotne zasnove [Rož99] in značilnosti merskega procesa. Merski pristop vsebinsko pokriva vsa potrebna področja, vendar so nabori še vedno preprosti, usklajeni in majhni. S tem omogočajo učinkovito izvajanje zastavljenega merskega procesa v majhnih organizacijah za razvoj IS.

5.6 Skupni repozitorij merskih podatkov

V uvodnem delu poglavja smo omenili še četrto raven vsebinske predstavitve merskega pristopa: merski repozitorij. Podrobna predstavitev te ravni presega zastavljene okvire disertacije, zato jo ob zaključku predstavitve merskega pristopa omenjamo samo na kratko.

Merski repozitorij nam predstavlja fizično implementacijo meta modela, ki smo ga temeljito predstavili v podpoglavju 5.3. Od tod sledi osnovna zahteva, da ta omogoča hranjenje in uporabo vseh vrst merskih podatkov na ustrezen način (recimo hranjenje podatkov na različnih ravneh, hranjenje zgodovinskih podatkov in podobno). Vendar naš pristop predvideva še večjo funkcionalnost merskega repozitorija. To je razvidno iz naše predstave o repozitoriju v tipični majhni organizaciji za razvoj IS (razdelek 3.4.2.8) ter predvsem iz značilnosti merskega procesa (podpoglavje 4.5).

V nadaljevanju navajamo glavne značilnosti merskega repozitorija kot jih predvideva naš merski pristop:

- V organizaciji obstaja centralen merski repozitorij, v katerem hranimo podatke na dveh ravneh: na ravni organizacije ter na ravni posameznih projektov.
- V repozitoriju hranimo različne vrste podatkov, ki so med seboj povezani:

  - Na ravni organizacije hranimo:
    - opise standardnih postopkov (standardov);
    - plane organizacije;
    - podatke o osebju;
    - podatke o procesu razvoja PO na ravni organizacije (s prilogami);
    - podatke o merjenju (opis merjenja, združeni merski podatki);
    - dokumente, predloge;
    - učno gradivo, izkušnje, opisi uporabe in drugo.

  - Na ravni projektov pa hranimo:
    - plane projekta;
    - podatke o procesu razvoja PO na ravni organizacije (s prilogami);
    - podatke o merjenju (opis merjenja, združeni merski podatki);
    - rezultate razvoja;
    - opis konfiguracije;
    - poročila in drugo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Pomemben del vsebine zajemajo podatki o podatkih, torej o sami vsebini repozitorija. Predvidevamo, da se nekateri izmed teh določijo s procesiranjem ostalih (recimo združevanjem podatkov kot v primeru tabele tipičnih razmerij obsega dela aktivnosti) in služijo za učinkovitejše delo z repozitorijem (glej [Rož99]).
- Del podatkov merskega dela repozitorija se nanaša na celotno organizacijo, del na trenutno izvajajoče projekte in del na pretekle projekte.
- Zaradi različnih vrst podatkov si merski repozitorij tipično predstavljamo kot strežnik, ki vsebuje dokumente različnih vrst. Do različnih vrst dostopamo na enoten način, uporabljamo pa jih različno.
- Dostop do različnih vrst podatkov v repozitoriju je različen, vendar je repozitorij organiziran kot PB za tisti del podatkov, kjer je to smiselno. PB je podprta s sistemom za upravljanje PB.
- Repozitorij je tudi sicer avtomatiziran s posebnim IS, ki podpira delovanje organizacije v več segmentih:
  - omogoča dostop do podatkov v skladu s pravicami članov,
  - omogoča izvajanje pregledov in generiranje ustreznih poročil,
  - omogoča vzdrževanje podatkov ter predvsem
  - podpira aktivnosti merskega procesa, kot so:
    - zajem merskih podatkov,
    - preverjanje kakovosti podatkov,
    - izvajanje ocenjevanja z uporabo statističnih metod (linearne regresije),
    - nenehen prikaz aktualnih napovedi in
    - izvedbo drugih analiz s prikazom grafičnih pokazateljev .
- Repozitorij je organiziran tako, da podpira uporabo razvojnih orodij in metodologij (lahko je organiziran tudi kot del razvojnega okolja). Poleg tega tudi dodatno vsebuje določena orodja, ki podpirajo naštete aktivnosti (recimo kot v [Pfl91, Hei99]).
- Repozitorij ustrezno ureja dostopne pravice za člane organizacije (uporabniška imena in gesla). V skladu s svojimi pooblastili lahko ti dostopajo do ustreznih podatkov in izvajajo aktivnosti nad podatki različnih delov repozitorija.

Centralni repozitorij predvidimo zato, da bi na tak način zagotovili povezanost in učinkovit dostop do podatkov vseh področij. V majhni organizaciji je tak repozitorij tudi smiselno uporabljati, ker ni (tolikšnih) težav z veliko količino podatkov. V bistvu pa bi lahko uporabili tudi drugačne rešitve. Primeren pristop je recimo tudi ločeno hranjenje različnih delov podatkov, ki se jih združuje po potrebi z uporabo podatkovnih skladišč (glej [Mah01]).

Ustrezna izvedba repozitorija je zelo pomembna za naš merski pristop, ki ustreza strogim zahtevam. Če naj posameznik merjenje operativno izvaja na relativno enostaven način (recimo z uporabo jasnih merskih pravil in majhnega nabora osnovnih merjenj), se mora zahtevnost skrivati v zahtevnejšem analiznem delu. Naš pristop v tem delu predvideva učinkovit IS merskega repozitorija. Ta mora poleg zajema merskih podatkov podpirati izvedbo različnih analiz, tako računskega dela (recimo ocenjevanja in napovedovanja z uporabo statističnih metod) kot priprave in prikaza ustreznih (grafičnih) pokazateljev. S tega stališča je zanimiv tudi zajem merskih podatkov z vnosnimi maskami, ki sproti preverjajo kakovost podatkov. Preko repozitorija se spremljajo tudi konfiguracije in oddajajo rezultati posameznih faz.

Ustrezna izvedba repozitorija omogoči, da lahko izvajalce merjenja razbremenimo in ga z njihovega vidika naredimo še enostavnejše. To je tudi ugotovitev, ki smo jo s predstavitvijo repozitorija želeli pokazati. Podrobnejše opisovanje presega začrtan okvir disertacije.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5.7 Zaključek poglavja

Z vsebinskim delom zaključujemo predstavitev merskega pristopa. Vsebino smo si ogledali na več ravneh in pri tem merski pristop dokončno določili. Skupaj z definiranim merskim procesom sedaj predstavlja celovit model za kvantitativno spremljanje procesa razvoja PO, ki je primeren za uporabo v majhnih organizacijah za razvoj IS ob predpostavki, da te uporabljajo prilagodljiv proces razvoja PO.

V okviru poglavja smo na vseh ravneh predlagali več izvirnih rešitev, s katerimi smo dosegli naslednje **ključne rezultate**:

- Definirali smo **merski sestav**, ki omogoča primeren opis vsebine merjenja. Pri tem smo merski sestav, ki je definiran v informacijskem modelu pristopa PSM, dopolnili s strukturo združenih meritev.
- Opisali smo **meta model na ravneh organizacije, projekta v teku in preteklih projektov**. V ER diagramih smo predvideli strukturo, ki podpira vse rešitve merskega procesa (recimo zbiranje merskih podatkov posameznikov, izvajanje napovedovanja, uporabo združenih rezultatov preteklih projektov in podobno).
- Predlagali smo **kombinirano metodo za določanje nabora** (vseh vrst) **meritev**. Prednost metode je v tem, da povzema dobre rešitve metod GQM in MQG: upošteva informacijske potrebe organizacije, vendar v skladu z našim merskim pristopom temelji na predpisanem naboru osnovnih meritev. Kombinirana metoda omogoča tudi določanje drugih naborov.
- Rezultati našega pristopa in uporabe kombinirane metode so **nabori merskih izhodišč, merskih entitet, atributov, meritev in pokazateljev**. V skladu z določili merskega procesa je definiran stalen nabor osnovnih meritev za posameznika, na podlagi teh pa z združevanjem ali izpeljavo določimo še preostale merske rezultate. Te uporabljajo pokazatelji, ki smo jih tudi predstavili ob koncu poglavja.
- Ob koncu poglavja smo okvirno predstvili še **merski repozitorij**. Predvidevamo, da bo ta avtomatiziral zahtevne dele merjenja in kot tak prispeval k večji uporabnosti merskega pristopa.

To je pomemben dosežek naše metode, ki je v skladu s principom iz podpoglavja 3.4. Z ustrezno izvedbo repozitorija poskrbimo, da je operativno izvajanje merjenja za člane projekta pravzaprav zelo preprosto, čeprav dosega visoke zahteve:

- Pri planiranju merjenja lahko zgolj privzamejo stalen nabor merskih izhodišč, iz katerih izhaja tudi nabor pokazateljev,
- Posamezniki ob vsaki aktivnosti izmerijo standardne stvari: čas trajanja, stroške, število odkritih in odpravljenih težav ter velikost rezultata, v kolikor so ga razvili.
- Vodja procesa in merjenja mora poskrbeti, da vsi udeleženi člani oddajo svoje plane izvedbe, na podlagi katerih z orodji repozitorija določi globalne in natančne plane.
- Napovedovanje se izvaja avtomatsko, brez truda vodje procesa in merjenja. Za ostale analize mora uporabljati orodja, ki avtomatizirajo pripravo, izvedbo in poročanje stalnih analiz.

Kot smo pokazali v sklepnem delu poglavja opisani nabori meritev ustrezno podpirajo vsa stalna merska izhodišča, ki izhajajo iz zahtev četrte stopnje modela CMM. Ti nabori tako predstavljajo otipljivo rešitev našega problema.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

6. ZAKLJUČEK

V zadnjem delu disertacije bomo najprej praktično preverili, kako predlagani model ustreza dejanskim potrebam majhnih organizacij za razvoj IS. Nato bomo povzeli rezultate in prispevke k znanosti, ki jih disertacija prinaša, ter predlagali možnosti za nadaljnje raziskave na tem področju. Disertacijo bomo zaključili s sklepno besedo.

6.1 Preizkus predlaganih rešitev v praksi

Predlagane rešitve smo želeli praktično preizkusiti tudi v realnem okolju, torej v majhni organizaciji za razvoj IS. Da bi popolnoma preverili predlagan model, bi morali v organizaciji uvesti najprej prilagodljiv proces razvoja PO ter nato še merjenje na opisan način. Kmalu se je pokazalo, da to žal ne bo mogoče. Oviri sta vsaj dve:

- Naš model temelji na predpostavki, da majhna organizacija z vgrajenim merskim pristopom ustreza zahtevam četrte stopnje modela CMM, torej mora njen proces razvoja PO pred uvajanjem merjenja ustrezati na tretji stopnji. Takšne organizacije trenutno v Sloveniji ni. Če bi želeli izbrano organizacijo dvigniti na takšno stopnjo zrelosti, bi potrebovali zelo veliko časa, saj po ugotovitvah SEI organizacija v povprečju potrebuje za pridobitev ene stopnje od 17 do 28 mesecev [SEI02].
- Poleg tega bi organizacija morala popolnoma povzeti naš model, kar pomeni smelo odločitev za organizacijo, ki posluje na konkurenčnem trgu razvoja PO. Uvajanje procesa in merjenja namreč zahteva veliko sprememb in posledično veliko časa [Ive03], pa tudi učinek bi se pokazal šele čez nekaj časa. Za objektivno oceno najpomembnejših merskih značilnosti našega modela bi morali obstajati tudi ustrezni podatki preteklih projektov, kar je malo verjetno. Realneje bi bilo pričakovati, da bo popolna funkcionalnost našega modela zaživela šele čez nekaj projektov.

Glede na časovne omejitve disertacije tak pristop ni izvedljiv. Zato smo morali izbrati tako vrsto preizkusa modela v praksi, ki bo omogočal ustrezno preveriti naš model, vendar ga lahko izvedemo v zastavljenem časovnem okviru.

Obiskali smo več majhnih organizacij za razvoj PO in vodstvu (ali članom z ustrezno vlogo v organizaciji) okvirno predstavili problematiko disertacije brez predlaganih rešitev. Z njimi smo se nato pogovorili o njihovih merskih izkušnjah in na koncu te preverili z vprašalnikom.

Vprašalnik (glej dodatek C) je sestavljen tako, da z vprašanji najprej zajame osnovne podatke o organizaciji, okvirno preveri poznavanje področja ter nato še mnenje o procesu razvoja PO v majhni organizaciji, primernih postopkih merjenja in vsebini merjenja. Odgovori so ponujeni vnaprej in jih mora anketiranec samo ovrednotiti z oceno od 1 (najslabše) do 5 (najboljše). Vprašanja pokrivajo vsa pomembnejša področja iz disertacije. Odgovori so seveda ponujeni tako, da dopuščajo tudi možnost, da anketiranec predlaga drugačne rešitve od tistih, ki jih v disertaciji predlagamo mi. Poleg tega so nekatere ideje zajete v več odgovorih, tako da vprašalnik omogoča tudi navzkrižno preverjanje. Nenazadnje ima anketiranec pri

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

vsakem vprašanju možnost, da dopiše svoje odgovore. Tak pristop omogoča hitro izpolnjevanje vprašalnika in objektivno zajame mnenje anketiranca.

Obiskali smo štiri organizacije za razvoj PO. Iz pogovorov smo razbrali, da se se vse zavedajo pomena kakovosti v procesu razvoja PO, čeprav nobena izmed njih ne dosega višjih stopenj zrelosti po modelu CMM. Da bi ustrezneje zajeli vtis o našem delu, smo izbrali precej raznolike organizacije:

1. **Prva organizacija** je s 65 razvijalci že nekoliko nad zgornjo mejo majhne organizacije. Ukvarja se izključno z razvojem PO in je v zagotavljanje kakovosti že precej investirala: imajo certifikat ISO-9001, proces razvoja PO pa so nadgrajevali tudi s pomočjo modela CMM.
2. **Druga organizacija** ima še nekoliko več zaposlenih, vendar je razvoj PO le ena od njihovih dejavnosti. Programsko opremo razvija 35 zaposlenih. Na podlagi načel ISO-9001 in druge stopnje modela CMM so s pomočjo naše fakultete oblikovali svoj proces razvoja PO (MAP [Keb02]), ki ga uspešno uporabljanjo v praksi.
3. **Tretja organizacija** je zelo majhna. Ima samo 13 zaposlenih, ki se ukvarjajo izključno z razvojem PO. Za razliko od zgoraj neštetih so v tej organizaciji zares izrazite značilnosti majhnosti: omejeni viri in velika odvisnost od zunanjih vplivov. Za zagotavljanje kakovosti skrbijo na neformalen način v skladu z načeli ISO-9001.
4. **Četrta organizacija** se ukvarja z razvojem (projektiranjem, avtomatizacijo) različnih sistemov, vendar v okviru tega razvijajo tudi potrebno PO. Zaposlenih je približno 50 ljudi, od katerih se jih le 10 ukvarja z razvojem PO. Za zagotavljanje kakovosti skrbijo z uporabo neformalnih pristopov, ki pa so usklajeni s standardom ISO-9001.

V nadaljevanju bomo predstavili glavne ugotovitve, ki izhajajo iz vprašalnika in pogovorov. Preverjali smo naslednja področja:

a) Splošno poznavanje modelov za zagotavljanje kakovosti in posebej modela CMM

Odgovori so pokazali, da imajo vsi anketiranci vsaj nekaj izkušenj s standardom ISO-9001 oziroma ga na tak ali drugačen način uporabljajo. Polovica pozna tudi model CMM, medtem ko je poznavanje drugih modelov slabše.

Dve izmed štirih organizacij imata praktične izkušnje z uporabo modela CMM. Popolnoma v skladu z izhodiščno idejo naše disertacije menita, da je model popoln in zelo natančen, vendar zahteven in kot tak ne najbolj primeren za majhno organizacijo. Organizaciji tudi menita, da njihove trenutne potrebe zadovoljujejo že ključna področja druge oziroma tretje stopnje modela CMM.

b) Mnenje o značilnostih procesa razvoja PO v majhni organizaciji

V okviru značilnosti procesa se je anketirancem zdelo najpomembneje, da je proces natančno in ustrezno definiran, da obvladuje spremembe zahtev ter da je prilagojen zmožnostim majhne organizacije. Pomembno je, da je proces enostaven za uporabo v praksi, ne pa, da je enostavno definiran ali povzet po znanem modelu. S tem anketiranci popolnoma potrjujejo naše teze.

Pri procesu se jim zdi še pomembno, da so vloge ustrezno določene, vendar ni smiselno, da bi bili člani specializirani samo za določene vloge. Za postopke morajo biti znani

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

rezultati, dokumentiranje naj bo omejeno, a ustrezno. Potrebno je izkoriščati tudi prednosti avtomatizacije, še zlasti pri podpornih aktivnostih. Zdi se jim zelo pomembno, da je mogoče plane sproti spreminjati zaradi sprememb zahtev ter to, da proces poleg razvoja podpira tudi uvajanje razvitih izdelkov v uporabo. Usposabljanje naj bo uvedeno na neformalen način. To so pravzaprav ključni poudarki našega prilagodljivega procesa razvoja PO.

Vendar so bili odgovori nepričakovano raznoliki pri potrebi po preverjanju kakovosti izdelkov, nekateri pa celo dvomijo v potrebo po enotnem procesu za vse projekte organizacije. Oboje sodi v domeno tretje in četrte stopnje modela CMM, ki je za anketirane organizacije očitno prezahtevna.

c) Poznavanje pristopov in metod za opis merjenja PO

To se ujema z ugotovitvijo, da anketiranci relativno slabo poznajo standardne pristope in metode s področja merjenja (recimo metodi GQM in MQG, standarde ISO, pristop PSM). Izjema je le prva organizacija, kjer poznajo pristop PSM.

d) Mnenje o izvajanju merjenja v majhni organizaciji za razvoj PO

Preverili smo tudi, kaj anketiranci menijo o merjenju procesa razvoja PO. Zdi se jim pomembno, da je merjenje natanko planirano in vsebovano v procesu razvoja PO. Potrebno je meriti vse pomembne aktivnosti z omejenim naborom meritev. Del podatkov naj se meri, možno pa je podatke tudi določati s sestavljanjem. Zelo pomembne so ustrezne analize, ki morajo temeljiti na preteklih podatkih, ki so zbrani v skupnem repozitoriju. Zanje naj skrbi posebej usposobljen merski analitik. V tem delu odgovori potrjujejo teze naše disertacije.

Vendar se odgovori na treh področjih tudi razlikujejo od naših tez. Anketiranci menijo, da ni potrebno, da vsi projekti povzemajo organizacijski način merjenja, temveč naj merjenje izhaja iz posebnosti projekta. Zdi se jim tudi nepraktično, da bi merjenje izvajali čisto vsi izvajalci za vsako aktivnost. Odveč se jim zdi tudi preverjanje kakovosti zbranih merskih podatkov. Vse to so dejansko zahteve tretje in četrte stopnje modela CMM, ki ga organizacije ne dosegajo in se jim zato ne zdi potreben. S tega stališča se jim zdi odveč tudi uvedba osebnega procesa razvoja PO, ki zahteva merjenje vsake aktivnosti posameznika.

e) Mnenje o vsebini merjenja

Anketiranci se strinjajo, da je potrebno meriti vse projekte, druge aktivnosti organizacije in pretekle projekte. Meriti si želijo aktivnosti, izdelke in vire za celoten projekt in za njegove sestavne dele. Enako predvideva tudi naš merski pristop.

Vendar anketiranci s svojimi trenutnimi merskimi postopki vsega ne zmorejo meriti. Nekateri ugotavljajo, da razen projektov ne znajo meriti drugih aktivnosti organizacije, drugi pa zmorejo meriti samo značilnosti celotnega projekta, ne pa tudi njegovih sestavnih delov. Na teze disertacije to ne vpliva, saj so se v predhodnem odgovoru vsi strinjali, da je potrebno meriti vse ravni.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Anketiranci so morali tudi našteti meritve, ki se zdijo potrebne in konkretno izvedljive v majhni organizaciji. Meritve, ki smo jih predvideli v okviru disertacije (trajanje aktivnosti, doseganje mejnikov, obseg dela, stroški, velikost izdelka, kakovost izdelka s pomočjo napak), so pri praktično vseh anketirancih dobile veliko podporo. Anketirancev niso zmedli navzkrižno povezani odgovori, so pa odgovarjali s stališča meritev, ki jih sami izvajajo v organizaciji. V drugi organizaciji recimo merijo natanko tiste značilnosti, ki jih predlagamo za merjenje tudi v našem merskem modelu.

Vendar med odgovori podjetij lahko odkrijemo nekaj razlik. Tako se nekaterim ne zdi pomembno spremljati velikosti izdelkov, temveč njegovo funkcionalnost. Pri tem zaradi prevelike zahtevnosti funkcionalnosti ne merijo klasično s funkcijskimi točkami [Alb83], temveč s pomočjo deleža pokritih zahtev[31]. Nekaterim se zdi pomembno spremljati poleg napak tudi nekatere značilnosti kakovosti izdelka (recimo vzdrževalnost ali zanesljivost), vendar za to uporabljajo nezanesljive subjektivne metrike. To je bil tudi razlog, zakaj teh meritev nismo uvrstili v naš nabor.

f) Mnenje o analizah merskih podatkov (pokazateljih)

Glede različnih vrst analiz so se anketiranci strinjali, da sta zagotovo potrebna ocenjevanje in (nekoliko manj) analiza performans. Analiza izvedljivosti se jim ni zdela tako pomembna. Za napovedovanje so menili, da je kot idejna rešitev zelo uporabno (bolj kot analiza performans), vendar težavno z vidika izvedbe. S tem so potrdili tezo, ki jo zagovarjamo tudi v disertaciji: napovedovanje je primerno, le če je ustrezno avtomatizirano.

Anketiranci so potrdili tudi nabor ključnih pokazateljev, ki smo ga predlagali v okviru našega merskega modela. Odstopanji sta bili le dve: v enem podjetju ne spremljajo posebej velikosti izdelka v primerjavi s planiranimi vrednostmi, ker tudi sicer velikosti ne merijo. Drugi anketiranec je imel pomisleke glede potrebnosti spremljanja stroškov in uspešnosti zagotavljanja kakovosti. Oba sta pokazatelja, ki izhajata iz ciljev četrte stopnje modela CMM.

g) Mnenje o primernem načinu opisa merjenja

Nazadnje so anketiranci presojali še način opisa merjenja. Vsi so se strinjali, da mora biti opis merjenja sestavni del procesa razvoja PO. Pri predstavitvi vsebine merjenja je potrebno opredeliti nabor merskih entitet, merskih izhodišč in pokazateljev. Prredstavitev nabora meritev se jim ne zdi tako pomembna. Med odgovori smo predlagali tudi idejo, da bi merjenje prikazali kot recept za vsako vlogo posebej, kar anketirancem ni bilo preveč všeč. Kot kažejo odgovori je naš način predstavitve z nabori ustrezen.

V splošnem lahko povzamemo, da rezultati ankete in zbrane informacije iz pogovorov potrjujejo izbrane rešitve, saj je bilo odstopanj malo. Praktično vsa odstopanja se nanašajo na tiste dele naših rešitev, ki so bile uvedene zaradi posebnih zahtev tretje ali četrte stopnje modela CMM, kar je bila izhodiščna zahteva naše naloge. Ker obiskane organizacije teh stopenj ne dosegajo, so jim takšne rešitve tuje in nepotrebne. Če ob odgovorih smiselno upoštevamo to omejitev lahko ugotovimo, da predlagane rešitve zelo dobro sovpadajo z videnjem majhnih organizacij.

[31] Meritev pove, kolikšen del začetnih zahtev uporabnikov je že realiziran s trenutno razvitimi izdelki. Spremljanje pokritosti zahtev predvideva tudi naša rešitev, vendar v okviru spremljanja zahtev.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

6.2 Rezultati disertacije

Ob zaključku želimo ponovno povzeti glavne rezultate disertacije. V pričujoči disertaciji smo razvili model za kvantitativno spremljanje razvoja PO v majhnih organizacijah za razvoj IS, ki ustreza zahtevam četrte stopnje modela CMM.

V skladu z zahtevami smo do končnega rezultata prišli v treh korakih, ki jih predstavljajo trije ključni rezultati našega dela:

- prilagodljiv proces razvoja PO, ki je bil zasnovan na podlagi prilagojenih zahtev modela CMM in tipične majhne organizacije za razvoj IS,
- merski proces, ki je bil utemeljen na pristopu PSM in prilagojen za majhne organizacije ter
- vsebinski del merskega pristopa, ki je bil določen na podlagi obeh procesov s pomočjo izvirne kombinirane metode za določanje nabora meritev.

Med razvojem smo uporabljali preverjene pristope in vgrajevali izvirne rešitve, da bi na koncu zagotovili pričakovan rezultat. To nam je uspelo z dosledno uporabo načel, ki so bila predstavljena v podpoglavju 1.5:

- Vsi rezultati so razviti na podlagi jasnih izhodišč in s pomočjo formaliziranih postopkov. Če je bilo mogoče, smo uporabili preverjene rešitve (recimo model TSP ali pristop PSM). Postopki razvoja so bili v vsakem primeru izpeljani dosledno in popolno.
- Kjer je bilo mogoče, smo v rezultatih upoštevali tudi praktični vidik.
- V rezultate smo vgradili izvirne ideje, ki omogočajo doseganje zastavljenih ciljev. Isto zamisel smo smiselno vključili v vse različne rezultate naloge.
- Vsi rezultati upoštevajo osnovne zahteve disertacije in drug drugega nadgrajujejo.
- Če je bilo mogoče, smo dobljene rezultate navzkrižno preverili.

Vsi trije ključni rezultati ustrezajo obema osnovnima zahtevama, vendar se moramo v skladu s prvim načelom (razmerje med zahtevami modela CMM in majhnostjo) zavedati, da smo zahteve morali medsebojno usklajevati. Tako zahtev modela CMM v majhni organizaciji ni bilo mogoče v celoti upoštevati, temveč smo jih morali prilagoditi. Enako tudi pričakovanj majhne organizacije po minimalnih naborih in zelo preprostih postopkih ni mogoče doseči, če želimo zadostiti zahtevam modela CMM. Dobljeni rezultati so po našem mnenju zelo dober kompromis, saj so nabori relativno majhni, merske aktivnosti pa uvajajo merjenje na jasen, določen in predvsem učinkovit način.

Da bi to dosegli smo v vsakem glavnem rezultatu uvedli inovativne principe. Prilagodljiv proces razvoja tako z večcikličnostjo, prilagodljivimi koraki in pravili prilagajanja omogoča dinamično prilagajanje spremembam v razvoju. Čeprav sam proces ni preprost, omogoča učinkovite prilagoditve in je s tega stališča izrazito uporaben v majhni organizaciji.

Podobno je z merskim procesom, ki je usklajen s prilagodljivim procesom razvoja PO: merski proces operativno prestavi merjenje na raven posameznika in ga zanj poenostavi. Uporabnike razbremeni z uvedbo združenih meritev ter učinkovitih analiz (predvsem napovedovanja) na podlagi učinkovite uporabe rezultatov preteklih projektov in statističnih metod. Predvidena

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

avtomatizacija repozitorija zajame zahtevnejši del merskih postopkov, s čemer postane izvajanje merjenja za uporabnika izrazito preprosto.

