



Št. naloge: 01648/2010

Datum: 15.03.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ŽARKO VIŠEKRUNA**

Naslov: **MIGRACIJA PODATKOV MICROSOFT SHAREPOINT V VELIKEM
SISTEMU
MICROSOFT SHAREPOINT DATA MIGRATION IN A LARGE SYSTEM**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Preučite področje podatkovne migracije na splošno in tudi specifično za Microsoftov sistem SharePoint. Identificirajte aktivnosti v postopku migracije in jih umestite v ustrezne faze v življenjskem ciklu projekta. Posebno pozornost posvetite strategijam za fizični prenos podatkov. Za konkretni problem migracije v velikem podjetju izberite ustrezne metode in postopke migracije, naredite načrt migracije, opredelite funkcionalne zahteve in izdelajte ustrezno orodje. Po izvedbi migracije kritično ovrednotite uspešnost izvedbe.

Mentor:

doc. dr. Mojca Ciglarič



Dekan:

prof. dr. Franc Solina

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žarko Višekruna

**MIGRACIJA PODATKOV MICROSOFT SHAREPOINT V
VELIKEM SISTEMU**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Ljubljana, 2010

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žarko Višekruna

**MIGRACIJA PODATKOV MICROSOFT SHAREPOINT V
VELIKEM SISTEMU**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentorica: doc. dr. Mojca Ciglarič

Ljubljana, 2010

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

Rezultati diplomskega dela so intelektualna lastnina Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje Fakultete za računalništvo in informatiko ter mentorja.

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Žarko Višekruna
z vpisno številko 63030078

sem avtor diplomskega dela z naslovom:

MIGRACIJA PODATKOV MICROSOFT SHAREPOINT V VELIKEM SISTEMU

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
doc. dr. Mojce Ciglarič
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 13.9.2010

Podpis avtorja: _____

ZAHVALA

Ob koncu študija se iskreno zahvaljujem svojim najdražjim, ki so mi ves čas študija stali ob strani, me spodbujali in se skupaj z mano veselili uspehov. Posebno zahvalo namenjam mentorici doc. dr. Mojci Ciglarič za strokovno pomoč in nasvete pri izdelavi diplomske naloge. Zahvaljujem se tudi vsem sodelavcem, še posebej Petru Javorniku in Juretu Šubari, ki sta mi nesebično pomagala na poti do cilja. Nenazadnje se zahvaljujem vsem, ki so v času študija obogatili moje znanje.

KAZALO

| | |
|---|-----------|
| POVZETEK | 1 |
| ABSTRACT | 2 |
| 1 UVOD | 3 |
| 2 PODATKOVNO MIGRIRANJE | 4 |
| 2.1 Opredelitev podatkovnega migriranja | 4 |
| 2.2 Pravila podatkovnega migriranja | 6 |
| 2.2.1 Podatkovno migriranje je poslovni in ne tehnološki problem | 6 |
| 2.2.2 Podjetje (kot naročnik) ve najbolje | 7 |
| 2.2.3 Nobena organizacija ne potrebuje, noče, niti ne bo plačala za popolne podatke | 7 |
| 2.2.4 Če nečesa ne moreš prešteti, potem to ne šteje | 8 |
| 3 FAZE IN AKTIVNOSTI MIGRACIJSKEGA PROJEKTA | 9 |
| 3.1 Prepoznavna in izbor vseh podatkovnih virov | 10 |
| 3.2 Proces spoznavanja podatkov | 10 |
| 3.3 Čiščenje podatkov | 11 |
| 3.4 Transformiranje podatkov..... | 11 |
| 3.5 Testiranje in validacija..... | 11 |
| 3.6 Fizični prenos podatkov in preklop sistemov | 12 |
| 3.6.1 Strategija za shranjevanje vmesnih rezultatov | 12 |
| 3.6.2 Strategija za sinhroniziranje podatkov | 13 |
| 3.6.3 Preklopna strategija | 14 |
| 3.6.3.1 »Big bang« | 15 |
| 3.6.3.2 Paralelno delovanje..... | 15 |
| 3.6.3.3 Inkrementalna migracija | 16 |
| 3.7 Revidiranje in dokumentiranje..... | 16 |
| 4 MIGRACIJA PODATKOV V MICROSOFTOVEM SISTEMU SHAREPOINT.. | 17 |
| 4.1 Zgodovina SharePointa..... | 17 |
| 4.2 SharePoint terminologija | 18 |
| 4.3 Pristopi k nadgradnji SharePointa | 19 |
| 4.4 Funkcionalne zahteve vsebinske migracije..... | 21 |
| 4.4.1 Granularnost migracije..... | 22 |
| 4.4.2 Reorganizacija portala tekom migracije..... | 22 |
| 4.4.3 Nadzor nad zbirkami mest in vsebinskimi podatkovnimi bazami..... | 22 |
| 4.4.4 Združevanje virov podatkov..... | 23 |
| 4.4.5 Preslikovanje spletnih naslovov | 23 |

| | | |
|------------|---|-----------|
| 5 | PRAKTIČNI PRIMER MIGRACIJE PODATKOV V MICROSOFTOVEM SISTEMU SHAREPOINT | 24 |
| 5.1 | Diagram obstoječega intranet portala | 24 |
| 5.2 | Diagram prenovljenega portala | 26 |
| 5.3 | Členitev projektnega plana po aktivnostih..... | 27 |
| 5.3.1 | Prepoznavna in izbor vseh podatkovnih virov..... | 27 |
| 5.3.2 | Proces spoznavanja podatkov | 27 |
| 5.3.3 | Čiščenje podatkov..... | 27 |
| 5.3.4 | Transformiranje podatkov | 27 |
| 5.3.5 | Testiranje in validacija..... | 28 |
| 5.3.6 | Fizični prenos podatkov | 28 |
| 5.3.7 | Revidiranje in dokumentiranje | 28 |
| 5.4 | Orodje za vsebinsko migracijo podatkov SharePoint | 28 |
| 5.4.1 | Preslikovanje spletnih naslovov | 30 |
| 5.5 | Prenos organizacijskih enot | 32 |
| 5.5.1 | Migracija Področja..... | 36 |
| 5.5.1.1 | Izvoz vsebine področij na disk, uvoz spletnih mest | 36 |
| 5.5.1.2 | Migracija seznamov in knjižnic..... | 37 |
| 5.5.1.3 | Migracija spletnih gradnikov..... | 41 |
| 5.5.1.4 | Migracija globalnih skupin, vlog | 44 |
| 5.5.1.5 | Migracija opozoril | 45 |
| 5.6 | Prenos spletnih mest..... | 47 |
| 5.6.1 | Priprava migracijskega okolja | 48 |
| 5.6.2 | Migracija podatkovne baze | 49 |
| 6 | POST-MORTEM ANALIZA | 51 |
| 7 | VIRI..... | 53 |
| 8 | PRILOGE..... | 54 |
| 8.1 | Seznam tabel..... | 54 |
| 8.2 | Seznam slik..... | 54 |

SEZNAM UPORABLJENIH KRATIC IN SIMBOLOV

| | |
|-------------|-------------------------------------|
| ERP | Enterprise Resource Planning |
| CRM | Customer Relationship Management |
| IT | Information Technology |
| SPS | SharePoint Portal Server |
| STS | SharePoint Team Services |
| WSS | Windows SharePoint Services |
| MCMS | Microsoft Content Management System |
| MOSS | Microsoft Office SharePoint Server |
| XML | Extensible Markup Language |
| SQL | Structured Query Language |

POVZETEK

Podatkovno migriranje je enkraten prenos podatkov, ki vključuje prestrukturiranje teh podatkov, hkrati pa ohranja njihov način uporabe. Namen diplomske naloge je prikaz teoretičnega ozadja podatkovnega migriranja in njegovo praktično aplikacijo na primeru večjega sistema SharePoint. Naloga je razdeljena v dva dela, pri čemer je prvi del teoretični. V njem je opredeljeno podatkovno migriranje, faze in aktivnosti migracijskega projekta ter migracija podatkov v Microsoftovem sistemu SharePoint. Drugi del naloge je posvečen praktičnemu prikazu migriranja podatkov v Microsoftovem sistemu SharePoint. Predstavljena so stanja sistema pred in po migraciji, migracijski plan, razdeljen po aktivnostih ter orodje za vsebinsko migracijo podatkov SharePoint. Omenjeno orodje prenaša podatke med sistemoma s pomočjo programskega vmesnika. Prikazani so koncepti, ki pri tem veljajo. Za konec je izvedba migracije podatkov kritično ovrednotena.

Ključne besede:

podatkovno migriranje, Microsoft SharePoint, portal, transformacija, migracijski projekt, informacijski sistem

ABSTRACT

Data migration is one-time movement of data that involves some sort of restructuring of that data while the usage of that data remains constant. The purpose of the thesis is the display of theoretical background of data migration and its practical application on a large scale SharePoint system. The thesis is composed of two parts, at which the first one is theoretical. In the theoretical part the data migration, phases and activities of a migration project and data migration in Microsoft SharePoint are defined. The second part is dedicated to practical display of data migration in a real world Microsoft SharePoint system. The states of the system before and after data migration are shown. The migration plan, divided among project activities and SharePoint content migration utility are also introduced. The utility that is mentioned uses Advanced Programming Interface (API) to transfer data from one system to another. Concepts for this process are shown. At the end of thesis the critical evaluation takes place.

Key words:

Data migration, Microsoft SharePoint, portal, transformation, migration project, information system

1 UVOD

Pravijo, da so spremembe edina stalnica v življenju. Dejstvo, ki prav gotovo velja, ko je govora o življenjski dobi informacijskih sistemov. Ta, sestavljena iz razvoja novejših in boljših rešitev na eni ter odmiranja in ukinjanja starejših na drugi strani, je namreč vse krajša. Nič nenavadnega ni, če se na vsake tri leta pojavi nova različica ERP, CRM, komunikacijskega portala ali katerekoli druge rešitve. Podjetja so do splavitev novih produktov načeloma indiferentna in se raje odločajo za vlaganje dodatnih sredstev v podaljševanje življenjske dobe svojih informacijskih sistemov. Vendar pa tega ne morejo početi v nedogled. Prehiter tehnološki razvoj, prihod novih standardov, ukinitve podpore obstoječim sistemom in preteča konkurenca terjajo svoj davek. S časom prevlada zavedanje, da tudi najboljši informacijski sistem ni tako zmogljiv, da bi lahko večno sledil razvoju in potrebam organizacije, ki ga uporablja.

Kaj storiti, ko informacijski sistem ne deluje več tako, kot bi v podjetju želeli? V takšnih primerih se po navadi podjetja odločajo med nakupom licenc že pripravljenega oz. lastnim razvojem novega sistema. Obe možnosti se ponašata s svojimi prednostmi in slabostmi, vendar to zaenkrat ni tako pomembno. Na celotno zadevo je potrebno pogledati iz drugega zornega kota. Podjetje kot dinamična tvorba, v sklopu svojega delovanja generira informacije, ki jih v najboljšem primeru hrani v svojih informacijskih sistemih. Ti podatki so za podjetje lahko vitalnega pomena in jih je potrebno ohraniti tudi po zamenjavi informacijskega sistema. Novi sistemi nudijo izboljšave predvsem v smeri funkcionalnosti, struktura podatkov nad katerimi operirajo, pa je redkokdaj združljiva z obstoječimi podatki podjetja. Da bi ohranili želene podatke je potrebno poleg projekta uvajanja nove rešitve zastaviti tudi projekt podatkovne migracije.

Podatkovno migriranje je pojem, s katerim označujemo prenos podatkov med sistemoma, pri čemer se struktura podatkov spremeni. Gre za kompleksen proces, ki zahteva drugačne pristope, tako v tehnološkem kot tudi v sociološkem smislu. Prva polovica diplomske naloge je posvečena razkrivanju teh posebnosti.

Druga polovica diplomske naloge je namenjena prikazu podatkovne migracije, izvedene na primeru večjega intranet portala.

2 PODATKOVNO MIGRIRANJE

V prvem delu diplomske naloge bom najprej predstavil osnove podatkovnega migriranja iz vidika teorije in prakse, v nadaljevanju pa podal štiri »zlata« pravila migriranja.

2.1 Opredelitev podatkovnega migriranja

»Podatkovno migriranje je enkraten prenos podatkov, ki vključuje prestrukturiranje teh podatkov, hkrati pa ohranja njihov način uporabe.« [2]

Zgoraj podano definicijo je mogoče razčleniti na posamezne lastnosti:

- Podatkovno migriranje je enkratni proces. Časovno gledano lahko traja dlje časa, vendar je končano, ko so preneseni vsi podatki. Aktivnost, ki je ponovljiva imenujemo podatkovna integracija (npr. prenos podatkov v podatkovno skladišče).
- Podatkovno migriranje vključuje prestrukturiranje podatkov na nek način: spremembe v zapisu formatov, združevanje polj, itd... Če prestrukturiranja ni, govorimo o premiku podatkov.
- Podatkovno migriranje ni samo sebi namen. Vedno obstaja povezava med migracijo podatkov in razvojem, migracijo ali posodobitvijo informacijskega sistema.
- Podatkovno migriranje ohranja način uporabe podatkov. To pomeni, da se podatke prenese iz operativnega sistema v operativni in ne npr. analitičnega (npr. Operativni sistemi so vir podatkov, ki se prenesejo v podatkovna skladišča. Podatki v skladiščih so nato uporabljeni v raznih analizah).
- Podatkovno migriranje lahko vključuje menjavo konteksta. Če moramo združiti več SAP sistemov v enega, potem se kontekst ne zamenja. Kontekst se spremeni, če npr. migriramo SAP sistem v CRM. Migracije, kjer se kontekst spremeni veljajo za kompleksnejše od tistih, pri katerih se le ta ohrani.

Da bi podatkovno migriranje razumeli še iz bolj praktičnega vidika, so v tabeli Tabela 2-1 zbrani nekateri primeri.

| Primer | Podatkovno migriranje |
|---|------------------------------|
| Prenos podatkov iz ene ali več podatkovnih baz v drugo podatkovno bazo, pri čemer podatkom ni treba spremeniti strukture. | NE |
| Prenos podatkov v podatkovno skladišče. | NE |
| Arhiviranje starih in neuporabljenih podatkov na sekundarno lokacijo, pri čemer podatkom ni treba spremeniti strukture. | NE |
| Migriranje podatkov iz enega sistema v drugega, npr. Iz Oracle v SAP. | DA |
| Posodobitev obstoječega sistema na novo verzijo, pri čemer se spremeni način shranjevanja teh podatkov. | DA |
| Implementacija novega sistema, pri čemer je vsaj del potrebnih podatkov shranjen v obstoječih aplikacijah. | DA |

Tabela 2-1: Praktični primeri migriranja

Migriranje podatkov ni preprosto opravilo. Tovrstni projekti redno prekoračijo časovni in finančni plan. Razlogov za to je ogromno, eden večjih dejstvo, da podatkovno migriranje ni samostojna disciplina. Posledice tega dejstva se odražajo kot manko ustreznih metodologij, standardov in specializiranih strokovnjakov na tem področju.

Organizacija »Bloor Research« je leta 2007 naredila obsežno raziskavo, v kateri so podali ocene o uspešnosti migracijskih projektov [3]. Rezultati pravijo, da naj bi leta 2007 za podatkovno migriranje, razvoj ustreznih programskih rešitev, potrebno strojno opremo, svetovanje in storitve na tem področju, itd. bilo porabljenih več kot 5 milijard dolarjev. Samo v 16 % podatkovno migriranje ni bilo krivec za prekoračitev časovnega in finančnega plana. Več kot 64 % projektov je prekoračilo časovni plan, finančnega pa 37 %. Večina odgovornih je kot glavnega krivca navedla napačno ocenitev obsega tovrstnega projekta. Zanimiv je tudi podatek, da je le vsako deseto podjetje pri svojem delu uporabljalo dodatna orodja, s katerimi bi si pomagali pri razumevanju podatkov in vzdrževanju njihove kvalitete.

2.2 Pravila podatkovnega migriranja

Podatkovno migriranje ni ravno področje, ki bi bilo bogato s standardi in metodologijami. Prav zato se pojavlja vse več pravil, izhajajočih iz izkušenj strokovnih posameznikov in skupin. Eden takšnih strokovnjakov je John Morris, ki je podal štiri pravila za zmanjševanje tveganj pri izvedbi migracijskih projektov. [4]

2.2.1 Podatkovno migriranje je poslovni in ne tehnološki problem

Postavite se v vlogo IT strokovnjaka, ki je pravkar dobil nalogo migrirati podatke iz starega v nov sistem. Vodstvo vam dodeli potrebne resurse in določi rok do zaključka projekta. Ker imate na videz vse pripravljeno, se lotite pisanja ustreznih orodij. Občasno delovanje orodij preverite na manjši množici testnih podatkov. Čez čas pa se pojavi prvi večji testni prenos, pri katerem gre vse narobe: polovica podatkov manjka ali pa je neustreznih. Izkaže se, da poleg starega sistema obstaja še cela kopica podatkovnih zbirk, ki so vitalnega pomena za delovanje poslovnega sistema. Le te so ustvarili uslužbenci, ker stari sistem ni ustrezal njihovim zahtevam. Vas na to nihče ni opozoril, vendar tega ne morete izkoristiti kot izgovor. Vi ste namreč tisti, ki je s svojim tehničnim znanjem nase prevzel odgovornost za uspešen prenos podatkov.

