UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Mavsar

UPORABNIŠKA IZKUŠNJA PRI PROJEKTNEM VODENJU

DIPLOMSKO DELO

NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: prof. dr. Marko Bajec

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00506/2010
Datum: 05.04.2010

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

Kandidat: **PRIMOŽ MAVSAR**

Naslov: **UPORABNIŠKA IZKUŠNJA PRI PROJEKTNEM VODENJU**
USER EXPERIENCE IN PROJECT MANAGEMENT

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Vključevanje uporabnikov v proces razvoja programske opreme ter uporabniška izkušnja, s katero označujemo interakcijo med uporabnikom (podjetjem, ki storitev naroča)in izvajalcem (podjetjem, ki storitev ponuja) je na področju razvoja programske opreme čedalje pomembnejše. V diplomskem delu preučite aktualne trende na področju načrtovanja uporabniške izkušnje. Posebej se osredotočite na pomen uporabniške izkušnje pri vodenju IT projektov.

Mentor:

prof. dr. Marko Bajec

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Nikolaj Zimic

Izvirnik izdane teme

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Primož Mavsar, z vpisno številko 63040097, sem avtor diplomskega dela z naslovom:

Uporabniška izkušnja pri projektnem vodenju

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Marka Bajca;
- 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 _____ Podpis avtorja: _____

ZAHVALA

Zahvalil bi se Tilen Travniku in Mitji Mavsarju za idejo in pomoč pri izvedbi diplomske naloge.

Prav tako bi se zahvalil mentorju prof. dr. Marko Bajecu za nasvete pri izdelavi diplomske naloge.

Kazalo

Povzetek in ključne besede 1

Abstract and keywords 2

1. Uvod 3

2. Uporabniška izkušnja 4

2.1 Kaj je uporabniška izkušnja? 4

2.2 Kratka zgodovina uporabniške izkušnje 6

2.3 Trenutni in prihajajoči trendi na področju uporabniške izkušnje 6

2.4 Področja uporabniške izkušnje 10

2.4.1 Strategija 11

2.4.2 Obseg 14

2.4.3 Struktura 17

2.4.4 Ogrodje 21

2.4.5 Površina 27

2.5 Metode uporabniške izkušnje 31

3. Vodenje projektov 35

3.1 Problemi in trenutno stanje na področju vodenja projektov 35

3.2 Metodologija vodenja projektov Waterfall 37

3.3 Agilno vodenje projektov kot odgovor drugim metodologijam 41

3.3.1 Glavni elementi metodologije Scrum 43

3.3.2 Kaj metodologija Scrum prinaša načrtovalcem uporabniške izkušnje 49

3.3.3 Integracija uporabniške izkušnje v metodologijo Scrum 53

4. Rezultati opravljene ankete 56

4.1 Zavedanje o uporabniški izkušnji je prisotno 56

4.2 Kako se podjetja lotijo projektnega vodenja 57

5. Sklepne ugotovitve 58

6. Priloge 59

Dodatek A Vprašalnik o vodenju projektov ter uporabniški izkušnji 59

7. Slike 64

8. Viri 64

Seznam uporabljenih kratic

CMS – *Content Management System*

GPS – *Global Positioning System*

HTML – *Hypertext Markup Language*

QA – *Quality assurance*

UI – *User Interface*

UX – *User Experience*

Povzetek in ključne besede

V diplomskem delu je opisan eden izmed najbolj priporočljivih načinov integracije uporabniške izkušnje in njenih načrtovalcev v celoten proces vodenja projektov. V prvem delu je predstavljeno področje uporabniške izkušnje, kaj je uporabniška izkušnja, trenutni in prihajajoči trendi, opisana so posamezna področja uporabniške izkušnje skupaj s priporočili in nasveti, na kaj je treba biti pozoren pri njenem načrtovanju. Prvi del se zaključi z navedbo nekaterih najpogosteje uporabljenih metod pri načrtovanju uporabniške izkušnje.

V drugem delu je govor o načinu integracije uporabniške izkušnje v proces vodenja projektov. Najprej so predstavljeni najpogostejši problemi, s katerimi se srečujemo pri vodenju projektov. V nadaljevanju pa sta predstavljeni dve metodologiji vodenja projektov: Waterfall in Scrum. Prva je predstavljena kot slab primer metodologije, ki brez modifikacij težko dopušča preprosto integracijo in delovanje načrtovalcev uporabniške izkušnje v procesu vodenja projekta. Na drugi strani pa je kot dober primer predstavljena metodologija Scrum, skupaj z njenimi glavnimi elementi ter prednostmi in izzivi, ki jih prinaša načrtovalcem uporabniške izkušnje.

Zaključek diplomskega dela vsebuje anketo in komentar njenih rezultatov. Anketa je bila izvedena med desetimi slovenskimi podjetji, ki se ukvarjajo s spletno produkcijo.