Vsebinski del merskega pristopa podpira in nadgradi oba dela. Merjenje je v skladu z zahtevami modela CMM opredeljeno na organizacijski ravni in kot tako določa stalen nabor merskih izhodišč za merjenje. S pomočjo učinkovite kombinirane metode za določanje meritev so izpeljani majhni nabori meritev in pokazateljev, ki so zadostni in popolni. Kombinirana metoda omogoča tudi nadgrajevanje merjenja v primeru posebnih potreb projektov. S tem se omogoči dodatno prilagodljivost zahtevam uporabnika.

Prilagodljiv proces razvoja, merski pristop in merski informacijski model so trije ključni deli našega modela za spremljanje kakovosti PO v majhni organizaciji za razvoj IS.

6.3 Prispevki k znanosti

V okviru doktorske disertacije je bilo poleg omenjenih rezultatov opravljenih več delnih raziskav, ki so zanimive za raziskovalce tega področja. Med te lahko štejemo naslednje:

- raziskavo možnosti za prilagoditev modela CMM zahtevam majhnih organizacij, katere rezultat je nabor načel za prilagajanje modela CMM;
- predlog pristopa za oblikovanje procesa razvoja na podlagi obstoječih modelov (modela PSP in TSP) ter prilagajanja modela CMM;
- definicijo in izvedbo prilagodljivega procesa razvoja s tremi značilnostmi: večcikličnim procesom, prilagodljivimi koraki in pravili prilagajanja problemom;
- prilagoditev merskega procesnega modela iz pristopa PSM potrebam majhnih organizacij za razvoj IS z več izvirnimi rešitvami, med katerimi izpostavljamo zlasti:

  - pristop s stalnim naborom meritev za vse aktivnosti posameznika,
  - uvedbo določanja (združevanja, izpeljave) meritev na višjih ravneh,
  - uvedbo dodatne vrste analize (napovedovanja),
  - opredelitev stalnega (organizacijskega) in posebnega (projektnega) dela merjenja,
  - uvedbo enostavnih, a učinkovitih statističnih principov v analize,
  - definicijo učinkovite metode ocenjevanja s kombiniranjem natančnih ocen posameznikov in rezultatov preteklih projektov ter
  - predlog avtomatiziranega skupnega repozitorija.

- določitev meta modela na treh ravneh;
- definicijo in prikaz uporabe kombinirane metode za izpeljavo meritev, ki temelji na principih metod GQM in MQG. Rezultat uporabe metode so združene in izpeljane meritve ter pokazatelji;
- na podlagi zahtev in izkušenj določen nabor osnovnih meritev, ki v skladu s predstavljenim merskim procesom zadošča za merjenje v skladu z zahtevami 4. stopnje modela CMM za majhno organizacijo;
- predlagana nabora združenih in izpeljanih meritev, ki sta bila določena z uporabo kombinirane metode ter
- nabor pokazateljev za vse štiri vrste analiz.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Prilagodljiv proces razvoja PO, merski proces in merski informacijski model predstavljajo izviren način merjenja v majhni organizaciji. V okviru tega smo določili še več drobnih inovacij, kot so recimo: uvedba tabel z združenimi rezultati preteklih projektov za učinkovitejšo izvedbo analiz, način grafičnega prikaza rezultatov napovedovanja, uvedba ponovnih pregledov v prilagodljiv proces in podobno.

Poleg tega je bilo v okviru raziskav določenih še nekaj pomembnih rezultatov, recimo: sistematično predstavljen nabor izkušenj področja, predstavljeno delovanje tipične majhne organizacije za razvoj IS, natančno opredeljena izvedba ocenjevanja, predstavljena navzkrižna prevejanja popolnosti določenih naborov, predlagan vprašalnik za preverjanje stanja na področju merjenja v majhnih organizacijah in podobno.

6.4 Predlogi za nadaljnje delo

V okviru disertacije smo se zaradi obsega dela omejili na definicijo in predstavitev modela. S praktičnega stališča se je opis našega pristopa zaključil z opisom pričakovanih značilnosti repozitorija, zato bi predlagali dve področji za nadaljnje delo:

- realizacijo repozitorija z opisanimi značilnostmi, s čemer bi podrobneje raziskali možnosti zamišljene izvedbe repozitorija ter
- preizkus predstavljenega modela v praksi, bodisi po delih ali v celoti. Slednje zahteva tako uporabo prilagodljivega procesa razvoja kot opisanega merskega pristopa. Pred tem bi bilo potrebno doreči nekatere praktične vidike uporabe opisanih pristopov.

Na tak način bi nadgradili predstavljene rezultate disertacije. Poleg tega bi lahko nadaljevali raziskave tudi v drugih smereh. Nekaj jih je naštetih v nadaljevanju:

- izvedli bi raziskavo možnosti napovedovanja z uporabo principov umetne inteligence,
- raziskali bi možnosti kombiniranega prikaza več pokazateljev naenkrat,
- z uporabo zahtevnejših statističnih metod bi skušali izboljšati učinkovitost napovedovanja (recimo na podlagi preteklih podatkov posameznika),
- nadaljevali bi raziskave prilagodljivega procesa in podobno.

6.5 Sklepna misel

Če hočejo biti velike organizacije za razvoj PO konkurenčne in uspešne, so prisiljene uporabljati ustrezne formalne pristope za zagotavljanje kakovosti PO. Brez tega enostavno ne gre: celovito obvladovanje razvoja PO je preveč zahtevno, zato je nujna uvedba principov kakovostnega razvoja PO na formaliziran način.

V majhnih organizacijah za razvoj PO temu ni tako. Organizacije so lažje obvladljive, zato se pri poslovanju bolje prilagajajo in učinkovito izkoriščajo svoje prednosti. Zaradi tega lahko delujejo, ne da bi posebej skrbele za kakovost razvoja. Če k temu dodamo še visoke stroške uvajanja formalnih pristopov za zagotavljanje kakovosti, smo le še korak od ugotovitve, da majhne organizacije formalnih pristopov ne potrebujejo.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Vendar je to lažen vtis. V resnici brez uporabe ustreznih pristopov za zagotavljanje kakovosti majhne organizacije mogoče lahko preživijo, ne morejo pa rasti in se razvijati. To so ugotovile tudi vse slovenske organizacije, s katerimi smo imeli stike ob praktičnem preverjanju našega merskega modela. Zato ni smiselno spraševati, če so formalni pristopi potrebni, temveč na kakšen način uvesti zagotavljanje kakovosti, da bo le-to uvedeno v majhno organizacijo na ustrezen, a vendar enostaven in praktično uporaben način.

Pričujoča disertacija je naš odgovor na to vprašanje. Menimo, da smo s predlaganim modelom za kvantitativno spremljanje razvoja PO zagotovili tako ustreznost kot praktično uporabnost. Z dosledno uporabo predlaganih rešitev ima majhna organizacija učinkovito oporo na svoji poti do glavnega cilja: nenehne rasti na vseh področjih razvoja PO.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

PRILOGA A: PRILAGODLJIV PROCES RAZVOJA PO

Prilagodljiv proces razvoja PO predstavljamo v treh delih. Najprej predstavljamo tabelo vlog in skupin, ki jih določa proces na ravni organizacije in projektov. Sledi predstavitev aktivnosti procesa na različnih ravneh. Nazadnje predstavimo še tabeli vseh izdelkov (rezultatov faz in ciklov), ki se pojavijo ob izvedbi procesa.

A.1 Vloge in skupine

Prilagodljiv proces določa vloge in skupine na ravni organizacije in na ravni projekta.

A.1.1 Vloge in skupine na ravni organizacije

Na ravni organizacije prilagodljiv proces razvoja PO določa naslednje vloge in skupine:

	Naziv	Namen
Vloga Č	Vsak član organizacije	- sodelovanje pri projektih - mersko spremljanje vsake aktivnosti - skrb za svojo usposobljenost - sodelovanje pri aktivnostih organizacije ali projekta (sestanki, pregledi, odbori)
Vloga O	Vodstvo organizacije	- vodenje organizacije v skladu z izbrano strategijo razvoja - sprejemanje poslovnih odločitev pri izbiri in izvedbi projektov - nadzor nad izvajanjem projektov - vodenje sestankov s člani organizacije
Skupina SEPG	Skupina za tehnološki proces organizacije	- razvoj, izgradnja, vzdrževanje in nadgradnja organizacijskega procesa razvoja PO in z njim povezanih stvari - skrb za repozitorij in njegov IS - zbiranje, hranjenje in analiza merskih podatkov na ravni organizacije
Skupina SQA	Skupina za zagotavljanje kakovosti	- skrb za kakovost aktivnosti in izdelkov v organizaciji - preverjanje ustreznosti raznih aktivnosti, ki se izvajajo v organizaciji - preverjanje aktivnosti razvoja PO v projektih in izdelkov, ki pri tem nastanejo - določanje standardov kakovosti, ki služijo kot merilo za preverjanje ustreznosti postopkov ali izdelkov - skrb za usposabljanje članov organizacije

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.1.2 Vloge na ravni projekta

Na ravni projekta prilagodljiv proces ne predvideva skupin, temveč več vlog. Vsi člani (razen projektnega vodje) lahko opravljajo več vlog istočasno.

	Naziv	Namen
Vloga V	Vodja projekta	- vodi projekt - zastopa projekt (dogovori) - nadzira projektno delo - ukrepa ob težavah - vodi za spremljanje tveganj - vodi upravljanje konfiguracij - vodi upravljanje zahtev - poroča vodstvu
Vloga Q	Vodja kakovosti (član SQA)	- planira kakovost - spremlja kakovost izvedbe - izvaja zunanje preglede - preverja kakovost z merskimi rezultati
Vloga G	Vodja procesa in merjenja (član SEPG)	- prilagaja organizacijski proces zahtevam projekta - zbira merske podatke - izvaja merske analize - predlaga ukrepe - sodeluje pri upravljanju zahtev - zbira predloge za izboljšavo
Vloga M	Vodja podpore	- skrbi za podporo (orodja) - vodi upravljanje konfiguracije - skrbi za projektno dokumentacijo - skrbi za vzdrževanje vgrajenega IS
Vloga C	Vodja za stike z naročnikom	- komunicira z naročnikom - vodi izobraževanje uporabnikov - načrtuje uporabniški vmesnik
Vloga P	Vodja planiranja	- ocenjuje porabo časa in opravila - pripravlja plane - preverja usklajenost planov z merskimi rezultati
Vloga T	Vodja testiranja	- vodi različna testiranja - poskrbi za odpravo napak
Vloga A	Vodja analize	- izvaja raziskavo zahtev - izvaja logično načrtovanje - skrbi za spremembe zahtev
Vloga DA	Vodja načrtovanja IS	- vodi fizično načrtovanje IS - skrbi za načrt IS (spremembe)
Vloga DB	Vodja načrtovanja PB	- vodi fizično načrtovanje PB - skrbi za načrt PB
Vloga RA	Vodja izgradnje IS	- pomaga pri določanju načrta - vodi izvedbo (kodiranje) - pripravlja dokumentacijo - vodi delno testiranje in odpravljanje napak - vodi namestitev IS
Vloga RB	Vodja izgradnje PB	- pomaga pri določanju načrta - vodi izvedbo PB - pripravlja dokumentacijo o PB - vodi testiranje PB in odpravljanje napak - vodi namestitev PB (prekodiranja)

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2 Opis prilagodljivega procesa razvoja PO

Prilagodljiv proces razvoja PO je formalno opisan z dvanajstimi navodili (glej razdelek 3.4.1). Proces na najvišji ravni predstavlja dodatek A.2.1.

A.2.1 Celoten proces

Proces:		**Dinamični Prilagodljivi Proces razvoja PO: CELOTEN PROCES**
Namen:		**Definicija celotnega procesa za majhne organizacije za razvoj IS**
Cikel:	**Ime cikla:**	**Faze:**
0.	**PODPORNE AKTIVNOSTI (izvajajo se neprenehoma)**	**0.1 Tedenski sestanki**
		0.2 Zbiranje merskih podatkov
		0.3 Upravljanje konfiguracije
		0.4 Upravljanje zahtev
		0.5 Nadgradnja procesa
		0.6 Spremljanje tveganj
	Rezultati:	Merski podatki, Trenutna konfiguracija, Spremembe zahtev, Predlogi za izboljšave procesa
1.	**ZAČETEK PROJEKTA**	**1.1 Priprava projekta**
		1.2 Strateško načrtovanje in idejna rešitev
		1.3 Planiranje projekta
		1.4 Vodenje konfiguracij in zahtev
		1.5 Poročilo vodstvu organizacije
		1.6 Zaključek priprave projekta
	Rezultati:	Inicializacijski dokument, Strateški dokument, Projektna knjiga (globalni plani)
2.	**RAZISKAVA ZAHTEV**	**2.1 Priprava raziskave zahtev**
		2.2 Zajem zahtev
		2.3 Analiza zahtev in logični model
		2.4 Obseg rešitve in potrditev zahtev
		2.5 Zaključek raziskave zahtev
	Rezultati:	Plan izvedbe analize, Standardi analize, Specifikacija zahtev, Projektna knjiga (dop.)
3.	**NAČRTOVANJE**	**3.1 Priprava načrtovanja**
		3.2 Konceptualni načrt in prototipi
		3.3 Tehnična rešitev in strategija implementacije
		3.4 Fizični načrt PB
		3.5 Fizični načrt IS
		3.6 Preverjanje načrtov in specifikacija načrtovanja
		3.7 Zaključek načrtovanja
	Rezultati:	Plan načrtovanja, Standardi načrtovanja, Specifikacija načrtovanja, Projektna knjiga (dop.)
4.	**IZGRADNJA**	**4.1 Priprava izgradnje**
		4.2 Izgradnja PB
		4.3 Podroben načrt modulov (po delih)
		4.4 Izgradnja modulov (po delih)
		4.5 Testiranje (po delih)
		4.6 Zaključek izgradnje
	Rezultati:	Plan izgradnje, Stand. izgradnje, Delno testirana PB, IS in dok., Projektna knjiga (dop.)
5.	**TESTIRANJE**	**5.1 Priprava testiranja**
		5.2 Sistemsko testiranje
		5.3 Priprava dokumentacije in usposabljanja
		5.4 Testiranje izgleda in dokumentacije
		5.5 Uporabniško testiranje
		5.6 Zaključek testiranja
	Rezultati:	Plan testiranja, Standardi testiranja, Testirana PB s testnimi podatki, Testiran IS, Popolnoma preverjena uporabniška dokumentacija, Projektna knjiga (dop.)
6.	**UPORABA**	**6.1 Priprava uporabe**
		6.2 Namestitev
		6.3 Vzdrževanje
		6.4 Zaključek uporabe
	Rezultati:	Plan namestitve in vzdrževanja, Standardi namestitve in vzdrževanja, Nameščena PB in IS, Dok. o namestitvi, Dokument o vzdrževanju, Projektna knjiga (dop.)
7.	**ZAKLJUČEK PROJEKTA**	**7.1 Arhiviranje projekta**
		7.2 Analiza projekta
		7.3 Poročilo vodstvu organizacije in zaključek projekta
	Rezultati:	Vse dokončano in arhivirano: vse verzije IS in PB, Knjiga razvoja projekta, Končno poročilo o projektu, Projektna knjiga
Izhodni pogoji:		**Nameščena PB in IS** **Dokončana in arhivirana vsa dokumentacija**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.2 Cikel 0: Podporne Aktivnosti

Dodatki med A.2.2 in A.2.9 prestavljajo posamezne cikle prilagodljivega procesa razvoja PO.

Proces:		**Dinamični Prilagodljivi Proces razvoja PO:** **CIKEL 0 – PODPORNE AKTIVNOSTI**
Namen:		Definicija aktivnosti, ki se izvajajo neprenehoma: sestanki, merjenje, ipd.
Faza:	**Ime faze:**	**Koraki:**
0.1	**TEDENSKI SESTANKI**	V: vodi sestanek enkrat tedensko
		P: poročaj o izpolnjevanju plana opravil
		Q: poročaj o izpolnjevanju plana kakovosti
		A: poročaj o anlizi zahtev
		D: poročaj o načrtovanju
		R: poročaj o izvedbi
		T: poročaj o poteku testiranja
		M: poročaj o vodenju konfiguracij in dokumentiranju
		C: poročaj o stikih z naročnikom
		G: poročaj o vprašanjih procesa
		Č: poročaj o osebnem izpolnjevanju načrtov
	Rezultati:	Zapisnik sestanka (sklepi)
0.2	**ZBIRANJE MERSKIH PODATKOV**	G: sproti zbiraj podatke o porabi časa članov pri opravilih
		G: sproti zbiraj podatke o kakovosti (napakah) članov pri opravilih
		G: sproti izvajaj analize podatkov
	Rezultati:	Merski podatki (repozitorij), analize (porabe časa, kakovosti, produktivnosti,...)
0.3	**UPRAVLJANJE KONFIGURACIJE**	M: sproti zbiraj zahtevke za uvrstitev izdelkov v trenutno konfiguracijo
		M: sproti preverjaj izpolnjevanje pogojev zahtevka
		M: v kolikor izdelek izpolnjuje pogoje, ga uvrsti v konfiguracijo
		M: tedensko poročaj o stanju konfiguracije
	Rezultati:	Opis trenutne konfiguracije, Zahtevki za vnos v trenutno konfiguracijo
0.4	**UPRAVLJANJE ZAHTEV**	A: sproti zbiraj zahtevke za spremembo zahtev
		A: sproti presodi pomembnost zahtevka, po potrebi z naročnikom
		A: razišči vpliv spremembe zahtev
		A: ukrepaj v skladu z nujnostjo spremembe in vplivom
		A: če je sprememba nujna, sproži nujne postopke na prvem tedenskem sestanku
		A: v kolikor je sprememba potrebna, sproži postopke ob naslednjem začetku cikla
		A: poskrbi, da bo nova zahteva ustrezno dokumentirana v vseh dokumentih
		A: zahtevaj uvrstitev nove Specifikacije zahtev v trenutno konfiguracijo
		A: tedensko poročaj o spremembah zahtev
	Rezultati:	Opis sprememb zahtev, Zahtevki za spremembe zahtev
0.5	**NADGRADNJA PROCESA**	G: sproti beleži predloge za izboljšavo, dobre prijeme, težave
		G: poročaj o predlogih ob koncu ciklov
		G: predlagaj ukrepe za izboljšavo procesa
		G: tedensko poročaj o stanju procesa
	Rezultati:	Predlogi za izboljšave procesa, pripombe,...
0.6	**SPREMLJANJE TVEGANJ**	V: sproti spremljaj tveganja
		V: ob pojavitvi tveganj poročaj na tedenskem sestanku
		V: ukrepaj v skladu s predvidenim postopkom
		V: tedensko poročaj o statusu tveganj
	Rezultati:	Poročila o statusu tveganj
Izhodni pogoji:		**Merski podatki** **Trenutna konfiguracija** **Spremembe zahtev** **Predlogi za izboljšave procesa**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.3 Cikel 1: Začetek projekta

Proces:		**Dinamični Prilagodljivi Proces razvoja PO:** **CIKEL 1 - ZAČETEK PROJEKTA**
Namen:		**Prvi cikel – vzpostavitev projekta, strateško načrtovanje in planiranje**
Faza:	**Ime faze:**	**Koraki:**
1.1	**PRIPRAVA PROJEKTA**	U: vodstvo organizacije pripravi začetni sestanek
		U: vodstvo predstavi okvirne zahteve z Inicializacijskim dokumentom
		V: vodi prvi sestanek projektne skupine
		V: razdeli vloge med člane projektne skupine
		V: po potrebi predvidi izobraževanje članov
		G: določi merljive glavne cilje projekta
		G: določi standardni način dela (sestanki, poročanje)
		G: določi razvojno strategijo projekta
		M: določi potrebna orodja in metode za razvoj
		M: dokumentiraj vse (cilje, vloge...) v Projektni knjigi
	Rezultati:	Inicializacijski dokument - okvirne zahteve - okvirni rok izdelave Projektna knjiga (osnutek) - vloge - cilji - način dela - razvojna strategija - orodja, metode - izobraževanje - ...
1.2	**STRATEŠKO NAČRTOVANJE IN IDEJNA REŠITEV**	A: kratko razišči trenutno poslovno okolje in obstoječi sistem
		A: predlagaj več idejnih rešitev problema
		C: predstavi idejne rešitve naročniku na prvem srečanju
		N: naročnik izbere eno idejno rešitev
		S: dokumentiraj opis okolja in sprejeto rešitev v Strateškem dokumentu
	Rezultati:	Strateški dokument - namen - kratek oris okolja in trenutnega sistema - opis izbrane idejne rešitve (vnER, FD, vnDFD) - opis razvojne strategije
1.3	**PLANIRANJE PROJEKTA**	A: razdeli celoten projekt na dele (kolikor je mogoče natančno)
		P: oceni velikost posameznih delov in skupnega procesnega in podatkovnega dela
		P: oceni porabo časa po fazah glede na ocenjeno velikost delov
		P: oceni stroške po fazah glede na ocenjeno porabo časa
		P: naredi (čim bolj natančen) plan opravil projekta (razvoja in implementacije)
		P: naredi (čim bolj natančen) plan poteka projekta (razvoja, implementacije)
		P: če je projekt manjši, dobi oceno porabe časa za opravila od članov
		P: če je projekt manjši, natanko planiraj opravila in porabo časa za vse člane
		P: če je projekt manjši, popravi plane članov, da dosežeš uravnoteženo obremenitev
		Q: planiraj kakovost projekta (merljivo)
		A: ugotovi tveganja pri izvajanju projekta (tudi kritične rač.vire)
		Q: planiraj načine spremljanja tveganj in reševanja težav
		Q: zunanje preveri vse plane
		P: popravi plane, če je treba
		M: dopolni Projektno knjigo z vsemi ocenami in plani
	Rezultati:	Projektna knjiga (dopolnjena) - ocena skupne velikosti obeh delov - ocena skupne porabe časa po fazah - globalni plan izvedbe (razvoja in implementacije) - globalni okvirni plan opravil (razvoja in implementacije) - plan opravil za člane (če mogoče) - plan porabe časa za člane (če mogoče) - plan kakovosti - seznam tveganj pri izvedbi in njihova rešitev

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

1.4	VODENJE KONFIGURACIJ IN ZAHTEV	M: določi izdelke, za katere velja vodenje konfiguracij
		M: planiraj vodenje konfiguracij
		M: določi nadzorni odbor konfiguracije
		A: določi nadzorni odbor zahtev
		M: seznani člane s postopki za vodenje konfiguracij
		A: seznani člane s postopki za vodenje zahtev
		Q: zunanje preveri plan in ga po potrebi popravi
		M: dopolni Projektno knjigo s planom konfiguracije in sestavo obeh odborov
	Rezultati:	Projektna knjiga (dopolnjena) - plan za vodenje konfiguracij - sestava odbora za vodenje konfiguracij - sestava odbora za vodenje zahtev
1.5	POROČILO VODSTVU ORGANIZACIJE	V: predstavi projekt in plane izvedbe vodstvu (Str. dokument, Projektno knjigo)
		V: prediskutiraj predloge
		V: po potrebi spremeni izhodišča in ponovno planiraj
	Rezultati:	Potrjena Projektna knjiga Potrjen Strateški dokument
1.6	ZAKLJUČEK PRIPRAVE PROJEKTA	M: zberi rezultate cikla
		Q: zunanje preveri rezultate
		G: opravi analizo izvedbe cikla
		G: predlagaj ukrepe – po potrebi ponovi del razvoja
		V: pripravi Poročilo o ciklu (potek, rezultati, predlogi za izboljšavo, ukrepi)
		V: poročaj vodstvu organizacije
		M: dodaj pripombe vodstva organizacije Poročilu o ciklu
		M: dopolni Projektno knjigo z merskimi podatki in poročilom cikla
	Rezultati:	Projektna knjiga (dopolnjena) - Poročilo o poteku cikla - Merski podatki (velikost, poraba časa, podatki o odkritih napakah)
Izhodni pogoji:		**Inicializacijski dokument** **Strateški dokument** **Projektna knjiga (globalni plani)**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.4 Cikel 2: Raziskava zahtev

Proces:		**Dinamični Prilagodljivi Proces razvoja PO:** **CIKEL 2 - RAZISKAVA ZAHTEV**
Namen:		**Drugi cikel – zajem zahtev, analiza in logično načrtovanje**
Faza:	**Ime faze:**	**Koraki:**
2.1	**PRIPRAVA RAZISKAVE ZAHTEV**	P: podrobno planiraj opravila cikla (korake)
		P: planiraj porabo časa vseh opravil
		Q: planiraj kakovost posameznih opravil
		P: dobi oceno porabe časa za opravila od članov
		P: planiraj opravila in porabo časa za vse člane
		P: popravi plane članov, da dosežeš uravnoteženo obremenitev
		V: predstavi plane cikla vodstvu organizacije
		V: pridobi njihovo mnenje
		P: po potrebi spremeni izhodišča in ponovno planiraj
		Q: določi standarde za analizo
	Rezultati:	Plan izvedbe analize Standardi za izvedbo analize (intervjuji, standard ER opisa, ...)
2.2	**ZAJEM ZAHTEV**	A: zajemi zahteve pri uporabniku
		A: zajemi zahteve iz obstoječega sistema
		A: beleži zahteve
	Rezultati:	Seznam zahtev Parcialni ERD, FD in DFD
2.3	**ANALIZA ZAHTEV IN LOGIČNI MODEL**	A: analiziraj zahteve
		A: razvij logični podatkovni model sistema (ERD)
		A: razvij logični procesni model sistema (FD, DFD)
		A: določi povezave med obema (zah/fun, zah/ent, ent/fun)
		Q: zunanje preveri pravilnost zahtev in logičnih modelov
	Rezultati:	Seznam zahtev Logični podatkovni model (lERD) Logični procesni model (FD, lDFD) Povezave med deli opisa (fun/zah, ent/zah, ent/fun)
2.4	**OBSEG REŠITVE IN POTRDITEV ZAHTEV**	M: dokumentiraj zahteve, logični model in povezave v Specifikaciji zahtev
		A: pripravi več predlogov obsega rešitve
		C: predstavi predloge naročniku
		N: naročnik izbere obseg rešitve
		A: preveri z naročnikom pravilnost Specifikacije zahtev
		A: zahtevaj vključitev Specifikacije zahtev v trenutno konfiguracijo
	Rezultati:	Specifikacija zahtev - seznam zahtev - logični podatkovni model (lERD) - logični procesni model (FD, lDFD) - fun/zah, ent/zah, ent/fun - izbran obseg rešitve
2.5	**ZAKLJUČEK RAZISKAVE ZAHTEV**	M: zberi rezultate cikla
		Q: zunanje preveri rezultate (popolnost, točnost, konsistentnost vseh)
		M: preveri, če so končani rezultati v trenutni konfiguraciji
		G: opravi analizo izvedbe cikla
		Q: opravi analizo kakovosti
		G: predlagaj ukrepe – po potrebi ponovi del razvoja
		V: pripravi Poročilo o ciklu (potek, rezultati, predlogi za izboljšavo, ukrepi)
		V: poročaj vodstvu organizacije
		M: dodaj pripombe vodstva organizacije Poročilu o ciklu
		M: dopolni Projektno knjigo s planom, merskimi podatki in poročilom cikla
	Rezultati:	Projektna knjiga (dopolnjena) - Poročilo o poteku cikla - Merski podatki (velikost, poraba časa, podatki o odkritih napakah)
Izhodni pogoji:		**Plan izvedbe analize** **Standardi analize** **Specifikacija zahtev** **Projektna knjiga (dop.)**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.5 Cikel 3: Načrtovanje