Opisan scenarij je klasičen primer, ki nakazuje pomembnost prvega pravila. Dejstvo je, da je podatkovno migriranje po navadi sestavni del IT projekta. Ker takšne projekte izvajajo IT strokovnjaki, se pojavijo tendence k iskanju tehnoloških rešitev (npr. raba specifičnih orodij za prečiščevanje in uvoz podatkov v sisteme), zanemari pa se poslovni vidik. IT strokovnjak na eni strani in podjetje kot naročnik na drugi vidita iste podatke, vendar jih obe strani interpretirata po svoje. Izvajalec jih vidi predvsem na tehnični ravni, kot množico bitov in bajtov, naročnik pa v njih prepozna vrednost. Tu se izkaže slabost tehnološkega pristopa k reševanju migracijskih problemov. Nikoli namreč ne velja, da bi izvajalec bolje poznal pomen obstoječih podatkov za naročniško podjetje, kot podjetje samo. Naloga naročnika je torej prepoznati vire in predstaviti vrednost podatkov (Kaj bomo migrirali?), naloga izvajalca pa, da vrednost teh podatkov skozi proces migriranja ohrani (Kako bomo to naredili?). Takšno medsebojno razmerje mora veljati skozi celoten proces migriranja. Če temu ni tako, naročnik hitro privzame pasivno vlogo. Izvajalec v tem primeru prevzame odgovornost za tehnično in poslovno kvaliteto podatkov, kar pa z veliko gotovostjo vodi v nezadovoljstvo obeh strani.

2.2.2 Podjetje (kot naročnik) ve najboljše

»Podjetje je poganjalo stari sistem in stari sistem je poganjal podjetje.« [4]

Podjetje je tisto, ki ve kje so shranjeni vsi podatki, pomembni za njegovo delovanje. Sledi, da je le podjetje samo lahko odgovorno za kvaliteto in ustreznost svojih podatkov. Čeprav se to zdi povsem logično, pa marsikdaj temu ni tako. Vedno obstaja tveganje, da izvajalec podleže pritiskom in začne prevzemati vso odgovornost odločitev na svoja ramena. Vsaka odločitev je bodisi dobra bodisi slaba, vendar se srž problema skriva drugje. Odločitev je lažje sprejeti, kot nositi njene posledice. Zavedati se moramo, da bo naročnik tisti, ki bo še dolgo po končani migraciji moral uporabljati nov sistem. Zaradi tega mora biti naročnik postavljen za nosilca celotnega procesa migriranja, kot tudi samega rezultata migracije. Problem je torej v lastništvu! Kot primer vzemimo naročnika, ki želi po končani migraciji obdržati poleg vsakega dokumenta še zgodovino le tega za pet različic nazaj. Izvajalec se mora v tem primeru podrediti željam naročnika, čeprav poseduje znanje in tehnologijo za prenos vseh različic. Nepodrejanje naročniku bi v večini primerov povzročilo večje stroške pri izvedbi projekta.

Čeprav mora imeti naročnik pri odločitvah glavno besedo, je vseeno potrebno poudariti vlogo IT strokovnjakov. Njihova naloga je, da tehnološko podprejo odločitve naročnika in pospešujejo proces migriranja. Odločitve so namreč pogosto pogojene z znanjem, ki ga naročnik poseduje, le to pa je razpršeno med več uslužbencev. Izvajalčeva naloga je torej tudi pomoč pri iskanju tega znanja.

2.2.3 Nobena organizacija ne potrebuje, noče, niti ne bo plačala za popolne podatke

Popolne kvalitete podatkov praktično ni mogoče doseči. To lahko dokažemo že z zelo preprostim primerom. Recimo, da imamo v stari podatkovni bazi tabelo oseb, pri čemer za vsako osebo hranimo njeno ime, naslov in telefonsko številko. V tabeli sta zapisa dveh oseb, ki se ujemata v vseh atributih. Ali gre za eno osebo? Sta osebi oče in sin ali pa je vse skupaj le naključje? Brez dodatnih informacij tega ne moremo vedeti. Zelo verjetno je, da se v devetnajstih poskusih od dvajsetih ne bomo odločili narobe, če enega od zapisov v novem sistemu preprosto zanemarimo, še vedno pa ostaja tisti zapis pri katerem smo naredili napako. Pri popolni kvaliteti podatkov se je potrebno zavedati, da se kvaliteta podatkov s časom slabša. Po raziskavah sodeč naj bi bil faktor slabitve približno 2 % na mesec. Kot primer

lahko vzamemo statistiko o porokah, smrtih in selitvah posameznikov o katerih hranimo podatke. Če o tovrstnih dogodkih nismo obveščeni, pomeni, da imamo v nekem določenem trenutku vsaj delno nepravilne podatke. K temu dodajmo še dejstvo, da se ogromno podatkov vnaša ročno in ker smo ljudje nagnjeni k napakam, postane popolna kvaliteta res utopičen pojem.

Upoštevajoč omenjena dejstva lahko rečemo, da pretirana težnja po iskanju popolne kvalitete podatkov projektu kvečjemu škodi. Nivo kvalitete podatkov mora biti določen z dogovorom med naročnikom in izvajalcem že pred začetkom migracije, tekom projekta pa budno spremljan in reguliran.

2.2.4 Če nečesa ne moreš prešteti, potem to ne šteje

V prejšnjem pravilu smo spoznali, da je za uspešnost migracije potrebno določiti nivo kvalitete podatkov, ki ga želimo doseči. Kvaliteto podatkov po navadi opišemo s pridevniki, kot so »dobra«, »odlična« ali celo »popolna«. Kdaj pa sploh vemo, da smo dosegli »odlično kvaliteto podatkov«? Četrto pravilo pravi, da na to vprašanje ne moremo odgovoriti, če podatkov primerno ne ovrednotimo. Pomembno je, da podamo oceno o stanju obstoječih podatkov. Tako dobimo pregled na tem kaj imamo in lažje določimo to, kar želimo. Da tekom migracije ne bi skrenili s poti do zelenih rezultatov je potrebno uvesti meritve tistih parametrov, ki se nam zdijo pomembni. Primeri merljivih parametrov so:

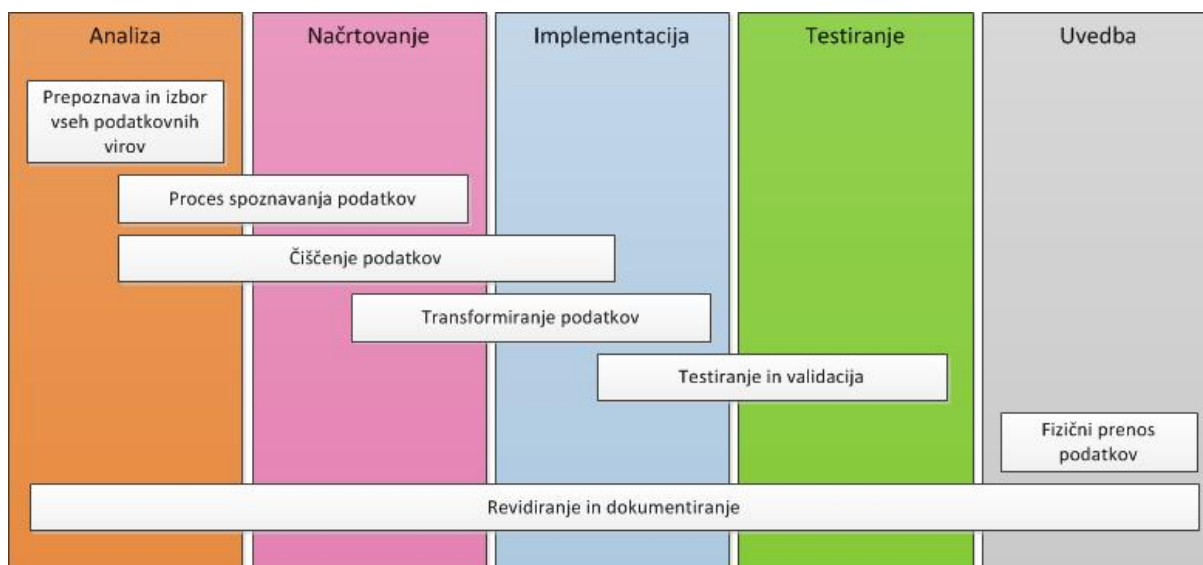
- število migriranih zapisov,
- število migriranih tabel,
- število migriranih zapisov (deljeno po tehnologiji),
- število migriranih zapisov (deljeno po aplikacijah),
- število zapisov neustrezne kakovosti,
- število napak pri migraciji,
- čas nedosegljivosti sistema zaradi migracije,
- potrebne prostorske kapacitete,
- število popravljenih zapisov.

3 FAZE IN AKTIVNOSTI MIGRACIJSKEGA PROJEKTA

Vsak dobro voden projekt gre v svojem življenjskem ciklu skozi določene faze, kot so analiza, načrtovanje, implementacija, testiranje in uvedba. Migracijski projekti v tem pogledu niso nič drugačni. Posebnost takšnih projektov je razvidna šele pri razčlenitvi posameznih faz po aktivnostih. Philip Howard [2] omenja sedem aktivnosti, ki so specifične za migracijske projekte. Te aktivnosti so:

- prepoznavna in izbor vseh podatkovnih virov,
- proces spoznavanja podatkov,
- čiščenje podatkov,
- transformiranje podatkov,
- fizični prenos podatkov,
- testiranje in validacija,
- revidiranje in dokumentiranje.

Za lažjo predstavo je na sliki Slika 3-1 prikazana umestitev omenjenih aktivnosti po razvojnih fazah.



Slika 3-1: Umestitev aktivnosti v projektne faze

3.1 Prepoznavna in izbor vseh podatkovnih virov

Metodologij in postopkov s katerimi bi si olajšali prepoznavo podatkovnih virov ni. To pa ne pomeni, da je ta aktivnost manj pomembna od ostalih. Lahko bi rekli, da velja ravno nasprotno. Napake, povzročene v tej stopnji projekta se nemalokrat pokažejo šele ob produkcijski migraciji, kjer se moramo soočiti z manjkajočimi oziroma nepopolnimi podatki. Že prvo zlato pravilo omenja, da so poslovni sistemi kompleksna okolja, v katerih posamezniki pomanjkljivosti informacijskih sistemov kompenzirajo z lastnimi shrambami podatkov. To pomeni, da se kot informatiki iskanja virov najlažje in najbolj učinkovito lotimo na tak način, da v proces vključimo končne uporabnike. Ko smo, vsaj delno, prepričani, da smo prepoznali vse vire podatkov, je potrebno nad njimi narediti analizo vrzeli (angl. »Gap Analysis«). Na tak način lažje ugotovimo, če najdeni podatkovni viri vsebujejo vse podatke, potrebne za nov sistem. Če nismo našli vseh potrebnih podatkov, pomeni, da smo pozabili na vsaj en vir oziroma, da ti podatki preprosto ne obstajajo.

3.2 Proces spoznavanja podatkov

Ko uspešno prepoznamo vse podatkovne vire, moramo določiti, katere podatke bomo iz posameznih virov črpali. Pri tem je zelo pomembno, da ob izbiri podatkov prepoznamo tudi njihove pomanjkljivosti in napake. Na tak način lažje ocenimo kvaliteto podatkov ter podamo časovno in finančno oceno, ki sta potrebni, da bi podatke popravili do zelenega nivoja.

Podatke moramo razumeti na vsaj treh ravneh, in sicer:

- **Na ravni polja (zapisa):** Pomembno je, da ugotovimo, če so vsi zapisi v ustreznih poljih, ali so vsa polja pravega tipa, ali obstaja ustrezno polje v novem sistemu,...
- **Na ravni tabele:** Preveriti je potrebno njihovo relevantnost za sistem, kako so definirani primarni ključi, tuji ključi,...
- **Na ravni povezanih tabel:** Odkriti moramo vse tiste povezave med tabelami, ki tvorijo poslovna pravila, potrebna pri implementaciji novega sistema.

Pri spoznavanju podatkov si v veliki meri lahko pomagamo s specifičnimi orodji. Včasih pa to ni dovolj. Problemi se lahko hitro pojavijo, ko imamo opravka z implicitnimi povezavami med podatki, oziroma ko podatki niso povezani na ravni podatkovne baze, temveč kar v sami aplikaciji. Delo nam lahko oteži tudi dejstvo, da stari sistemi v večji meri niso bili nikoli

ustrezno dokumentirani. Prav tako je pomembno, da podatkov ne gledamo le v kontekstu enega vira. Če imamo več virov, moramo odkriti povezave, ki obstajajo med njimi. Zakaj so te povezave tako pomembne? Zato, ker lahko predstavljajo poslovna pravila, ki jih moramo implementirati v novem sistemu.

3.3 Čiščenje podatkov

Po definiciji iz Wikipedie [1] je čiščenje podatkov proces zaznavanja in popravljanja (ali izločevanja) pokvarjenih ali netočnih zapisov na ravni polj, tabel ali podatkovne baze. Tehnično je definicija povsem ustrezna, vendar se je na tem mestu vredno spomniti na tretje pravilo migriranja, ki pravi, da nobena organizacija ne potrebuje, noče, niti ne bo plačala za popolne podatke. Velja torej, da je nivo kvalitete, ki ga želimo doseči, stvar medsebojnega dogovora med naročnikom in izvajalcem. Način s katerim ta nivo dosežemo pa je v domeni uporabljenih metodologij, ki presegajo tematiko pričujočega diplomskega dela.

3.4 Transformiranje podatkov

Transformiranje podatkov je dvofazni proces, pri čemer moramo v prvi fazi ugotoviti katere transformacije potrebujemo, v drugi pa te transformacije implementirati. Izkušensko pravilo pravi, da je faza ugotavljanja težje izvedljiva kot sama implementacija. Razlog tiči predvsem v manku ustreznih orodij, kar pa posledično pomeni, da nam ne preostane drugega, kot da potrebne transformacije poiščemo ročno, tj. s preizkušanjem različnih možnosti, tehnološkimi testi in svojim šestim čutom. Ko definiramo vsa potrebna poslovna pravila za prenos podatkov, postane kodiranje ustrezne rešitve veliko lažje.

3.5 Testiranje in validacija

Tekom implementacije, pa tudi že v zgodnejših fazah je potrebno testirati rezultate dela. S tem se izognemo fizičnim napakam, ki so po navadi sintaktične narave. Še bolj pomembne so logične napake, nastale zaradi površnega preslikovanja podatkov.

3.6 Fizični prenos podatkov in preklon sistemov

Na kakšen način bomo prenašali podatke je eno izmed pomembnejših vprašanj, ki si jih moramo zastaviti pri migriranju. Odgovorov na to vprašanje je več, vsi pa izhajajo iz poslovnega modela naročnika. Če izvajamo migracijo za podjetje, katerega sistemi neprestano obratujejo, moramo izbrati drugačne strategije, kot pri podjetju, ki sistem uporablja manj intenzivno. Strategije, ki jih moramo izdelati so:

- **strategija za shranjevanje vmesnih rezultatov** (angl. »Transitional Storage Strategy«),
- **strategija za sinhroniziranje podatkov** (angl. »Synchronization Strategy«),
- **preklopna strategija** (angl. »Business Cutover Strategy«). [5,6]

3.6.1 Strategija za shranjevanje vmesnih rezultatov

Strategija za shranjevanje vmesnih rezultatov opisuje način, po katerem bomo shranili podatke, ki bodo prehajali med novim in starim sistemom. Možni sta dve rešitvi:

- **Začasno vmesno shranjevanje** (angl. »Temporary Storage Solution«):
Podatke iz starega sistema izvlečemo v vmesno skladišče, ter jih predelamo do te mere, da so primerni za uvoz v novi sistem. Rešitev z vmesnim shranjevanjem za poslovanje ni vedno najbolj ustrezna, saj nad podatki v vmesnem skladišču ni mogoče izvajati poizvedb ali kakšnih drugih operacij.



Slika 3-2: Začasno vmesno shranjevanje

- **Direkten prenos** (angl. »Real-time Solution«):
Ta pristop odstrani potrebe po vmesnem shranjevanju. Podatki so namreč izvlečeni, preneseni in uvoženi v realnem času. Strategija direktnega prenosa se večinoma uporablja takrat, ko si nedosegljivosti ali omejenega delovanja sistemov ne moremo privoščiti. Ker je novi sistem tekom prenosa že na voljo, obstaja večja možnost, da bo kateri od uporabnikov zahteval podatke, ki še niso bili preneseni. Potrebno je razviti

dodatno logiko, ki bo takšne zahteve pošiljala staremu sistemu, ta pa bo želene podatke vrnil novemu sistemu.