Ključne besede:

uporabniška izkušnja, načrtovanje uporabniške izkušnje, vodenje projektov, metodologija Waterfall, metodologija Scrum

Abstract and keywords

This thesis describes one of the most recommended ways of integration user experience and its designers in the process of project management. The first part describes the domain of the user experience, what user experience is, which are the current and upcoming trends, describes the fields of user experience, along with recommendations and advices on its planning. The first part ends with an indication of the most used methods in designing the user experience.

The second part describes best practices of user experience integration in to project management. First I describe the most common problems faced in project management. Then I present two project management methodologies: Waterfall and Scrum. The first one is presented as a bad example of the methodology, in which user experience experts are, without modification of methodology, hard to operate and integrate in the process of project management. On the other hand Scrum methodology is presented as a good example of this integration, along with its main elements. This part ends with presentation of advantages and challenges that this methodology brings to the user experience designers.

The end of thesis contains a survey and its results. Survey included 10 Slovenian companies involved in web production.

Key words:

user experience, user experience design, project management, Waterfall methodology, Scrum methodology

1. Uvod

Dandanes se veliko podjetij ukvarja z izdelavo spletnih strani in spletnih aplikacij. Uspešnost končnega izdelka pa je odvisna od mnogih dejavnikov. Poleg osnovne ideje in vsebine močno vplivata na končno uspešnost projekta tudi dobro in organizirano vodenje projekta ter predvsem to, kako končan izdelek služi njegovim uporabnikom.

Izbrana metodologija vodenja projekta ima lahko močan vpliv na stroške, potreben čas razvoja ter na koncu tudi na samo uporabniško izkušnjo končnega izdelka. V veliko podjetjih vodenje projekta poteka nekako tako: najprej od naročnika zberejo vse funkcionalne in vsebinske zahteve, nato se podlagi zbranih zahtev pripravijo načrt in grobo strukturo, ki jo nato prejme oblikovalec in na njeni osnovi pripravi končen videz, tega pa v zadnjem koraku programer pretvori v funkcionalno in delujočo kodo. Ker pa realnost ni tako idealna, se pogosto v času izvedbe projekta pojavijo težave, ki jih v začetnih fazah načrtovanja nismo predvideli. Pojavijo se lahko nove, boljše ideje za posamezne funkcionalnosti, ki bi jih bilo greh zavreči. Vse to zahteva posege v prejšnje, že definirane in zaključene faze, kar nas na koncu pripelje do povečanja stroškov, podaljšanja rokov izdelave ter do padca koncentracije ter motivacije celotne ekipe za nadaljnje delo pri projektu.

Zelo pomemben vpliv na uspešnost projekta pa ima tudi njegova uporabniška izkušnja in to, kako končan izdelek zadovoljuje cilje in navade njegovih uporabnikov. Pri posameznih fazah načrtovanja, oblikovanja in programiranja se velikokrat premalo pozornosti posveča uporabniku in predvsem njegovim, pogosto tudi čudnim načinom uporabe. Pri veliko projektih je glavno vodilo zadovoljstvo naročnika, za katerega se stran ali aplikacija izdeluje, ne pa zadovoljstvo končnega uporabnika, ki mu je izdelek dejansko namenjen.

Uporabniška izkušnja in njeno načrtovanje zahtevata nenehno spreminjanje in odzivanje na raziskave ter rezultate testov. V času razvoja projekta morda ob testiranju določene funkcionalnosti pri uporabnikih ugotovimo, da se funkcionalnost ne obnaša tako, kot smo predvideli, in tako ne služi prvotnem namenu. Uporabniki pa se ob uporabi zmedejo in hitro zapustijo spletno stran ali zaprejo aplikacijo. Tega seveda na začetku, pri načrtovanju nismo mogli predvideti, zato je zelo pomembno, da so načrtovalci uporabniške izkušnje prisotni med celotnim razvojem ter pri uporabnikih sproti testirajo posamezne dele spletne strani ali spletne aplikacije. Uporaba metodologije vodenja projektov, ki se je sposobna preprosto odzvati na

spremembe, ki se pojavijo med razvojem projekta, je tako lahko ključni dejavnik uspešnosti končnega izdelka.

Zakaj torej vse definirati vnaprej, če pa vemo, da realnost ni idealna in da bo do sprememb zagotovo prišlo? Zakaj uporabljati metodologijo, ki se težko odziva na spremembe v poznejšem razvoju? Kako poskrbeti, da bodo uporabniki pri uporabi spletne strani ali aplikacije imeli kar najboljšo izkušnjo? Kaj sploh je uporabniška izkušnja? Odgovori na ta in še mnoga druga vprašanja sledijo v nadaljevanju.