Proces:		Dinamični Prilagodljivi Proces razvoja PO: CIKEL 3 - NAČRTOVANJE
Namen:		Tretji cikel – fizično načrtovanje PB in IS
Faza:	Ime faze:	Koraki:
3.1	PRIPRAVA NAČRTOVANJA	V: ponovno preveri cilje projekta (na podlagi analize prejšnjega cikla)
		V: po potrebi dodaj projektu nove člane ali prerazporedi vloge
		P: po potrebi spremeni globalni projektni plan (opravil, porabe časa)
		P: podrobno planiraj opravila cikla (korake)
		P: planiraj porabo časa vseh opravil
		Q: planiraj kakovost posameznih opravil
		P: dobi oceno porabe časa za opravila od članov
		P: planiraj opravila in porabo časa za vse člane
		P: popravi plane članov, da dosežeš uravnoteženo obremenitev
		V: predstavi plane cikla vodstvu organizacije
		V: pridobi njihovo mnenje
		P: po potrebi spremeni izhodišča in ponovno planiraj
		Q: ponovno oceni tveganja
		Q: določi standarde za načrtovanje
	Rezultati:	Plan izvedbe načrtovanja Standardi za načrtovanje (GUI, zaslon, pomoč, dokumentacija)
3.2	KONCEPTUALNI NAČRT IN PROTOTIPI	DA: določi konceptualni načrt (sestavo)
		DA: oblikuj več predlogov tehničnih rešitev
		RA: razvij prototipe zanje
		DA: oblikuj več strategij implementacije (testiranja, namestitve, vzdrževanja)
	Rezultati:	Konceptualni načrt Več predlogov tehničnih rešitev Prototipi Več strategij implementacije
3.3	TEHNIČNA REŠITEV IN STRATEGIJA IMPLEMENTACIJE	C: predstavi naročniku tehnične rešitve
		C: prikaži naročniku prototipe
		N: naročnik izbere eno
		C: prikaži naročniku več strategij implementacije (testiranja, namestitve, vzdrževanja)
		N: naročnik izbere eno
		P: naredi plane testiranj (po delih, sistemskega, ...)
		P: naredi plan namestitve
		P: naredi plan vzdrževanja
		Q: zunanje preveri plane
	Rezultati:	Izbrana tehnična rešitev Izbrana strategija implementacije
3.4	FIZIČNI NAČRT PB	DB: določi fizični načrt PB
		DB: razvij fizične načrte pomožnih podatkov (za testiranje, nameščanje, vzdrževanje)
	Rezultati:	Fizični načrt PB Fizični načrt pomožnih podatkov
3.5	FIZIČNI NAČRT IS	DA: določi fizični načrt IS
		DA: razvij fizične načrte pomožnih programov (testiranje, nameščanje, vzdrževanje)
	Rezultati:	Fizični načrt IS Fizični načrt pomožnih programov

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

3.6	**PREVERJANJE NAČRTOV IN SPECIFIKACIJA NAČRTOVANJA**	DAB: preveri usklajenost fizičnih načrtov IS in PB
		DAB: preveri popolnost vseh pomožnih postopkov in materialov
		Q: zunanje preveri vse načrte
		M: dokumentiraj konceptualni načrt in fizične načrte v Specifikaciji načrtovanja
		DAB: zahtevaj vključitev Specifikacije načrtovanja v trenutno konfiguracijo
	Rezultati:	Specifikacija načrtovanja - standardi (GUI, koda, zaslon, pomoč, dokumentacija) - fizični načrt PB (tabele, relacije, atributi, pogledi ...) - fizični načrt IS (konceptualni načrt, opis modulov, poročila, navigacija, fun/mod, zah/mod, mod/atr) - fizični načrt pomožnih izdelkov - izbrana tehnična rešitev - strategija implementacije
3.7	**ZAKLJUČEK NAČRTOVANJA**	M: zberi rezultate cikla
		Q: zunanje preveri rezultate (popolnost, točnost, konsistentnost vseh)
		M: preveri, če so končani rezultati v trenutni konfiguraciji
		G: opravi analizo izvedbe cikla
		Q: opravi analizo kakovosti
		G: predlagaj ukrepe – po potrebi ponovi del razvoja
		V: pripravi Poročilo o ciklu (potek, rezultati, predlogi za izboljšavo, ukrepi)
		V: poročaj vodstvu organizacije
		M: dodaj pripombe vodstva organizacije Poročilu o ciklu
		M: dopolni Projektno knjigo s planom, merskimi podatki in poročilom cikla
	Rezultati:	Projektna knjiga (dopolnjena) - Poročilo o poteku cikla - Merski podatki (velikost, poraba časa, podatki o odkritih napakah)
Izhodni pogoji:		**Plan načrtovanja** **Standardi načrtovanja** **Specifikacija načrtovanja** **Projektna knjiga (dop)**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.6 Cikel 4: Izgradnja

Proces:		Dinamični Prilagodljivi Proces razvoja PO: CIKEL 4 - IZGRADNJA
Namen:		Četrti cikel – podrobno načrtovanje, izgradnja PB in IS ter testiranje po delih
Faza:	Ime faze:	Koraki:
4.1	PRIPRAVA IZGRADNJE	V: ponovno preveri cilje projekta (zaradi sprememb zahtev ali poteka projekta)
		V: po potrebi dodaj projektu nove člane ali prerazporedi vloge
		P: po potrebi spremeni globalni projektni plan (opravil, porabe časa)
		P: podrobno planiraj opravila cikla (korake)
		P: planiraj porabo časa vseh opravil
		Q: planiraj kakovost posameznih opravil
		P: dobi oceno porabe časa za opravila od članov
		P: planiraj opravila in porabo časa za vse člane
		P: popravi plane članov, da dosežeš uravnoteženo obremenitev
		Q: ponovno oceni tveganja
		V: predstavi plane cikla vodstvu organizacije
		V: pridobi njihovo mnenje
		P: po potrebi spremeni izhodišča in ponovno planiraj
		Q: določi standarde za kodiranje
	Rezultati:	Plan izgradnje Standardi izgradnje (kodiranje, zasnova fizične PB, ...)
4.2	IZGRADNJA PB	RB: naredi fizično PB
		RB: napolni jo s testnimi podatki
		RB: pripravi pomožne dele PB in podatke (za testiranje, nameščanje, vzdrževanje)
		RB: pripravi dokumentacijo o PB
		RB: zahtevaj vključitev PB v trenutno konfiguracijo
	Rezultati:	PB s testnimi podatki (zgrajena PB z ustreznimi tabelami ...) Pomožni deli PB in podatki Dokumentacija o PB
4.3	PODROBEN NAČRT MODULOV (PO DELIH)	RA: izdelaj podroben načrt dela IS ali pomožnega programa
		Q: zunanje preveri podrobne načrte (razmerja med posameznimi deli)
		RA: zahtevaj vključitev podrobnih načrtov IS v trenutno konfiguracijo
	Rezultati:	Podroben načrt vseh delov IS Podroben načrt vseh pomožnih programov (za testiranje, nameščanje, vzdrževanje)
4.4	IZGRADNJA MODULOV (PO DELIH)	RA: izgradi kodo dela IS (modula) v skladu z zahtevami
		RA: izgradi pomožne programe (za testiranje, nameščanje, vzdrževanje)
		RA: prevajaj, dokler ni brez napak
		RA: odpravi napake
		RA: pripravi dokumentacijo o vseh delih IS
	Rezultati:	IS (uspešno prevedena po delih) Pomožni programi (za testiranje, nameščanje, vzdrževanje) Dokumentacija o IS
4.5	TESTIRANJE (PO DELIH)	RA: izvedi testiranje vseh delov IS na testnih podatkih PB
		RA: izvedi testiranje pomožnih programov in delov PB oziroma podatkov
		RA: testiraj, dokler ni brez napak
		RAB: odpravi napake (v delih IS in v PB)
		RA: preveri dokumentacijo (opis delovanja)
		RAB: pripravi Poročilo o testiranju po delih
		Q: zunanje preveri dele IS in dokumentacijo
		RAB: zahtevaj vključitev testiranih delov v trenutno konfiguracijo
	Rezultati:	Delno testirana PB s testnimi podatki IS testiran po delih (koda v modulih-poročilih, funkcijah...) Testirani pomožni izdelki (programi, deli PB, podatki) Preverjena dokumentacija po delih (opis delovanja) Poročilo o testiranju po delih

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

4.6	ZAKLJUČEK IZGRADNJE	M: zberi rezultate cikla
		Q: zunanje preveri rezultate (popolnost, točnost, konsistentnost vseh)
		M: preveri, če so končani rezultati v trenutni konfiguraciji
		G: opravi analizo izvedbe cikla
		Q: opravi analizo kakovosti
		G: predlagaj ukrepe – po potrebi ponovi del razvoja
		V: pripravi Poročilo o ciklu (potek, rezultati, predlogi za izboljšavo, ukrepi)
		V: poročaj vodstvu organizacije
		M: dodaj pripombe vodstva organizacije Poročilu o ciklu
		M: dopolni Projektno knjigo s planom, merskimi podatki in poročilom cikla
	Rezultati:	Projektna knjiga (dopolnjena) - Poročilo o poteku cikla - Merski podatki (velikost, poraba časa, podatki o odkritih napakah)
Izhodni pogoji:		**Plan izgradnje** **Standardi izgradnje** **Delno testiran PB, IS in dokumentacija** **Projektna knjiga (dop)**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.7 Cikel 5: Testiranje

Proces:		Dinamični Prilagodljivi Proces razvoja PO: CIKEL 5 - TESTIRANJE
Namen:		Peti cikel – sistemsko in uporabniško testiranje IS, PB in dokumentacije
Faza:	Ime faze:	Koraki:
5.1	PRIPRAVA TESTIRANJA	V: ponovno preveri cilje projekta (zaradi sprememb zahtev ali poteka projekta)
		V: po potrebi dodaj projektu nove člane ali prerazporedi vloge
		P: po potrebi spremeni globalni projektni plan (opravil, porabe časa)
		P: podrobno planiraj opravila testiranja (IS, PB, podatkov, pom. izdelkov)
		P: planiraj porabo časa vseh opravil
		Q: planiraj kakovost posameznih opravil
		P: dobi oceno porabe časa za opravila od članov
		P: planiraj opravila in porabo časa za vse člane
		P: popravi plane članov, da dosežeš uravnoteženo obremenitev
		Q: ponovno oceni tveganja
		V: predstavi plane cikla vodstvu organizacije
		V: pridobi njihovo mnenje
		P: po potrebi spremeni izhodišča in ponovno planiraj
		V: ponovno preveri cilje projekta (zaradi sprememb zahtev ali poteka projekta)
	Rezultati:	Plan testiranja Standardi za testiranje
5.2	SISTEMSKO TESTIRANJE	T: izvedi združitveni test IS na testnih podatkih PB
		T: izvedi test performans (pomožni podatki)
		T: izvedi test delovanja pri kritični obremenitvi
		T: izvedi test arhiviranja in obnove PB
		T: izvedi test prenosa podatkov v PB (pomožni programi)
		T: izvedi test varnosti (pomožni podatki)
		RAB: odpravi napake
		T: ponavljaj, dokler ni brez napak
		T: pripravi Poročilo o testiranju
		T: zahtevaj vključitev testiranega IS v trenutno konfiguracijo
	Rezultati:	Sistemsko testirana PB s testnimi podatki Sistemsko testiran IS (kot celota) Poročilo o sistemskem testiranju
5.3	PRIPRAVA DOKUMENTACIJE IN USPOSABLJANJA	RAB: pripravi uporabniško dokumentacijo (navodila za uporabo)
		C: pripravi učno dokumentacijo (učni proces, učni primeri)
		Q: zunanje preglej uporabniško in učno dokumentacijo
	Rezultati:	Uporabniška dokumentacija Učna dokumentacija
5.4	TESTIRANJE IZGLEDA IN DOKUMENTACIJE	T: testiraj uporabniški vmesnik in navigacijo
		T: testiraj sistem pomoči
		T: preglej glavne rezultate (Strateški dokument, Specifikacijo zahtev/načrtovanja)
		T: preveri sistemsko dokumentacijo
		T: preveri uporabniško dokumentacijo
		T: preveri učno dokumentacijo
		RAB: odpravi napake in pomanjkljivosti
		T: ponavljaj, dokler ni brez napak
		T: pripravi Poročilo o testiranju
		T: zahtevaj vključitev IS in dokumentacije v trenutno konfiguracijo
	Rezultati:	Testiran izgled IS Preverjena vsa dokumentacija Poročilo o testiranju izgleda in dokumentacije

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

5.5	UPORABNIŠKO TESTIRANJE	C: pripravi predstavitev IS naročniku
		C: pripravi osnovno izobraževanje (pomožni materiali, učna dokumentacija)
		T: izvedi laboratorijski uporabniški test
		T: izvedi uporabniški test prijaznosti (uporabniška dokumentacija, pomoč)
		RAB: odpravi napake
		T: ponavljaj, dokler ni brez napak
		T: pripravi Poročilo o testiranju
		T: zahtevaj vključitev IS v končno konfiguracijo
	Rezultati:	Uporabniško testirana PB s testnimi podatki Uporabniško testiran IS Uporabniško preverjena uporabniška dokumentacija Poročilo o uporabniškem testiranju
5.6	ZAKLJUČEK TESTIRANJA	M: zberi rezultate cikla
		Q: zunanje preveri rezultate (popolnost, točnost, konsistentnost vseh)
		M: preveri, če so končani rezultati v trenutni konfiguraciji
		G: opravi analizo izvedbe cikla
		Q: opravi analizo kakovosti
		G: predlagaj ukrepe – po potrebi ponovi del razvoja
		V: pripravi Poročilo o ciklu (potek, rezultati, predlogi za izboljšavo, ukrepi)
		V: poročaj vodstvu organizacije
		M: dodaj pripombe vodstva organizacije Poročilu o ciklu
		M: dopolni Projektno knjigo s planom, merskimi podatki in poročilom cikla
	Rezultati:	Projektna knjiga (dopolnjena) - Poročilo o poteku cikla - Merski podatki (velikost, poraba časa, podatki o odkritih napakah)
Izhodni pogoji:		**Plan testiranja** **Standardi testiranja** **Testirana PB s testnimi podatki** **Testiran IS**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.8 Cikel 6: Uporaba

Proces:		Dinamični Prilagodljivi Proces razvoja PO: CIKEL 6 – UPORABA
Namen:		Šesti cikel – namestitev in vzdrževanje
Faza:	Ime faze:	Koraki:
6.1	PRIPRAVA UPORABE	V: ponovno preveri cilje projekta (zaradi sprememb zahtev ali poteka projekta)
		V: po potrebi dodaj projektu nove člane ali prerazporedi vloge
		P: po potrebi spremeni globalni projektni plan (opravil, porabe časa)
		P: podrobno planiraj opravila namestitve in vzdrževanja
		P: planiraj porabo časa vseh opravil
		Q: planiraj kakovost posameznih opravil
		P: dobi oceno porabe časa za opravila od članov
		P: planiraj opravila in porabo časa za vse člane
		P: popravi plane članov, da dosežeš uravnoteženo obremenitev
		Q: ponovno oceni tveganja
		V: predstavi plane cikla vodstvu organizacije
		V: pridobi njihovo mnenje
		P: po potrebi spremeni izhodišča in ponovno planiraj
		Q: določi standarde namestitve v skladu s strategijo
		Q: določi standarde vzdrževanja
	Rezultati:	Plan namestitve Plan vzdrževanja Standardi namestitve Standardi vzdrževanja
6.2	NAMESTITEV	RAB: pripravi izvedbo namestitve (pomožne izdelke)
		RAB: izvedi namestitev pri uporabniku (lahko daljše obdobje)
		RB: napolni PB s pravimi podatki
		T: opravi uporabniški test v realnem okolju
		RAB: pripravi IS in PB za delo (zadnje pomanjkljivosti, nastavitve)
		V, N: pripravi in podpiši Dokument o namestitvi
	Rezultati:	Nameščena PB s pravimi podatki Nameščen IS z dokumentacijo Dokument o namestitvi
6.3	VZDRŽEVANJE	V: preveri strategijo vzdrževanja z naročnikom
		V, N: pripravi in podpiši Dokument o vzdrževanju
		M: zastavi sistem sprotne pomoči
		M: zastavi spremljanje delovanja sistema pri uporabniku (pomožni izdelki)
		*M: redno izvajaj preglede in pregled podatkov o delovanju (v skladu s strategijo)
		*M: redno zbiraj predloge za spremembe, izboljšave
		*M: po potrebi izvajaj vzdrževalna dela pri uporabniku (v skladu s strategijo)
	Rezultati:	Nameščena PB s pravimi podatki Nameščen IS z dokumentacijo Dokument o vzdrževanju Podatki o delovanju sistema Predlogi za spremembe, izboljšave
6.4	ZAKLJUČEK UPORABE	M: zberi rezultate cikla
		Q: zunanje preveri rezultate (popolnost, točnost, konsistentnost vseh)
		G: opravi analizo izvedbe cikla
		Q: opravi analizo kakovosti
		G: predlagaj ukrepe – po potrebi ponovi del namestitve ali vzdrževanja
		V: pripravi Poročilo o ciklu (potek, rezultati, predlogi za izboljšavo, ukrepi)
		V: poročaj vodstvu organizacije
		M: dodaj pripombe vodstva organizacije Poročilu o ciklu
		M: dopolni Projektno knjigo s planom, merskimi podatki in poročilom cikla
	Rezultati:	Projektna knjiga (dopolnjena) - Poročilo o poteku cikla - Merski podatki (velikost, poraba časa, podatki o odkritih napakah)
Izhodni pogoji:		**Plan namestitve in vzdrževanja** **Standardi namestitve in vzdrževanja** **Nameščena PB in IS** **Dokument o namestitvi** **Dokument o vzdrževanju** **Projektna knjiga (dop.)**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.9 Cikel 7: Zaključek projekta

Proces:		Dinamični Prilagodljivi Proces razvoja PO: CIKEL 7 – ZAKLJUČEK PROJEKTA
Namen:		Sedmi cikel – arhiviranje, končna analiza in zaključek projekta
Faza:	Ime faze:	Koraki:
7.1	ARHIVIRANJE PROJEKTA	M: zberi vse verzije IS in PB
		M: arhiviraj vse verzije IS, PB in pomožnih materialov
		M: zberi vse rezultate razvoja
		M: oblikuj celovito dokumentacijo o razvoju projekta – Knjiga razvoja projekta
		M: arhiviraj Knjigo razvoja projekta
	Rezultati:	Arhivirane (vse) verzije IS in PB Arhivirana Knjiga razvoja projekta
7.2	ANALIZA PROJEKTA	G: ponovno preglej vse merske podatke v Projektni knjigi
		G: izvedi analizo poteka projekta
		Q: izvedi analizo kakovosti projekta
		Q: izvedi analizo članov projektne skupine
		G: zberi predloge za izboljšavo procesa
		V: shrani rezultate v Končno poročilo o projektu
	Rezultati:	Končno poročilo o projektu
7.3	POROČILO VODSTVU ORGANIZACIJE IN ZAKLJUČEK PROJEKTA	V: poročaj vodstvu organizacije o izvedbi projekta, rezultatih, predstavi analizo
		V: dopolni Končno poročilo o projektu s pripombami vodstva organizacije
		M: dopolni Projektno knjigo s Končnim poročilom o projektu
		M: arhiviraj dokončano Projektno knjigo
		V: Zaključi projekt (zaključni sestanek)
	Rezultati:	Arhivirana dokončana Projektna knjiga
Izhodni pogoji:		**Vse dokončano in arhivirano:** - **vse verzije IS in PB** - **Knjiga razvoja projekta** - **Končno poročilo o projektu** - **Projektna knjiga**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.10 Enostaven korak (posameznik)

Dodatki med A.2.10 in A.2.12 predstavljajo tri načine izvedbe koraka.

Proces:		**ENOSTAVEN KORAK (posameznik)**
Namen:		**Izvedba enostavnega koraka posameznika (zahteva enostavno planiranje, izvedbo in merjenje)**
Faza:	**Ime faze:**	**Koraki:**
1.	**ZAČETEK NALOGE**	dobi opis enostavne aktivnosti
2.	**PLANIRANJE**	oceni velikost rešitve – z uporabo ustrezne metode (OV)
		oceni porabo časa izvedbe aktivnosti - z uporabo metode (OČ)
		če smiselno, oceni stroške izvedbe aktivnosti – z uporabo metode (OS)
		če smiselno, oceni predvideno število napak v rešitvi – z uporabo metode (ON)
		shrani vse ocene (P)
3.	**REŠEVANJE**	izvedi aktivnost - v skladu s standardnim postopkom rešitve (SPR)
		beleži porabo časa (Č)
		beleži število napak (N)
		ponovno preveri rešitev - v skladu s standardom za pregled rešitve (PR)
		beleži porabo časa (Č)
		beleži število napak (N)
		odpravi najdene napake
		beleži porabo časa (P)
		beleži število napak (P)
4.	**ZAKLJUČEK NALOGE**	seštej in shrani porabo časa (P)
		določi in shrani končno velikost rešitve (P)
		določi in shrani stroške aktivnosti (P)
		seštej in shrani število napak (P)
		izroči dokončano rešitev in poročilo o reševanju naloge (P)
Rezultati:		**Rešena naloga** **P – poročilo o reševanju naloge** **Č – beleženje porabe časa** **N – beleženje odkritih napak po standardnih razredih (SN)**
Pomožni dokumenti, pripomočki:		**OV – metoda za oceno velikosti** **OČ – metoda za oceno časa po fazah** **ON – metoda za oceno skupnega števila napak (velikosti, zgodovina)** **OS – metoda za oceno stroškov** **SPR – standarden postopek za reševanje** **PR – standard za pregled rešitve** **SN – standardni razredi napak**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.11 Zahtevnejši korak (posameznik)

Proces:		ZAHTEVNEJŠI KORAK (posameznik)
Namen:		Izvedba zahtevnejšega koraka posameznika (zahteva natančno planiranje, načrtovanje, izvedbo, testiranje in merjenje)
Faza:	Ime faze:	Koraki:
1.	ZAČETEK NALOGE	dobi opis problema
		razjasni, kaj zahteva problem
		po potrebi razdeli problem na podprobleme
2.	PLANIRANJE	oceni skupno velikost rešitve – z uporabo ustrezne metode (OV)
		določi 90% verjetnostno območje velikosti
		shrani oceno velikosti in meji verjetnostnega območja (P)
		oceni porabo časa po fazah reševanja - z uporabo metode (OČ)
		določi 90% verjetnostno območje porabe časa
		shrani oceno porabe časa in meji verjetnostnega območja (P)
		planiraj aktivnosti za rešitev problema (PO)
		planiraj porabo časa za posamezne aktivnosti (PČ)
		oceni predvidene stroške – z uporabo metode (OS)
		oceni predvideno število napak v rešitvi – z uporabo metode (ON)
		oceni število vnešenih in odpravljenih napak po fazah – z uporabo metode (ONF)
		shrani vse ocene napak (P)
3.	NAČRTOVANJE	določi načrt rešitve – v skladu s standardnim postopkom načrtovanja (SPN)
		beleži porabo časa (Č)
		ponovno preveri načrt - v skladu s standardom za pregled načrta (PN)
		Odpravi najdene napake
		beleži porabo časa (Č)
		beleži število napak načrtovanja (N)
4.	REŠEVANJE	reši nalogo - v skladu z načrtom in standardnim postopkom rešitve (SPR)
		beleži porabo časa (Č)
		beleži število napak (N)
		ponovno preveri rešitev - v skladu s standardom za pregled rešitve (PR)
		Odpravi najdene napake
		beleži porabo časa (Č)
		beleži število napak (N)
5.	TESTIRANJE	testiraj rešitev, dokler ni brez napak
		odpravljaj napake
		beleži porabo časa (Č)
		beleži število napak (N)
		shrani opise in rezultate testiranja (P)
6.	ZAKLJUČEK NALOGE	seštej in shrani porabo časa po fazah (P)
		določi in shrani končno velikost rešitve (P)
		določi in shrani stroške (P)
		seštej in shrani število vnešenih napak po fazah (P)
		seštej in shrani število odpravljenih napak po fazah (P)
		Zabeleži predloge za izboljšavo, dobre prijeme, težave (P)
		izroči dokončano rešitev in popolno poročilo o reševanju naloge (P)
Rezultati:		Rešena naloga P – poročilo o reševanju naloge Č – beleženje porabe časa N – beleženje odkritih napak po standardnih razredih (SN) PI – predlogi za izboljšavo, težave, dobri prijemi
Pomožni dokumenti, pripomočki:		OV – metoda za oceno velikosti OČ – metoda za oceno časa po fazah ON – metoda za oceno skupnega števila napak (velikosti, zgodovina) ONČ – metoda za oceno vnešenih in odpravljenih napak po fazah OS – metoda za oceno stroškov PA – plan opravil PČ – plan porabe časa SPN – standarden postopek za izdelavo načrta PN – standard za pregled načrta SPR – standarden postopek za reševanje PR – standard za pregled rešitve SN – standardni razredi napak

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.2.12 Zahteven korak (več članov)

Proces:		ZAHTEVEN KORAK (več članov)
Namen:		Izvedba zahtevnega koraka več članov (zahteva natančno planiranje, načrtovanje, delitev dela, izvedbo po delih, sestavljanje, testiranje in merjenje)
Faza:	Ime faze:	Koraki:
1.	ZAČETEK NALOGE	VČ: dobi opis problema
		VČ: razjasni, kaj zahteva problem
		VČ: dogovori se za strategijo reševanja
		VČ: razbij problem na podprobleme
2.	SKUPNO PLANIRANJE	V: oceni skupno velikost rešitve – z uporabo ustrezne metode (OV)
		V: določi 90% verjetnostno območje velikosti
		V: shrani oceno velikosti in meji verjetnostnega območja (PS)
		V: oceni skupno porabo časa po fazah - z uporabo metode (OČ)
		V: določi 90% verjetnostno območje porabe časa
		V: shrani oceno porabe časa in meji verjetnostnega območja (PS)
		V: planiraj vse aktivnosti projekta (PAS)
		V: planiraj skupno porabo časa projekta (PČS)
		V: oceni predvidene stroške – z uporabo metode (OS)
		V: oceni predvideno število napak v rešitvi – z uporabo metode (ON)
		V: oceni število vnešenih/odpravljenih napak po fazah – z metodo (ONF)
		V: shrani vse ocene napak (PS)
3.	REŠEVANJE	V: naredi zasnovo končne sestavljene rešitve – v skladu s SPR
		VČ: beleži porabo časa (ČS, Č)
		Č: v skladu z ENOSTAVNIM ali ZAHTEVNEJŠIM KORAKOM POSAMEZNIKA reši svoj del podproblema
		V: zberi poročila (P) vseh članov (PS)
		V: sestavi vse dele rešitve v skupno rešitev
		VČ: temeljito preveri skupno rešitev - v skladu s PR
		Č: odpravi najdene napake
		VČ: beleži porabo časa (ČS, Č)
		VČ: beleži število napak (NS, N)
4.	TESTIRANJE	VČ: po potrebi testiraj skupno rešitev, dokler ni brez napak
		Č: odpravljaj napake
		VČ: beleži porabo časa (ČS, Č)
		VČ: beleži število napak (NS, N)
		V: shrani opise in rezultate testiranja (PS)
5.	ZAKLJUČEK NALOGE	določi skupno porabo časa članov iz poročil članov (P)
		seštej s porabo časa združevanja rešitve (ČS)
		shrani porabo časa po fazah (PS)
		določi in shrani končno velikost rešitve (PS)
		določi in shrani skupne stroške (PS)
		določi skupno število vnešenih napak po fazah in poročil članov (P)
		shrani število vnešenih napak po fazah (PS)
		določi skupno število odpravljenih napak po fazah in poročil članov (P)
		seštej in shrani število odpravljenih napak po fazah (PS)
		določi vse predloge za izboljšavo, dobre prijeme, težave iz poročil članov (P)
		shrani predloge za izboljšavo, dobre prijeme, težave (PS)
		izroči končano skupno rešitev in poročilo o reševanju naloge za skupino (PS)
Rezultati:		**Skupna rešitev naloge** **P – poročilo o reševanju naloge člana** **PS – poročilo o reševanju naloge skupine** **Č – beleženje porabe časa člana** **ČS - beleženje porabe časa skupine** **N – beleženje odkritih napak po standardnih razredih (SN) za člana** **NS – beleženje odkritih napak po standardnih razredih (SN) za skupino** **PI – predlogi za izboljšavo, težave, dobri prijemi**
Pomožni dokumenti, pripomočki:		**OV – metoda za oceno velikosti** **OČ – metoda za oceno časa po fazah** **ON – metoda za oceno skupnega števila napak (velikosti, zgodovina)** **ONČ – metoda za oceno vnešenih in odpravljenih napak po fazah** **OS – metoda za oceno stroškov** **PAS– plan opravil skupine** **PČS – plan porabe časa skupine** **SPN – standarden postopek za izdelavo načrta** **PN – standard za pregled načrta** **SPR – standarden postopek za reševanje** **PR – standard za pregled rešitve** **SN – standardni razredi napak**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.3 Rezultati prilagodljivega procesa

Rezultati prilagodljivega procesa so treh vrst. Prvi opisujejo značilnosti izvajanja projekta, medtem ko drugi prikazujejo sam proces razvoja. Tretja vrsta so nastali izdelki (vmesni in končni), ki sodijo bodisi v procesni ali podatkovni del. Kot četrto vrsto predstavljamo še uporabljeno dokumentacijo, ki sicer ne sodi med izdelke.