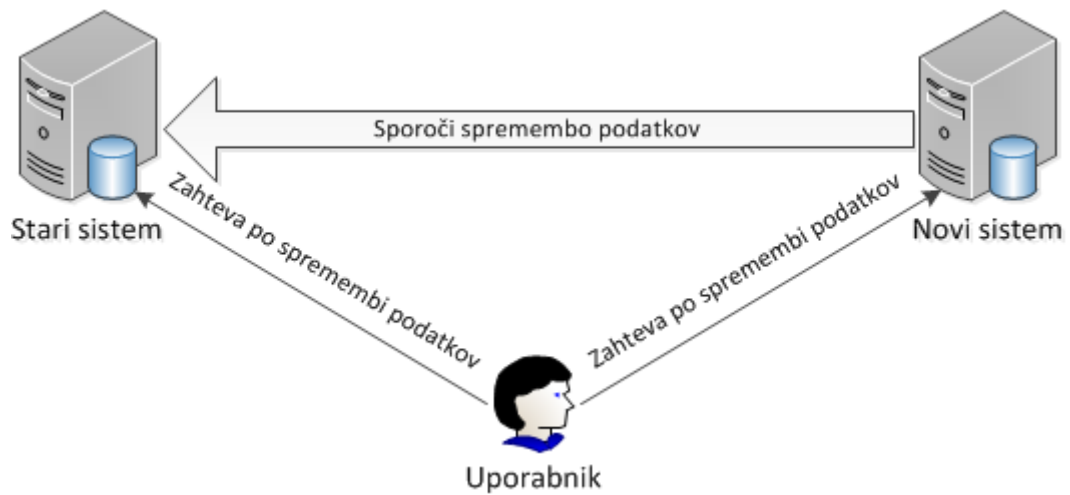
Slika 3-3: Direktni prenos

3.6.2 Strategija za sinhroniziranje podatkov

Da bi izpeljali preklonni strategiji kot sta paralelno delovanje in inkrementalna migracija, moramo osnovati pravila za sinhronizacijo podatkov med starim in novim sistemom. Pri delnih prenosih so namreč podatki razdeljeni med oba sistema, kar pa ne sme vplivati na njihovo uporabo. Vsak podatek, prenesen ali ne, mora biti na voljo tako staremu kot novemu sistemu. Zaradi tega moramo osnovati dodatno sistemsko logiko, oziroma podatkom dodati oznake, s katerimi določimo njihov status (Ali so že bili preneseni). Zopet sta na voljo dve možnosti, in sicer:

- **Enosmerna sinhronizacija** (angl. »Mono-directional«):

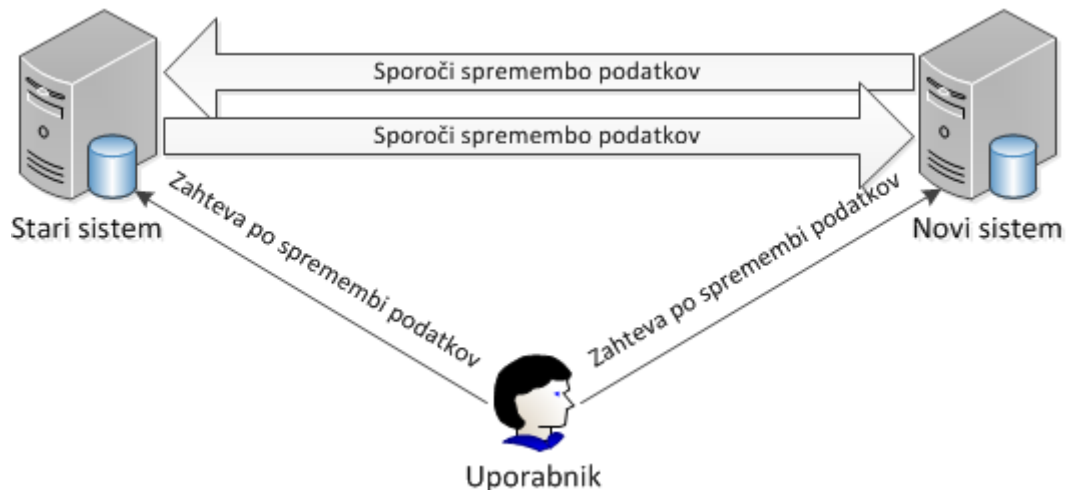
Strategija, pri kateri novi sistem vsakršne spremembe nad svojimi podatki sporoči tudi staremu sistemu, medtem ko obratno ne velja. S tem, ko novi sistem ponudimo uporabnikom na voljo, vsakršne spremembe nad podatki pa sporočimo tudi staremu sistemu, si v bistvu zagotovimo dobro razveljavitevno strategijo. V primeru, da novi sistem ne deluje po pričakovanjih, lahko preprosto preklopimo nazaj na starega.



Slika 3-4: Enosmerna sinhronizacija

- **Dvosmerna sinhronizacija** (angl. »Bi-directional«):

Strategija, ki se najbolj pogosto uporablja v navezi z inkrementalno migracijo. Značilnost omenjene migracije je postopen prenos podatkov. Na tak način kaj hitro pridemo do situacije, ko prenesemo množico podatkov, vseh funkcionalnih modulov, ki te podatke uporabljajo pa ne. Da bi ohranili delovne procese obeh sistemov, moramo zagotoviti, da se vsakršne spremembe nad podatki enega sistema sporočijo drugemu sistemu in obratno.



Slika 3-5: Dvosmerna sinhronizacija

3.6.3 Preklopna strategija

Zelo pomembno je, kako izvedemo dejanski preklon med starim in novim sistemom. Na to vpliva več dejavnikov. Če izvajamo projekt v močno reguliranem podjetju s strani raznih agencij, moramo ubrati pristop, pri katerem lahko sproti in natančno testiramo rezultate prenosa. Po drugi strani se lahko odločimo za pristop pri katerem prenesemo vse podatke

naenkrat. Tak primer je možen, če migracija ne traja predolgo, oziroma nedosegljivost sistema ne ogroža blagostanja samega podjetja. Možne strategije so tri, in sicer:

- »Big Bang«,
- paralelno delovanje,
- inkrementalna migracija.

3.6.3.1 »Big bang«

Značilnost »Big bang« strategije je, da se stari sistem uporablja, dokler se ne odločimo za migracijo vseh podatkov. Času od začetka do konca prenosa podatkov pravimo preklopna faza. Po končanem prenosu se stari sistem odklopi ter začne uporaba novega sistema. Takšna strategija ima dve večji prednosti pred ostalimi:

- Za razliko od ostalih strategij se pri »Big bang« izognemo težavam, ki nastanejo zaradi medsebojnih odvisnosti med starim in novim sistemom.
- Zmanjšamo strojne in systemske potrebe na minimum, saj stari in novi sistem istočasno tečeta le v preklopni fazi.

V praksi se »Big bang« strategija v čisti obliki ne pojavlja pogosto. Eden izmed pomembnejših razlogov je predvsem dejstvo, da stari in novi sistem v preklopni fazi nista na voljo za uporabo. Najbolj smotrno je, da preklopno fazo izvedemo v času, ko je sistem najmanj obremenjen, npr. ob koncu tedna. Pogosto pa tudi to ni dovolj. Podjetja, ki svojo dejavnost vršijo s pomočjo internetnih sistemov, si neodzivnosti sistema preprosto ne morejo privoščiti. Druga slabost tovrstne strategije se pokaže šele po določenem času, ko je novi sistem že v uporabi. Nepremostljive napake v delovanju, najdene tako pozno, lahko projekt resno ogrozijo. Da bi tveganja karseda omilili, je potrebno izdelati ustrezno razveljavitveno strategijo. Tako lahko sistem za določen čas povrnemo v prejšnje stanje in začnemo s ponovnimi pripravami na sam prenos.

3.6.3.2 Paralelno delovanje

Za razliko od »Big bang« strategije se pri paralelnem delovanju po končani migraciji stari sistem še nekaj časa uporablja. Novi sistem deluje sočasno in je na voljo za testiranje ustreznosti delovanja. Oba sistema tečeta, dokler se ne prepričamo, da novi sistem deluje pravilno. Po zadostnem testiranju se stari sistem ugasne.

Prednost takega pristopa je v tem, da smo po dokončnem preklopu prepričani v ustreznost novega sistema. Prav tako minimiziramo čas preklopne faze, saj je večina podatkov že v novem sistemu.

Slabost paralelnega delovanja predstavlja poraba resursov. Med te ne štejemo le strojne in sistemske opreme, ki je zmožna poganjati oba sistema hkrati, temveč tudi ustrezne kadre. Oni so namreč tisti, ki preverjajo delovanje novega sistema. Prav tako med slabosti štejemo večjo kompleksnost takšnega pristopa. Ker sistema vsaj nekaj časa tečeta hkrati, moramo razviti ustrezno strategijo za sinhroniziranje podatkov.

3.6.3.3 Inkrementalna migracija

Pri inkrementalni migraciji gre za podoben pristop kot pri paralelnem delovanju. Razlika je v tem, da podatke migriramo postopoma. Na tak način lahko ocenimo kateri moduli so za poslovanje kritičnega pomena in jih prenesemo najprej. Ker sistem prenašamo po kosih, lahko hitreje preverimo ustreznost delovanja posameznega modula in ga damo na voljo v uporabo. Tako imamo v kratkem času na voljo funkcionalnosti novega sistema, ki jih lahko praktično uporabimo pri poslovanju. Prednost je torej hitrejše povračilo naložbe. Največja prednost pa je hkrati tudi podlaga za največjo slabost takšnega pristopa. Zavedati se moramo, da je modulov, ki uporabljajo iste podatke lahko več. Problem nastane, ko podatke prenesemo, vseh ustreznih modulov pa ne. V tem primeru moramo podati dodatna pravila, ki določajo vir podatkov za tiste module, ki niso bili preneseni.

3.7 Revidiranje in dokumentiranje

Revidiranje in dokumentiranje je aktivnost, ki se pojavlja v vseh fazah projektnega razvoja. Potreba po dokumentiranju načrtov in opravljenega dela se namreč vseskozi pojavlja. Veliko lažje je regulirati tok projekta, če veš kakšne so njegove zahteve, koliko dela bo za izpolnitev zahtev potrebno in katere od njih so že bile opravljene.

4 MIGRACIJA PODATKOV V MICROSOFTOVEM SISTEMU SHAREPOINT

V drugem delu diplomske naloge bom namenil nekaj besed zgodovini Microsoftovega sistema SharePoint. V nadaljevanju bom, zaradi lažjega razumevanja, podal opis nekaterih terminov, za konec pa predstavil pristope, po katerih lahko izvedemo nadgradnjo sistema SharePoint na naslednjo različico. Eden izmed omenjenih pristopov je vsebinska migracija, pri kateri bom izpostavil njene funkcionalne zahteve.

4.1 Zgodovina SharePointa

Tehnologija SharePoint ima precej kratko zgodovino. Začetki segajo v leto 1999, ko so pri Microsoftu izdali produkt z imenom Site Server. Ta se je že ponašal z nekaterimi funkcionalnostmi, ki jih danes prepoznamo kot del SharePointa, npr. personalizacija strani, upravljanje z dokumenti, indeksiranje ter iskanje. Bolj kot celovita platforma je bil Site Server skupek orodij, ki pač niso našla mesta v sklopu ostalih produktov. Pravi pohod na portalsko tržišče se je začel dve leti kasneje z izidom SharePoint Portal Server 2001 (SPS 2001) platforme. Kljub temu, da se je slednji obnašal kot pravo portalsko ogrodje, je imel precejšnje pomankljivosti. Skalabilnosti ni podpiral pretirano dobro, prav tako pa je bil neprijazen do razvijalcev. Istega leta so pri Microsoftu izdali še dodatek za Office 2000 z imenom SharePoint Team Services (STS), s katerim so nadgradili funkcionalnosti za spletno sodelovanje.

Leta 2003 se je zgodil še en pomemben mejnik. STS je bil nadgrajen v Windows SharePoint Services 2.0 (WSS 2.0). Tako je nastalo ogrodje, ki je nudilo množico storitev za sodelovanje in podlago za gradnjo spletnih aplikacij, upravljanje z dokumenti, delo z dokumentnimi knjižicami in sezname ter spletnimi gradniki, povezovanje z Microsoft Office paketom, itd. Na osnovi WSS 2.0 je nastal tudi novi SharePoint Portal Server 2003 (SPS 2003), ki je v večji meri odpravil težave z razširljivostjo. Za razvijalce je bilo prav tako poskrbljeno z dodano podporo za SharePoint v orodja kot je Visual Studio. Kljub temu, da je bil opravljen ogromen korak v primerjavi z prejšnjo različico SharePoint portala, se je slednji še vedno ponašal z mnogo pomanjkljivostmi. Ena od pomembnejših je prav gotovo dejstvo, da WSS 2.0 kot tudi SPS 2003 nista podpirala delovnih tokov. Vsebine tako ni bilo mogoče vpeti v poslovne

procesu. Kot primer vzemimo delovni tok za novice, ki jih skoraj vsako podjetje z intranet portalom nudi v branje svojim zaposlencem. S SPS 2003 ni bilo mogoče realizirati korakov, ki bi vsebino novice prenesli od avtorja k uredniku v potrditev ter naprej k administratorju v objavo. Tovrstno pomanjkljivost so pri Microsoftu dokaj ne-elegantno obšli z vpeljavo še enega sistema, ki je bil SharePointu komplementaren. Ta sistem, katerega večji del je bil namenjen delu z vsebino, se je imenoval Microsoft Content Management System (MCMS).

Z leti je SharePoint vse bolj dobival na veljavi v svetu portalov. Uspešne serije preprosto ne gre ukiniti, tako da je leta 2007 izšla nova različica WSS, imenovana kar WSS 3.0. Istega leta, seveda bolj pompozno, pa so izdali še Microsoft Office SharePoint Server (MOSS 2007). Slednji je razširil že obstoječe funkcionalnosti ter prinesel marsikaj novega. Za povprečno portalsko rešitev tako nista bila več potrebna dva sistema. Zmogljivosti MCMS so namreč vgradili kar v MOSS 2007. Poleg tega se je nova različica strežnika med drugim ponašala s predlogami za wiki ter bloge, podporo mobilnim aparatom, spletnimi gradniki za mreženje ter agregacijo podatkov.

SharePoint je s tretjo izdajo postal ena najpopularnejših platform za portalske rešitve, kar so potrdili tudi pri vodilni raziskovalni družbi Gartner, ko so ga uvrstili v magični kvadrant vodilnih rešitev. Še bolj zanimive so nekatere analogije, ki so se tekom let pojavile. Nekateri namreč pravijo, da naj bi SharePoint za korporativni svet predstavljal to, kar Facebook predstavlja za svetovno populacijo. Tovrstne primerjave gre seveda jemati s ščepcem soli, pa vendar pomembnosti SharePointa ne moremo zanikati.

4.2 SharePoint terminologija

Termini, s katerimi operiramo v svetu SharePointa so nekoliko drugačni od običajnih pojmov, uporabljenih pri spletnih aplikacijah. Kot taki lahko nepoznavalcu povzročijo ogromno oviro pri razumevanju pričujoče diplomske naloge, kot tudi vseh ostalih del s podobno tematiko. Za lažji prehod na SharePoint sem na tem mestu zbral nekaj najpomembnejših terminov in jih poskušal na kratko opisati. Vsi omenjeni termini igrajo vidno vlogo tudi v nadaljnjih poglavjih, pri čemer nekatere od njih poskušam predstaviti tudi iz kakšnega drugega zornega kota. Najpomembnejši termini so:

- **Vsebinska podatkovna baza** (angl. »Content Database«) – Podatkovna baza, ki jo SharePoint uporablja za shranjevanje dokumentov, sprememb predlog mest, itd.
- **Spletna aplikacija** (angl. »Web application«) – Spletna aplikacija, ki gostuje na »Internet Information Services« strežniku in je prilagojena za delo s SharePoint platformo.
- **Spletno mesto** (angl. »Site«) – Spletna stran v zbirki mest.
- **Področje** (angl. »Area«) – logična struktura, podobna spletnemu mestu. Pojem področij najdemo le pri SPS 2003.
- **Zbirka mest** (angl. »Site Collection«) – Za večino uporabnikov je zbirka mest kar korensko spletno mesto. Iz programerskega vidika to ni res, saj je korensko mesto le eno izmed mest v zbirki (Predstavljeno z objektom »SPWeb«). Lahko rečemo, da je zbirka mest več kot le množica mest, saj vsebuje tudi konfiguracijske nastavitve, ki so skupne vsem mestom. Kot primer lahko navedem nastavitve uporabnikov in grup, iskalnika, nabor predlog mest in funkcij spletnih mest (angl. »Site Collection Features«).
- **Seznam** (angl. »List«) – Vsebuje informacije, zložene v vrstice in stolpce, torej tabele. Vsaka vrstica predstavlja posamezen zapis (»SPListItem«), vsak stolpec (»SPField») pa vrednost posameznega atributa za določen zapis.
- **Spletni gradnik** (angl. »Web Part«) – modul z lastno logiko in izgledom, ki gradi posamezne spletne strani. Njihova značilnost je, da jih je možno povezovati in personalizirati, tj. prirediti njihov izgled in v neki meri funkcionalnost za vsakega uporabnika posebej. Spletne gradnike lahko dodajamo na stran dinamično v posebej prirejena odložišča (angl. »WebPart Zone«).
- **Polje** (angl. »Field«) – Posamezen stolpec v seznamu.
- **Grupa** (angl. »Group«) – Množica SharePoint uporabnikov.