A.3.1 Opis projekta

Vse izdelke, ki se nanašajo na opis projekta, shranjujemo v Projektno knjigo. Sestava je razvidna iz spodnje tabele. Sestavni deli izdelkov so navedeni samo kot primer – lahko bi jih določili tudi bolj podrobno.

Združen izdelek	Izdelki	Sestavni deli
Projektna knjiga	**Opis projektne skupine, ciljev, načina dela, razvojne strategije**	
	Skupne ocene velikosti, porabe časa	
	Globalni plan opravil in porabe časa	
	(Plan opravil in porabe časa za člane)	
	Globalni plan kakovosti	
	Plan obvladovanja tveganj	
	Plan vodenja konfiguracij in sestava odbora za vodenje konfiguracij	
	(Več) Planov izvedbe ciklov	Plan opravil in porabe časa cikla
		Plan opravil in porabe časa za vse člane
		Plan kakovosti opravil (če potrebno)
	(Več) Poročil o ciklih	Potek cikla – opravljene aktivnosti
		Doseženi rezultati (izdelki)
		Predlogi za izboljšavo, dobre prakse, ...
		Opis izrednih ukrepov
		Pripombe vodstva ob zaključku cikla
	(Več) Merskih podatkov cikla	Porabo časa po fazah, aktivnostih in članih
		Velikosti razvitih izdelkov
		Števila napak (fazah, aktivnostih, vrstah, ...)
		Stroški (po članih, fazah, ...)
		Izpeljane pokazatelje (produktivnosti, ...)
	(Več) Poročil o testiranju	Testiranju po delih
		Sistemskem testiranju
		Testiranju izgleda in dokumentacije
		Uporabniškem testiranju
	Končno poročilo o projektu	Analiza izvedbe projekta (rezultati, potek)
		Analiza kakovosti (dosežena kakovost)
		Analiza dela članov projektne skupine
		Predloge za izboljšave procesa
		Pripombe vodstva ob zaključku cikla

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.3.2 Opis procesa

V Projektni knjigi so shranjeni tudi izdelki, ki se nanašajo na opis procesa. Poleg teh opisujejo proces tudi podatki o merjenju, ki se zbirajo v repozitoriju.

Združen izdelek	Izdelki	Sestavni deli
Projektna knjiga	**Skupne ocene velikosti, porabe časa**	
	Globalni plan opravil in porabe časa	
	(Plan opravil in porabe časa za člane)	
	Globalni plan kakovosti	
	(Več) Planov izvedbe ciklov	Plan opravil in porabe časa cikla
		Plan opravil in porabe časa za vse člane
		Plan kakovosti opravil (če potrebno)
	(Več) Merskih podatkov cikla	Porabo časa po fazah, aktivnostih in članih
		Velikosti razvitih izdelkov
		Števila napak (fazah, aktivnostih, vrstah, ...)
		Stroški (po članih, fazah, ...)
		Izpeljane pokazatelje (produktivnosti, ...)
	(Več) Poročil o testiranju	Testiranju po delih
		Sistemskem testiranju
		Testiranju izgleda in dokumentacije
		Uporabniškem testiranju
	Končno poročilo o projektu	Analiza izvedbe projekta (rezultati, potek)
		Analiza kakovosti (dosežena kakovost)
		Predloge za izboljšave procesa
Merski podatki projekta v repozitoriju	**Merski podatki posameznih članov**	Poraba časa za izvedbo aktivnosti
		Obseg dela za izvedbo aktivnosti
		Velikost rezultata aktivnosti (izdelka)
		Stroški izvedbe aktivnosti
		Število odkritih in odpravljenih napak
	Podatki o delovanju sistema v času vzdrževanja	
	Predlogi za spremembe in izboljšave v času vzdrževanja	

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.3.3 Izdelki

Glavni rezultati izvajanja procesa so izdelki. Ločimo tri vrste izdelkov:

- opise rezultatov razvoja (ob zaključku projekta so združeni v Knjigo razvoja projekta);
- končne izdelke, ki so bodisi izvedba IS ali PB, in
- različne vrste dokumentacije, ki je namenjena uporabniku oziroma naročniku.

Združen izdelek	Izdelki	Sestavni deli
Knjiga razvoja projekta	Inicializacijski dokument	okvirne zahteve
		okvirni rok izdelave
	Strateški dokument	namen
		oris okolja, trenutnega sistema
		opis izbrane idejne rešitve (vnER, FD, vnDFD)
		opis razvojne strategije
	Specifikacija zahtev	logični podatkovni model (lERD)
		logični proc. model (FD, lDFD)
		seznam zahtev
		fun/zah, ent/zah, ent/fun
		izbran obseg rešitve
	Specifikacija načrtovanja	standardi (GUI, koda, pomoč, ...)
		fizični načrt PB (tabele, relacije)
		fizični načrt IS
		fizični načrt pomožnih izdelkov
		izbrana tehnična rešitev
		strategija implementacije
	Podrobni načrti	vseh delov IS
		vseh pomožnih programov
	Dokument o namestitvi	
	Dokument o vzdrževanju	
Informacijski Sistem (IS)	Prototipi IS	
	IS (uspešno preveden po delih)	
	IS testiran po delih	
	Sistemsko testiran IS	
	Testiran izgled IS	
	Uporabniško testiran IS	
	Nameščen IS z dokumentacijo	
	Pomožni programi (za testiranje, nameščanje, vzdrž.)	
	Testirani pomožni programi	
Podatkovna Baza (PB)	PB s testnimi podatki (zgrajena PB s tabelami ...)	
	Delno testirana PB s testnimi podatki	
	Sistemsko testirana PB s testnimi podatki	
	Uporabniško testirana PB s testnimi podatki	
	Nameščena PB s pravimi podatki	
	Pomožni deli PB in podatki	
	Testirani pomožni deli PB in podatki	
Dokumentacija - za uporabnika	Dokumentacija o PB	
	Dokumentacija o IS	
	Preverjena dokumentacija po delih (opis delovanja)	
	Uporabniška dokumentacija (navodila za uporabo)	
	Učna dokumentacija (učni proces, učni primeri)	
	Preverjena vsa dokumentacija skupaj	
	Uporabniško preverjena uporabniška dokumentacija	

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

A.3.4 Pomožna dokumentacija

Med projektnim razvojem uporabljamo tudi pomožno dokumentacijo (standarde, zabeležke in obrazce). Čeprav ta ne sodi med rezultate uporabe procesa, je s prilagodljivim procesom tesno povezana in jo zato posebej predstavljamo. Podobno bi lahko določili tudi dokumentacijo na ravni organizacije (recimo opise standardnih postopkov), vendar to presega okvire naše naloge.

Združen izdelek	Izdelki	Sestavni deli
Standardi (projekt)	**Standardni postopki izvajanja**	Standardni način dela (sestanki, poročanje, ...)
		Standardi za izvedbo analize (intervjuji, normalizacija, ...)
		Standardi za načrtovanje (delitev na module, določanje tabel, ...)
		Standardi izgradnje (postopek prevajanja, postopek gradnje PB, ...)
		Standardi za testiranje (postopki, ponavljanja, ...)
		Standardi namestitve (način, učni proces, ...)
		Standardi vzdrževanja (redni pregledi, beleženje sprememb, ...)
	Standarden izgled	Standardni način dela (izgled poročil, merskih podatkov, ...)
		Standardi za izvedbo analize (izgled vprašalnika, ERD, ...)
		Standardi za načrtovanje (opis modula, opis tabele, ...)
		Standardi izgradnje (kodiranje, vrste napak ob prevajanju, ...)
		Standardi za testiranje (izgled poročila, vrste napak, ...)
		Standardi namestitve (dokument o namestitvi, učni primer, ...)
		Standardi vzdrževanja (dokument o vzdrževanju, izgled predloga, ...)
Predloge (projekt)	**Obrazci**	Plani (opravil, porabe časa, kakovosti, merjenja, spremljanja tveganj, ...)
		Poročila (poteka cikla, testiranja, ...)
		Predlogi (za izboljšave, spremembe konfiguracije, spremembe zahtev, ...)
	Beležke	Spremljanje porabe časa
		Števila napak

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

PRILOGA B: MERSKI PRISTOP

V tej prilogi predstavljamo naš merski pristop, ki obsega dva dela. **Merski procesni model** definira način, kako se merjenje izvaja. Drugi del je **merski informacijski model**, ki opredeljuje način za opis vsebine merjenja.

B.1: Merski procesni model

Merski proces je nadgradnja merskega procesnega modela iz pristopa PSM [McG01]. Predstavljamo ga z opisom aktivnosti, njihovo umeščenostjo v prilagodljiv proces razvoja PO in predstavitvijo analiz, iz katerih so razvidne nekatere posebnosti našega pristopa.

B.1.1 Glavne aktivnosti, podaktivnosti in koraki

Čeprav merski procesni model formalno obsega enake aktivnosti kot pristop PSM, vsebuje posebno izvedbo merjenja, ki je prilagojena zahtevam majhnih organizacij.

Proces:		**MERSKI PROCES**
Namen:		**Definicija celotnega merskega procesa, ki je sestavni del procesa razvoja PO**
	Glavna aktivnost:	**Podaktivnosti in koraki:**
1.	**UVAJANJE MERJENJA**	**Pridobivanje podpore organizacije**
		Določanje odgovornosti
		Skrb za potrebne vire
2.	**PLANIRANJE MERJENJA**	**Definiranje merskih izhodišč in določanje njihove prioritete** ○ Definiranje merskih izhodišč ○ Razvrščanje merskih izhodišč ○ Določanje prioritete merskih izhodišč
		Izbira in opis meritev ○ Izbira razreda meritev ○ Izbira primernih meritev ○ Opis podatkovnih zahtev
		Vključevanje merjenja v proces razvoja PO ○ Opis razvojnega okolja ○ Določanje priložnosti za merjenje ○ Opis zahtev za vgraditev meritev
3.	**IZVAJANJE MERJENJA**	**Zbiranje in obdelava podatkov** ○ Zbiranje podatkov ○ Preverjanje podatkov ○ Normalizacija podatkov
		Analiza merskih izhodišč ○ Ocenjevanje ○ Analiza izvedljivosti ○ Analiza performans ○ Napovedovanje
		Odločanje ○ Poročanje o rezultatih ○ Izbiranje različic ○ Ukrepanje
4.	**OVREDNOTENJE MERJENJA**	**Ovrednotenje meritev in indikatorjev**
		Ovrednotenje merskega procesa
		Dopolnjevanje podatkovne baze izkušenj
		Določanje in implementacija izboljšav

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.1.2 Umeščenost merskega procesa v proces razvoja PO

Spodnja tabela prikazuje aktivnosti prilagodljivega procesa razvoja PO, v okviru katerih se izvaja merjenje. Del merjenja se izvaja tudi v okviru aktivnosti majhne organizacije, ki ga proces razvoja PO ne podrobno definira (organizacijska raven delovanja).

Glavna aktivnost:	Podaktivnosti (in koraki):	Aktivnost prilagodljivega procesa razvoja PO
UVAJANJE MERJENJA		**Organizacijske aktivnosti skupine SEPG**
PLANIRANJE MERJENJA	**Definiranje merskih izhodišč in določanje prioritet**	**Organizacijska aktivnost skupine SEPG** **Začetek projekta – Priprava projekta (1.1)** **Vsi cikli – priprave ciklov**
	Izbira in opis meritev	**Organizacijska aktivnost skupine SEPG** **Začetek projekta – Priprava projekta (1.1)** **Vsi cikli – Priprave ciklov**
	Vključevanje merjenja v proces razvoja PO	**Organizacijska aktivnost skupine SEPG** **Začetek projekta – Priprava projekta (1.1)** **Cikli – Priprave ciklov** **Podporne aktivnosti – Nadgradnja procesa (0.5)**
IZVAJANJE MERJENJA	**Zbiranje in obdelava podatkov**	**Ves čas izvajanja projekta** **Podporne aktivnosti – Zbiranje merskih podatkov**
	Analiza merskih izhodišč - I ○ Ocenjevanje ○ Analiza izvedljivosti	**Začetek projekta – Priprava projekta (1.1)** **Vsi cikli – Priprave ciklov**
	Analiza merskih izhodišč - II ○ Analiza performans ○ Napovedovanje	**Ves čas izvajanja projekta** **Vsi cikli – Zaključki ciklov** **Zaključek projekta – Analiza projekta**
	Odločanje	**Tedenski sestanki (0.1)** **Začetek projekta – Poročilo vodstvu (1.5)** **Vsi cikli – Priprave ciklov** **Vsi cikli –Zaključki ciklov** **Zaključek projekta – Poročilo vodstvu (7.3)** **Organizacijske aktivnosti – sestanki**
OVREDNOTENJE MERJENJA		**Organizacijske aktivnosti skupine SEPG** **Organizacijske aktivnosti skupine SQA**

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.1.3 Analize

Pomembne razlike med našim pristopom in pristopom PSM so vidne pri izvajanju analiz. To je še zlasti razvidno iz opisa pokazateljev, ki predstavljajo rezultate izvedbe analiz. Ti so podrobneje predstavljeni v dodatku B.2.5.

Proces:		**ANALIZA MERSKIH IZHODIŠČ**
Namen:		**Definicija štirih vrst analiz, ki jih uporablja merski proces**
	Vrsta analize:	**Izvedba analize:**
1.	**OCENJEVANJE**	**Izbira pristopa**
		Prilagoditev pristopa
		Izvedba ocenjevanja ○ Ocenjevanje velikosti končnega izdelka ○ Ocenjevanje obsega dela ○ Ocenjevanje porabe časa ○ Ocenjevanje stroškov ○ Ocenjevanje kakovosti
		Ovrednotenje ocen
2.	**ANALIZA IZVEDLJIVOSTI**	**Ovrednotenje planov**
		Ocenjevanje vpliva
		Določanje alternativ
3.	**ANALIZA PERFORMANS**	**Primerjava planiranih vrednosti z dejanskimi**
		Ocenjevanje vpliva
		Napovedovanje rezultatov
		Ovrednotenje alternativ
4.	**NAPOVEDOVANJE**	**Stalno napovedovanje globalnih vrednosti** ○ Velikost končnega izdelka ○ Velikosti izdelkov ciklov ○ Obseg dela celotnega projekta ○ Obseg dela posameznih ciklov in faz ○ Dolžina trajanja in datum zaključka projekta ○ Dolžina trajanja in mejniki posameznih ciklov in faz ○ Stroški projekta in stroški faz in ciklov ○ Kakovost končnega izdelka ○ Kakovost na štiri načine
		Predstavitev napovedanih vrednosti
Izhodni pogoji:		

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2: Merski informacijski model

Vsebina našega merskega pristopa je predstavljena z meta modelom in opisom naborov splošnih merskih izhodišč, različnih vrst meritev in pokazateljev.

B.2.1 Meta model

Meta model predstavlja strukturo podatkov o merjenju: podatkov, ki opisujejo merjenje, in podatkov, ki predstavljajo rezultate merjenja (izmerjene vrednosti). Naš pristop ga določa na treh ravneh: organizacijski, na ravni projekta v teku in na ravni preteklih projektov. Struktura je razvidna iz več diagramov entiteta razmerje, ki so predstavljeni v podpoglavju 5.3. Na tem mestu povzemamo entitete vseh treh ravni in njihove opise.

Entiteta	Raven	Opis
Analiza	Organizacijska	Nabor postopkov za določanje pokazateljev iz meritev
Analizni model	Organizacijska	Opis načina določanja in prikaza pokazatelja
Cikel	Organizacijska	Cikli procesa razvoja PO
Cikel projekta	Tekoči projekt	Cikli projektnega procesa razvoja PO
Član projekta	Tekoči projekt	Nabor članov projektne skupine
Definirana meritev	Organizacijska	Nabor definiranih (osnovnih in izpeljanih) meritev
Del meritve (izp./pok.)	Organizacijska	Posamezni deli izpeljane meritve ali pokazatelja
Del meritve (združene)	Organizacijska	Posamezni deli združene meritve ali pokazatelja
Dodatek	Tekoči projekt	Priloga, ki pojasnjuje določen rezultat analize – običajno besedilo
Določena vrednost	Tekoči projekt	Rezultati združenih in izpeljanih meritev, izračunane s pomočjo združevalnih struktur ali merskih funkcij – vrednosti, izražene v ustreznih enotah
Entiteta	Organizacijska	Objekti, ki jih merimo (na različnih ravneh podrobnosti)
Entiteta projekta	Tekoči projekt	Vsi merski objekti, ki jih lahko merimo v okviru projekta
Faza	Organizacijska	Faze, ki pripadajo posameznim ciklom procesa
Faza projekta	Tekoči projekt	Faze, ki se izvajajo v okviru projekta
Izmerjena vrednost	Tekoči projekt	Vrednosti osnovnih meritev, ki jih razvijalci izmerijo ob izvedbi vsakega koraka
Izpeljana meritev	Organizacijska	Meritev, določena iz več (osnovnih/izpeljanih) meritev
Izvedba koraka	Tekoči projekt	Opis deleža koraka, ki ga izvaja en izvajalec
Korak	Organizacijska	Koraki, ki so del posameznih faz procesa
Korak projekta	Tekoči projekt	Koraki, ki izvajajo v okviru projekta
Mer. izhodišča/Pokazatelj	Organizacijska	Pokazatelji za določena merska izhodišča
Mer. izhodišče/Pokazatelj	Tekoči projekt	Povezava med pokazatelji in (stalnimi in posebnimi) merskimi izhodišči
Meritev projekta	Tekoči projekt	Nabor (osnovnih, združenih in izpeljanih) meritev projekta
Merljiv atribut	Organizacijska	Značilnost, ki jo lahko kvantitativno ocenimo
Merska enota	Organizacijska	Enota za merjenje
Merska funkcija	Organizacijska	Opis načina izvedbe izpeljane meritve
Merska metoda	Organizacijska	Opis načina, kako izvedemo osnovno meritev
Merska skala	Organizacijska	Nabor dovoljenih vrednosti rezultatov osnovne meritve
Napoved projekta	Tekoči projekt	Napovedi (vrednost in meji 90% območja zaupanja) za stalen nabor atributov, ki so najboljše v danem trenutku
Normalizacijska tabela	Organizacijska	Razmerje med vrednostjo nekaterih osnovnih meritev in končno velikostjo s stopnjo odvisnosti (glede na pretekle projekte)
Ocena projekta	Tekoči projekt	Ocene (vrednost in meji 90% območja zaupanja) za stalen nabor atributov, ki se določajo ob vsakem planiranju
Odločitveni kriterij	Organizacijska	Kriterij, ki na podlagi merskega izhodišča določa merilo za ustreznost rezultata
Oseba	Organizacijska	Opis oseb, ki so (bile) zaposlene v organizaciji
Osnovna meritev	Organizacijska	Neposredna meritev merljivega atributa entitete
Pokazatelj	Organizacijska	Vrednost, določena na podlagi dveh ali več meritev in merskega izhodišča
Pokazatelj projekta	Tekoči projekt	Nabor pokazateljev, ki jih uporablja projekt
Posebno mersko izhodišče	Tekoči projekt	Nabor merskih izhodišč, ki naslavljajo posebne potrebe projekta
Proces	Organizacijska	Opis organizacijskega procesa razvoja PO – en element
Projekt	Tekoči projekt	Opis projekta – omogoča združevanje podatkov več projektov
Razmerja izdelkov faz	Pretekli projekti	Razmerje med vrednostjo nekaterih osnovnih meritev in končno velikostjo
Razmerja obsega dela faz	Pretekli projekti	V odstotkih izraženo razmerje potrebnega obsega dela za izvedbo faz in ciklov
Razmerja stroškov faz	Pretekli projekti	V odstotkih izraženo razmerje dodatnih stroškov za izvedbo faz in ciklov
Razred meritev	Organizacijska	Poraba časa in napredovanje, velikost izdelkov, viri in stroški, kakovost izdelkov
Rezultat analize projekta	Tekoči projekt	Predstavitev pokazateljev - rezultatov pomembnejših (opis in naslov shranitve) analiz izvedljivosti in analiz performans, določenih s pomočjo analiznih modelov in odločitvenih kriterijev (običajno v obliki)

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Entiteta (nadaljevanje)	Raven	Opis
Rezultat cikla	Organizacijska	Izdelki, ki so dokončani ob zaključku cikla
Rezultat faze	Organizacijska	Vmesni izdelki, ki so dokončani ob zaključku faze
Rezultat faze projekta	Tekoči projekt	Izdelki faz, ki so razviti med izvajanjem projekta
Rezultat koraka projekta	Tekoči projekt	Vmesni izdelki posameznih korakov projekta
Rezultati projekta	Pretekli projekti	Povzetek glavnih rezultatov za celoten projekt
Skupina	Organizacijska	Opis vseh skupin na ravni organizacije
Stalna merska izhodišča	Organizacijska	Nabor merskih izhodišč, ki jih vedno merimo
Tip analize	Organizacijska	Ocenjevanje, izvedljivost, performanse, napovedovanje
Tip dodatka	Tekoči projekt	Vrsta dodatka, recimo pojasnilo, predlog, opis predpostavk ali kaj podobnega
Tip entitete	Organizacijska	Aktivnost, izdelek, projekt ali vir
Tip kriterija	Organizacijska	Omejitev, statistična metoda, hevristika, ...
Tip merske metode	Organizacijska	Objektivna ali subjektivna
Tip merske skale	Organizacijska	Nominalna, ordinalna, intervalna, razmernostna, ...
Tip merskih izhodišč	Organizacijska	Cilj, težava, tveganje ali pomanjkanje informacij
Tip usposabljanja	Organizacijska	Opis vrste usposabljanja
Tipična razmerja obsega dela	Organizacijska	V odstotkih izraženo razmerje potrebnega obsega dela za izvedbo posameznih faz in ciklov glede na podatke preteklih projektov
Tipična razmerja stroškov	Organizacijska	V odstotkih izraženo razmerje dodatnih stroškov za izvedbo posameznih faz in ciklov glede na podatke preteklih projektov
Usposabljanje	Organizacijska	Seznam usposabljanj po posameznih osebah
Vloga	Organizacijska	Vse vloge, ki jih predpisuje proces razvoja PO
Združena meritev	Organizacijska	Meritev atributa, ki je določena iz več osnovnih meritev
Združevalna struktura	Organizacijska	Opis načina združevanja osnovnih v združene meritve

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.2 Razredi meritev

Naš merski pristop razvršča lastnosti (atribute) merskih entitet v štiri razrede:

- Poraba časa in napredovanje;
- Viri in stroški;
- Velikost izdelkov;
- Kakovost izdelkov.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.3 Stalen nabor merskih izhodišč

Stalen nabor merskih izhodišč je določen na podlagi omejitev in zahtev majhne organizacije za razvoj IS ter ciljev posameznih ključnih področij modela CMM. Prvi del prispeva splošne cilje, medtem ko drugi del definira nabor merskih izhodišč. Drugi del merskih izhodišč je prikazan razdrobljeno po stopnjah modela CMM. Predstavljeni so cilji ključnih področij, ustrezna merska izhodišča ter merske entitete, na katere se izhodišča nanašajo.