4.3 Pristopi k nadgradnji SharePointa

Microsoft ob vsaki novi izdaji SharePoint Portala objavi dokumente, v katerih so zabeleženi možni pristopi k nadgradnji starejših portalov na novejšo različico. Odločitev o izbiri ustreznega pristopa ni trivialna. Potrebno je namreč upoštevati mnogo dejavnikov, ki nam lahko močno zagrenijo življenje, če jih pri odločanju izpustimo. Kot orodje za osnovno orientacijo, sem s pomočjo dokumenta »Upgrading to Office SharePoint Server 2007« [7]

sestavil tabelo večine uradno podprtih pristopov, tem pa dodal še enega neuradnega, tj. vsebinsko migracijo.

| Pristop | Opis | Prednosti | Slabosti | Primeren okvirni scenarij |
|--|---|--|--|--|
| Nadgradnja »na mestu« (angl. »In-place Upgrade«) | Novejši sistem namestimo kar preko starega. Vsebina in konfiguracijski podatki so posodobljeni naenkrat. | Najlažji pristop. Spletna mesta ohranijo izvirne spletne naslove. Nadgradijo se obstoječe podatkovne baze in portalska aplikacija na obstoječi strojni opremi. | Portal v času nadgradnje ni dosegljiv. Po nadgradnji ni mogoče povrniti prejšnjega stanja. | Portal na enem strežniku ali v manjši farmi. |
| Postopna nadgradnja (angl. »Gradual Upgrade«) | Namesti novo različico ob boku starejše in ne čez, tako kot pri nadgradnji na mestu. Sistemski administrator določi, katere zbirke spletnih mest je potrebno nadgraditi in časovni plan same izvedbe. | Omogoča večjo granularnost nadgradnje, na ravni zbirke mest. Spletna mesta ohranijo izvirne spletne naslove. Možna je povrnitev prejšnjega stanja. Uporabljena je ista strojna oprema. | Bolj kompleksen in potraten pristop. Tekom nadgradnje je potrebno zagotoviti preusmeritve spletnih naslovov, kar povzroča težave določenim integriranim aplikacijam. | Nadgradnja srednje velikih ali velikih strežniških farm. |

| Pristop | Opis | Prednosti | Slabosti | Primeren okvirni scenarij |
|---|---|---|---|---|
| Migracija podatkovne baze (angl. »Database Migration«) | Sistemski administrator mora namestiti novejšo različico portala na ločeno farmo ali strojno opremo, nato pa ročno migrirati podatkovne baze v novo okolje. | Omogoča premik na novo farmo ali strojno opremo. SPS 2003 ostane dosegljiv in hkrati nedotaknjen tekom migracije. | Kompleksen proces, pri katerem je zaradi več ročnega dela povečana možnost za napake. Zahtevani so dodatni koraki za ohranitev izvirnih spletnih naslovov. Potrebna je nova strežniška farma in dvakrat večja diskovna kapaciteta za SQL Server bazo. | Premik na novo strojno opremo ali drugačno arhitekturno zasnovno rešitve. Takrat, ko želimo karseda izkoristiti zmogljivosti novega sistema in ohraniti karseda malo združljivostnih funkcionalnosti. Premik iz WSS 2.0 na MOSS 2007 platformo. |
| Vsebinska migracija (angl. »Content Migration«) | Tipična migracija. Vsebina starega sistema se prek programskega vmesnika klonira v nov sistem. | Omogoča premik na novo farmo ali strojno opremo. Stari in novi sistem sta povsem neodvisna. | Zelo kompleksen proces, saj je pogosto potreben razvoj specifičnih orodij za prenos vsebine. | Vsi scenariji, omenjeni pri ostalih pristopih. |

Tabela 4-1: Pregled podprtih pristopov k nadgradnji SharePoint platforme

4.4 Funkcionalne zahteve vsebinske migracije

Nadgradnja »na mestu«, postopna nadgradnja in migracija podatkovne baze so pristopi, katere Microsoft omogoča, ko gre za nadgradnjo sistema SharePoint. Za mnogo SharePoint konfiguracij, ki se uporabljajo v praksi, omenjeni pristopi predstavljajo zadovoljiv in hiter

način za nadgradnjo. Včasih pa se zgodi, da z rezultati nadgradnje, ki jih prinese uporaba tovrstnih pristopov, nismo zadovoljni. V takšnih primerih moramo ubrati pristop vsebinske migracije. To storimo tako, da kupimo že pripravljeno orodje, npr. »SharePoint Site Migration Manager« podjetja Metalogix ali »Tsunami Deployer for SharePoint 2007 Migration«. Lahko pa razvijemo svoje. Nakup že pripravljenega orodja se namreč ne izplača vedno. Zelo pogosto se zgodi, da moramo na projektu realizirati stvari katerim tudi najboljše že pripravljeno orodje ni kos. Tako moramo funkcionalnost kupljenega orodja dopolnjevati z lastnim razvojem.

Ko govorimo o vsebinski migraciji v navezi s sistemom SharePoint, je potrebno izpostaviti nekaj funkcionalnih zahtev, ki se ob tem porajajo.

4.4.1 Granularnost migracije

Dobro zasnovano orodje za vsebinsko migracijo mora omogočati večji nadzor nad tem kaj, ter tudi kdaj mora biti določen blok podatkov prenesen. Velikost bloka je odvisna od sistema, katerega nadgrajujemo. Nadgradnja »na mestu« blokovne porazdelitve ne omogoča, postopna nadgradnja deluje na ravni zbirke mest, velikost bloka pri migraciji podatkovne baze pa je odvisna od vsebine podatkovne baze.

4.4.2 Reorganizacija portala tekom migracije

Dobro zasnovano orodje za vsebinsko migracijo mora omogočati prestrukturiranje SharePoint portala, najsi bo to na ravni zbirke mest, spletnih mest, seznamov ali celo zapisov v samem seznamu. Pristopi, ki jih nudi Microsoft tovrstne funkcionalnosti ne podpirajo.

4.4.3 Nadzor nad zbirkami mest in vsebinskimi podatkovnimi bazami

Pri vpeljavi velikih informacijskih sistemov je zaželeno, da podamo karseda natančno oceno o predvideni širitvi le teh za določeno obdobje. Takšna ocena je zelo pomembna, saj lahko na njeni podlagi določimo število potrebnih strežnikov in diskovnih enot zdaj in v prihodnosti. Pri tem nam je SharePoint lahko v pomoč. Zavedati se moramo, da je SharePoint platforma za spletne portale. Ob stvaritvi nove instance portala se hkrati ustvari tudi pripadajoča vsebinska podatkovna baza v kateri so shranjeni podatki vseh zbirk spletnih mest. Pri manjših portalih ni nič narobe, če vse podatke hranimo le v eni podatkovni bazi, medtem ko pri večjih to ne drži. V takšnih primerih je skoraj nujno, da portal razdelimo v več smiselno zaokroženih delov, pri čemer vsakemu delu dodelimo po eno vsebinsko podatkovno bazo. Posamezen del je skupek več zbirk mest. Rast portala je tako lažje obvladovati, saj nam ob vpeljavi večje diskovne

enote ni potrebno prenesti podatkovne baze celotnega portala, temveč le podatkovne baze tistih delov, ki najhitreje rastejo.

4.4.4 Združevanje virov podatkov

Nemalokrat se zgodi, da želimo več neodvisnih postavitev SharePoint portalov združiti v enega. Dobro orodje za vsebinsko migracijo mora biti zgrajeno dovolj generično, da ga je možno uporabiti pri vsaki od omenjenih SharePoint postavitev, brez ali z manjšimi modifikacijami.

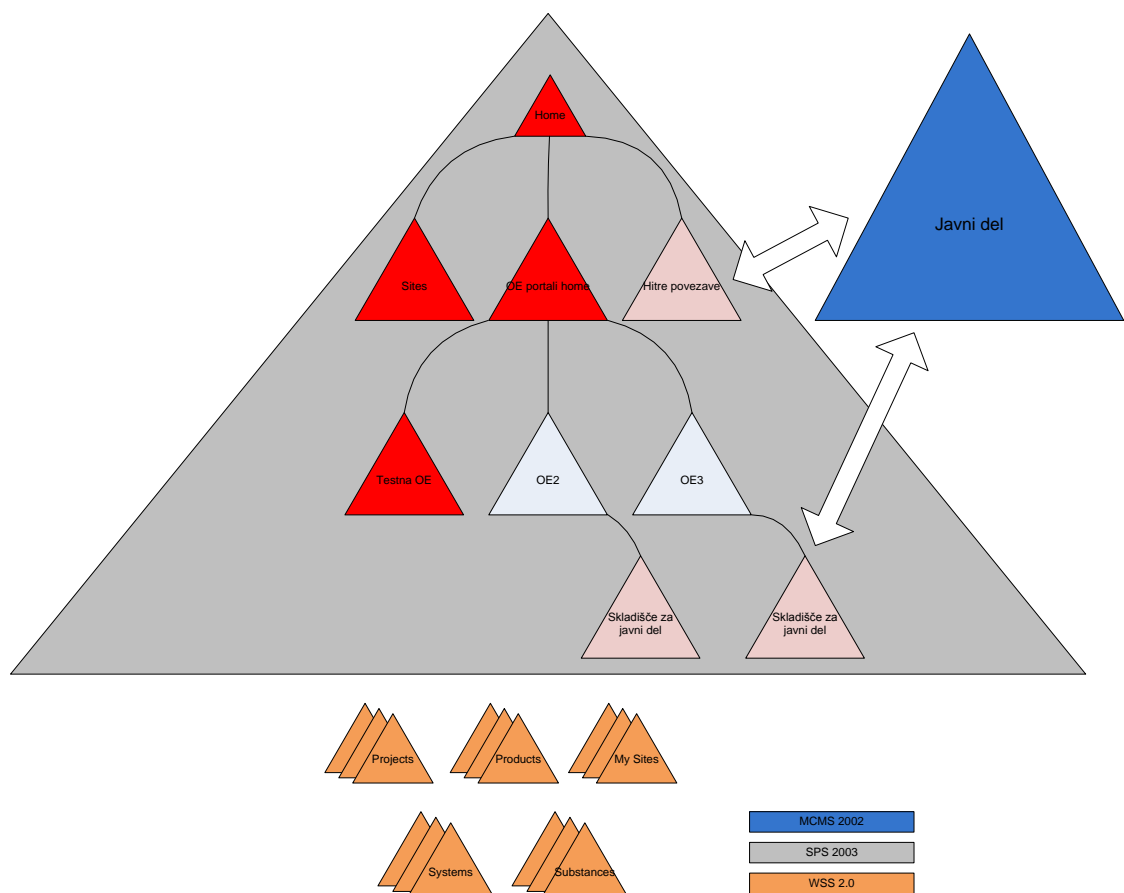
4.4.5 Preslikovanje spletnih naslovov

Orodje za vsebinsko migracijo mora podpirati ohranjanje izvirnih spletnih naslovov, kot tudi njihovo spreminjanje.

5 PRAKTIČNI PRIMER MIGRACIJE PODATKOV V MICROSOFTOVEM SISTEMU SHAREPOINT

Projektna ekipa, katere član sem tudi sam, je bila soočena z izzivom nadgradnje SPS 2003 intranet portala na MOSS 2007 platformo. Obstoječi portal je bil v uporabi od leta 2005 in imel preko 4000 aktivnih uporabnikov. Podatkovna baza je hranila več kot 200 gigabajtov podatkov. V zapisih, ki sledijo bom opisal kako smo se projekta lotili in ga tudi uspešno pripeljali do konca. Za začetek bom razkril strukturo obstoječega in prenovljenega portala. Kako smo prehod med strukturama portala izpeljali bom okvirno opisal s členitvijo projektnega plana po aktivnostih, značilnih za migracije. Prav tako bom nekaj stavkov namenil orodjem in pristopom, uporabljenim na projektu. Poudarek bo predvsem na orodju za vsebinsko migracijo, ki smo ga razvili za prenos organizacijskih enot. Za konec pa še nekaj o prenosu WSS 2.0 spletnih mest.

5.1 Diagram obstoječega intranet portala



Slika 5-1: Diagram obstoječega intranet portala

Vstopno točko v portal predstavlja modro označen trikotnik – Javni del. Dovolj bo, če omenim, da je podlaga zanj MCMS 2002, tehnologija komplementarna SPS 2003 platformi. Podrobnosti prenosa javnega dela so vezane na migracijo MCMS okolja, kar pa presega obseg pričujočega diplomskega dela in jih zato ne bom obravnaval.

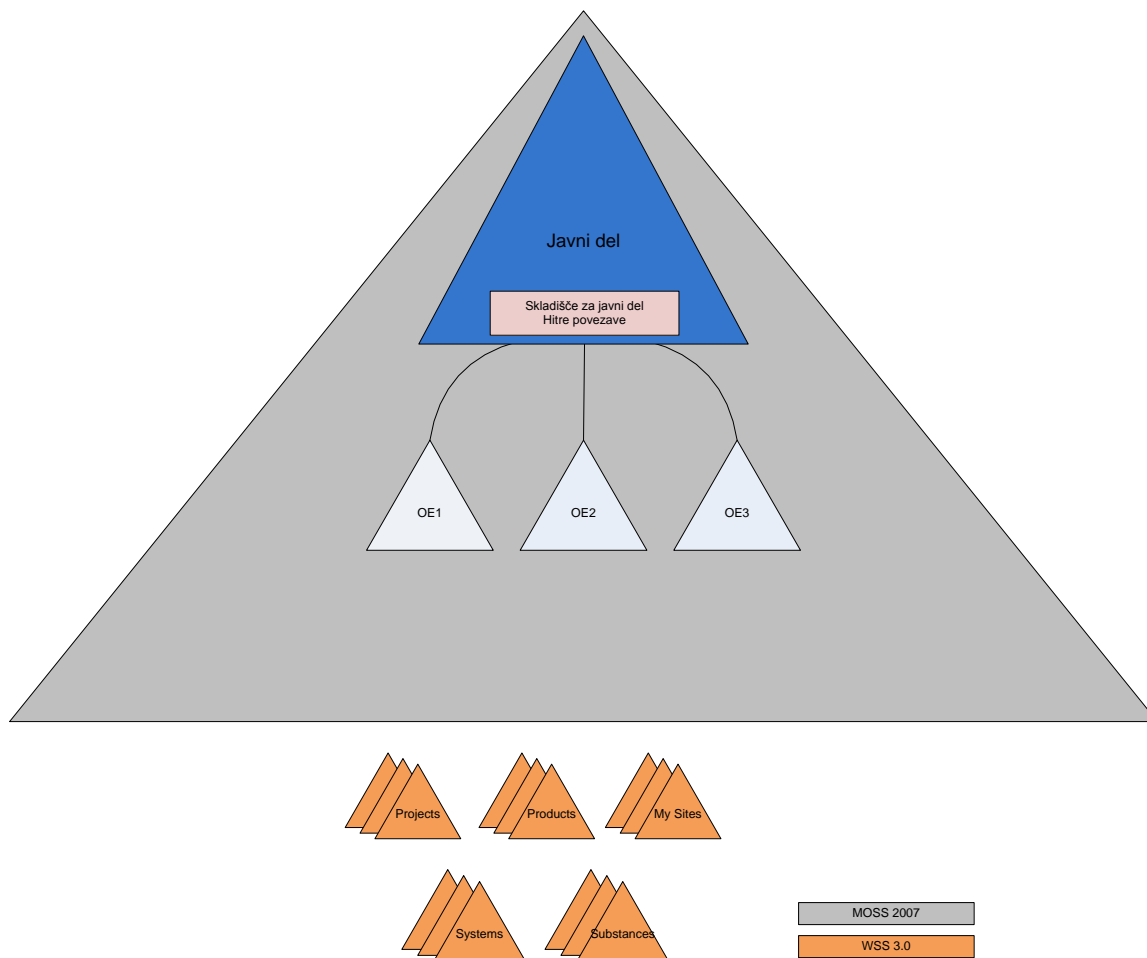
SPS 2003 portal, na sliki predstavljen s sivim trikotnikom, ima 4 nivoje. Prvi nivo – »Home« predstavlja spletno mesto za nastavitve portala, ter ga zato ni potrebno prenesti. Drugi nivo pa ima tri področja:

- »Sites« - področje, ki se ne uporablja in ga zato ni potrebno prenesti.
- »OE Portali - Home« - korenko mesto organizacijskih enot. Po našem planu se vsaka organizacijska enota preslika v eno od zbirk mest. Omenjenega korenkega mesta torej ni potrebno prenesti, le njegova podpodročja.
- »Hitre povezave« - Množica povezav na interna in eksterna spletna mesta. Po našem planu se povezave doda na vstopno stran MOSS 2007 portala.