B.2.3.1 Majhne organizacije za razvoj informacijskih sistemov

Cilj	Opis cilja
Splošni cilj 1	Upoštevati je treba drugačno strukturo procesa razvoja PO in pri merjenjih ustrezno podpreti posamezne dele razvoja.
Splošni cilj 2	Na ustrezen način moramo meriti tako aktivnosti razvoja kot aktivnosti uvajanja rezultatov v uporabo.
Splošni cilj 3	Razlikovati moramo med merjenjem samih izdelkov in merjenjem dela izdelkov, ki te pripravi za namestitev.
Splošni cilj 4	Merjenje mora omogočati spremljanje tako procesnega kot podatkovnega vidika izdelkov.
Splošni cilj 5	Za pomembne aktivnosti predvidimo posebna merjenja.
Splošni cilj 6	Merjenja morajo odražati povezave med določenimi vmesnimi izdelki in te povezave uporabljati pri analizah.
Splošni cilj 7	Merjenja morajo upoštevati standardne pristope, ki jih uporabljajo programska orodja.
Splošni cilj 8	Merjenje naj bo prilagojeno zahtevam in zmožnostim majhne organizacije.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.3.2 Druga stopnja modela CMM: Ponovljiv proces

2.1	Upravljanje z zahtevami	
Cilj 1	Zahteve sistema glede PO so pod nadzorom in predstavljajo izhodišče za tehnično realizacijo in vodenje projekta.	
Cilj 2	Plani razvoja, izdelki in aktivnosti so konsistentni z zahtevami.	
Mer. Izh. 2.1.1	Seznam zahtev	Spremljaj status zahtev
Mer. Izh. 2.1.2	Seznam zahtev	Spremljaj spremembe zahtev
Mer. Izh. 2.1.3	Aktivnosti upravljanja z zahtevami	Spremljaj status aktivnosti upravljanja z zahtevami

2.2	Planiranje projektov	
Cilj 1	Ocene glede PO so dokumentirane za potrebe planiranja in spremljanja opravil na projektu.	
Cilj 2	Aktivnosti na projektu in zadolžitve so planirane in dokumentirane.	
Cilj 3	Prizadete skupine in posamezniki soglašajo z zadolžitvami, ki so povezane s projektom.	
Mer. Izh. 2.2.1	Vse aktivnosti	Spremljaj razpored planirane izvedbe opravil projekta (ocena)
Mer. Izh. 2.2.2	Vse aktivnosti	Spremljaj potreben obseg dela za izvedbo opravil
Mer. Izh. 2.2.3	Aktivnosti planiranja	Spremljaj status aktivnosti planiranja

2.3	Zasledovanje in nadzor nad projekti	
Cilj 1	Dejanski rezultati se zasledujejo v primerjavi s planom.	
Cilj 2	V primeru večjih odstopanj od plana se izvajajo aktivnosti za izboljšanje stanja.	
Cilj 3	Spremembe zadolžitev so dogovorjene med prizadetimi skupinami in posamezniki.	
Mer. Izh. 2.3.1	Končni izdelki	Spremljaj velikost končnih izdelkov
Mer. Izh. 2.3.2	Vse aktivnosti	Spremljaj obseg dela projekta
Mer. Izh. 2.3.3	Vse aktivnosti	Spremljaj potek projekta
Mer. Izh. 2.3.4	Vse aktivnosti	Spremljaj težave med izvajanjem projekta
Mer. Izh. 2.3.5	Aktivnosti zasledovanja	Spremljaj status aktivnosti zasledovanja in nadzora nad projekti

2.4	Urejanje razmerij s poddobavitelji
Cilj 1	Organizacija za razvoj PO (prvi dobavitelj) izbira usposobljene pod dobavitelje.
Cilj 2	Organizacija za razvoj PO in pod dobavitelji se dogovorijo o medsebojnih obveznostih.
Cilj 3	Organizacija za razvoj PO in dobavitelj vzdržujeta stalne stike.
Cilj 4	Organizacija za razvoj PO zasleduje rezultate dela pri podizvajalcu in primerja rezultate z obveznostmi tega.
Se ne uporablja!	

2.5	Zagotavljanje kakovosti PO	
Cilj 1	Aktivnosti za zagotavljanje kakovosti so planirane.	
Cilj 2	Skladnost izdelkov in aktivnost za uporabljenimi standardi, postopki in zahtevami se presoja objektivno.	
Cilj 3	Prizadete skupine in posamezniki so obveščeni o aktivnostih za zagotavljanje kakovosti in rezultati teh.	
Cilj 4	Vodstvo organizacije rešuje vsa neskladja, ki jih ni moč rešiti v okviru projekta.	
Mer. Izh. 2.5.1	SQA aktivnosti	Spremljaj preverjanja in preglede procesa in izdelkov
Mer. Izh. 2.5.2	SQA aktivnosti	Spremljaj težave na področju zagotavljanja kakovosti
Mer. Izh. 2.5.3	SQA aktivnosti	Spremljaj status aktivnosti za zagotavljanje kakovosti (SQA aktivnosti)

2.6	Upravljanje s konfiguracijo	
Cilj 1	Aktivnosti, povezane z upravljanjem s konfiguracijo, so planirane.	
Cilj 2	Določeni so rezultati dela, nad katerimi poteka nadzor in se zagotavlja razpoložljivost.	
Cilj 3	Spremembe teh izdelkov so nadzorovane.	
Cilj 4	Prizadete skupine in posamezniki so obveščeni o trenutnem stanju in vsebini posameznih izdelkov.	
Mer. Izh. 2.6.1	Opis konfiguracije	Spremljaj status konfiguracij
Mer. Izh. 2.6.2	Opis konfiguracije	Spremljaj spremembe konfiguracij
Mer. Izh. 2.6.3	Opis konfiguracije	Spremljaj odstopanja v konfiguraciji
Mer. Izh. 2.6.4	Aktivnosti upravljanja konfiguracije	Spremljaj status aktivnosti za upravljanje s konfiguracijo

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.3.3 Tretja stopnja modela CMM: Definiran proces

3.1	Osredotočenje na proces, ki ga uporablja organizacija	
Cilj 1	Aktivnosti za razvoj PO in izboljšanje procesa razvoj PO so usklajene v celotni organizaciji.	
Cilj 2	Znane so prednosti in pomanjkljivosti procesa razvoja PO, ki se trenutno uporablja.	
Cilj 3	Izdelan je plan aktivnosti za razvoj in izboljšanje procesa razvoja PO na ravni organizacije.	
Mer. Izh. 3.1.1	Aktivnosti za proces	Spremljaj in napoveduj razpored izvajanja aktivnosti za koordinacijo in izboljšanje procesa razvoja PO
Mer. Izh. 3.1.2	Aktivnosti za proces	Spremljaj in napoveduj stroške izvedbe aktivnosti
Mer. Izh. 3.1.3	SQA poročila	Spremljaj in napoveduj rezultate preverjanj aktivnosti za koordinacijo in izboljšanje procesa razvoja PO
Mer. Izh. 3.1.4	Aktivnosti za proces	Spremljaj in napoveduj status aktivnosti za koordinacijo in izboljšanje procesa razvoja PO

3.2	Definicija procesa razvoja PO organizacije	
Cilj 1	Obstaja standarden proces razvoja PO, ki se razvija in vzdržuje za potrebe celotne organizacije.	
Cilj 2	Informacije o načinih uporabe (prilagajanju) standardnega procesa razvoja PO za posamezne projekte se zbirajo, pregledujejo in so vsem dostopne.	
Mer. Izh. 3.2.1	Aktivnosti za proces	Spremljaj in napoveduj razpored izvajanja aktivnosti za razvoj in vzdrževanje standardnega procesa razvoja PO
Mer. Izh. 3.2.2	Aktivnosti za proces	Spremljaj in napoveduj obseg dela in stroške izvedbe aktivisti za razvoj in vzdrževanje standardnega procesa razvoja PO
Mer. Izh. 3.2.3	Aktivnosti za proces	Spremljaj in napoveduj status aktivnosti za razvoj in vzdrževanje standardnega procesa razvoja PO

3.3	Program izobraževanja	
Cilj 1	Aktivnosti za izobraževanje se planirajo.	
Cilj 2	Zagotovljeno je izobraževanje za tista področja, ki so potrebna pri vodenju projektov in izvajanju tehničnih opravil.	
Cilj 3	Posamezniki v vseh skupinah (še zlasti v razvojni skupini) dobijo potrebno izobraževanje za svoje delo.	
Mer. Izh. 3.3.1	Osebje	Spremljaj in napoveduj udeležbo pri izobraževanju
Mer. Izh. 3.3.2	Ankete udeležencev	Spremljaj in napoveduj rezultate anket udeležencev
Mer. Izh. 3.3.3	Aktivnosti izobraževanja	Spremljaj in napoveduj status aktivnosti izobraževanja

3.4	Integrirano vodenje projektov razvoja PO	
Cilj 1	Proces razvoja PO vsakega projekta je izpeljan iz standardnega procesa organizacije.	
Cilj 2	Planiranje in vodenje projekta poteka v skladu s procesom razvoja PO, ki je bil definiran za ta projekt.	
Mer. Izh. 3.4.1	Vsi izdelki faz in ciklov	Spremljaj in napoveduj velikost posameznih izdelkov
Mer. Izh. 3.4.2	Vse aktivnosti	Spremljaj in napoveduj potreben obseg dela za posamezne aktivnosti
Mer. Izh. 3.4.3	Osebje	Spremljaj in napoveduj vire pri izvajanju projektov
Mer. Izh. 3.4.4	Vse aktivnosti	Spremljaj in napoveduj stroške pri izvajanju projektov
Mer. Izh. 3.4.5	Vse aktivnosti	Spremljaj in napoveduj časovni razpored izvajanja projektov
Mer. Izh. 3.4.6	Vse aktivnosti	Spremljaj in napoveduj napake pri izvajanju projektov
Mer. Izh. 3.4.7	Aktivnosti vodenja projektov	Spremljaj in napoveduj status aktivnosti integriranega vodenje projektov

3.5	Tehnologija izdelkov PO	
Cilj 1	Tehnične naloge so definirane, integrirane in se konsistentno izvajajo v okviru procesa razvoja PO.	
Cilj 2	Vsi izdelki PO so med seboj usklajeni (konsistentni).	
Mer. Izh. 3.5.1	Vsi izdelki faz in ciklov	Spremljaj in napoveduj funkcionalnost izdelkov
Mer. Izh. 3.5.2	Vsi izdelki faz in ciklov	Spremljaj in napoveduj kakovost izdelkov
Mer. Izh. 3.5.3	Aktivnosti razvoja	Spremljaj in napoveduj status aktivnosti tehnologije izdelkov

3.6	Koordinacija med različnimi skupinami	
Cilj 1	Zahteve naročnika so usklajene med vsemi prizadetimi skupinami.	
Cilj 2	Zadolžitve posameznih skupin so usklajene med vsemi prizadetimi skupinami.	
Cilj 3	Določene skupine odkrivajo, zasledujejo in razrešujejo vprašanja, ki se nanašajo na medsebojne odnose.	
Mer. Izh. 3.6.1	Aktivnosti sodelovanja	Spremljaj in napoveduj stroške sodelovanja z zunanjimi skupinami
Mer. Izh. 3.6.2	Aktivnosti sodelovanja	Spremljaj in napoveduj status aktivnosti sodelovanja z zunanjimi skupinami

3.7	Pregledi med kolegi	
Cilj 1	Pregledi so planirani.	
Cilj 2	Pomanjkljivosti v izdelkih so odkrite in odstranjene.	
Mer. Izh. 3.7.1	Aktivnosti preverjanja	Spremljaj in napoveduj status aktivnosti preverjanja (namesto pregledov med kolegi)

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.3.4 Četrta stopnja modela CMM: Upravljan proces

4.1	**Kvantitativno vodenje procesov**	
Cilj 1	Aktivnosti za kvantitativno upravljanje procesov so planirane.	
Cilj 2	Uveden je kvantitativni nadzor nad rezultati procesov, ki so bili uporabljeni na projektu.	
Cilj 3	Zmožnost procesa v organizaciji je mogoče izraziti na kvantitativen način.	
Mer. Izh. 4.1.1	Vse aktivnosti	(Recimo) Spremljaj in nadzoruj produktivnost
Mer. Izh. 4.1.2	Vsi izdelki	(Recimo) Spremljaj in nadzoruj ponovna uporabljivost
Mer. Izh. 4.1.3	Vse aktivnosti	(Recimo) Spremljaj in nadzoruj učinkovitost izvedbe
Mer. Izh. 4.1.4	Aktivnosti merjenja	Spremljaj in nadzoruj status aktivnosti kvantitativnega vodenja procesov

4.2	**Upravljanje za kakovost**	
Cilj 1	Aktivnosti za upravljanje kakovosti so planirane.	
Cilj 2	Definirani so merljivi cilji in njihove prioritete za kakovost izdelkov.	
Cilj 3	Napredek pri doseganju ciljev je kvantificiran in voden.	
Mer. Izh. 4.2.1	Vse aktivnosti	Spremljaj in nadzoruj kakovost s pomočjo razporeditve napak
Mer. Izh. 4.2.2	SQA aktivnosti	Spremljaj in nadzoruj status aktivnosti upravljanja kakovosti

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.4 Nabor meritev

Naš merski pristop določa tri vrste meritev:

- osnovne meritve,
- združene meritve in
- izpeljane meritve

B.2.4.1 Nabor osnovnih meritev

Stalen nabor osnovnih meritev obsega merjenje, ki ga izvajajo razvijalci pri vsakem koraku ali delu koraka. Nabor je sestavljen tako, da pokriva ključne (merljive) značilnosti aktivnosti, virov in rezultata aktivnosti, v kolikor ta nastane. Rezultati osnovnih meritev so podlaga za združene in izpeljane meritve, ki se ne merijo, temveč določajo.

Osnovne meritve so predstavljene po razredih meritev. Za vsako meritev je predstavljena entiteta ali vrsta entitete, atribut in enota merjenja. Zaradi boljše predstave so osnovne meritve velikosti izdelkov predstavljene tudi za rezultate posameznih faz procesa.

Poraba časa in napredovanje

Razred meritev	Entiteta	Atribut	Meritev	Enota
Poraba časa	Del koraka (Korak)	Datum in čas začetka Datum in čas zaključka	Dolžina trajanja	minuta (ura, dan)
Napredovanje	Vsi izdelki cikla ali faze	Zaključen izdelek (mejnik)	Zadnji dosežen mejnik	1

Viri in stroški

Razred meritev	Entiteta	Atribut	Meritev	Enota
Viri	Del koraka (Korak)	Dejanski čas za izvedbo	Obseg dela aktivnosti	človek-minuta (človek-ura, človek-dan)
Stroški	Del koraka (Korak)	Dodatni stroški	Znesek	SIT

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Velikost izdelkov

Razred meritev	Vrsta Entitete	Atribut	Osnovna meritev	Enota
Velikost izdelkov – rezultati faz in ciklov	**Dokument, predlog, opis, poročilo**	Skupna dolžina	Število vrstic ali strani	vrstica ali stran
	Seznam zahtev, standard (raziskave zahtev, načrtovanja, kodiranja, ipd.)	Količina delov Dolžina vsakega dela Skupna dolžina	Število delov Število vrstic Število vrstic ali strani	del vrstica vrstica ali stran
	Diagram (ERD, DFD)	Količina vsake vrste elementov Dolžina celotnega opisa	Število vsake vrste elementov Število vrstic ali strani	element vrstica ali stran
	Diagram sestave (FD, konceptualni načrt, struktura menujev, diagram navigacije)	Količina ravni Količina elementov Dolžina celotnega opisa	Število ravni Število elementov Število vrstic ali strani	raven element vrstica ali stran
	Matrika	Količina elementov	Število elementov	element
	Prototip	Količina delov Dolžina pr. kode za vsak del Dolžina pr. kode skupaj	Število delov Število vrstic kode Število vrstic kode	del vrstica kode vrstica kode
	Fizični načrt (procesiranja in PB), **del IS**	Količina vsake vrste elementov	Število vsake vrste elementov	element
	Modul (deli programa, poročila, obrazci), **sprožilec, bazna procedura**	Dolžina prog. kode	Število vrstic kode	vrstica kode
	Zaslon	Količina elementov	Število elementov	element
	PB	Količina vsake vrste elementov	Število vsake vrste elementov	element
	Podatki v PB (testni, pravi), **arhivirani deli**	Obseg porabljenega prostora	Poraba pomnilnika	MB
	Dokumentacija (sistemska uporabniška, učna)	Količina delov Dolžina vsakega dela Skupna dolžina	Število delov Število vrstic ali strani Število vrstic ali strani	del vrstica ali stran vrstica ali stran
Velikost izdelkov – opis projekt	**Projektna knjiga** (opis, plani, poročila)	Dolžina vsakega dela Skupna dolžina	Število vrstic ali strani Število vrstic ali strani	vrstica ali stran vrstica ali stran
Velikost izdelkov – opis procesiranja	**Merski podatki** (izmerjene vrednosti, informacijski izdelki)	Količina podatkov Količina inf. izdelkov Dolžina inf. izdelkov	Število merjenih vrednosti Število inf izdelkov Število vrstic ali strani	meritev inf. izdelkov vrstica ali stran
	Poročilo o testiranju	Dolžina poročila	Število vrstic ali strani	vrstica ali stran
	Predlogi, zahtevki	Količina predlogov ali zahtevkov	Število predlogov ali zahtevkov	predlogov ali zahtevkov

Tabela v nadaljevanju podrobneje prikazuje osnovne meritve velikosti z vidika rezultatov faz in ciklov procesa razvoja PO. Prvi stolpec predstavlja izdelke posameznih ciklov, kot jih lahko razberemo iz dodatka A.3. Drugi stolpec predstavlja število izdelkov cikla, ki nastanejo v okviru enega projekta, tretji pa meritve velikosti celotnega izdelka. Četrti, peti in šesti stolpec predstavljajo izdelke faz, ki sestavljajo posamezne izdelke ciklov. Ti izdelki so lahko različnih vrst. Iz petega stolpca lahko razberemo, koliko izdelkov izbrane vrste se lahko nahaja v izdelku cikla. Šesti stolpec predstavlja osnovne meritve za te izdelke. Nekateri izdelki faz se delijo še naprej. Sestavni deli in meritve zanje so predstavljeni v zadnjih dveh stolpcih.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Izdelek cikla		Osnovna meritev	Sestavni deli		Osnovna meritev	Sestavni deli	Osnovna meritev
Inicializacijski dokument	1	Število vrstic					
Strateški dokument	1	Število strani	VnERD	1	Število entitet		
					Število razmerij		
					Število vrstic opisa ERD		
			VnDFD	1	Število el. procesov		
					Število pod. skladišč		
					Število zun. entitet		
					Število pod. tokov		
					Število vrstic opisa DFD		
			VnFD	1	Število ravni		
					Število funkcij		
					Število vrstic opisa FD		
Standardi raziskave zahtev	V e č	Skupno število	Standard	1	Število vrstic		
		Skupno število vrstic					
Specifikacija zahtev	1	Število strani	Seznam zahtev	1	Število zahtev	Zahteva	Število vrstic zahteve
					Skupno število vrstic zahtev		
			LERD	1	Število entitet		
					Število razmerij		
					Število atributov		
					Število vrstic opisa ERD		
			LDFD	1	Število el. procesov		
					Število pod. skladišč		
					Število zun. entitet		
					Število pod. tokov		
					Število vrstic opisa DFD		
			LFD	1	Število ravni		
					Število funkcij		
					Število vrstic opisa FD		
			Matrika funkcije/ zahteve	1	Število elementov		
			Matrika entitete/zahteve	1	Število elementov		
			Matrika entitete/funkcije	1	Število elementov		
Standardi načrtovanja	V e č	Skupno število	Standard	1	Število vrstic		
		Skupno število vrstic					
Specifikacija načrtovanja	1	Število strani	Konceptualni načrt	1	Število ravni		
					Število modulov		
					Število vrstic opisa modulov		
			Predlogi tehnične rešitve	V e č	Število predlogov		
					Skupno število vrstic		
			Prototipi	V e č	Število prototipov Skupno število vrstic kode vseh prototipov	Prototip	Število vrstic kode prototipa
			Strategija implementacije	1	Število vrstic		
			Fizični načrt IS	1	Število modulov Število zaslonov	Modul	Št.vrstic opisa mod.
						Zasloni	Št.elem. zaslona
						Diagram navigacije	Število ravni
							Število elementov
						Struktura menujev	Število ravni
							Število elementov
			Fizični načrt PB	1	Število tabel		
					Število atributov		
					Število razmerij		
					Število pogledov		
			Matrika funkcije/moduli	1	Število elementov		

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Standardi izgradnje	V e č	Skupno število Skupno število vrstic	Standard	1	Število vrstic		
IS in vsak pomožni del, ki služi za testiranje, namestitev ali vzdrževanje (cikli 4, 5, 6)	V e č	Število modulov Skupno število vrstic programske kode	Del IS	V e č	Število vrstic kode Število zaslonov Skupno število elementov zaslonov Število menujev Skupno število elementov menujev	Moduli Zasloni Menuji	Število vrstic kode Modula Število ravni Število elementov Zaslona Število ravni Število elementov Menuja
PB in vsak pomožni del PB, ki služi za testiranje, namestitev ali vzdrževanje (cikli 4, 5 in 6)	V e č	Skupno število vseh elementov	Del PB	V e č	Število tabel Število atributov Število razmerij Število pogledov Število indeksov Število sekvenc Število sprožilcev Število baznih procedur Skupno število vrstic kode baznih procedur in sprožilcev	Sprožilci Bazne procedure	Število vrstic kode sprožilca Število vrstic kode bazne procedure
Podatki (testni in pravi)	V e č		Podatki PB	1	Velikost porabe prostora		
Dokumentacije (sistemska, učna, uporabniška)	V e č	Skupno število strani vse dokumentacije	Dokumentacija	1	Število strani		
Arhivirani deli (vsi)	V e č		Arhivirani del	1	Velikost porabe prostora		
Projektna knjiga (se dopolnjuje v vsakem ciklu)	1	Skupno število vrstic	Opis projekta	1	Število vrstic		
			Plani (globalni in po ciklih)	V e č	Število vrstic vseh planov	Plan opravil (globalni ali za člane) Plan porabe časa (globalni ali za člane) Plan kakovosti Plan merjenja Drugo (tveganja, cilji)	Število vrstic Število vrstic Število vrstic Število vrstic Število vrstic
			Poročila (o ciklih)	V e č	Število vrstic vseh poročil	Poročilo	Število vrstic
			Končno poročilo	1	Število vrstic		
Dokumenti (namestitev, vzdrževanje)	V e č	Skupno število vrstic	Dokument	1	Število vrstic		
Merski podatki (v vsakem ciklu in v končnem poročilu)	V e č	Skupno število meritev Skupno število inf. izdelkov Skupno število vrstic inf. izdelkov	(Delne) meritve	V e č	Število meritev		
			Informacijski izdelki	V e č	Število izdelkov	Informacijski izdelek	Število vrstic izdelka
Poročila o testiranju (cikli 4, 5, 6)	V e č	Skupno število vrstic	Poročilo o testiranju	1	Število vrstic		
Predlogi za izboljšavo (vsak cikel)	V e č	Skupno število					
Zahtevki za spremembe (vsak cikel)	V e č	Skupno število					

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Kakovost izdelkov

Razred meritev	Entiteta	Atribut	Meritev	Enota
Kakovost izdelkov	Del koraka (Korak)	Število odkritih napak	Število vseh odkritih napak	napaka
	Del koraka (Korak)	Število odkritih napak po fazah vnosa	Število odkritih napak po fazah vnosa	napaka
	Del koraka (Korak)	Število odkritih napak po razredih napak	Število odkritih napak po razredih napak	napaka
	Del koraka (Korak)	Število odpravljenih napak	Število vseh odpravljenih napak	napaka
	Del koraka (Korak)	Število odpravljenih napak po fazah vnosa	Število odpravljenih napak po fazah vnosa	napaka
	Del koraka (Korak)	Število odpravljenih napak po razredih napak	Število odpravljenih napak po razredih napak	napaka
	Del koraka (Korak)	Potreben čas za odpravo napak	Skupen obseg dela odpravljanja napak	napaka

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.4.2 Nabor združenih meritev

Združene meritve omogočajo, da rezultate določimo iz rezultatov osnovnih meritev z uporabo združevalnih struktur. Naš merski pristop omogoča združevanje na naslednje načine:

- ZR - po rezultatih (kateri izdelki sestavljajo druge izdelke);
- ZF - po funkcijah (vse aktivnosti z isto funkcijo, recimo vsa preverjanja kakovosti);
- ZA - po aktivnostih (recimo koraki sestavljajo faze, faze sestavljajo cikle);
- ZV - po vidiku uporabe (podatkovni in procesni vidik);
- ZI - po izvajalcih (aktivnosti, ki jih izvaja isti razvijalec).

Nabor združenih meritev ni stalno določen, temveč meritve z uporabo združevalnih struktur določamo in uporabljamo po potrebi. V nadaljevanju je predstavljen nabor tistih združenih meritev, ki jih potrebujemo glede na določena stalna merska izhodišča. V okviru posebnih merskih izhodišč lahko določimo še druge združene meritve.

Za razliko od osnovnih meritev so tokrat vse meritve prikazane v eni tabeli, dodan pa je tudi stolpec tip, ki pojasnjuje združevalno strukturo.

Razred meritev	Entiteta	Atribut	Meritev	Tip	Enota
Poraba časa	Cikel, Faza (Korak)	Datum in čas začetka Datum in čas zaključka	Dolžina trajanja	ZA	minuta (ura, dan)
Velikost izdelka rezultati faz	**Specifikacija** (zahtev, načrtovanja)	Skupna dolžina	Število strani	ZR	stran
	Inf. sistem	Skupna dolžina	Število vrstic prog. kode	ZR	vrstica kode
	PB	Skupna količina elementov	Število vseh elementov	ZR	element
	Dokumentacija	Skupna dolžina	Število strani	ZR	stran
Velikost izdelka opis projekta	**Projektna knjiga**	Skupna dolžina	Število strani	ZR	stran
Velikost izdelka opis procesiranja	**Merski podatki**	Skupna količina podatkov Skupna dolžina inf. izdelkov	Število merjenih vrednosti Število strani	ZA ZR	meritev stran
Viri	Cikel, Faza (Korak)	Potreben obseg dela za izvedbo	Obseg dela aktivnosti	ZA	človek-ura, človek-dan
	Aktivnosti razvoja	Potreben obseg dela za izvedbo	Obseg dela aktivnosti	ZF	človek-ura, človek-dan
	Aktivnosti za odpravo napak	Potreben obseg dela za odpravljanje napak	Obseg dela aktivnosti	ZF	človek-ura, človek-dan
	Aktivnosti razvoja	Potreben obseg dela za izvedbo	Obseg dela aktivnosti	ZF	človek-ura, človek-dan
Stroški	Cikel, Faza (Korak)	Dodatni stroški	Znesek	ZA	SIT
Kakovost izdelkov	Faza ali cikel	Število odkritih napak aktivnosti	Št. odkritih napak aktivn.	ZA	Napaka
	Izdelek faze ali cikla	Število odkritih napak izdelka	Št. odkritih napak izdelka	ZR	Napaka
	Izdelki faze ali cikla	Število odkritih napak procesnih ali podatkovnih izdelkov	Število odkritih napak v izdelku	ZV	Napaka
	Faza ali cikel	Število odkritih napak izbranega razvijalca	Število odkritih napak izbranega razvijalca	ZI	Napaka
	Faza ali cikel	Število odpravljenih napak aktivnosti	Število odpravljenih napak aktivnosti	ZA	Napaka
	Izdelek faze ali cikla	Število odpravljenih napak izdelka	Število odpravljenih napak izdelka	ZR	Napaka
	Izdelki faze ali cikla	Število odpravljenih napak (proc. ali pod.) izdelkov	Število odpravljenih napak v izdelku	ZV	Napaka
	Faza ali cikel	Število odpravljenih napak izbranega razvijalca	Število odpravljenih napak izbranega razvijalca	ZI	Napaka
	Faza ali cikel	Potreben čas za odpravo napak aktivnosti	Skupen obseg dela odpravljanja napak akt.	ZA	Napaka
	Izdelek faze ali cikla	Potreben čas za odpravo napak izdelka	Skupen obseg dela odpravljanja napak izd.	ZR	Napaka
	Izdelki faze ali cikla	Potreben čas za odpravo napak procesnih ali podatkovnih izdelkov	Skupen obseg dela odpravljanja napak v (proc. ali pod.) izdelku	ZV	Napaka
	Faza ali cikel	Potreben čas za odpravo napak izbranega razvijalca	Skupen obseg dela odpravljanja napak izbranega razvijalca	ZI	Napaka

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.4.3 Nabor izpeljanih meritev

Z izpeljanimi meritvami določimo rezultate, ki upoštevajo značilnosti več različnih vrst entitet. Določimo jih na podlagi dveh ali več osnovnih, združenih ali izpeljanih meritev.