Tretji nivo predstavljajo področja organizacijskih enot. Prenos »Testne OE« ni potreben. Zadnji nivo predstavljajo področja, ki so del OE portalov, namenjena pa skladišču vsebin za javni del. Podrobnosti prenosa organizacijskih enot bom razkril v poglavju 5.5.

Oranžni trikotniki predstavljajo produkte, projekte, sisteme, surovine ter spletna mesta uporabnikov. Platforma zanje je WSS 2.0, pri čemer je vsak produkt, projekt ... svoja zbirka spletnih mest. Podrobnosti o prenosu WSS 2.0 zbirk spletnih mest bom obravnaval v poglavju 5.6.

5.2 Diagram prenovljenega portala



Slika 5-2: Diagram prenovljenega portala

Javni del se postavi na novo z MOSS 2007 predlogami, kot ločeno zbirko mest. Vsebina, ki je bila prej na SPS 2003 (Hitre povezave, sezname, knjižnice, itd.), se sedaj naredi na pripadajočem mestu javnega dela. S tem se ukine nepotrebno ločevanje vsebine in dvojna administracija.

Portale organizacijskih enot se prenese v spletni naslov, ki odraža ime organizacijske enote, vsakega kot svojo zbirko mest.

Produkte, projekte, sisteme, surovine, spletna mesta uporabnikov se prenese v enake spletne naslove na MOSS 2007, vsako kot svojo zbirko mest.

5.3 Členitev projektnega plana po aktivnostih

V poglavju 3 smo spoznali aktivnosti, ki ločijo migracijske od splošnih informacijskih projektov. Da ne bi ostali le pri teoriji, si pogledjmo, kakšen vpliv je imela vsaka od aktivnosti na naš projekt.

5.3.1 Prepoznavna in izbor vseh podatkovnih virov

Specifično za naš primer je bilo iskanje vseh podatkovnih virov precej preprosto opravilo. Podatki organizacijskih enot in WSS spletnih mest so bili shranjeni v SharePoint vsebinski podatkovni bazi, novice iz javnega dela pa v podatkovni bazi sistema MCMS.

5.3.2 Proces spoznavanja podatkov

Podatkov SharePointa nismo spoznavali neposredno, tako da bi analizirali podatkovno bazo. Bolj primeren in hiter način smo našli v analizi objektov, ki jih SharePoint izpostavi preko programskega vmesnika. Vsak objekt je na nek način projekcija množice podatkov iz baze.

5.3.3 Čiščenje podatkov

Večjo čistko smo izvedli pri sestavi seznama vseh področij potrebnih pri migraciji. Stranka je iz popisa odstranila vsa področja, ki jih v novem sistemu niso želeli. Čiščenja so bili deležni tudi dokumenti in opozorila. Za vsak dokument smo prenesli le pet prejšnjih različic, medtem ko so opozorila obdržali le aktivni uporabniki.

5.3.4 Transformiranje podatkov

Transformacije podatkov, ki smo jih morali izvesti pri prenosu organizacijskih enot in WSS spletnih mest so jasno prikazane v tabeli Tabela 5-3. Razlika je že pri prvem objektu – področju. Teh na MOSS 2007 več ni, zato jih je bilo potrebno preslikati v spletna mesta. Hkrati s področji so ukinili tudi nekatere spletne gradnike. Tiste, ki jih je stranka na vsak način želela obdržati, smo morali razviti na novo ali pa najti ustrezno zamenjavo. Objektni model obeh različic sistema SharePoint se razlikuje tudi pri opozorilih. Varnostni model se je prav tako spremenil, zato smo morali poskrbeti za dodatne preslikave. Več o varnosti v poglavju 5.5.1.4.

Prenos javnega dela sicer ni sestavni del diplomske naloge, pa vendar naj omenim, da je bilo treba novice iz MCMS preslikati v spletne strani na MOSS 2007.

5.3.5 Testiranje in validacija

Tekom razvoja smo redno testirali kvaliteto kode, vsake toliko pa tudi ustreznost prenosa na testni množici podatkov. V ta namen je stranka vzpostavila dve ločeni testni okolji, pri čemer je bilo prvo kar kopija obstoječega portala, drugo pa je poganjal ciljni sistem MOSS 2007.

5.3.6 Fizični prenos podatkov

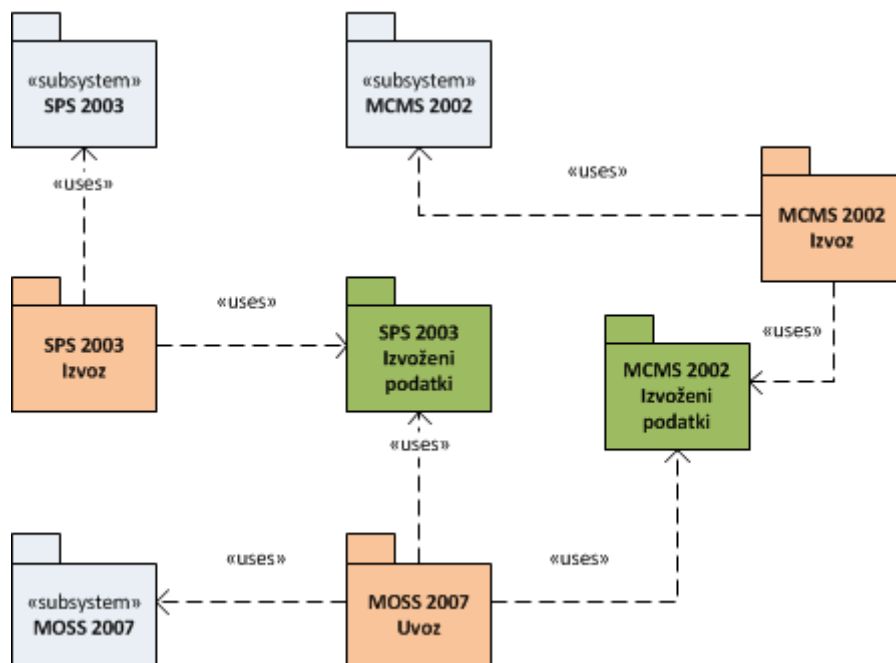
Po zadostnem testiranju smo prišli do točke, ko je bilo treba razvita orodja uporabiti za prenos podatkov v produkcijo. Okvirno smo plan migracije sestavili tako, da bi migracija potekala v času najmanjše obremenitve obstoječega sistema. Za preklopno strategijo smo izbrali »Big bang«, zato strategije za sinhronizacijo podatkov nismo potrebovali, saj se je takoj za migracijo začela uporaba novega sistema, starega pa pustilo v omejenem načinu delovanja. Migracija javnega dela se je zaradi časovne zahtevnosti izvedla najprej. Prenos spletnih mest in organizacijskih enot pa smo izvedli vzporedno, nekaj dni kasneje.

5.3.7 Revidiranje in dokumentiranje

Že v startu projekta je nastalo več dokumentov, kot so funkcionalna specifikacija in plan migracije. Tekom razvoja orodij smo vso spisano kodo opremili z ustreznimi komentarji. Po koncu migracije so nastale analize o poteku projekta.

5.4 Orodje za vsebinsko migracijo podatkov SharePoint

V poglavju 4.3 smo spoznali pristope, s pomočjo katerih je možno nadgraditi obstoječ SharePoint sistem z njegovim naslednikom. V našem primeru smo se odločili, da bomo WSS spletna mesta portala prenesli po principu migracije podatkovne baze, organizacijske enote in javni del pa z vsebinsko migracijo. Pri tem smo se odločili, da orodje za vsebinsko migracijo podatkov razvijemo sami.



Slika 5-3: Visokonivojski diagram orodja za vsebinsko migracijo

Slika 5-3 prikazuje odvisnosti med komponentami, ki so igrale vidno vlogo pri vsebinski migraciji. Izhodišče predstavljata dva sistema, SPS 2003 in MCMS 2002. Podatke iz teh sistemov smo izvozili z ločenima orodjema in jih začasno shranili na disk, nato pa uvozili v MOSS 2007. Tudi uvoz sta v resnici dve orodji, čeprav na sliki to zaradi lažje predstave ni posebej označeno.

Orodje, ki smo ga razvili ustreza vsem funkcionalnim zahtevam vsebinske migracije iz poglavja 4.4, pri čemer si vsaka od njih zasluži dodaten komentar.

Prva zahteva vsebinske migracije je vezana na granularnost. V primeru našega orodja smo velikost bloka omejili na velikost SPS 2003 področja. Možen je prenos enega ali več področij.

Orodje za prenos podatkov naj bi omogočalo tudi reorganizacijo portala tekom migracije, na nivoju zapisov, seznamov, mest, itd. Orodje, ki smo ga razvili za potrebe našega projekta to omogoča.

Nadzor nad vsebinskimi podatkovnimi bazami je prav tako pomemben. Za potrebe našega projekta smo kreirali tri takšne baze. Tako smo naredili fizični mejnik med javnim delom portala, organizacijskimi enotami ter WSS spletnimi mesti. Skalabilnost portala je bila s tem

zagotovljena, prav tako pa se je čas migracije skrajšal. Ker so sklopi med sabo povsem ločeni, smo jih lahko migrirali vzporedno.

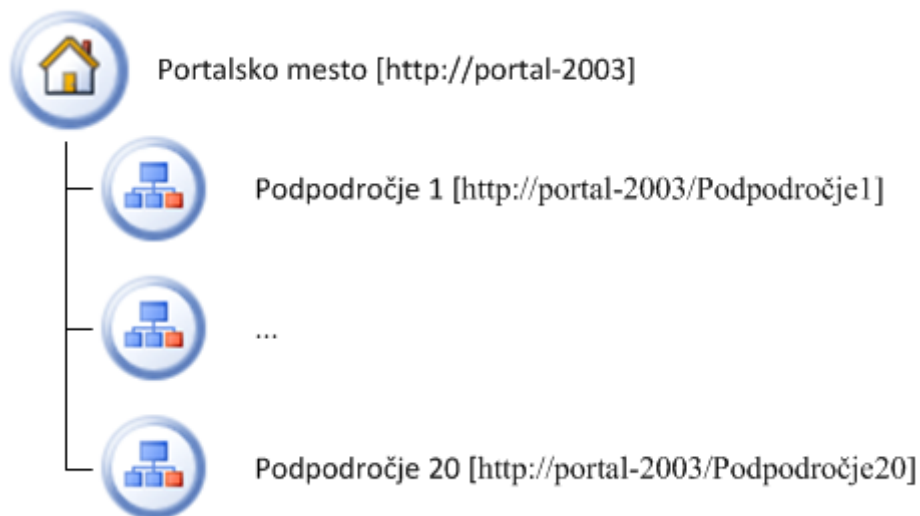
Združevanje virov bi z našim orodjem lahko zagotovili, saj je spisano dokaj generično. Zelo verjetno bi ga morali nekoliko prirediti, tako da bi ustrezala posameznim konfiguracijam SharePoint portalov, ki bi jih želeli združiti, pa vendar.

Zadnja od zahtev je vezana le na SPS 2003 in se tiče preslikovanja spletnih naslovov. Da bi lažje razumeli bistvo problema, je potrebno pojasniti, kako preslikovanje poteka.

5.4.1 Preslikovanje spletnih naslovov

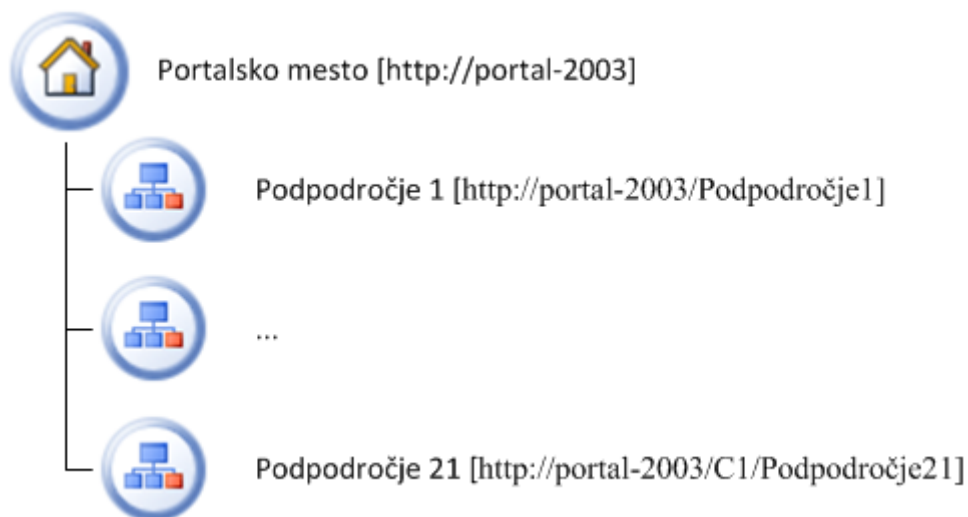
Ob pozornem branju ste lahko opazili, da vsi uradni pristopi k nadgradnji ohranjajo izvirne spletne naslove. Včasih takšna funkcionalnost ni zaželeno. Na primeru pogledjmo zakaj je bilo temu tako.

Nek SPS 2003 portal ima naslednjo strukturo:



Slika 5-4: Primer hierarhije portala z 20 podpodročji

Če ustvarimo enaindvajseto podpodročje, se zanj kreira sledeč spletni naslov:



Slika 5-5: Primer hierarhije portala z 21 podpodročji

SPS 2003 je v spletni naslov vrnil »C1«. Z vrivanjem znakov v spletne naslove so pri Microsoftu občli omejitev WSS 2 platforme, ki ni omogočala premikanja fizičnih spletnih mest v hierarhiji portala. Tako spremenjenemu naslovu pravimo logični spletni naslov, ta pa ne odraža fizične umestitve v hierarhiji portala. Namenjen je prikazu področja. Vsako področje ima poleg logičnega naslova shranjen tudi seznam spletnih mest, ki se nahajajo nad njim v fizični hierarhiji - prednikov. Premik področja v hierarhiji ohranja logični naslov, spremeni se le seznam vseh starševskih spletnih mest.

| Fizična umestitev (seznam staršev) | Logični naslov |
|---|------------------|
| Koren | / |
| Koren: PodPodročje1 | /C1/PodPodročje1 |
| Koren: PodPodročje1:PodPodročje2 | /C2/PodPodročje2 |
| Koren: PodPodročje1:PodPodročje2:PodPodročje3 | /C3/PodPodročje3 |

Tabela 5-1: Primeri spletnih naslovov pri SPS 2003

S prihodom WSS 3 so pojem področij ukinili. Za prehod iz SPS 2003 na MOSS 2007 je bilo potrebno definirati preslikavo logičnih naslovov v fizične. Preslikava je preprosta, saj se spletni naslov sestavi s pomočjo seznama prednikov določenega področja:

| Fizična umestitev (seznam prednikov) | MOSS 2007 spletni naslov |
|---|---|
| Koren | / |
| Koren: PodPodročje1 | /PodPodročje1 |
| Koren: PodPodročje1:PodPodročje2 | /PodPodročje1/PodPodročje2 |
| Koren:PodPodročje1:PodPodročje2 :PodPodročje3 | /PodPodročje1/PodPodročje2/Podpodročje3 |

Tabela 5-2: Primeri preslikovanja spletnih naslovov

Problem se pojavi, ko je področje postavljeno zelo globoko v hierarhiji portala. Preslikan spletni naslov, ki ga tvorijo elementi seznama prednikov, kaj kmalu postane daljši od 256 znakov. SharePoint v takšnih primerih javi napako in področje preslika na naslov začasne shrambe. Shramba ni nič drugega kot spletno mesto z zelo kratkim spletnim naslovom. Datoteke v začasni shrambi moramo kasneje ročno prenesti na naslove, ki ustrezajo omejitvam dolžine. Pri našem projektu smo vse preslikave naslovov določili sami. Novi naslovi so bili sestavljeni tako, da so bili kratki in intuitivni. Število napak zaradi predolgega naslova ob migraciji zaradi tega skoraj ni bilo.

Kot že omenjeno, naj bi uradni pristopi k nadgradnji ohranjali izvirne spletne naslove. Po takšni logiki potem velja, da lahko po novem portalu krmarimo s pomočjo starih naslovov. Razlog za to je zopet precej preprost. Ob prehodu iz SPS 2003 na MOSS 2007 se v podatkovni bazi kreira nova tabela. V to tabelo se ob preslikavi vsakega naslova doda po en zapis, ki določa stari logični ter novi fizični naslov. Pri vpisu naslova v brskalnik MOSS 2007 iz tabele preslikav izbere zapis z logičnim naslovom »/C2/PodPodročje1«, prebere fizični naslov »/PodPodročje1« in naredi ustrezno preusmeritev. Z našim orodjem za vsebinsko migracijo smo se preslikovanju naslovov izognili. Vse podatke smo najprej izvozili iz starega sistema, jih po potrebi spremenili, ter nato uvozili v novi sistem. Posledično sta bila sistema povsem avtonomna, zato preslikovanje ni bilo potrebno.