Nabor izpeljanih meritev, ki izhaja iz stalnih merskih izhodišč, določa pet izpeljanih meritev. Z dodatnimi posebnimi merskimi izhodišči ga lahko po potrebi dopolnimo.

Izpeljane meritve so predstavljene z opisom izpeljave, udeleženih meritev in entitet, nad katerimi izvajamo udeležene meritve.

Razred meritev	Opis	Merska funkcija	Udeležene meritve	Entiteta	Enota
Produktivnost	Razmerje obsega dela in velikosti	deljenje	Obseg dela Velikost rezultata	Aktivnosti Izdelki Vidiki Razvijalci	Odvisno od enote velikosti
Izkoristek odpravljanja napak	Razmerje med številom odpravljenih in številom obstoječih napak	deljenje	Število odpravljenih napak Število obstoječih napak	Aktivnosti Izdelki Vidiki Razvijalci	Odstotki
Strošek zagotavljanja kakovosti	Razmerje med obsegom dela aktivnosti zagotavljanja kakovosti in obsegom dela aktivnosti razvoja	deljenje	Obseg dela aktivnosti zagotavljanja kakovosti Obseg dela aktivnosti razvoja	Aktivnosti Vidiki	Odstotki
Strošek odpravljanja napak	Razmerje med obsegom dela odpravljanja napak in obsegom dela aktivnosti razvoja	deljenje	Obseg dela odpravljanja napak Obseg dela aktivnosti razvoja	Aktivnosti Vidiki	Odstotki
Razmerje stroškov kakovosti	Razmerje med obsegom dela odpravljanja napak in obsegom dela aktivnosti zagotavljanja kakovostjo	deljenje	Obseg dela odpravljanja napak Obseg dela aktivnosti zagotavljanja kakovosti	Aktivnosti Vidiki	Odstotki

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.5 Nabor pokazateljev

V nadaljevanju je prikazan nabor pokazateljev po vrstah analiz, ki jim pripadajo. Prikazan je samo stalen del nabora pokazateljev, ki je določen na podlagi stalnih merskih izhodišč. Naš merski pristop je zastavljen tako, da je glavnina pokazateljev stalnega nabora iz analiz ocenjevanja in napovedovanja, ker predvidevamo, da je ta del avtomatiziran in enostavnejši za uporabo. V okviru posebnih merskih izhodišč lahko posamezni projekti nadgradijo nabor z novimi pokazatelji, ki pa bodo predvsem vezani na analize izvedljivosti in performans.

B.2.5.1 Ocenjevanje

Atribut (razred)	Opis pokazatelja (ocenjene vrednosti)	Dodatni postopki
Velikost končnega izdelka (velikost izdelkov)	Velikost procesnega dela končnega izdelka (z dokumentacijo)	Območje 90% zaupanja z linearno regresijo, popravljanje glede na dejanske rezultate
	Velikost podatkovnega dela končnega izdelka (z dokumentacijo)	Območje 90% zaupanja z linearno regresijo, popravljanje glede na dejanske rezultate
Obseg dela (viri in stroški)	Obseg dela celotnega projekta – procesni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Obseg dela celotnega projekta – podatkovni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Obseg dela posameznih ciklov – procesni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Obseg dela posameznih ciklov – podatkovni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Obseg dela posameznih aktivnosti (faz in korakov)	Na podlagi ocen razvijalcev
Poraba časa (poraba časa in napredovanje)	Poraba časa celotnega projekta – procesni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Poraba časa celotnega projekta – podatkovni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Poraba časa posameznih ciklov in faz – procesni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Poraba časa posameznih ciklov in faz – podatkovni del	Območje 90% zaupanja z linearno regresijo, usklajevanje glede na ocene razvijalcev
	Poraba časa posameznih faz in korakov	Na podlagi ocen razvijalcev
Stroški (viri in stroški)	Stroški celotnega projekta (iz obsega dela in dodatni)	Območje 90% zaupanja z linearno regresijo, popravljanje glede na dejanske rezultate
	Stroški posameznih ciklov in faz (iz obsega dela in dodatni)	Območje 90% zaupanja z linearno regresijo, popravljanje glede na dejanske rezultate
Kakovost izdelkov (kakovost izdelkov)	Število vnešenih napak v procesni del končnega izdelka	Območje 90% zaupanja z linearno regresijo, popravljanje glede na nat. planiranje kakovosti
	Število odpravljenih napak v procesni del končnega izdelka	Območje 90% zaupanja z linearno regresijo, popravljanje glede na nat. planiranje kakovosti
	Število vnešenih napak v podatkovni del končnega izdelka	Območje 90% zaupanja z linearno regresijo, popravljanje glede na nat. planiranje kakovosti
	Število odpravljenih napak v podatkovni del končnega izdelka	Območje 90% zaupanja z linearno regresijo, popravljanje glede na nat. planiranje kakovosti
	Število vnešenih napak s podatkovnega in procesnega vidika	Na podlagi ocen kakovosti aktivnosti
	Število odpravljenih napak s podatkovnega in procesnega vidika	Na podlagi ocen kakovosti aktivnosti
	Število vnešenih napak po fazah tekočega cikla	Na podlagi ocen kakovosti aktivnosti
	Število odpravljenih napak po fazah tekočega cikla	Na podlagi ocen kakovosti aktivnosti
	Število vnešenih napak po rezultatih tekočega cikla	Na podlagi ocen kakovosti aktivnosti
	Število odpravljenih napak po rezultatih cikla	Na podlagi ocen kakovosti aktivnosti
	Število vnešenih napak po izvajalcih	Na podlagi ocen kakovosti aktivnosti
	Število odpravljenih napak po izvajalcih	Na podlagi ocen kakovosti aktivnosti

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.5.2 Analiza izvedljivosti

V okviru stalnega nabora pokazateljev ni pokazateljev izvedljivosti, ker jih z vidika stalnih merskih izhodišč nadomeščajo pokazatelji napovedovanja.

B.2.5.3 Analiza performans

Tudi analiza performans je v našem merskem pristopu pretežno nadomeščena z napovedovanjem in se uporablja predvsem tedaj, ko napovedi ne dajejo jasne slike. V ta namen predvidevamo pokazatelje statusa aktivnosti, izdelkov in virov. Ti omogočajo podroben vpogled v vse informacije o merskih entitetah na različnih ravneh. Na ta način neposredno preverimo tudi rezultate osnovnih in združenih meritev zanje in za njihove sestavne dele.

Dodatni trije pokazatelji so določeni za spremljanje produktivnosti in kakovosti procesa razvoja, ki izhajajo iz splošnih merskih izhodišč. Z uporabo posebnih merskih izhodišč lahko po potrebi določimo tudi še veliko drugih.

Atribut (razred merjenja)	Opis pokazatelja (vsebina)
Status merskih aktivnosti (ciklov, faz, korakov in podkorakov)	Vse informacije o izbrani aktivnosti in njenih sestavnih delih (v tekstovni ali grafični obliki)
Status izdelkov (rezultatov izvedbe faz in ciklov)	Vse informacije o izbranem izdelku in njegovih sestavnih delih (v tekstovni ali grafični obliki)
Status virov (skupin, osebja – izvajalcev)	Vse informacije o izbranem viru (v tekstovni ali grafični obliki)
Produktivnost po aktivnostih	Produktivnost po aktivnostih na različnih ravneh in vidikih: • za same aktivnosti (cikli, faze, koraki), • za aktivnosti izvajalcev, • za aktivnosti razvoja rezultatov in drugo.
Izkoristek odpravljanja napak	Izkoristek odpravljanja napak na različnih ravneh in vidikih: • za same aktivnosti (cikli, faze, koraki), • za aktivnosti izvajalcev, • za aktivnosti razvoja rezultatov in drugo.
Strošek kakovosti	Kombiniran pokazatelj treh pogledov za aktivnosti na različnih ravneh (cikli, faze, koraki)

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.5.4 Napovedovanje

Napovedi so ena pomembnejših značilnosti našega merskega pristopa. Z njimi podpiramo glavnino potreb po merjenju v majhni organizaciji.

Atribut (razred merjenja)	Opis pokazatelja (vsebina)
Velikost končnega izdelka (Velikost izdelkov)	- zadnja ocena velikosti procesnega dela končnega izdelka z območjem zaupanja - napoved velikosti procesnega dela končnega izdelka z območjem zaupanja - zadnja ocena podatkovnega dela končnega izdelka z območjem zaupanja - napoved velikosti podatkovnega dela končnega izdelka z območjem zaupanja
Velikost izdelkov ciklov in faz - projekcija (Velikost izdelkov)	Za vse izbrane atribute glavnih izdelkov posameznih ciklov: - napoved velikosti atributa izdelka z območjem zaupanja
Obseg dela celotnega projekta (Viri in stroški)	- zadnja ocena obsega dela za izvedbo procesnega dela z območjem zaupanja - napoved obsega dela za izvedbo procesnega dela z območjem zaupanja - zadnja ocena obsega dela za izvedbo podatkovnega dela z območjem zaupanja - napoved obsega dela za izvedbo podatkovnega dela z območjem zaupanja
Obseg dela posameznih ciklov in faz - projekcija (Viri in stroški)	Za vsak cikel: - zadnja ocena obsega dela za izvedbo procesnega dela cikla z območjem zaupanja - napoved obsega dela za izvedbo procesnega dela cikla z območjem zaupanja - zadnja ocena obsega dela za izvedbo podatkovnega dela cikla z območjem zaupanja - napoved obsega dela za izvedbo podatkovnega dela cikla z območjem zaupanja Za vsako fazo, kjer je to mogoče: - zadnja ocena obsega dela za izvedbo procesnega dela faze z območjem zaupanja - napoved obsega dela za izvedbo procesnega dela faze z območjem zaupanja - zadnja ocena obsega dela za izvedbo podatkovnega dela faze z območjem zaupanja - napoved obsega dela za izvedbo podatkovnega dela faze z območjem zaupanja
Dolžina trajanja in datum zaključka projekta (Poraba časa in napredovanje)	- zadnja ocena dolžine trajanja projekta z območjem zaupanja - napoved dolžine trajanja projekta z območjem zaupanja - zadnja ocena datuma zaključka projekta z območjem zaupanja - napoved datuma zaključka projekta z območjem zaupanja
Dolžina trajanja in mejniki posameznih ciklov in faz (Poraba časa in napredovanje)	Za vsak cikel: - zadnja ocena dolžine trajanja z območjem zaupanja - napoved dolžine trajanja z območjem zaupanja Za vsako fazo, kjer je to mogoče: - zadnja ocena dolžine trajanja z območjem zaupanja - napoved dolžine trajanja z območjem zaupanja Za vsak znan mejnik (ujema se z zaključki faz): - zadnja ocena mejnika z območjem zaupanja - napoved mejnika z območjem zaupanja
Stroški projekta (Viri in stroški)	- zadnja ocena celotnih stroškov projekta z območjem zaupanja - napoved celotnih stroškov projekta z območjem zaupanja
Stroški posameznih faz in ciklov - projekcija (Viri in stroški)	Za vsak cikel: - zadnja ocena stroškov cikla z območjem zaupanja - napoved stroškov cikla z območjem zaupanja Za vsako fazo, kjer je to mogoče: - zadnja ocena stroškov faze z območjem zaupanja - napoved stroškov faze z območjem zaupanja
Kakovost končnega izdelka (Kakovost izdelkov)	- zadnja ocena skupnega števila vnešenih napak pri izvedbi procesnega dela z območjem zaupanja - napoved skupnega števila vnešenih napak pri izvedbi procesnega dela z območjem zaupanja - zadnja ocena skupnega števila odpravljenih napak procesnega dela z območjem zaupanja - napoved skupnega števila odpravljenih napak pri izvedbi procesnega dela z območjem zaupanja - zadnja ocena skupnega števila vnešenih napak podatkovnega dela z območjem zaupanja - napoved skupnega števila vnešenih napak pri izvedbi podatkovnega dela z območjem zaupanja - zadnja ocena skupnega števila odpravljenih napak podatkovnega dela z območjem zaupanja - napoved skupnega števila odpravljenih napak podatkovnega dela z območjem zaupanja
Kakovost z vidika procesnega in podatkovnega dela (Kakovost izdelkov)	Podobno kot pri kakovosti končnega izdelka, le napovedano na drugačni podlagi: - zadnja ocena skupnega števila vnešenih napak pri izvedbi procesnega dela projekta - napoved skupnega števila vnešenih napak pri izvedbi procesnega dela projekta - zadnja ocena skupnega števila odpravljenih napak pri izvedbi procesnega dela projekta - napoved skupnega števila odpravljenih napak pri izvedbi procesnega dela projekta - zadnja ocena skupnega števila vnešenih napak pri izvedbi podatkovnega dela projekta - napoved skupnega števila vnešenih napak pri izvedbi podatkovnega dela projekta - zadnja ocena skupnega števila odpravljenih napak pri izvedbi podatkovnega dela projekta - napoved skupnega števila odpravljenih napak pri izvedbi podatkovnega dela projekta

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Atribut (razred merjenja)	Opis pokazatelja (vsebina)
Kakovost z vidika posameznih faz in ciklov (Kakovost izdelkov)	Za vsakega izmed ciklov: - zadnja ocena števila vnešenih napak cikla - napoved števila vnešenih napak cikla upanja - zadnja ocena števila odpravljenih napak cikla - napoved števila odpravljenih napak cikla Za vsako fazo, kjer je to mogoče: - zadnja ocena števila vnešenih napak faze - napoved števila vnešenih napak faze - zadnja ocena števila odpravljenih napak faze - napoved števila odpravljenih napak faze
Kakovost z vidika posameznih rezultatov (Kakovost izdelkov)	Za vsak rezultat (izdelek) faze ali cikla, kjer je to mogoče: - zadnja ocena števila vnešenih napak pri izdelavi izdelka - napoved števila vnešenih napak pri izdelavi izdelka - zadnja ocena števila odpravljenih napak pri izdelavi izdelka - napoved števila odpravljenih napak pri izdelavi izdelka
Kakovost z vidika izvajalcev (Kakovost izdelkov)	Za vsakega razvijalca oziroma člana projektne skupine: - zadnja ocena skupnega števila vnešenih napak - napoved skupnega števila vnešenih napak - zadnja skupnega ocena števila odpravljenih napak - napoved skupnega števila odpravljenih napak

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.6 Povezava med stalnimi merskimi izhodišči in pokazatelji

V zadnjem delu predstavitve vsebine merskega pristopa opisujemo povezavo med izbranim naborom merskih izhodišč ter pokazatelji.

B.2.6.1 Stalna merska izhodišča in pokazatelji

Spodnja tabela prikazuje, kateri pokazatelji ustrezajo posameznim stalnim merskim izhodiščem. Pri tem neposredno navajamo vsa tista izhodišča, ki izhajajo iz zahtev modela CMM. Splošni cilji so upoštevani posredno (glej dodatek B.2.3).

Oznaka	Stalno mersko izhodišče	Pokazatelj
2.1	**Upravljanje z zahtevami**	
2.1.1	Spremljaj status zahtev	Status izdelka (seznama zahtev)
2.1.2	Spremljaj spremembe zahtev	Status izdelka (seznama zahtev) Kakovost z vidika posameznih faz in ciklov
2.1.3	Spremljaj status aktivnosti upravljanja z zahtevami	Status aktivnosti
2.2	**Planiranje projektov**	
2.2.1	Spremljaj razpored planirane izvedbe opravil projekta (ocena)	Dolžina trajanja in mejniki ciklov in faz
2.2.2	Spremljaj potreben obseg dela za izvedbo opravil	Obseg dela posameznih ciklov in faz – projekcija
2.2.3	Spremljaj status aktivnosti planiranja	Status aktivnosti
2.3	**Zasledovanje in nadzor nad projekti**	
2.3.1	Spremljaj velikost končnih izdelkov	Velikost končnega izdelka
2.3.2	Spremljaj obseg dela projekta	Obseg dela celotnega projekta
2.3.3	Spremljaj potek projekta	Dolžina trajanja in datum zaključka projekta
2.3.4	Spremljaj težave med izvajanjem projekta	Kakovost končnega izdelka Kakovost s proc. in pod. vidika Kakovost z vidika posameznih faz in ciklov
2.3.5	Spremljaj status aktivnosti zasledovanja in nadzora nad projekti	Status aktivnosti
2.4	**Urejanje razmerij s poddobavitelji**	
2.5	**Zagotavljanje kakovosti PO**	
2.5.1	Spremljaj preverjanja in preglede procesa in izdelkov	Status aktivnosti Stroški kakovosti
2.5.2	Spremljaj težave na področju zagotavljanja kakovosti	Kakovost končnega izdelka Kakovost s proc. in pod. vidika Kakovost z vidika posameznih faz in ciklov
2.5.3	Spremljaj status aktivnosti za zagotavljanje kakovosti (SQA aktivnosti)	Status aktivnosti
2.6	**Upravljanje s konfiguracijo**	
2.6.1	Spremljaj status konfiguracij	Status izdelka
2.6.2	Spremljaj spremembe konfiguracij	Status izdelka
2.6.3	Spremljaj odstopanja v konfiguraciji	Status izdelka Kakovost z vidika posameznih faz in ciklov
2.6.4	Spremljaj status aktivnosti za upravljanje s konfiguracijo	Status aktivnosti
3.1	**Osredotočenje na proces, ki ga uporablja organizacija**	
3.1.1	Spremljaj in napoveduj razpored izvajanja aktivnosti za koordinacijo in izboljšanje procesa razvoja PO	Status aktivnosti Dolžina trajanja in mejniki ciklov in faz
3.1.2	Spremljaj in napoveduj stroške izvedbe aktivnosti	Stroški projekta Stroški posameznih faz in ciklov – projekcija
3.1.3	Spremljaj in napoveduj rezultate preverjanj aktivnosti za koordinacijo in izboljšanje procesa razvoja PO	Status izdelka Status aktivnosti
3.1.4	Spremljaj in napoveduj status aktivnosti za koordinacijo in izboljšanje procesa razvoja PO	Status aktivnosti
3.2	**Definicija procesa razvoja PO organizacije**	
3.2.1	Spremljaj in napoveduj razpored izvajanja aktivnosti za razvoj in vzdrževanje standardnega procesa razvoja PO	Status aktivnosti Dolžina trajanja in mejniki ciklov in faz
3.2.2	Spremljaj in napoveduj obseg dela in stroške izvedbe aktivisti za razvoj in vzdrževanje standardnega procesa razvoja PO	Status aktivnosti Obseg dela posameznih ciklov in faz
3.2.3	Spremljaj in napoveduj status aktivnosti za razvoj in vzdrževanje standardnega procesa razvoja PO	Status aktivnosti
3.3.	**Program izobraževanja**	
3.3.1	Spremljaj in napoveduj udeležbo pri izobraževanju	Status virov
3.3.2	Spremljaj in napoveduj rezultate anket udeležencev	Status izdelka
3.3.3	Spremljaj in napoveduj status aktivnosti izobraževanja	Status aktivnosti

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Oznaka	Stalno mersko izhodišče (nadaljevanje)	Pokazatelj
3.4	**Integrirano vodenje projektov razvoja PO**	
3.4.1	Spremljaj in napoveduj velikost posameznih izdelkov	Velikost izdelkov ciklov in faz - projekcija Status izdelka
3.4.2	Spremljaj in napoveduj potreben obseg dela za posamezne aktivnosti	Obseg dela posameznih ciklov in faz – projekcija Status aktivnosti
3.4.3	Spremljaj in napoveduj vire pri izvajanju projektov	Status virov Status aktivnosti
3.4.4	Spremljaj in napoveduj stroške pri izvajanju projektov	Stroški posameznih faz in ciklov – projekcija
3.4.5	Spremljaj in napoveduj časovni razpored izvajanja projektov	Dolžina trajanja in mejniki ciklov in faz
3.4.6	Spremljaj in napoveduj napake pri izvajanju projektov	Kakovost s proc. in pod. vidika Kakovost z vidika posameznih faz in ciklov
3.4.7	Spremljaj in napoveduj status aktivnosti integriranega vodenje projektov	Status aktivnosti
3.5	**Tehnologija izdelkov PO**	
3.5.1	Spremljaj in napoveduj funkcionalnost izdelkov	Status izdelkov
3.5.2	Spremljaj in napoveduj kakovost izdelkov	Kakovost z vidika posameznih rezultatov
3.5.3	Spremljaj in napoveduj status aktivnosti tehnologije izdelkov	Status aktivnosti
3.6	**Koordinacija med različnimi skupinami**	
3.6.1	Spremljaj in napoveduj stroške sodelovanja z zunanjimi skupinami	Status virov Stroški posameznih faz in ciklov – projekcija
3.6.2	Spremljaj in napoveduj status aktivnosti sodelovanja z zunanjimi skupinami	Status aktivnosti
3.7	**Pregledi med kolegi**	
3.7.1	Spremljaj in napoveduj status aktivnosti preverjanja	Status aktivnosti Kakovost z vidika izvajalcev
4.1	**Kvantitativno vodenje procesov**	
4.1.1	(Recimo) Spremljaj in nadzoruj produktivnost	Produktivnost po aktivnostih
4.1.2	(Recimo) Spremljaj in nadzoruj ponovna uporabljivost	Status izdelkov
4.1.3	(Recimo) Spremljaj in nadzoruj učinkovitost izvedbe	Status aktivnosti Produktivnost po aktivnostih
4.1.4	Spremljaj in nadzoruj status aktivnosti kvantitativnega vodenja procesov	Status aktivnosti
4.2	**Upravljanje za kakovost**	
4.2.1	Spremljaj in nadzoruj kakovost s pomočjo razporeditve napak	Izkoristek odpravljanja napak Stroški kakovosti
4.2.2	Spremljaj in nadzoruj status aktivnosti upravljanja kakovosti	Status aktivnosti

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

B.2.6.2 Pokazatelji in stalna merska izhodišča

V drugem delu predstavljamo vse pokazatelje, razrede meritev in oznake stalnih merskih izhodišč, ki jih z njihovo pomočjo lahko preverjamo. Oznake so povzete po dodatkih B.2.3.2 do B.2.3.4, razvidne pa so tudi iz dodatka B.2.6.1. Prva številka predstavlja stopnjo CMM, druga ključno področje in tretja zaporedno številko merskega izhodišča.

Pokazatelj	Razred meritve	Oznaka stalnega merskega izhodišča
Ocena		
Velikost končnega izdelka	Velikost izdelkov	2.2.1 (2.3.1, 3.4.1)
Obseg dela celotnega projekta, ciklov in faz, korakov	Viri in stroški	2.2.2 (2.3.2, 3.2.2, 3.4.2)
Poraba časa in napredovanje projekta, ciklov, faz	Poraba časa in napredovanje	2.2.1 (2.3.3, 3.2.2, 3.4.2)
Stroški projekta, ciklov in faz	Viri in stroški	3.1.2 (3.4.4)
Kakovost končnega izdelka z različnih vidikov	Kakovost izdelkov	2.1.2 (2.3.4, 2.5.2, 2.6.3, 3.4.6, 3.5.2, 3.7.1)
Analiza performans		
Status aktivnosti	Vsi razredi	2.1.3, 2.2.3. 2.3.5, 2.5.1, 2.5.3, 2.6.4, 3.1.1, 3.1.3, 3.1.4, 3.2.1, 3.2.2., 3.2.3, 3.3.3, 3.4.2, 3.4.3, 3.4.7, 3.5.3, 3.6.2, 3.7.1, 4.1.3, 4.1.4, 4.2.2
Status izdelka	Vsi razredi	2.1.1, 2.1.2, 2.6.1, 2.6.2, 2.6.3, 3.1.3, 3.3.2, 3.4.1, 3.5.1, 4.1.2
Status vira	Viri in stroški	3.3.1, 3.4.3, 3.6.11
Produktivnost po aktivnostih	Viri in stroški	4.4.1, 4.4.3
Izkoristek odpravljanja napak	Kakovost izdelkov	4.2.1
Strošek kakovosti	Kakovost izdelkov	2.5.1, 4.2.1
Napoved		
Velikost končnega izdelka	Velikost izdelkov	2.3.1
Velikost izdelkov ciklov in faz – projekcija	Velikost izdelkov	3.4.1
Obseg dela celotnega projekta	Viri in stroški	2.3.2
Obseg dela posameznih ciklov in faz – projekcija	Viri in stroški	2.2.2, 3.2.2, 3.4.2
Dolžina trajanja in datum zaključka projekta	Poraba časa in napredovanje	2.3.3
Dolžina trajanja in mejniki ciklov in faz	Poraba časa in napredovanje	2.2.1, 3.1.1, 3.2.1, 3.4.5
Stroški projekta	Viri in stroški	3.1.2
Stroški posameznih faz in ciklov – projekcija	Viri in stroški	3.1.2, 3.4.4
Kakovost končnega izdelka	Kakovost izdelkov	2.3.4, 2.5.2
Kakovost z vidika procesnega in podatkovnega dela	Kakovost izdelkov	2.3.4, 2.5.2, 3.4.6
Kakovost z vidika posameznih faz in ciklov	Kakovost izdelkov	2.1.2, 2.3.4, 2.5.2, 2.6.3, 3.4.6
Kakovost z vidika posameznih rezultatov	Kakovost izdelkov	3.5.2
Kakovost z vidika izvajalcev	Kakovost izdelkov	3.7.1

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

PRILOGA C: VPRAŠALNIK

Merski model smo praktično preverili v več manjših organizacijah za razvoj PO. Pri tem so sogovorniki po okvirni predstavitvi le-tega izpolnili vprašalnik, ki je predstavljen v nadaljevanju. Vprašalnik omogoča enostavno in hitro izpolnjevanje. Z več navzkrižnimi vprašanji preverja vse pomembnejše zamisli, ki so vgrajene model za kvantitativno spremljanje razvoja PO oziroma v vse tri njegove sestavne dele: prilagodljiv proces razvoja PO, merski proces in vsebino merjenja.

Spoštovani!

V svoji doktorski disertaciji bom definiral model za kvantitativno spremljanje razvoja programske opreme v majhni organizaciji za razvoj informacijskih sistemov. V okviru tega bom opisal tako postopke kot vsebino merjenja razvoja programske opreme, oboje pa bom tudi umestil v ustrezen proces razvoja programske opreme.

Pred Vami je krajši vprašalnik, s katerim bi rad preveril nekatere teze iz vseh treh delov modela. Prosil bi Vas, da si vzamete nekaj minut časa, ga skrbno preberete ter mi iskreno odgovorite na zastavljena vprašanja. Za pomoč se Vam iskreno zahvaljujem.

Vprašanja so zastavljena tako, da omogočajo kar najlažje izpolnjevanje. Odgovori na večino vprašanj so zato kar ocene od 1 do 5, kjer so pomeni posameznih ocen naslednji:

- 1 - najslabše, popolnoma nepotrebno, odločno nasprotujete
- 2 - slabo, nepotrebno, v glavnem se ne strinjate
- 3 - povprečno, vseeno, deloma se strinjate, deloma ne
- 4 - zelo dobro, koristno, v glavnem se strinjate
- 5 - najboljše, nujno, popolnoma se strinjate

Pri ostalih vprašanjih je treba odgovore napisati na črto. V okviru vsakega vprašanja je tudi možnost, da odgovore dopolnite še s svojimi predlogi.