5.5 Prenos organizacijskih enot

Organizacijske enote so prav gotovo predstavljale največji del obstoječega portala. Iz poslovnega vidika je vsaka izmed enot odražala fizični oddelek v organizacijski strukturi stranke. Tehnično gledano pa je bila vsaka enota SharePoint področje s pripadajočimi

področji. Kot že rečeno, smo omenjeni del portala prenesli po principu vsebinske migracije z orodjem, razvitim prav za ta namen. Podlago za naše orodje je prispeval kar SharePoint, ki večino funkcionalnosti izpostavi prek lastnega objektnega modela. Tako smo se neposrednim dostopom do podatkovne baze izognili. V praksi je takšen pristop zelo pogost in se smatra kot najbolj pravilen. Delati preslikavo med bazama, ki ju nimamo pod kontrolo, torej ne poznamo vseh relacijskih razmerij, namreč zahteva reverzni inženiring obeh verzij iste aplikacije. Gre za dolgotrajen, mukotrpen, predvsem pa drag proces. Bolj pravilno je, da objekte starega sistema preko programskega vmesnika analiziramo in jih kloniramo v nov sistem, prav tako s pomočjo programskega vmesnika.

Z našim orodjem smo izvedli naslednje preslikave iz domene objektnega modela:

| SPS 2003 objekt | MOSS 2007 objekt |
|------------------------------------|------------------------------------|
| Področje (»Area«) | Spletno mesto (»SPWeb«) |
| Seznam (»SPList«) | Seznam (»SPList«) |
| Zapis seznama (»SPListItem«) | Zapis seznama (»SPListItem«) |
| Polje elementa seznama (»SPField«) | Polje elementa seznama (»SPField«) |
| Dokument (»SPFile«) | Dokument (»SPFile«) |
| Opozorilo (»Alert«) | Opozorilo (»SPAlert«) |

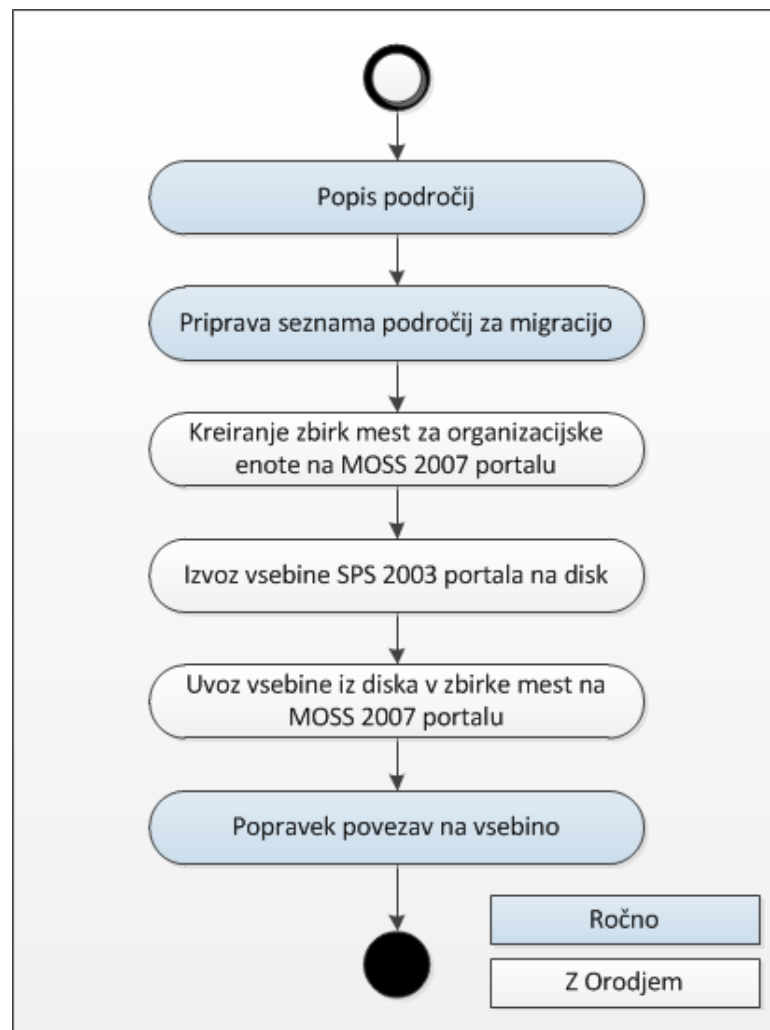
Tabela 5-3: Preslikave objektov

V sodelovanju s stranko so nastala še nekatera pravila, katera je bilo potrebno pri migraciji upoštevati:

| Pravilo | Komentar |
|---|---|
| Pri prenosu vsebine se morajo ohraniti pravice dostopa in zadnjih pet verzij dokumenta, kjer je bilo verzioniranje vklopljeno. | Na tak način se bistveno zmanjša prostorska zahtevnost portala. |
| Ohraniti se morajo tudi spletni gradniki in povezave med njimi, ki so del skupnega pogleda (angl. »Shared View«). Osebni pogledi (angl. »Personal View«) se ne prenašajo. | Skupni pogled je enak za vse uporabnike portala medtem ko je osebni pogled prirejen posameznikom. Prenos slednjega se ne izvede iz več razlogov. Osebne poglede so imeli tudi uporabniki, ki jih več ni, kar za prenos predstavlja smeti. Prav tako bi bilo ustreznost prenosa težko preveriti, saj bi morali vključiti vse posameznike, ki so osebne poglede uporabljali. |
| »Office 2003 AddIn« spletni gradniki se ne prenašajo avtomatsko. | Tovrstni gradniki v MOSS 2007 ne obstajajo. |
| Obstoječe povezave znotraj vsebine dokumentov se pri migraciji vsebin ne ohranjajo in jih je potrebno (ročno) popraviti. | Popis vseh lokacij, kje so povezave je nemogoč (elektronska pošta, MOSS dokumenti v raznih formatih, Internet Explorer zaznamki, bližnjice, lokalni dokumenti v raznih formatih, itd.) Pristop z ohranitvami povezav je cenovno predrag in bi slabo vplival na povrnitev naložbe. Pristop z dodatno aplikacijo za preusmerjanje pa bi še vedno potreboval popravek povezav ali pa bi problematiko samo zamaknil v naslednjo nadgradnjo, kjer bi se njena težavnost podvojila. |

Tabela 5-4: Dodatna pravila pri migraciji organizacijskih enot

Migracijo organizacijskih enot ni šlo začeti drugače, kot s popisom področij v obstoječem portalu. Ker bi ročni popis trajal bistveno predolgo in bil podvržen nemalo napakam, smo vsa področja ter pripadajoče attribute pridobili kar s pomočjo ustrezne poizvedbe na podatkovni bazi. Tako nastali seznam smo nato posredovali stranki, kjer so seznam pripravili za migracijo. Izmed vseh področij so izbrali tista, ki so jih želeli ohraniti, ter vsakemu izmed njih določili spletni naslov na MOSS 2007 portalu. Ne zdi se mi odveč, če omenim, da je vsak spletni naslov jasno nakazoval pripadnost organizacijski enoti in hkrati podrejenost spletnim mestom v hierarhiji portala. Takšno naslavljanje je bilo posledica medsebojnega dogovora med stranko in nami, saj smo tako lahko za vsako organizacijsko enoto ročno ustvarili novo zbirko spletnih mest na MOSS 2007. Razbitje organizacijskih enot na zbirke mest smo izkoristili tudi pri razvoju orodja za uvoz vsebine. Vsako organizacijsko enoto smo namreč uvažali posebej in tako zmanjšali tveganje ob nepravilnem prenosu. Zaradi napake pri eni organizacijski enoti ni bilo potrebno ponovno migrirati celotnega portala.



Slika 5-6: Prenos organizacijskih enot po korakih

5.5.1 Migracija Področja

Migracija področja je pojem, ki obsega mnogo več, kot lahko iz same besedne zveze razberemo. Spoznali smo že, da je katerikoli SharePoint portal sestavljen iz množice spletnih mest. Tudi področje je v svojem bistvu spletno mesto, čeprav ga SharePoint marsikdaj obravnava drugače. Migracija področja tako ne obsega le prenos metapodatkov o področju samem, temveč tudi prenos vseh področnih strani, seznamov, spletnih gradnikov ter pravic.

5.5.1.1 Izvoz vsebine področij na disk, uvoz spletnih mest

Vsako izmed področij je bilo izvoženo po enakem principu, ki ga najbolje opišejo sledeče vrstice psevdokode:

Izvoz področij

- 1: **Foreach** Področje **from** Seznam področij:
 - 2: Izvozi_MetaPodatke_Področja_Na_Disk;
 - 3: Izvozi_Sezname_Področja_Na_Disk;
 - 4: Izvozi_Spletne_Gradnike_Področja_Na_Disk;
 - 5: Izvozi_Uporabniške_Pravice_Področja_Na_Disk;
-

Ker je izvoz seznamov, spletnih gradnikov ter uporabniških pravic obravnavan posebej, bom za zdaj namenil nekaj besed izvozu metapodatkov področja. SharePoint za vsako področje hrani opisne vrednosti, ki določajo njegov naslov, datum nastanka, unikatni identifikator, itd. SharePoint tovrstne podatke izpostavi preko svojega objektnega modela kot attribute razreda »Area«. Na tak način smo tudi z našim orodjem do njih dostopali, ter jih prepisali v strukturo, kateri smo predpisali enake attribute. Tako imenovane entitetne strukture smo naredili za vsak migriran element SharePointa. Po prepisu vseh potrebnih metapodatkov se je na disku ustvarila nova mapa v katero je bila entitetna struktura področja shranjena kot XML datoteka.

Uvoz področij je po zaporednih korakih dokaj podoben izvozu:

Uvoz področij

- 1: **Foreach** Struktura **from** Seznam struktur:
- 2: Preberi_Strukturo_Na_Disku (struktura);
- 3: Kreiraj_Novo_Spletno_Mesto_In_Uvozi_Metapodatke;
- 4: Uvozi_Sezname_V_Spletno_Mesto;
- 5: Uvozi_Spletne_Gradnike_V_Spletno_Mesto;
- 6: Uvozi_Uporabniške_Pravice_Za_Spletno_Mesto;

Vsa področja organizacijskih enot na SPS 2003 portalu so imela enako grafično podobo in kar je še pomembnejše, enako razmestitev odložišč za spletne gradnike. Ker smo postavitev elementov želeli ohraniti tudi na MOSS 2007 portalu nam ni preostalo nič drugega, kot da razvijemo novo definicijo spletnega mesta. Ta je nastala kar na podlagi definicije področja. Velja namreč, da je področje definirano s področnimi stranmi ter WSS 2.0 spletnim mestom, pri čemer strani določajo uporabniški vmesnik, spletno mesto pa področju omogoča vso svojo funkcionalnost.

5.5.1.2 Migracija seznamov in knjižnic

Vsak element SharePointa je seznam ali zapis v seznamu. Drzna trditev, ki pa ni tako daleč od resnice. Najbolj očiten primer so seveda dokumenti v dokumentnih knjižnicah. Po drugi strani pa redko kdo na tak način razmišlja o spletnih straneh mesta. Tudi ta so shranjena kot zapisi v za njih namenskem seznamu. To, kar vidimo kot grafični vmesnik ni nič drugega, kot poseben pogled na zapis v seznamu strani. Filozofijo na stran, seznam je prav gotovo najmočnejši gradnik SharePointa. Obstaja več tipov seznamov, lahko bi rekli, da za vsako priložnost vsaj eden. Za shranjevanje dokumentov se uporabljajo knjižnice, razpravam služijo table, itd. Različni sezname imajo svoje posebnosti, a so si hkrati tudi v marsičem podobni, preprosto zato, ker so vsi izpeljani iz istega osnovnega seznama.

Brez izjeme vsi sezname svoje zapise predstavijo na enak način. Tako kot pri podatkovnih bazah, kjer ima vsak zapis relacije nabor atributov, ima seznam množico polj. V posamezna polja lahko shranimo skoraj vse, z izjemo datotek. Vsi sezname poznajo tudi poglede. S

pogledi lahko omejujemo prikazovanje polj na uporabniškem vmesniku, dodamo grupiranje, sortiranje, itd.

Korektna migracija seznamov zahteva prenos vseh naštetih pojmov. Izvoz seznamov smo ponovili za vsako področje posebej. Za vsak seznam je bilo potrebno izvesti naslednje operacije:

- **Izvoz metapodatkov seznama** – Teh podatkov ni bilo veliko. Omembe vredna sta predvsem tip seznama ter seznam identifikatorjev vseh seznamov, na katere se je trenutni seznam skliceval preko iskalnih polj.
- **Izvoz vseh podprtih polj** – Ko kreiramo nov seznam, se hkrati ustvarijo ustrezna polja, pri čemer je možno tudi naknadno dodajanje novih. Polja smo izvozili s klicem SharePoint spletne storitve, katere odgovor je bila celotna definicijska shema vseh polj. Razlog za takšno spremembo je dokaj preprost. Vsak, ki se je srečal z objektnim programiranjem pozna pojem dedovanja. Vsa polja dedujejo svoje lastnosti posredno ali neposredno iz tako imenovanega osnovnega polja. Dedovano polje vsebuje lastnosti svojih staršev ter lastnosti, ki jih samo definira. V našem primeru bi morali tako za vsak tip polja implementirati po eno entiteto, kar pa se ni zdelo smotrno.
- **Izvoz pogledov** – Kot rečeno, pogledi spreminjajo uporabniški vmesnik seznama. Tako kot pri poljih se nekaj pogledov ustvari skupaj s seznamom, možno pa je tudi naknadno definiranje novih. Izvoz pogledov je zahteval sprehod čez zbirko, ki jo hrani vsak seznam zase. V ustrezno entiteto smo prepisali vse konfiguracijske podatke ter imena polj nad katerimi je pogled deloval. Izvoženi so bili vsi pogledi z izjemo datotečnega. Tega kreira SharePoint hkrati s knjižnico in ga ni mogoče spreminjati. Njegov obstoj je bil tako že vnaprej zagotovljen na MOSS 2007 platformi.
- **Izvoz zapisov** – Načeloma gre za izvoz vrednosti polj. Izjema so dokumentne knjižnice, ki poleg polj vsebujejo še kazalec na datoteko. Več o tem kasneje.

Pri izvozu metapodatkov seznama sem bežno omenil seznam identifikatorjev vseh seznamov na katere se nek seznam sklicuje. SharePoint podpira poseben tip polja, imenovan iskalno polje. Z iskalnim poljem lahko v seznamu prikažemo vsebino kateregakoli polja iz drugega seznama. Funkcionalnost iskalnih polj smo želeli na novem portalu obdržati. Tako smo pred začetkom uvoza seznamov v spletno mesto entitete iz diska najprej sortirali po naslednjem postopku:

- Sezname brez kazalcev na druge sezname se morajo uvoziti prvi.
- Sezname s kazalci se sortirajo tako, da bodo ob uvozu seznama vsi referirani sezname že uvoženi.

Podobno, kot smo to storili pri izvozu, lahko tudi uvoz vsakega od seznamov razbijemo na več zaporednih operacij:

- **Branje strukture iz diska.**
- **Kreiranje novega seznama** – Osnovo za nov seznam je predstavljal atribut tip, prebran iz strukture na disku. Nov seznam se je dodal v zbirko seznamov, ki ga hrani spletno mesto.
- **Uvoz polj** – Tako SPS 2003 kot MOSS 2007 vsakemu seznamu dodata privzeta polja, npr. »ID«, »Title«, itd. To so sistemska polja, ki so pri obeh različicah SharePointa enaka. Pri uvozu smo se takšnim poljem poskušali ogniti in se osredotočiti na tista, ki so bila dodana na sezname naknadno. Sistemska polja namreč nastavlja SharePoint. Pri uvozu naknadno dodanih polj je potrebno omeniti še eno stvar. Vsako polje ima dve imeni, prikazno ter interno. Prikazno ime vidimo na uporabniškem vmesniku in se lahko spreminja. Interno ime je nespremenljivo in unikatno za seznam. SharePoint interno ime uporablja kot referenco v drugih objektih. Včasih se zgodi, da ne moremo ustvariti novega polja z želenim internim imenom. Naš primer pri tem ni bil izjema. Zaradi tega smo problematična polja ustvarili z izmišljenimi imeni, pri katerih se ta težava ni pojavila. Ker je imelo novonastalo polje drugačno ime, smo to morali zabeležiti v preslikovalni slovar. Vsak zapis slovarja je bil tako sestavljen iz starega in novega internega imena.
- **Uvoz pogledov** – Za vsako entiteto pogleda smo preverili, če ta že obstaja na seznamu. Naj vas še enkrat spomnim, da ima vsak seznam že privzet nabor pogledov. Če pogled še ni obstajal, se je pač dodalo novega z vsemi potrebnimi konfiguracijskimi podatki. Na vsak pogled so se nato dodala še interna imena vseh potrebnih polj, upoštevajoč predhodno sestavljen preslikovalni slovar. Vsi pogledi so se po uvozu ponašali z novim identifikatorjem. Ker kazalce na poglede uporabljajo tudi nekateri spletni gradniki, smo sestavili preslikovalni slovar, katerega posamezen zapis je tvoril par starega in novega identifikatorja pogleda.