1. **Priimek in ime**: _____
2. **Podjetje**: _____
3. **Število zaposlenih v podjetju**: _____
4. **Število zaposlenih, ki se neposredno ukvarjajo z razvojem PO:** _____
5. **Koliko let se strokovno uvarjate z razvojem PO**: _____
6. **Poznate model stopenj zrelosti (Capability Maturity Model-CMM)?** 1 2 3 4 5
7. **Imate z modelom tudi CMM praktične izkušnje?** 1 2 3 4 5
8. **Poznate ali imate izkušnje z drugimi pristopi za zagotavljanje kakovosti PO? Katerimi?**

_____

9. **Če poznate model CMM, kako bi ga ovrednotili <u>v primerjavi</u> z drugimi pristopi:**
   - **Po popolnosti** 1 2 3 4 5
   - **Po zahtevnosti uporabe** 1 2 3 4 5
   - **Po obsežnosti** 1 2 3 4 5
   - **Po natančnosti definicije** 1 2 3 4 5
   - **Po uporabnosti za majhno organizacijo** 1 2 3 4 5

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

10. Vaše mnenje o uporabi modela CMM v majhni organizaciji (vašem podjetju)?

_____

11. Kako bi ovrednotili pomen spodnjih značilnosti procesa razvoja PO v majhni organizaciji? Če sodite, da pomembne značilnosti manjkajo, jih dopišite!

• Usklajenost s preverjenim modelom	1	2	3	4	5
• Natančna in ustrezna definiranost	1	2	3	4	5
• Majhna in enostavna definicija procesa	1	2	3	4	5
• Enostavnost uporabe procesa v praksi	1	2	3	4	5
• Obvladovanje sprememb zahtev	1	2	3	4	5
• Prilagojenost zmožnostim organizacije	1	2	3	4	5
• Prilagojenost specifični vrsti PO	1	2	3	4	5
• Podpora z ustreznimi orodji	1	2	3	4	5
• _____	1	2	3	4	5
• _____	1	2	3	4	5

12. Kako bi ovrednotili pomen spodnjih trditev, ki so povezane z razvojem PO v majhni organizaciji?

• Obvezno natanko določimo vse skupine/vloge	1	2	3	4	5
• Vsi projekti uporabljajo enoten org. proces	1	2	3	4	5
• Člani so specializirani samo za določene vloge	1	2	3	4	5
• Vse aktivnosti so natanko planirane	1	2	3	4	5
• Za vsak postopek je predpisan formalen rezultat	1	2	3	4	5
• Obseg dokumentacije je omejen	1	2	3	4	5
• Postopki so podprti z ustreznimi orodji	1	2	3	4	5
• Avtomatizacija poveča učinkovitost postopkov	1	2	3	4	5
• Formalno preverjamo kakovost na več ravneh	1	2	3	4	5
• Kakovost spremljamo s pomočjo odkritih napak	1	2	3	4	5
• Sprotno se prilagajamo spremembam zahtev	1	2	3	4	5
• Smiselno je, da se plani občasno preverjajo in prilagajajo	1	2	3	4	5
• Usposabljanje je formalizirano na ravni org.	1	2	3	4	5
• Vse podporne aktivnosti so enostavne in učinkovite	1	2	3	4	5
• Dodatne pomembne značilnosti razvoja PO, ki bi jih izpostavili:					

_____

13. Na področju merjenja razvoja PO je znanih več standardnih pristopov in metod. Ali v okviru vašega podjetja uporabljate principe katero izmed spodaj naštetih?

• metode GQM (Goal-Question-Metric)	1	2	3	4	5
• metode MQG (Metric-Question-Goal)	1	2	3	4	5
• standarda ISO/IEC 15939 – Merski proces za PO	1	2	3	4	5
• pristopa Praktično merjenje PO (Practical Software Measurement)	1	2	3	4	5
• drugih znanih pristopov: _____					

14. Kako bi ovrednotili pomen spodnjih trditev o merjenju razvoja PO v majhni organizaciji?

• Merjenje je definirano s preverjenim modelom	1	2	3	4	5
• Način merjenja je enoten v organizaciji, projekti ga povzemajo	1	2	3	4	5
• Projekti imajo poleg splošnih tudi specifične merske zahteve	1	2	3	4	5
• Merjenje je natanko planirano in je del procesa razvoja PO	1	2	3	4	5
• Merjenje izvajajo čisto vsi razvijalci	1	2	3	4	5
• Vsi člani morajo biti usposobljeni za merjenje	1	2	3	4	5
• Za obdelave merskih podatkov skrbi posebej usposobljen član	1	2	3	4	5
• Merjenje je vgrajeno v vsako aktivnost procesa	1	2	3	4	5
• Nabor meritev je natanko določen	1	2	3	4	5
• Zadostuje omejen nabor ustreznih meritev	1	2	3	4	5

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Za večje aktivnosti (faze, projekt) lahko merske podatke določimo s sestavljanjem rezultatov manjših aktivnosti (korakov) 1 2 3 4 5
- Merjenje je avtomatizirano (zbiranje, hramba) 1 2 3 4 5
- Formalno preverjamo kakovost mer. podatkov 1 2 3 4 5
- Merski podatki se zbirajo/obdelujejo centralno 1 2 3 4 5
- Podatke preteklih projektov hranimo in rabimo 1 2 3 4 5
- Analize temeljijo na podatkih pret. projektov 1 2 3 4 5
- Analize se izvajajo učinkovito in enostavno 1 2 3 4 5
- Izvedba analiz je čimbolj avtomatizirana 1 2 3 4 5
- **Dodatne pomembne značilnosti:** _____

15. Naš pristop predvideva merjenje tako na ravni organizacije kot na ravni posameznih projektov. Kaj menite o <u>potrebnosti</u> merjenja razvoja PO na različnih ravneh?

- Na ravni organizacije je potrebno (zaželjeno) meriti:
  - vsak projekt 1 2 3 4 5
  - vse druge aktivnosti organizacije 1 2 3 4 5
  - pretekle, že končane projekte 1 2 3 4 5
- Na ravni projekta je pomembno meriti:
  1. značilnosti celotnega projekta:
     - Aktivnosti (celoten projekt) 1 2 3 4 5
     - Izdelke (končne rezultate) 1 2 3 4 5
     - Vire (proj. skupino, skup. stroške) 1 2 3 4 5
  2. značilnosti sestavnih delov projekta (posameznih faz, korakov):
     - Aktivnosti (cikel, faza, korak) 1 2 3 4 5
     - Izdelke (delni rezultati, moduli) 1 2 3 4 5
     - Vire (razvijalce, stroške) 1 2 3 4 5

16. Kaj menite o <u>izvedljivosti (smiselnosti)</u> različnih ravni merjenja v majhni organizaciji?

- Na ravni organizacije je mogoče (smiselno) meriti:
  - vsak projekt 1 2 3 4 5
  - vse druge aktivnosti organizacije 1 2 3 4 5
  - pretekle, že končane projekte 1 2 3 4 5
- Na ravni projekta lahko merimo:
  1. značilnosti celotnega projekta:
     - Aktivnosti (celoten projekt) 1 2 3 4 5
     - Izdelke (končne rezultate) 1 2 3 4 5
     - Vire (proj.skupino, skup.stroške) 1 2 3 4 5
  2. značilnosti sestavnih delov projekta (posameznih faz, korakov):
     - Aktivnosti (cikel, faza, korak) 1 2 3 4 5
     - Izdelke (delni rezultati, moduli) 1 2 3 4 5
     - Vire (razvijalce, stroške) 1 2 3 4 5

17. Mersko spremljamo postopke (aktivnosti), izdelke in vire. Kaj menite o <u>pomembnosti (potrebnosti)</u> merjenja posameznih značilnosti teh v majhni organizaciji za razvoj PO?

- **Aktivnosti:**
  - Trajanje aktivnosti 1 2 3 4 5
  - Doseganje mejnikov 1 2 3 4 5
  - Obseg dela pri izvedbi aktivnosti 1 2 3 4 5
  - Produktivnost pri izvedbi aktivnosti 1 2 3 4 5
  - Stroški izvedbe aktivnosti 1 2 3 4 5
  - Kakovost izvedbe aktivnosti:
    - Vnešene napake 1 2 3 4 5
    - Odkrite napake 1 2 3 4 5
    - Obseg dela pri odkrivanju napak 1 2 3 4 5
- **Izdelki:**
  - Velikost izdelka:
    - Prog. koda (št. vrstic) 1 2 3 4 5
    - Dokumenti (št. strani) 1 2 3 4 5
    - Diagrami, matrike (št. elementov) 1 2 3 4 5

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

- Funkcionalnost izdelka 1 2 3 4 5
- Strukturiranost izdelka 1 2 3 4 5
- Zahtevnost izdelka 1 2 3 4 5
- Obseg dela pri izdelavi izdelka 1 2 3 4 5
- Produktivnost pri izdelavi izdelka 1 2 3 4 5
- Stroški izdelave izdelka 1 2 3 4 5
- Kakovost izdelka:
  - Vnešene napake 1 2 3 4 5
  - Odkrite napake 1 2 3 4 5
  - Obseg dela pri odkrivanju napak 1 2 3 4 5
  - Uporabnost 1 2 3 4 5
  - Vzdrževalnost 1 2 3 4 5
  - Zanesljivost 1 2 3 4 5

- **Viri:**
  - Udeležba (velikost) 1 2 3 4 5
  - Kakovost virov (recimo izobrazba) 1 2 3 4 5
  - Obseg dela virov (članov, skupin) 1 2 3 4 5
  - Produktivnost posameznih virov 1 2 3 4 5
  - Stroški članov in skupin 1 2 3 4 5
- **Dodatne značilnosti, ki niso naštete:** _____

18. Merjenje katerih izmed naštetih značilnosti je po vašem mnenju v majhni organizaciji <u>smiselno oziroma izvedljivo</u>?

- **Aktivnosti:**
  - Trajanje aktivnosti 1 2 3 4 5
  - Doseganje mejnikov 1 2 3 4 5
  - Obseg dela pri izvedbi aktivnosti 1 2 3 4 5
  - Produktivnost pri izvedbi aktivnosti 1 2 3 4 5
  - Stroški izvedbe aktivnosti 1 2 3 4 5
  - Kakovost izvedbe aktivnosti:
    - Vnešene napake 1 2 3 4 5
    - Odkrite napake 1 2 3 4 5
    - Obseg dela pri odkrivanju napak 1 2 3 4 5
- **Izdelki:**
  - Velikost izdelka:
    - Prog. koda (št. vrstic) 1 2 3 4 5
    - Dokumenti (št. strani) 1 2 3 4 5
    - Diagrami, matrike (št. elementov) 1 2 3 4 5
  - Funkcionalnost izdelka 1 2 3 4 5
  - Strukturiranost izdelka 1 2 3 4 5
  - Zahtevnost izdelka 1 2 3 4 5
  - Obseg dela pri izdelavi izdelka 1 2 3 4 5
  - Produktivnost pri izdelavi izdelka 1 2 3 4 5
  - Stroški izdelave izdelka 1 2 3 4 5
  - Kakovost izdelka:
    - Vnešene napake 1 2 3 4 5
    - Odkrite napake 1 2 3 4 5
    - Obseg dela pri odkrivanju napak 1 2 3 4 5
    - Uporabnost 1 2 3 4 5
    - Vzdrževalnost 1 2 3 4 5
    - Zanesljivost 1 2 3 4 5
- **Viri:**
  - Udeležba (velikost) 1 2 3 4 5
  - Kakovost virov (recimo izobrazba) 1 2 3 4 5
  - Obseg dela virov (članov, skupin) 1 2 3 4 5
  - Produktivnost posameznih virov 1 2 3 4 5
  - Stroški članov in skupin 1 2 3 4 5

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

19. Katere značilnosti (izmed teh in druge) merite (ali nameravate meriti) v vaši organizaciji?

Aktivnosti: _____
Izdelki: _____
Viri: _____

20. Z merjenjem teh značilnosti zbiramo merske podatke in na podlagi teh izvajamo različne analize. Kaj menite o <u>potrebnosti posameznih vrst analiz</u> v majhni organizaciji?

- **Ocenjevanje:** pri planiranju iz podatkov preteklih projektov in zastavljenih ciljev z uporabo statističnih metod ocenimo izbran nabor vrednosti 1 2 3 4 5
- **Analizo izvedljivosti:** po planiranju na podlagi podatkov preteklih projektov preverimo izvedljivost zastavljenih planov 1 2 3 4 5
- **Analizo performans**: med izvajanjem projekta zbiramo merske podatke in jih primerjamo z ocenami iz planov 1 2 3 4 5
- **Napovedovanje:** med izvajanjem projekta iz podatkov preteklih projektov, planov in zbranih merskih podatkov projeciramo izbran nabor značilnosti merskih entitet od trenutnega stanja do konca projekta 1 2 3 4 5

21. Pokazatelji so rezultati analiz. Gre za nazoren (grafični) prikaz kombinacije več merskih podatkov, ocen in podatkov preteklih projektov z izbranega vidika, ki neposredno odgovarja na zastavljene merske cilje organizacije. Kateri pokazatelji so po Vašem primerni za majhno organizacijo, da celovito in jedrnato predstavijo vse potrebne informacije?

- Trenutni status vseh merskih entitet : ocene iz planov 1 2 3 4 5
- Velikost izdelkov : ocene velikosti iz planov 1 2 3 4 5
- Funkcionalnost izdelkov : ocene iz planov 1 2 3 4 5
- Porabo časa : ocene porabe časa iz planov 1 2 3 4 5
- Doseganje mejnikov : ocene iz planov 1 2 3 4 5
- Obseg dela : ocene obsega dela iz planov 1 2 3 4 5
- Produktivnost : ocene produktivnosti iz planov 1 2 3 4 5
- Stroški (skupni) : ocene stroškov iz planov 1 2 3 4 5
- Strošek zagotavljanja kakovosti : planirani stroški 1 2 3 4 5
- Uspešnost zagotavljanja kakovosti : plan kakovosti 1 2 3 4 5
- _____ 1 2 3 4 5
- _____ 1 2 3 4 5

22. Na kakšen način naj bo (je) definirano (opisano) merjenje razvoja PO v majhni organizaciji, da bo to čimbolj enostavno in praktično uporabno?

- Merjenje je opisano kot del procesa razvoja PO 1 2 3 4 5
- Merski proces je predstavljen posebej 1 2 3 4 5
- Merski proces je za vsako vlogo podan kot recept 1 2 3 4 5
- Vsebina merjenja je predstavljena z več nabori 1 2 3 4 5
- Opis merjenja vsebuje nabor merskih ciljev 1 2 3 4 5
- Opis merjenja vsebuje merske entitete in atribute 1 2 3 4 5
- Opis merjenja vsebuje nabor vseh meritev 1 2 3 4 5
- Opis merjenja vsebuje nabor pokazateljev (analiz) 1 2 3 4 5
- _____ 1 2 3 4 5
- _____ 1 2 3 4 5

23. Pripombe:_____
_____
_____

Za trud in pomoč se Vam najlepše zahvaljujem!

Igor Rožanc

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

LITERATURA

Ada84 Adams, E. N.
Optimizing preventive service of software products.
IBM J. Research and Development, zv. 28, št. 1, str. 2 – 14, januar 1984.

Alb83 Albrecht, A. J., Gaffney, J. E.
Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation.
IEEE Transactions on Software Engineering, zv. 9, št. 6, str. 639 – 648, november 1983.

Ali02 Allison, A. S. W.
Meaningful Metrics - Conveying the context of your numbers.
dokument na spletu, STQE magazine, januar/februar 2002.
http://www.stickyminds.com

Ama98 Amam, K. E., Drouin, J-N., Melo, W.
SPICE: The Theory and Practice of Software Improvement and Capability Determination.
IEEE Computer Society Press, Los Alamitos (ZDA), 1998.

Ant01 Anton, A. I., Carter, R. A., Srikanth, H., Sureka, A., Williams, L. A., Yang, K., Yang, L.
Tailored CMM for a Small e-Commerce Company – Level 2: Repeatable.
teh. poročilo, North Carolina State University (ZDA), 23. avgust 2001.
TR-2001-09

Bas86 Basili, V. R., Selby, R.W., Hutchens, D. H.
Experimentation in Software Engineering.
IEEE Transactions on Software Engineering, zv. 12, št. 7, str. 733 –743, julij 1986.

Bas88 Basili, V. R., Rombach, H. D.
The TAME Project: Towards Improvement-Oriented Software Environments.
IEEE Transactions on Software Engineering, zv. 14, št. 6, str. 758 – 773, junij 1988.

Boe78 Boehm, B. W., Brown, J. R., McLeod, G., Lipow, M., Merrit, Ml.
Characteristics of Software Quality.
TRW Series of Software Technology, North Holland Publishing, Amsterdam (NIZ), 1978.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Boe81 Boehm, B. W.
Software Engineering Economics.
Prentice-Hall, Englewood Clifs (ZDA), 1981.

Boe91 Boehm, B. W.
Software Risk Management: Principles and Practices.
IEEE Software, zv. 8, št. 1, str. 32 – 41, januar 1991.

Boe95 Boehm, B. W., Clark, B., Horowitz, E., Westland, C., Madachy R., Selby, R.
Cost models for future life cycle processes: COCOMO 2.0.
Annals of Software Engineering, zv. 1, št. 1, str. 57 – 94, november 1995.

Bro92 Brocklehurst, S., Littlewood, B.
New Ways to Get Accurate Reliability Measures.
IEEE Software, zv. 9, št. 4, str. 34 – 42, julij 1992.

Bro94 Brodman, J. G., Johnson, D. L.
What Small Businesses and Small Organizations Say about CMM.
zbornik 16. mednarodne konference Software Engineering, str. 331 – 340,
Sorento (ITA), 16.-21. maj 1994.

BSi02 British Standard Institute (BSi).
Getting the measure of TickIT- Guidance and Information about the emerging ISO measurement standards for improving software process and how they relate to ISO 9001: 2000.
dokument na spletu, TickIT office, London (VB), februar 2002.
http:// www.tickit.org/measures.pdf

Car92 Carleton, A. D., Park, R. E., Goethert, W. B., Florac, W. A., Bailey, E. K., Pfleeger, S. L.
Software Measurement for DoD Systems:Recomendations for Initial Core Measures.
teh.poročilo, Software Engineering Institute, Pittsburg (ZDA), september 1992.
CMU/SEI-92-TR-019

Cha86 Charette, R.N.
Software Engineering Environments: Concepts and Technology.
McGraw-Hill, New York (ZDA), 1986.
ISBN 0-07-010645-2

Col94 Coleman, D., Ash, D., Lowther, B., Oman, P.
Using metrics to evaluate software system maintainability.
Computer, zv. 27, št. 8, str. 44 – 49, avgust 1994.

Dah97 Dahlberg, T., Jarvinen, J.
Chalenges to IS Quality.
Information and Software Technology, zv. 39, št. 12, str. 809 – 818,
december 1997.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Das92 Daskalantonakis, M. K. (Motorola).
A Practical View of Sw Measurement and Inplementation Experiences within Motorola.
IEEE Transactions on Software Engineering, zv. 18, št. 11, str. 998 – 1010, november 1992.

Das94 Daskalantonakis, M. K. (Motorola).
Achieving Higher SEI levels.
IEEE Software, let. 11, št. 4, str. 17 – 24, julij 1994.

Dav02 Davis, N., McHale, J.
Relating the Team Software Process (TSP) to the Capability Maturity Model for Software (SW-CMM).
teh.poročilo, Software Engineering Institute, Pittsburg (ZDA), junij 2002.
CMU/SEI-2002-TR-008

Dek00 Dekkers, C. A., Gunter, I.
Using "Backfiring" to Accurately Size Software - More Wishful Thinking than Science?
dokument na spletu, STQE magazine, september 2000.
http://www.stickyminds.com

Dek01 Dekkers, C. A.
The Secrets of Highly Successful Measurement Programs.
dokument na spletu, STQE magazine, oktober 2001.
http://www.stickyminds.com

DeM86 De Marco, T.
Controlling Software Projects.
Prentice Hall, druga izdaja, New York (ZDA), 1986.

DeM95 De Marco, T.
Mad about Measurement.
poglavje v knjigi »Why Does Software Cost So Much?«, Dorset House Publishing, str. 11 – 25, New York (ZDA), 1995.

Dem86 Deming, W. E.
Out of Crisis.
MIT Press, Cambridge (ZDA), 1986.

Dio93 Dion, R. (Raytheon).
Process Improvement and the Corporate Balance Sheet.
IEEE Software, zv. 10, št. 4, str. 28 – 35, julij-avgust 1993.

DoD98 Department of Defense and US Army.
Practical Software Measurement - A Fundation for Objective Project Management. Version 3.1a.
dokumentacija na spletu, Joint Group for Systems Engineering, Newport (ZDA), april 1998.
http://www.psmsc.com

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

DoD02 Department of Defense and US Army.
Practical Software Measurement - A Fundation for Objective Project Management. Version 4.0b.
dokument na spletu, Joint Group for Systems Engineering, Newport (ZDA), oktober 2002.
http://www.psmsc.com

Dol97 Dolado, J.J.
A Study of the Relationships among Albrecht and Mark II Function Points, Lines of Code 4GL and Effort.
J.Systems Software, Elsevier Science Inc., št. 37, str. 161-173, 1997.

Dor97 Dorsey, P., Koletzke, P.
Oracle Designer/2000 Handbook.
Osborne McGraw-Hill, Barkeley (ZDA), 1997.
ISBN 0-07-882229-7

Dor97a Dorsey, P., Koletzke, P.
Case Application Development Method (CADM) using Designer/2000.
dokument na spletu, Dulcain Inc., Oracle, 1997.
http://www.dulcian.com/papers/

Dor98 Dorsey, P., Coen, T.
CADM (Case Application Development Method) & CDM (Custom Development Method).
dokument na spletu, Dulcain Inc., Oracle, 1998.
http://www.dulcian.com/papers/

Ebe97 Ebert, C., Morschel, I.
Metric for quality analysis and improvement of object-oriented software.
Information and Software Technology, zv. 39, št. 7, str. 497 – 410, julij 1997.

Fen94 Fenton, N. E., Pfleeger, S.L., Glass, R. L.
Science and Substance: a challenge to software engineers.
IEEE Software, zv. 11, št. 4, str. 86 – 95, julij 1994.

Fen97 Fenton, N. E., Pfleeger, S.L.
Software Metrics: A Rigorous & Practical Approach.
PWS Publishing Company, druga izdaja, Boston (VB), 1997.
ISBN 0-534-95425-1

Fen99 Fenton, N. E.
Software Measurement Programs: Leveraging metrics basics into a functional measurement plan.
dokument na spletu, STQE magazine, maj/junij 1999.
http://www.stickyminds.com

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Fen00 Fenton, N. E, Neil, M.
Bayesian Belief Nets: A causual model for predicting defect rates and defect requirements.
dokument na spletu, STQE magazine, januar/februar 2000.
http://www.stickyminds.com

Fen00a Fenton, N. E, Neil, M.
Software Metrics: Roadmap.
zbornik 22. International Conference on Software Engineering, ACM Press, str. 357-370, 2000.
ISBN 1-58113-253-0

Fen01 Fenton, N. E
Viewpoint Article: Conducting and Presenting Empirical Software Engineering.
Empirical Software Engineering, št. 6, str. 195-200, 2001.

Flo92 Florac, W. A. et al.
Software Quality Measurement: A Framework for Counting Problems and Defects.
teh.poročilo, Software Engineering Institute, Pittsburg (ZDA), september 1992.
CMU/SEI-92-TR-022

Flo97 Florac, W. A., Park, R. E., Carleton, A. D.
Practical Software Measurement: Measuring for Process Measurement and Improvement.
učno gradivo, Software Engineering Institute, Pittsburg (ZDA), april 1997.
CMU/SEI-97-HB-003

Goe92 Goethert, W. B., Bailey, E. K., Busby, M. B. et al.
Software Effort and Schedule Measurement: A Framework for Counting Staff-hours and Reporting Schedule Information.
teh.poročilo, Software Engineering Institute, Pittsburg (ZDA), september 1992.
CMU/SEI-92-TR-021

Goo95 Goodland, M., Slater, C.
SSADM Version 4: A Practical Approach.
McGraw-Hill, London (VB), 1995.

Gra94 Grady, R. (Hewlett-Packard).
Successfully Applying Software Metrics.
Computer, zv. 27, št. 9, str. 18 – 25, september 1994.

Gra94a Grady, R., Van Slack, T.
Key Lessons in Achieving Widespread Inspection Use.
IEEE Software, zv. 11, št. 4, str. 46 – 57, julij 1994.

Haa94 Haase, V., Messnarz, R., Koch, G., Kugler, H. J., Dicrinis, P.
BOOTSTRAP: Fine-tuning process assesment.
IEEE Software, zv. 11, št. 4, str. 25 – 35, julij 1994.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Hal77 Halstead, M.H.
Elements of Software Science.
Elsevier North-Holland, New York (ZDA), 1977**.**

Hen94 Henry, J., Henry, S., Kafura, D., Matheson, L.
Improving Software Maintenance at Martin Marietta.
IEEE Software, zv. 9, št. 4, str. 67 – 75, julij 1994.

Has01 Hastings, T. E., Sajeev, A. S. M.
A Vector-Based Approach to Software Size Measurement and Effort Estimation.
IEEE Transactions on Software Engineering, zv. 27, št. 4, str. 337 – 350, april 2001.

Hei99 Heires, J. T.
Software Estimation Model Shoot-Out: SLIM Estimate vs. PRICE-S.
dokument na spletu, STQE magazine, november 1999.
http://www.stickyminds.com

Her94 Herbsleb, J.
Benefits of CMM – based Software Process Improvement: Initial results.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), 1994.
CMU/SEI-94-TR-13

Her98 Heričko, M.
Kakovost objektno-orientiranega razvoja programskih sistemov.
doktorska disertacija, Univerza v Mariboru, Fakulteta za elektotehniko, računalništvo in informatiko, 1998.

Het93 Hetzel, W.
Making Software Measurement Work: Building an Effective Software Measurement Program.
QED Publishing, Boston (ZDA), 1993.
ISBN 0-471-56568-7

Hor90 Horn, R. E.
Developing Procedures, Policies, and Documentation.
Information Mapping Inc., Waltham (ZDA), 1990.

Hor98 Horgan, G., Khaddaj, S., Forte, P.
Construction of an FPA-type metric for early lifecycle estimation.
Information and Software Technology, zv. 40, številka 8, str. 409 – 416, avgust 1998.

Hum89 Humphrey, W.S.
Managing the Software Process.
Addison-Wesley Publishing Company, prva izdaja, Reading (ZDA), 1989.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Hum91 Humphrey, W.S., Snyder, T.R., Willis, R. R.
Software Process Improvement at Hughes Aircraft.
IEEE Software, zv. 8, št. 4, str. 11 – 23, julij 1991.

Hum95 Humphrey, W.S.
The Discipline for Software Engineering.
Addison-Wesley Publishing Company, Reading (ZDA), 1995.
ISBN 0-201-54610-8.

Hum97 Humphrey, W.S.
Introduction to the Personal Software Process.
Addison-Wesley Publishing Company, Reading (ZDA), 1997.
ISBN 0-201-47719-X.

Hum97a Humphrey, W.S.
Managing Technical People: Innovation, Teamwork, and The Software Process.
Addison-Wesley Publishing Company, Reading (ZDA), 1997.
ISBN 0-201-54597-7.

Hum98 Humphrey, W.S.
Three Dimensions of Process Improvement - Part I: Process Maturity.
dokument na spletu, CrossTalk, str. 14 -17, februar 1998.
http://stsc.hill.af.mil/CrossTalk/1998/feb/

Hum98a Humphrey, W.S.
Three Dimensions of Process Improvement - Part II: The Personal Software Process.
dokument na spletu, CrossTalk, str. 13 -15, marec 1998.
http://stsc.hill.af.mil/CrossTalk/1998/mar/

Hum98b Humphrey, W.S.
Three Dimensions of Process Improvement - Part III: The Team Software Process.
dokument na spletu, CrossTalk, str. 14 -17, april 1998.
http://stsc.hill.af.mil/CrossTalk/1998/apr/

Hum99 Humphrey, W.S.
Pathways to Process Maturity: The Personal Software Process and Team Software Process.
SEI Interactive, zv. 2, št. 2, junij 1999.

Hum00 Humphrey, W.S.
Introduction to the Team Software Process.
Addison-Wesley Publishing Company, Reading (ZDA), 2000.
ISBN 0-201-47719-X.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Hum00a Humphrey, W.S.
The Personal Software Process (PSP).
teh.poročilo, Software Engineering Institute, Pittsburg (ZDA), november 2000.
CMU/SEI-2000-TR-022

Hum00b Humphrey, W.S.
The Team Software Process (TSP).
teh.poročilo, Software Engineering Institute, Pittsburg (ZDA), november 2000.
CMU/SEI-2000-TR-023

IEE92 The Institute of Electrical and Electronic Engineers (IEEE).
IEEE STD 610 – Standard Glossary of Software Engineering Terminology.
standard, The Institute of Electrical and Electronic Engineers, New York (ZDA), 1992.

ISO92 International Organization for Standardization (ISO).
ISO 9001, Sistemi kakovosti - modeli zagotavljanja kakovosti v načrtovanju/razvoju, proizvodnji, vgradnji in servisiranju.
standard, Urad za standardizacijo in meroslovje, Ljubljana (SLO), 1992.

ISO94 International Organization for Standardization (ISO).
ISO 9000-3, Standardi upravljanja in zagotavljanja kakovosti 3. del: Smernice za uporabo ISO 9001 pri razvoju, dobavi in vzdrževanju programske opreme.
standard, Urad za standardizacijo in meroslovje, Ljubljana (SLO), 1994.

ISO01 International Organization for Standardization (ISO).
ISO/IEC 15939 Information Technology – Software Measurement Process. Committee Draft.
predlog standarda, 2001.

Ive03 Iversen, J., Mathiassen, L.
Cultivation and engineering of a software metrics program.
Information Systems Journal, zv.13, št. 1, str. 3 – 19, januar 2003.

Jef93 Jeffery, R., Berry, M.
A framework for Evaluation and Prediction of Metrics Program Success.
zbornik 1. mednarodnega simpozija Software Metrics, str. 28 – 39, Los Alamitos (ZDA), 1993.

Joh98 Johnson, D. L., Brodman, J. G.
Applying the CMM to Small Organizations and Small Projects.
zbornik mednarodne konference Software Engineering Process Group, Chicago (ZDA), 9.-12. marec 1998.

Kah95 Kahoe, R., Jarvis, A.
ISO 9000-3: A Tool for Software Product and Process Improvement.
Springer-Verlag, prva izdaja, New York (ZDA), 1995.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Kan95 Kan, S. H.
Metrics and Models in Software Quality Engineering.
Addison-Wesley Publishing Company, prva izdaja, Reading (ZDA), 1995.

Keb02 Keber, B.
Zasnova in uporaba agilnih procesov.
zbornik 7. konference OTS'2002, str. 165 -175, Maribor (SLO),
junij 2002.

Kil01 Kilpi, T. (Nokia).
Implementing a Software Program at Nokia.
IEEE Software, zv.18, št. 6, str. 72 – 77, november-december 2001.

Kit01 Kitchenham, B. A., Huges, R. T., Linkman S. G.
Modeling Software Measurement Data.
IEEE Transactions on Software Engineering, zv. 27, št.. 9, str. 788 - 804,
september 2001.

Kul00 Kulik, P.
Software Metrics "State of the Art" - 2000.
dokument na spletu, STQE magazine, december 2000.
http://www.stickyminds.com

Kuv95 Kuvaja, P., Bicego, A., Dorling A.
SPICE: The Software Process Assessment Model.
zbornik konference ESI-ISCN'95: Measurement and Training Based Process Improvement, Conference Proceedings, Dunaj (AVS), 1995.

Lar00 Laryd, A., Orci, T.
Dynamic Capability Maturity Model for Small Organisations.
Zbornik mednarodne konference ASSE2000, Buenos Aires (ARG), 2000.

Lay00 Layman, B., Schaan, K.
Measurement maturity at CMM level 3.
zbornik mednarodne konference SM/ASM: Software Management and Applications of Software Measurement, San Jose (ZDA), 6.-10. marec 2000.

LeV00 Le Vie Jr., D. S.
Documentation Metrics - What Do You Really Want to Measure?
STC's INTERCOM Magazine, zv. 47, št. 10, december 2000.

Low96 Lowe, D. E., Cox, G.M. (Hewlett-Packard).
Implementing the Capability Maturity Model for Software Development.
Hewlett-Packard Journal, zv. 47, št. 4, str. 6-14, avgust 1996.

Mac01 MacGarry, M. A.
Applaying PSM and Insight within a Small Organization.
zbornik 5. letne konference PSM User's Group Conference, Keystone (ZDA),
22. - 26. julij 2001.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Mah99 Mahnič, V.
Modeli za zagotavljanje kakovosti programske opreme.
interno učno gradivo, Fakulteta za računalništvo in informatiko, Ljubljana (SLO), 1999.

Mah01 Mahnič, V., Rožanc, I.
Data Quality: A Prerequisite for Successful Data Warehouse Implementation.
Informatica, Special issue: The Changing University, and the Role of Information Technology, zv. 25, št. 2, str. 183 – 188, julij 2001.
ISSN 0350-5596

Mar81 Martin, J., Finkelstein, C.
Information engineering.
Pearson professional Education, New York (ZDA), november 1981.
ISBN 0-134- 64462- X

Mar85 Martin, J., McClure, C.
Diagramming techniques for analysts and programmers.
Prentice Hall, New Jersey (ZDA), 1985.

Maz94 Mazza C., Fairclough, J., Melton, B., De Pablo, D.,
Scheffer, A., Stevens, R.l.
Software Engineering Standards.
Prentice Hall Inc., prva izdaja, New Jersey (ZDA), 1994.

McA93 McAndrews, D.R.
Establishing a Software Measurement Process.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), 1993.
CMU/SEI-93-TR-16

McA00 McAndrews, D.R.
The Team Software Process (TSP): An overview and preliminary results of using disciplined practicies.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), julij 2000.
CMU/SEI-2000-TR-015

McC76 McCabe, T. J.
A Complexity Measure.
IEEE Transactions on Software Engineering, zv. 2, št. 4, str. 308 – 320, december 1976.

McC77 McCall, J. A., Richards, P. K., Walters, G.F.
Factors on Software Quality.
teh. poročilo, Rome Air Development Center, Rim (ITA), 1977.
RADC TR-77-369

McG01 McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J., Hal, F.
Practical Software Measurement: Objective Information for Decision Makers.
Addison-Wesley Longman, Reading (ZDA), 2001.
ISBN 0-201-71516-3

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Men98 Mendes, E., Harrison, R., Hall, W.
Reusability and maintainability in hypermedia applications for education.
Information and Software Technology, zv. 40, št. 14, str. 841 – 850, december 1998.

Men00 Mendonca, M. G., Basili, V. R.
Validation of an Approach for Improving Existing Measurement Frameworks.
IEEE Transactions on Software Engineering, zv. 26, št. 6, str. 484 – 499, junij 2000.

Mil88 Mills, E.E.
Software Metrics: SEI Curriculum Module SEI-CM-12-1.1.
učno gradivo, Software Engineering Institute, Pittsburg (ZDA) december 1988.
SEI-CM-12-1.1

Mun92 Munson, J.C., Khoshgoftaar, T. M.
Measuring dynamic program complexity.
IEEE Software, zv. 9, št. 6, str. 48 – 55, november 1992.

Ohl98 Ohlsson, M. C., Wohlin, C., Regnell B.
A project effort estimation study.
Information and Software Technology, zv. 40, št. 14, str. 831 – 840, december 1998.

Olo99 Olofsson, S.
Tailoring P-CMM for Small Companies.
Dokument na spletu, Umeå University (ŠVE), 1999.
http:// www.cs.umu.se/~olofsson/xjob/p-cmm.html

Oma93 Oman, P. W.
A case study in SQA audits.
Software Quality J., zv. 2, št. 1, str. 13-27, marec 1993.

Oma97 Oman, P. W., Pfleeger, S. L.
Applying Software Metrics.
IEEE Computer Society Press, prva izdaja, Los Alamitos (ZDA), 1997.
ISBN 0-8186-7645-0

Orc00 Orci, T., Laryd, A.
Dynamic CMM for Small Organisations - implementation aspects.
zbornik mednarodne konference EUROSPI 2000, Kopenhagen (DAN), 7. april 2000.

Par92 Park, R. E. et al.
Software Size Measurement: A Framework for Counting Source Statements.
teh.poročilo, Software Engineering Institute, Pittsburg (ZDA), september 1992.
CMU/SEI-92-TR-020

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Par96 Park, R. E., Goethert W. B., Florac W. A.
Goal-Driven Software Measurement - A Guidebook.
učno gradivo, Software Engineering Institute, Pittsburg (ZDA), avgust 1996.
CMU/SEI-96-HB-002

Pau93 Paulk, M. C., Curtis, B., Chrissis, M. B., Weber, C. V.
Capability Maturity Model for Software, Version 1.1.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), 1993.
SEI-93-TR-024

Pau93a Paulk, M. C., Weber, C. V., Garcia, S. M., Chrissis, M. B., Bush, M.
Key Practicies of the Capability Maturity Mode, Version 1.1.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), 1993.
SEI-93-TR-025

Pau94 Paulish, D. J., Carleton, A.D. (Siemens).
Case Studies of Software-Process-Impovement Measurement.
Computer, zv. 27, št. 9, str. 50 – 57, september 1994.

Pau94a Paulk, M. C.
A Comparison of ISO 9001 and the Capability Maturity Model for Software.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), 1994.
CMU/SEI-94-TR-012

Pau95 Paulk, M. C.
How ISO 9001 Compares with the CMM.
IEEE Software, zv. 12, št. 1, str. 74 - 83, januar 1995.

Pau95a Paulk, M. C., Weber, C. V., Curtis, B., Chrissis, M. B.,
The Capability Maturity Model for Software: Guidelines for Improving the Software Process.
Addison-Wesley, Reading (ZDA), 1995.

Pau99 Paulk, M. C.
Using the Software CMM in Small Projects and Organizations.
dokument na spletu, Software Engineering Institute, Pittsburg (ZDA),
februar 1999.
http://www.sei.cmu.edu

Pau01 Paulk, M. C.
Extreme Programming from a CMM Perspective.
IEEE Software, zv. 18, str. 19 - 26, november/december 2001.

Pfl90 Pfleeger, S. L., McGowan, C. (Contel).
Software Metrics in the Proces Maturity Framework.
J. Systems and Software, zv. 12, št.3, str. 255 – 261, Amsterdam (NIZ), 1990.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Pfl91 Pfleeger, S. L., Fitzgerald.
Software Metrics Tool kit: Support for Selection, Collection and Analysis.
Information and Software Technology, zv. 33, št. 7, str. 477 – 482, september 1991.

Pfl92 Pfleeger, S. L., Fitzgerald, J.C., Rippy. D.A.
Using Multiple Metrics for Analysis of Improvement.
Software Quality J., zv. 1, št. 1, str. 27 – 36, marec 1992.

Pfl94 Pfleeger, S. L., Fenton, N. E., Page, S.
Evaluating Software Engineering Standards.
Computer, zv. 27, št. 9, str. 71 – 79, september 1994.

Pig98 Pighin, M.
An empirical quality measure based on complexity values.
Information and Software Technology, zv. 40, št. 14, str. 861 – 870, december 1998.

Pit99 Pitts, D.
Why Is Software Measurement Hard?
zbornik konference SM/ASM, San Jose (ZDA), 15. – 19. februar 1999.

Piv98 Pivka, M.
Control Mechanisms for Assuring Better IS Quality.
Informatica, zv. 22, št. 3, str. 309 – 317, oktober 1998.

Por90 Porter, A.A., Selby, R.W.
Empiricaly Guided Software Development Using Metric Based Classification Trees.
IEEE Software, zv. 7, št. 2, str. 46 - 54, marec 1990.

Pud01 Pudussery, A.
Starting a Metric Program.
dokument na spletu, STQE magazine, oktober 2001.
http://www.stickyminds.com

Put78 Putnam, L. H.,
A General Empirical Solution to the Software Sizing and Estimating Problem.
IEEE Transactions on Software Engineering, zv. 4, št. 4, str. 345 - 361, julij 1978.

Ray99 Raynus, J.
Software Process Improvement with CMM.
Artech House, Boston (VB), 1999.
ISBN 0-89006-644-2

Red00 Reddy, S. K.
CMM Level 2 - an Interpretation for Small Projects.
zbornik 7. mednarodne konference SPIN, Otawa (CAN), november 2000.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Ric02 Richins, K., Jarzombek, J.
Maturing Measurement Capability: The CMMI Measurement Thread and Racognized Industry Guidance - Standards.
zbornik 6. letne PSMUG konference, Keystone (ZDA), 22. – 26. julij 2002.

Rif91 Rifkin, S., Cox, C.
Measurement in Practice.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), 1991.
SEI-91-TR-016

Rou95 Rout, T. P.
SPICE: A Framework for Software Process Assessment.
Software Process Improvement and Practice, avgust 1995.

Roz97 Rozman, I., Vajde Horvat, R., Győrkős, J., Heričko, M.
PROCESSUS – Integration of SEI CMM and ISO quality models.
Software Quality Journal, št. 6, str. 37 – 63, 1997.

Roz00 Rozum, J. A.
Software Measurement Companion Guide.
učno gradivo na spletu, Carnegie Mellon University, Pittsburg (ZDA), 2000.
http:// www.distance.cmu.edu

Rož99 Rožanc, I.
Zasnova merskega modela za zagotavljanje kakovosti v procesu razvoja poslovnih aplikacij.
zbornik 8. konference ERK'99, zv. B: str. 69 -72, Portorož (SLO), september 1999.

Rož00 Rožanc, I., Mahnič, V.
Principi za zagotavljanje kakovosti podatkov in njihova uporaba pri merskem modelu.
zbornik 9. konference ERK 2000, zv. B, str. 71 - 74, Portorož (SLO), september 2000.

Rož01 Rožanc, I., Mahnič, V.
Merjenje velikosti programske opreme.
zbornik 10. konference ERK 2001, zv. B, str. 51 - 54, Portorož (SLO), september 2001.

Rož02 Rožanc, I., Mahnič, V.
Zagotavljanje kakovosti programske opreme v majhnih organizacijah.
čaka na objavo v Elektrotehniškem vestniku, 2002.

Rož02a Rožanc, I., Mahnič, V.
Uporaba osebnega in skupinskega procesa razvoja programske opreme v majhnih organizacijah.
zbornik 11. konference ERK 2002, Portorož (SLO), zv. B, str. 75 - 78, september 2002.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Sch92 Schneidewind, N. F.
Methodology for Validating Software Metrics.
IEEE Transactions on Software Engineering, zv. 18, št. 5, str. 410 – 422, maj 1992.

Sch00 Schrank, M. J. (The MITRE Corporation).
Does the CMM® Make $ense for Small Businesses?
zbornik mednarodne konference IRS Small Business, New Carrollton (ZDA), 21. junij 2000.

Sch01 Schneidewind, N. F.
Knowledge Requirements for Software Quality Measurement.
Empirical Software Engineering, št. 6, str. 201 – 205, 2001.

SEI97 Software Engineering Institute (SEI).
Process Maturity Profile for the Software Community 1997 Mid-Year Update.
dokument na spletu, Software Engineering Institute, Pittsburg (ZDA), 1997.
http://www.sei.cmu.edu/sema

SEI00 Software Engineering Institute (SEI).
Mapping of SW-CMM V1.1 to CMMI-SE/SW V1.0..
dokument na spletu, Software Engineering Institute, Pittsburg (ZDA), 2000.
http://www.sei.cmu.edu/cmmi

SEI02 Software Engineering Institute (SEI).
Process Maturity Profile of the Software Community 2002 Mid-Year Update.
dokument na spletu, Software Engineering Institute, Pittsburg (ZDA), avgust 2002.
http://www.sei.cmu.edu/sema

SEI02a CMMI Development Team.
Capability Maturity Model Integration CMMI(SM), Version 1.1 - CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE-SW-IPPD-SS, V1.1). Continous Representation.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), marec 2002. CMU/SEI-2002-TR-011

SEI02b CMMI Development Team.
Capability Maturity Model Integration CMMI(SM), Version 1.1 - CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE-SW-IPPD-SS, V1.1). Staged Representation.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), marec 2002. CMU/SEI-2002-TR-012

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

SEI02c CMMI Development Team.
Capability Maturity Model Integration CMMI(SM), Version 1.1 - CMMI for Software Engineering (CMMI-SW, V1.1). Continous Representation.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), avgust 2002. CMU/SEI-2002-TR-028

SEI02d CMMI Development Team.
Capability Maturity Model Integration CMMI(SM), Version 1.1 - CMMI for Software Engineering (CMMI-SW, V1.1). Staged Representation.
teh. poročilo, Software Engineering Institute, Pittsburg (ZDA), avgust 2002. CMU/SEI-2002-TR-029

SEL95 NASA Software Engineering Laboratory.
Software Measurement Guidebook (Revision 1).
učno gradivo, Goddard Space Flight Center, Greenbelt (ZDA), junij 1995. SEL-94-102

Som92 Sommerville, I.
Software Engineering.
Addison-Wesley Publishing Company, četrta izdaja, New York (ZDA), 1992.

Sta94 Stark, G., Durst, R. C., Vowell, C. V. (NASA Johnson Space Center).
Using Metrics in Management Decision Making.
Computer, zv. 27, št. 9, str. 42 – 48, september 1994.

Str01 Strike, K., El Emam, K., Madhavji, N.
Software Cost Estimation with Incomplete Data.
IEEE Transactions on Software Engineering, zv. 27, št. 10, str. 890 – 908, oktober 2001.

Sym91 Symons, C. R.
Software Sizing and Estimating: MkII FPA.
John Willey & Sons, Chichester (ZDA), 1991.

Van95 Van den Berg, K.
Software Measurement and Functional Programming.
doktorska disertacija, Univerza v Twentu, Enschede (NIZ), 1995.
ISBN 90-9008251-4

Web99 Webb, D.
Using the TSP on the TaskView Project.
dokument na spletu, CrossTalk, str. 3 - 10, 1999 februar 1999.
http://stsc.hill.af.mil/CrossTalk/1999/feb/

Wel94 Weller, E. F. (Bull HN Information Systems).
Using Metrics to Manage Software Projects.
Computer, zv. 27, št. 9, str. 27 – 33, september 1994.

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

Wel01 Weller, E.
Measurement in the CMM.
dokument na spletu, STQE magazine, 26. april 2001.
http://www.stickyminds.com

Wie96 Wiegers, K. E.
Misconceptions of the Capability Maturity Model.
Software Development, zv. 4, št. 11, str. 57 – 64, november 1996.

Wig97 Wigle, G. B., Yamamura G. (Boeing).
Practicies of an SEI CMM Level 5 SEPG.
dokument na spletu, CrossTalk, november 1997.
http://www.stsc.hill.afmil/CrossTalk/1997/nov/

Wit97 Wittig, G., Finnie, G.
Estimating software development effort with connectionist models.
Information and Software Technology, zv. 39, št. 7, str. 469 – 476, julij 1997.

Zha98 Zhao M., Wohlin, C., Ohlsson, N., Xie M.
A Comparison between software design and code metrics for the predicition of software fault content.
Information and Software Technology, zv. 40, številka 14, str. 801 – 810, december 1998.

Zub01 Zubrow, D.
The measurement and Analysis Process Area in CMMI.
dokument na spletu, Software Engineering Institute, 2001.
http://www.sei.cmu.edu/cmii/

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

POMEMBNEJŠE SPLETNE STRANI

Bootstrap	Metodologija BOOTSTRAP http://www.boostrap-institute.com
BSi	British Standard Institute - BSI http:// www.bsi-global.com
CMM	Capability Maturity Model – CMM (SEI) http://www.sei.cmu.edu/cmm
CMMI	Capability Maturity Model Integration – CMMI (SEI) http://www.sei.cmu.edu/cmmi
IEEE	The Institute of Electrical and Electronic Engineers - IEEE http://www.ieee.org
ISO	International Organisation for Standardization - ISO http://www.iso.org
PSM	Practical Software Measurement (PSM) http://www.psmsc.com
PSP	Personal Software Model – PSP (SEI) http://www.sei.cmu.edu/psp
SEI	Software Engineering Institute (SEI) http://www.sei.cmu.edu
SEL	Software Engineering Laboratory – SEL (NASA) http://sel.gsfc.nasa.gov
SEMA	Software Engineering Measurement and Analysis – SEMA (SEI) http://www.sei.cmu.edu/sema
SPICE	Metodologija SPICE http://www-sqi.cit.gu.edu.au/spice/
TickIT	Software Quality Management System Certification http://www.tickit.org
TSP	Team Software Model – PSP (SEI) http://www.sei.cmu.edu/tsp

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

SLOVAR POMEMBNEJŠIH POJMOV

aktivnost – opravilo ali del procesa razvoja PO (korak, faza ali cikel)
analiza izvedljivosti – ena izmed analiz v prilagodljivem procesu razvoja PO
analiza performans – ena izmed analiz v prilagodljivem procesu razvoja PO
atribut – značilnost merske entitete, ki je zanimiva za merjenje
centralni repozitorij – centralna podatkovna zbirka različnih podatkov organizacije
cilj – formalen cilj na ključnem področju modela CMM, ki je osnova za mersko izhodišče
cikel – aktivnost v prilagodljivem procesu razvoja PO; sestavlja ga več faz
CMM – model stopenj zrelosti (ang. Capability Maturity Model)
CMMI – sestavljeni model stopenj zrelosti (ang. Capability Maturity Model Integrated)
delovna faza – faza v prilagodljivem procesu razvoja PO, ki izvaja razvojno aktivnost
določena vrednost – rezultat meritev, ki ga določimo z zdrževanjem ali izpeljavo
ER diagram – diagram entiteta-razmerje za predstavitev podatkovnega modela
faza – aktivnost v prilagodljivem procesu, ki ima določen rezultat; sestavlja jo več korakov
GQM – ciljno usmerjena metoda za določanje meritev (ang. Goal-Question-Metric)
informacijski izdelek – rezultat izvedbe analize (ocena, napoved, grafični prikaz)
IS – informacijski sistem, končni rezultat uporabe prilagodljivega procesa razvoja PO
ISO/IEC 15939 – standard za opis merskega procesa pri razvoju PO
izkoristek odpravljanja napak – pokazatelj, ki kaže uspešnost zagotavljanja kakovosti
izmerjena vrednost – rezultat izvedbe neposredne (osnovne) meritve
izpeljana meritev – meritev, ki je izračunana iz več drugih meritev
kakovost izdelkov – značilnost, ki jo merimo s številom vnešenih in odpravljenih napak
kombinirana metoda – naša metoda za določanje meritev; kombinacija GQM in MQG
korak – elementarna aktivnost v prilagodljivem procesu, ki ima izvajalca; sestavni del faze
linearna regresija – statistična metoda; služi za ocenjevanje na podlagi zgod. podatkov
majhna organizacija za razvoj IS – organizacija za razvoj IS z manj kot 50 zaposlenimi
merska entiteta – objekt (v procesu razvoja PO), ki ga merimo (aktivnost, vir ali izdelek)
merski model – formalen model, ki opisuje merjenje v organizaciji za razvoj PO
Merski Informacijski Model – del pristopa PSM, ki opisuje strukturo opisa merjenja
merski podatki (rezultati) – izmerjene ali določene vrednosti meritev
Merski Procesni Model – del pristopa PSM, ki opisuje postopke merjenja
merski repozitorij – del centralnega repozitorija, namenjen hranjenju podatkov o merjenju
mersko izhodišče – formaliziran opis cilja merjenja v našem merskem pristopu
meta model – model za opis podatkov o merjenju s predstavitvijo razmerij med njimi
meta podatki – podatki, ki opisujejo merjenje
MQG – objektno usmerjena metoda za določanje meritev (ang. Metric-Question-Goal)

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

napovedovanje – naša izvirna analiza v prilagodljivem procesu razvoja PO
napredovanje – značilnost, ki jo merimo z doseganjem mejnikov
obseg dela – značilnost, ki jo merimo v okviru vsake aktivnosti
območje zaupanja – statistično določeno območje, v katerem pričakujemo rezultat
ocenjevanje – ena izmed analiz v prilagodljivem razvoju procesa PO
osnovna meritev – neposredna meritev merske entitete, ki jo izvaja posameznik
PB – podatkovna baza, podatkovni del končnega rezultata
pokazatelj – rezultat analize, prikaz ene ali več meritev uporabniku z določenega vidika
ponovljiva faza – faza, ki se ponavlja ob vsakem začetku ali zaključku cikla
poraba časa – značilnost aktivnosti, ki jo merimo v okviru vsake aktivnosti
posebno mersko izhodišče – merski cilj, ki je določen na podlagi posebnih zahtev projekta
prilagodljiv proces razvoja PO – naš proces razvoja PO za majhne organizacije za razvoj IS
produktivnost – pomemben pokazatelj, ki ga spremljamo
proces razvoja PO – množica aktivnosti, orodij in praks organizacije s ciljem razviti PO
PSM – pristop Praktično merjenje PO (ang. Practical Software Measurement)
PSP – model Osebni proces razvoja PO (ang. Personal Software Process)
PO – programska oprema
skupina SEPG – skupina za tehnologijo programske opreme na ravni organizacije
skupina SQA – skupina za zagotavljanje kakovosti na ravni organizacije
splošni cilj – cilj na podlagi zahtev majhnih organizacij, ki je osnova za mersko izhodišče
stalno mersko izhodišče – merski cilj, ki je enak za vse projekte organizacije
status merske entitete – pokazatelj trenutnega stanja aktivnosti, izdelka ali vira
strošek – pomembna značilnost virov, ki jo mersko spremljamo
strošek kakovosti – pokazatelj; razmerje aktivnosti zagotavljanja kakovosti in razvoja
SUPB – sistem za upravljanje podatkovnih baz
TSP – model Skupinski proces razvoja PO (ang. Team Software Process)
velikost izdelka – pomembna značilnost izdelkov, ki jo merimo
združena meritev – meritev, ki je določena z združevanjem več podobnih meritev
združevalna struktura – princip združevanja pri združeni meritvi

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko

IZJAVA

Spodaj podpisani Igor Rožanc izjavljam, da sem doktorsko disertacijo izdelal samostojno pod vodstvom mentorja prof. dr. Viljana Mahniča.

Ljubljana, 13. maj 2003 Igor Rožanc

© Univerza v Ljubljani, Fakulteta za računalništvo in informatiko