- **Uvoz zapisov** – Če odmislim nekatere izjeme, je uvoz zapisov v seznam precej enostavno opravilo. S sprehodom čez vse entitete zapisov smo jih zaporedno dodali v seznam in jim v polja prepisali ustrezne vrednosti.

Posebnost dokumentnih knjižnic

Vse kar je bilo do sedaj rečeno velja tudi za dokumentne knjižnice. Razlikujejo se le pri izvozu ter uvozu zapisov. Navadni sezname namreč ne poznajo datotek, map in verzioniranja. V našem primeru je bilo potrebno za vsak dokument izvoziti še pet njegovih predhodnih različic.

Izvoz zapisov je tako še vedno sestavljen iz izvoza vrednosti polj, temu pa je potrebno dodati izvoz datotek in njihovih verzij. Vsak zapis dokumentne knjižnice ima kazalec na zadnjo različico dokumenta, ta pa kazalec na seznam prejšnjih različic. Posledično izvoz dokumenta in njegovih različic ni bil pretirano zahteven, saj je zahteval le sprehod po toku kazalcev. Vsi dokumenti so bili shranjeni na disk, njihova imena ter opisni podatki pa zabeleženi v entiteti zapisa.

Pri uvozu dokumentnih knjižnic so se izvršile vse operacije za uvoz seznama. Potreben je bil le še uvoz vseh izvoženih različic posameznega dokumenta, pri čemer smo morali paziti na vrstni red različic.

Posebnost table za razprave

Tabla za razprave je seznam, ki poskuša posnemati povprečen spletni forum. Tisti, ki se na forumih udeležuje na tak ali drugačen način ve, da so razprave med sabo vizualno ločene. S tem je mogoče posamezni razpravi lažje slediti. Tabla za razprave ni v tem pogledu nič drugačna.

Izvoz tovrstnega seznama je potekal enako kot za ostale sezname, medtem ko smo morali uvoz nekoliko prirediti. Tabla za razprave vsebuje posebno polje, ki označuje pripadnost posamezni razpravi. To polje je sistemske narave in se imenuje »ThreadID«. To vedoč, smo lahko strukturo razprav ohranili na sledeč način: Za vsako entiteto zapisa smo preverili, če že obstaja razprava z enako vrednostjo »ThreadID« polja. Če ni, smo z namenskimi metodami SharePointa dodali novo razpravo, v nasprotnem primeru pa dodali nov odgovor obstoječi. Za vsak novo dodan zapis smo nato še nastavili vrednosti vsem poljem.

5.5.1.3 Migracija spletnih gradnikov

Spletni gradniki so elementi SharePointa, s katerimi dinamično spreminjamo izgled spletnih strani mesta. Poznamo več tipov gradnikov, pri čemer se vsak ponaša z lastnim izgledom ter funkcionalnostjo. Najpogosteje uporabljen tip je prav gotovo gradnik za delo s seznamami. Omogoča hiter vpogled v vsebino seznamov in knjižnic preko uporabniških spletnih strani. Urejanje besedil omogoča urejevalnik besedil, prikaz novic pa gradnik – poizvedba po vsebini. Spletne gradnike je možno med seboj povezovati. Dogodki, proženi na izvornem gradniku po določenem scenariju vplivajo na ponorni gradnik. Tipičen primer takšnega obnašanja predstavlja filtriranje zapisov enega gradnika po vrednostih, pridobljenih iz drugega.

Spletne gradnike lahko na spletno stran dodamo v posebej označena območja – odložišča. Vsako odložišče ima določeno pozicijo in je skladno s tem tudi poimenovano. Odložišče lahko vsebuje poljubno število spletnih gradnikov.



Slika 5-7: Razmestitev odložišč spletnih gradnikov

Večino spletnih gradnikov smo izvozili po enakem kopitu. Naj spomnim, da ima vsako področje v definiciji navedene področne strani, katerim lahko dodajamo spletne gradnike. V našem primeru je bila na vsako področje vezana le ena takšna stran. Da bi izvozili vse gradnike, ki so bili na področni strani prisotni, smo izvedli naslednje operacije:

- **Pridobitev seznama vseh spletnih gradnikov področne strani** – Do omenjenega seznama ni težko priti. Potreben je le klic namenske funkcije, ki za področno stran vrne nabor prisotnih spletnih gradnikov.
- **Izvoz metapodatkov spletnega gradnika** – Ker je tipov spletnih gradnikov ogromno, smo se izvoza metapodatkov gradnika lotili nekoliko drugače. SharePoint že sam po sebi omogoča izvoz kateregakoli spletnega gradnika. Mehanizem deluje tako, da se za spletni gradnik vrši dostop do systemske strani, ki vsebuje logiko za izvoz vrednosti vseh atributov. V sklopu naše rešitve smo to s pridom izkoristili. Za vsak spletni gradnik iz seznama smo prek kode posredovali zahtevo prej omenjeni systemski strani, njen odgovor pa shranili v entiteto spletnega gradnika.
- **Izvoz dodatnih podatkov za posebne primere** – Nekaterih spletnih gradnikov SPS 2003 v novejši različici platforme ni več možno najti. V našem primeru sta tako izginila gradnik za prikaz urednika področja (angl. »AreaDetailsWebPart«), kot tudi gradnik za prikaz področnih novic (angl. »ListingSummary«). Do rešitve tovrstnega problema je možno priti z razvojem enakovrednega gradnika na novi platformi, ali pa s poiskanjem najboljšega približka. Mi smo se odločili za slednji pristop. Pri izvozu marginalnih primerov, kot sta prej omenjena, smo izvedli še prepis dodatnih podatkov. Z izvozom gradnikov preko systemske strani namreč teh podatkov ni bilo moč pridobiti. Tako gradnik za prikaz urednika področja kot tudi gradnik za prikaz področnih novic svoje podatke pridobita iz instance področja, na kateri sta prisotna. Za oba gradnika smo sestavili novo entiteto, vrednosti iz področja pa prepisali v njene attribute.

Definicijo mesta smo ustvarili po definiciji področja. Na tak način se je razmestitev odložišč ohranila, uvoz spletnih gradnikov pa poenostavil. Za vsako ustvarjeno spletno mesto smo morali iz diska prebrati vse gradnike in jih uvoziti s pomočjo naslednjih operacij:

- **Odstranitev odvečnih elementov v shemi spletnega gradnika** – Shemo smo pridobili z izvozom spletnih gradnikov preko systemske strani. Pogosto so se

izvozili tudi podatki, ki jih ni bilo mogoče neposredno uvoziti v MOSS 2007. Kot primer lahko navedem html kodo glave in noge gradnika za prikaz seznamov. Ta ni enaka kodi, ki se za isti gradnik generira na MOSS 2007. Takšne odseke smo iz sheme odstranili, saj v nasprotnem primeru uvoz ne bi bil mogoč.

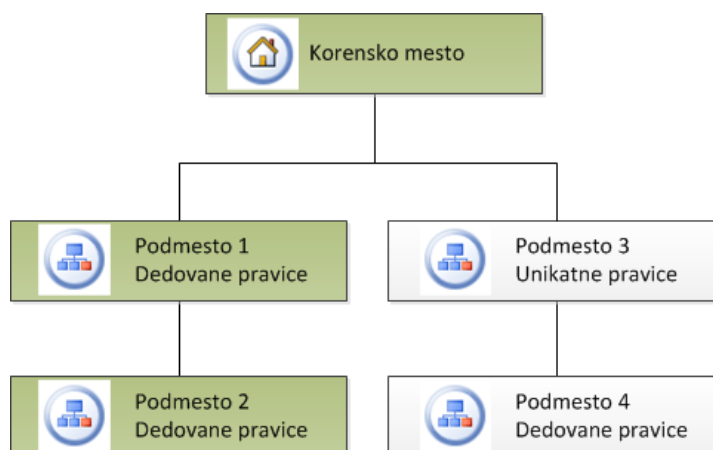
- **Stvaritev novega spletnega gradnika na podlagi obdelane sheme** – S klicem namenske metode je možno uvoziti nov spletni gradnik na podlagi sheme. SharePoint sam prepozna za kateri gradnik gre, ga ustvari in vrne kazalec nanj. Ker se je za vsak spletni gradnik generiral nov identifikator, smo morali to sproti beležiti. V ta namen smo sestavili preslikovalni slovar, katerega zapisi so predstavljali par starega in novega identifikatorja.
- **Umestitev spletnega gradnika v pripadajoče odložišče** – S tem, ko smo dobili kazalec na nov spletni gradnik, smo ga lahko s klicem še ene metode umestili v pravo odložišče.
- **Konfiguracija posebnih spletnih gradnikov** – Vsi gradniki so na nek način posebni, pa vendar naj na tem mestu izpostavim samo dva. Prvi je gradnik za delo s seznamami. Omogoča pregled vsebine seznamov in knjižnic, tako da uporabi enega od pogledov, ki so na izvornem seznamu definirani. Pri stvaritvi novega tovrstnega gradnika se je hkrati uvozil tudi kazalec na pogled, definiran na starem seznamu. V dodatnem koraku smo ta kazalec zamenjali s pomočjo preslikovalnega slovarja vseh pogledov. Za posebnega se je izkazal tudi gradnik za prikaz področnih novic. Na MOSS 2007 namreč tega gradnika ni več, zato smo zanj morali najti ustrezno zamenjavo in jo tudi našli v obliki gradnika za hitre povezave. Uvoz s pomočjo sheme ni bil mogoč, zato smo morali za omenjeni gradnik spisati dodatno logiko, ki je ustvarila povsem ne-konfigurirano instanco gradnika za hitre povezave, nato pa v zanki uvozila področne novice, vsako kot hitro povezavo.
- **Dodajanje povezav med spletnimi gradniki** – Po uvozu vseh spletnih gradnikov smo se še enkrat sprehodili čez celoten seznam. Za vsak gradnik smo preverili, ali mora vsebovati povezave na druge gradnike. V primeru, da je povezava na starem portalu obstajala, se je za dotični gradnik še enkrat prebrala entiteta iz diska. Povezave smo hranili kot atribut entitete, pri čemer je vsako povezavo tvoril trojec: identifikator izvora – tip povezave – identifikator ponora. S pomočjo preslikovalnega slovarja ni bilo težko pridobiti preslikane identifikatorje, tip

povezave pa definira kar SharePoint. Vse potrebne vrednosti smo podali namenski metodi, ki je spletna gradnika povezala.

5.5.1.4 Migracija globalnih skupin, vlog

Varnostni mehanizmi SharePointa omogočajo omejevanje dostopa do svojih resursov, najsi bo to na nivoju celotnega portala, spletnega mesta, seznama ali celo posameznega zapisa v njem. Tovrstna funkcionalnost je precej kompleksna, podrobnosti pa za potrebe diplomske naloge nepomembne. Kljub temu bom zavoľo lažjega razumevanja podal osnovno sliko.

Vsaka SharePoint zbirka mest ima definirano privzeto množico skupin, npr. administrator, bralec in član. Na ravni celotne platforme so definirane osnovne pravice, kot so »Dovoli ogled vsebine« ter »Dovoli urejanje dokumentov«. Vsaka pravica določa možnost uporabe ene ali več funkcionalnosti portala. Pri SPS 2003 velja, da nabor osnovnih pravic skupaj z določeno skupino ali uporabnikom tvori tako imenovano vlogo za nek varnostni objekt, npr. mesto, seznam, itd. MOSS 2007 se v tem pogledu razlikuje. Osnovne pravice niso več vezane neposredno na vlogo. Namesto tega so vpeljali še pojem nivoja pravic, pri čemer je vsak nivo pravic podmnožica množice osnovnih pravic. Poleg tega SharePoint pozna dedovanje pravic. Če eksplicitno ne zahtevamo prekinitve, se pravice prenesejo iz korenkega mesta v vsa podmesta. Ob prekinitvi je treba skupinam pravice spremeniti. Tako spremenjene pravice se prenesejo v vsa podmesta, do naslednje prekinitve.



Slika 5-8: Dedovanje pravic

Pri organizacijskih enotah je bil edini varnostni objekt področje. Omenjene podatke smo iz SPS 2003 portala pridobili v dveh korakih. Kot prvo se je globalno izvedel izvoz vseh skupin

mesta in pripadajočih članov. Ker je imel SPS 2003 le eno portalsko mesto (ostalo so tvorila področja), je bilo potrebno izvoz skupin narediti le enkrat za celoten portal. V drugem koraku je bilo potrebno za vsako področje preveriti, ali pravice deduje oz. jih prekinja. Obe možnosti smo z orodjem zabeležili in shranili v entiteto.

Tudi uvoz je bil opravljen v dveh korakih. Skupine in pripadajoče uporabnike je bilo potrebno uvoziti še pred uvozom prvega spletnega mesta. Drugi korak se je izvršil za vsako spletno mesto posebej. Vsakič je bilo potrebno preveriti, ali spletno mesto deduje pravice od starša. Če temu ni bilo tako, je orodje našlo izmed predpripravljenih prvi nivo, ki je vseboval manj pravic kot je bilo potrebno. Rezultat preseka med poiskanim nivojem ter potrebnimi pravicami smo nato dodali kot nov nivo pravic.

5.5.1.5 Migracija opozoril

SharePoint je dokumentni sistem, namenjen množici uporabnikom, ki med sabo sodelujejo. Uporabniki imajo na voljo dokumente, ki jih skupaj urejajo. Ker se je izkazalo za praktično, da bi o vsaki spremembi določenega dokumenta (lahko tudi celotnega seznama ali področja) obvestili vse sodelujoče, so pri Microsoftu vpeljali sistem sporočil – opozoril.

Študija tehnologije, ki se skriva za opozorila je pokazala, da teh ne bo mogoče prenesti v sklopu posameznega področja. SPS 2003 podatke o opozorilih hrani v profilih uporabnikov. Profili so vmesne entitete, ki del podatkov pridobijo iz profilov v aktivnem imeniku, drugi del pa zapolni Sharepoint. Tako zbrani profili tvorijo bazen, iz katerega posamezni portali črpajo svoje uporabnike.

Ko uporabnik želi prejemati opozorila, mora za določen dokument, knjižnico ali področje ustvariti prijavo. To stori tako, da vpiše konfiguracijske podatke, kot so tip opozorila, frekvenca obveščanja, itd. Vsakič, ko se uporabnik prijavi na opozorila za nek dokument, se ustvari referenca med njegovim profilom in dokumentom. Vsak uporabnik je seveda lahko prijavljen na poljubno število opozoril.

| Profil | ID_Elementa; Frekvenca; Tip |
|---------------------|---|
| Profil uporabnika 1 | #ID_Dokumenta1;#ID_Knjžnice1;Dnevno; Ob spremembi |
| | #ID_Knjžnice2; Tedensko; Vse spremembe |
| Profil uporabnika 2 | #ID_Področja; Mesečno; Ob brisanju |
| | ... |

Tabela 5-5: Struktura opozoril, na katera je uporabnik prijavljen

V našem primeru smo vsa opozorila izvozili naenkrat, po končanem izvozu področij. S preprostim sprehodom čez vse profile uporabnikov smo kreirali strukturo, ki je nadvse podobna tisti v tabeli Tabela 5-5.

MOSS 2007 z opozorili ravna drugače. Ta več niso shranjena v profilih uporabnikov, ampak kot atribut spletnega mesta, kjer naj bi bilo opozorilo prijavljeno. Pri uvozu smo se zaradi tega morali soočiti z marsikaterim problemom. Tekom uvoza je SharePoint vsem elementom, vključno s spletnimi mesti, dodelil nov identifikator. Izvožena struktura opozoril je bila takšnih identifikatorjev polna, a kaj ko so se vsi nanašali na dokumente, knjižnice in področja SPS 2003 portala. Da bi težavo obšli, smo morali ustvariti preslikovalni mehanizem, ki je povezoval stare identifikatorje z novimi. Za potrebe opozoril so tako nastali trije preslikovalni slovarji, kateri so se polnili skozi celotno migracijo posamezne organizacijske enote:

- **Slovar seznamov** – Vsak zapis je predstavljal par, pri čemer je prvi zapis določal identifikator starega seznama, drugi zapis pa identifikator novega seznama.
- **Slovar seznamov in spletnih mest** – Vsak zapis je predstavljal par, pri čemer je prvi zapis določal identifikator novega seznama, drugi zapis pa identifikator spletnega mesta, v katerem se je novi seznam nahajal.
- **Slovar zapisov seznama** – Vsak zapis je predstavljal par, pri čemer je prvi zapis določal identifikator novega seznama, drugi zapis pa zvezo med identifikatorjema zapisov v starem in novem seznamu.

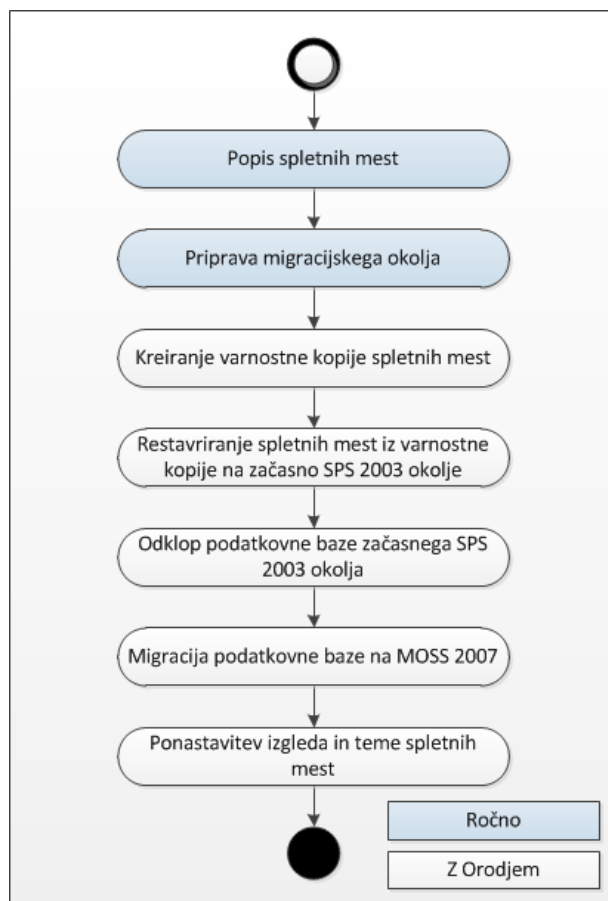
Teoretično bi s tako definiranimi preslikavami bilo mogoče v novem sistemu najti dokument na katerega se nanaša opozorilo, a le pod enim pogojem. Stari in novi sistem bi morala ohranjati isto strukturo mest. V našem primeru temu ni bilo tako, saj smo organizacijske enote, ki so bile prej združene v eno portalsko mesto razbili v množico

zbirk mest. Če k temu dodamo dejstvo, da smo vsako organizacijsko enoto uvažali posebej, je jasno, da so prej omenjeni slovarji imeli podatke le za trenutno zbirko mest. posledično nam ni ostalo drugega, kot da za vsako organizacijsko enoto izvedemo uvoz z istim naborom opozoril. S tem smo dosegli, da se je v vsako zbirko mest uvozila le tista podmnožica opozoril, ki jo je bilo mogoče s preslikovalnimi slovarji prepoznati.

5.6 Prenos spletnih mest

Del obstoječega intranet portala je bil namenjen zbirkam WSS 2.0 spletnih mest. Vsaka takšna zbirka mest je bila ustvarjena s posebnim namenom. Nekatere izmed njih so služile kot predstavitevna mesta za produkte, spet druge kot namenski prostor za delo s projekti, sistemi ter substancami. Omenjenih mest ni bilo mogoče vgraditi v obstoječo hierarhijo organizacijskih enot iz dveh razlogov:

- Produktno, sistemsko ali katerokoli drugo mesto je nastalo kot plod sodelovanja posameznikov iz več različnih organizacijskih enot.
- WSS platforma ponuja predpripravljene predloge mest, ki ustrezajo sodelovalnim potrebam, npr. predloga srečanj (angl. »Meetings«). Ustvarjanje novih mest je tako precej poenostavljeno.



Slika 5-9: Migracija WSS spletnih mest po korakih

Vsa WSS mesta so bila prenesena po pristopu, imenovanem migracija podatkovne baze. Pred dejanskim prenosom je bilo potrebno sestaviti seznam vseh zbirk mest, katere je stranka želela obdržati. Tako nastali seznam je služil kot podlaga za pripravo migracijskega okolja. V to aktivnost sem združil postavitev SPS 2003 okolja na ločenem strežniku, pripravo novih predlog mest za MOSS 2007 ter sestavo preslikovalnih pravil.

5.6.1 Priprava migracijskega okolja

Za vsak tip mesta (produkt, projekt,...) je bilo na MOSS 2007 potrebno definirati novo definicijo. Tehnično gledano je definicija XML dokument v katerem je definirano spletno mesto, pripadajoči seznam, spletni gradniki, podmesta, itd. Podrobnosti o načinih njihovega kreiranja bom namenoma izpustil, saj jih zlahka najdemo na spletu, okviren postopek pa je sledeč:

- Iz obstoječe definicije mesta razberemo podatke o straneh, seznamih in pripadajočih poljih, postavitvah spletnih gradnikov ...

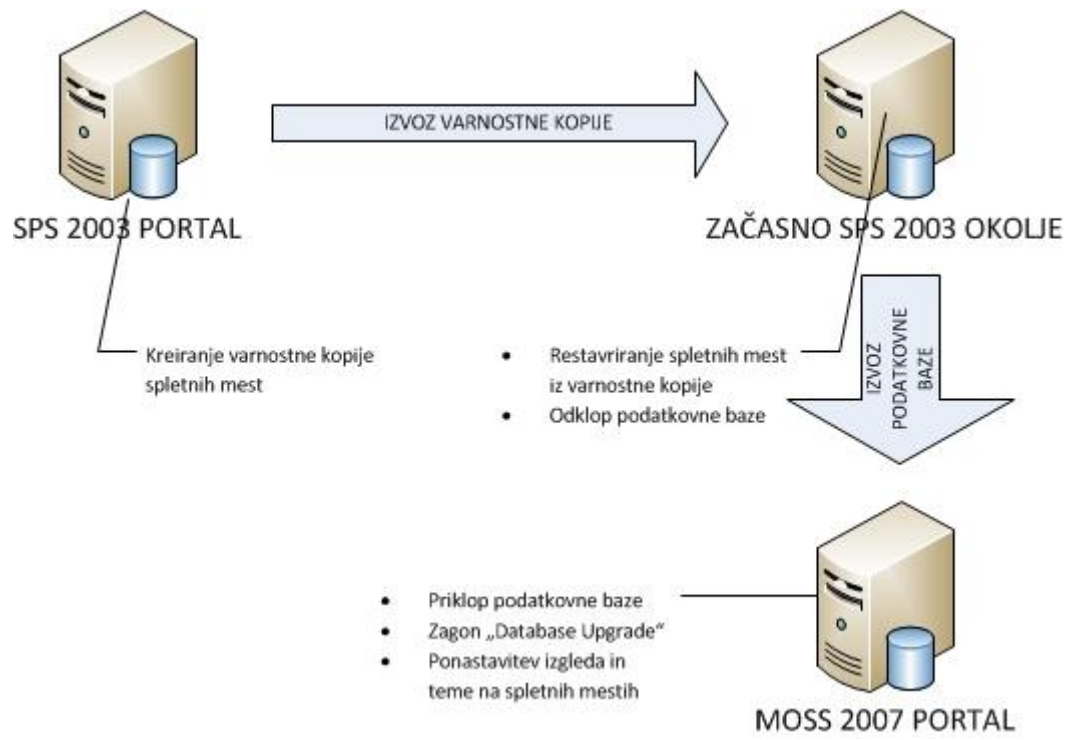
- Za vsak seznam ustvarimo definicijo seznama (angl. »List Definition«), torej predpišemo vsa polja, določimo nastavitve ter obrazce za dodajanje novih ter pregled obstoječih elementov v seznamu.
- V okviru MOSS 2007 ustvarimo novo definicijo mesta, v katero vključimo reference na novonastale predloge seznama, definiramo spletne gradnike ter določimo hierarhijo podmest in strani.
- S sistemskim orodjem novonastalo definicijo registriramo v MOSS 2007.

Za pravilen prenos podatkov same definicije mest niso dovolj. Potrebno je definirati pravila po katerih se elementi stare definicije preslikajo v elemente nove definicije. V MOSS 2007 je preslikovalni mehanizem že vgrajen. Deluje po principu XML dokumentov, v katerih so privzeto navedeni le tisti elementi definicij, ki so bili na voljo ob namestitvi SharePointa. Sezname naših definicij so bili narejeni po meri, zato je bilo potrebno za vsakega izmed njih dodati novo pravilo, ki določa:

- izvoren element stare definicije mesta,
- ciljni element nove definicije mesta.

5.6.2 Migracija podatkovne baze

Obstoječi intranet portal je vse podatke shranjeval v eno vsebinsko bazo. Tako so bili podatki spletnih mest združeni s podatki organizacijskih enot ter javnega dela. V novem portalu smo želeli spletnim mestom zagotoviti lastno podatkovno bazo. Da bi to dosegli, smo s sistemskim orodjem naredili varnostne kopije spletnih mest, ter jih izvozili v začasno SPS 2003 okolje. V začasnem okolju smo izvožena mesta restavriral v eno vsebinsko podatkovno bazo. Tako nastalo bazo smo nato odklopili in jo priklopili nazaj v MOSS 2007 okolje. Ostal je le še zagon postopka za migracijo podatkovne baze s sistemskim orodjem, ki je upoštevaajoč predloge mest ter preslikovalnih pravil kreiralo produktna, projektna, sistemska ter ostala mesta in jih povežalo s podatki v priklopljeni vsebinski podatkovni bazi.



Slika 5-10: Migracija podatkovne baze

6 POST-MORTEM ANALIZA

Po sedmih mesecih dela, več tisoč spisanih vrsticah kode in treh testnih migracijah je prišel čas za izvedbo prave migracije. Po planu naj bi celoten prenos trajal tri dni, od petka do nedelje. V tem času bi obstoječi portal deloval v bralnem načinu. Tako bi zagotovo vedeli, v kakšnem stanju je obstoječi portal pred prenosom in kakšno stanje želimo vzpostaviti na novi platformi. Za preklopno strategijo bi uporabili pristop s prvinami »Big bang« ter paralelnega delovanja. Starega sistema ne bi povsem ugasnili, temveč bi ta po končani migraciji še vedno deloval v omejenem načinu, objavljen pa bi bil na drugem spletnem naslovu. MOSS 2007 portal bi zasedel spletni naslov starega portala in bi bil uporabnikom takoj na voljo za uporabo.

Sodeč po planu, bi morala biti prava migracija precej rutinsko opravilo. Ni bilo povsem tako, pa vendar smo se prebili do zadnjega dne, ko je bil na vrsti prenos organizacijskih enot. Ta naj bi trajal približno štirinajst ur, ravno dovolj, da se migracija ne bi zavlekla čez predviden časovni plan. Žal se je kmalu izkazalo, da to ne bo mogoče. Pri skoraj polovici organizacijskih enot so se pojavile napake, zaradi katerih bi morali prenos ponoviti. Prenos največje izmed njih je trajal kar šest ur, prenos ostalih pa v povprečju dve uri na enoto. To je bil poglavitni razlog za prekinitve migracije. Kaj je šlo narobe? V poglavju 5.5 sem omenil seznam področij, ki ga je stranka obdelala in pripravila za migracijo. Med nastankom dokumenta z vsemi preslikavami področij ter izvedbo prave migracije je minilo nekaj časa. V tem času se je stari portal širil. Dodana so bila nova področja, ki so sicer bila v dokumentu zabeležena, a ne povsem pravilno. Predniki novo dodanih področij se niso zabeležili. Pri prenosu organizacijskih enot so se takšna področja prenesla kar pod korensko mesto zbirke mest. Seveda je bilo to narobe, saj se je s tem podrla hierarhija mest in posledično dedovanje pravic. V redkih primerih pa se je vsebina področij, ki so imela enako ime, celo združila.

Omenjeni dogodki jasno prikazujejo posledice neupoštevanja pravil migriranja. S tem, ko smo preveč stavili na tehnološko izvedbo migracije, smo zanemarili človeški faktor. Uredniki vsebin starega portala so bili v celoten proces premalo vključeni, zato so hitro prevzeli vlogo opazovalca. Da bi se v prihodnje neskladjem izognili, je stranka nadzor nad širjenjem portala poostrila. Ogromna napaka je bila storjena tudi z naše strani. Pred pravo migracijo smo izvedli več testnih migracij na vzorčnih delih starega portala. Ker je bila zadnja testna

migracija prestana z odliko, se nam ni zdelo potrebno izvesti še eno, tokrat na celotnem portalu. V primeru, da bi tako obsežno testno migracijo izvedli, bi se z veliko gotovostjo ognili problemu novih področij. Obogateni z izkušnjami iz prvega poskusa smo naša orodja ter seznam področij popravili, nato pa celoten postopek migracije čez mesec dni ponovili in tudi uspešno prestali.

Po zaključeni migraciji se je v sodelovanju s stranko vzpostavila vzdrževalna pogodba. Pri tem je vredno omeniti, da je bilo največ popravkov potrebnih pri WSS spletnih mestih. Naj spomnim, da smo za prenos spletnih mest uporabili pristop, imenovan migracija podatkovne baze. Uporaba tega pristopa se je izkazala za precej neustrezno. Vse skupine mest starega portala so se namreč preslikale v nivoje pravic, pri čemer je vsak nivo vseboval pravice, ki jih je imela skupina. Da bi težavo odpravili, smo morali razviti še eno namensko orodje. Težav s spletnimi mesti vseeno še ni bilo konec. Dokumentne knjižnice privzeto omogočajo uporabo map. Mapa je poseben tip zapisa v seznamu in ne vsebuje istih polj, kot jih dokumenti. Po migraciji to ni bilo več res. Mapam so se pripisala polja, značilna le za dokumente. Rešitev smo sicer našli, vzroka za spremembo v delovanju map pa do tega trenutka še ne. Upal bi si reči, da smo s tem, ko smo spletna mesta prenesli po pristopu migracije podatkovne baze in ne orodjem za vsebinsko migracijo naredili precejšnjo napako.

Razvoj namenskih orodij za vsebinsko migracijo je prinesel tudi marsikaj dobrega. Ne samo, da smo vsi člani projektne ekipe pridobili poglobljeno znanje o SharePoint platformi. Tekom projekta smo razvili orodje, ki po zmogljivosti prav gotovo konkurira vsaj nekaterim obstoječim orodjem za vsebinsko migracijo SharePointa. Razvoj smo sicer pogojili s potrebami specifičnega projekta, vendar bi isto orodje lahko uporabili tudi na katerem drugem praktičnem primeru. Res pa je, da je od izida SPS 2003 minilo precej let. Številna podjetja so že pred leti na tak ali drugačen način prešla na MOSS 2007 platformo, nemalo pa je tudi takih, ki že razmišljajo o prehodu na novi SharePoint Server 2010. Ko bodo takšne misli prešle v dejanja, bomo na to prav gotovo pripravljeni.

7 VIRI

[1] (2010) Data Cleansing.

Dostopno na: http://en.wikipedia.org/wiki/Data_cleansing

[2] (2007) Data Migation. A White Paper by Bloor Research.

Dostopno na: <http://www.Bloorresearch.com/download/freepaper/952.html>

[3] (2007) Data Migration in the Global 2000. Research, forecasts and survey results.

Dostopno na: <http://www.rever.eu/white-papers/876-Data-Migration-Survey.pdf>

[4] John Morris, *Practical Data Migration*. Swindon: BCS, 2008, pogl. 2.

5] John Morris, *Practical Data Migration*. Swindon: BCS, 2008, pogl. 12.

[6] (2009) The Data Migration go-live strategy - what is it and why does it matter.

Dostopno na: <http://www.datamigrationpro.com/data-migration-articles/2009/3/26/the-data-migration-go-live-strategy-what-is-it-and-why-does.html>

[7] (2009) Upgrading to Office SharePoint Server 2007.

Dostopno na: <http://go.microsoft.com/fwlink/?LinkID=85556>

8 PRILOGE

8.1 Seznam tabel

| | |
|--|----|
| Tabela 2-1: Praktični primeri migriranja | 5 |
| Tabela 5-1: Pregled podprtih pristopov k nadgradnji SharePoint platforme | 21 |
| Tabela 5-2: Primeri spletnih naslovov pri SPS 2003 | 31 |
| Tabela 5-3: Primeri preslikovanja spletnih naslovov | 32 |
| Tabela 6-1: Preslikave objektov | 33 |
| Tabela 6-2: Dodatna pravila pri migraciji organizacijskih enot | 34 |
| Tabela 6-3: Struktura opozoril, na katera je uporabnik prijavljen | 46 |

8.2 Seznam slik

| | |
|---|----|
| Slika 3-1: Umestitev aktivnosti v projektne faze..... | 9 |
| Slika 3-2: Začasno vmesno shranjevanje | 12 |
| Slika 3-3: Direktni prenos..... | 13 |
| Slika 3-4: Enosmerna sinhronizacija | 14 |
| Slika 3-5: Dvosmerna sinhronizacija | 14 |
| Slika 4-1: Primer hierarhije portala z 20 podpodročji | 30 |
| Slika 4-2: Primer hierarhije portala z 21 podpodročji | 31 |
| Slika 5-1: Diagram obstoječega intranet portala | 24 |
| Slika 5-2: Diagram prenovljenega portala | 26 |
| Slika 5-3: Visokonivojski diagram orodja za vsebinsko migracijo | 29 |
| Slika 5-4: Prenos organizacijskih enot po korakih | 35 |
| Slika 5-5: Razmestitev odložišč spletnih gradnikov..... | 41 |
| Slika 5-6: Dedovanje pravic | 44 |
| Slika 5-7: Migracija WSS spletnih mest po korakih..... | 48 |
| Slika 5-8: Migracija podatkovne baze | 50 |