2. Uporabniška izkušnja

Splet, katerega avtor je Tim Berners-Lee, je bil v začetku namenjen znanstvenikom in fizikom po celotnem svetu, kot orodje za lažjo izmenjavo in dostopnost do njihovih raziskav. Takrat je malokdo videl njegov dejanski potencial. Z razvojem tehnologije so spletu, spletnim brskalnikom in spletnim strežnikom dodajali vedno več opravilnih zmožnosti, te pa so z bliskovito hitrostjo privabljale nove in nove uporabnike, ki od spletne strani ali aplikacije tudi vedno več pričakujejo. Splet je med svojim razvojem postal vedno bolj dvosmeren medij, ki ni služil le shrambi ogromne količine podatkov. Z razvojem spletnih forumov, spletnih bank, spletnih skupnosti itd. se je od uporabnika zahtevalo veliko več kot le klikanje. S tem je prišla težnja po karseda dodelani uporabniški izkušnji ter izpopolnjenemu uporabniškemu vmesniku *(angl. user interface – UI)*. Ker pa vsak uporabnik gleda skozi svoje oči in si uporabo funkcij razlaga po svoje, je treba zavoljo dobre uporabniške izkušnje pri vsakem projektu imeti nekoga, ki zna poslušati, in razumeti njegove potrebe in načine uporabe. Zbrane ugotovitve pa naj bi znal sestaviti v uporabnikom jasno, nedvoumno in delujočo celoto.

Zaradi teh potreb se je razvilo področje uporabniške izkušnje ali s tujko *user experience* (*UX*), ki se mu danes posveča vedno več pozornosti.

2.1 Kaj je uporabniška izkušnja?

V podjetju Nielsen Norman Group, ki velja za eno vodilnih na področju svetovanja o uporabniški izkušnji in uporabnosti in katerega ustanovitelji so Donald Norman, Jakob Nielsen in Bruce Tognazzini, vsi dobro poznani profesionalni načrtovalci uporabniške izkušnje, so to definirali z naslednjimi besedami: »Uporabniška izkušnja zajema vse vidike interakcije končnega uporabnika s podjetjem, njegovimi stiritvami in izdelki. Prvi pogoj za

dobro uporabniško izkušnjo je izpolnitev vseh potreb uporabnika, brez njegovega razburjenja. Temu pogoju sledita preprostost in eleganca končnega izdelka, ki ga imajo uporabniki z veseljem v lasti in uporabljajo. Pri resnično dobri uporabniški izkušnji ne gre le za to, da uporabnikom ponudimo tisto, kar pravijo, da potrebujejo. Če želimo doseči visokokakovostno uporabniško izkušnjo v celotni ponudbi podjetja, je treba združiti storitve več disciplin, vključno s tehničnimi, marketinškimi in disciplinami grafičnega in industrijskega oblikovanja ter oblikovanja vmesnika.« [7] Torej uporabniško izkušnjo definira celoten skupek interakcij uporabnika z določenim podjetjem. Vzemimo primer, ko bi uporabnik pri pridobivanju informacij o nakupu izdelka po telefonu naletel na neprijaznega svetovalca. Ta bi ga z zdolgočasenim glasom preusmeril na spletno stran, kjer bi se s težavo prebil skozi vse korake nakupa. Na koncu pa bi na izdelek, za katerega je bilo navedeno, da bo dobavljiv v 3 delovnih dneh, čakal dva tedna. Na koncu bo uporabniška izkušnja s celotnim podjetjem pri njem pustila slab vtis.

Standard ISO 9241-210 opredeljuje uporabniško izkušnjo kot zaznavanje osebe in njenih odzivov, ki izhajajo iz uporabe ali predvidene uporabe izdelka, sistema ali storitve. Torej je uporabniška izkušnja subjektivna in se osredotoča na uporabo izdelka. Dodatne opombe pri opredelitvi standarda ISO pojasnjujejo, da uporabniška izkušnja vključuje vsa uporabnikova čustva, prepričanja, želje, predstave, telesne in duševne odzive, vedenja in dosežke, ki so se pojavljali pred uporabo, med njo in po njej.

Uporabniška izkušnja je torej sama po sebi subjektiven pojem, saj govori o individualnih pogledih uporabnikov, ki pa se s časom spreminjajo. Ravno zaradi tega je uporabniška izkušnja zelo dinamično področje, ki se z razvojem in učenjem uporabnikov hitro spreminja.

2.2 Kratka zgodovina uporabniške izkušnje

Izraz uporabniška izkušnja je v sredini leta 1990 prvi uporabil Donald Norman, eden izmed ustanoviteljev podjetja Nilsen Norman Group ter izkušen načrtovalec uporabniške izkušnje. Na vse večje zanimanje za uporabniško izkušnjo pa so nato vplivali predvsem ti dejavniki:

- razvoj področij načrtovanja spletnih strani, kjer je bilo pomembno združiti interese predstavnikov različnih interesnih skupin (oddelek za trženje, oddelek za širitev in prepoznavnost blagovne znamke, oddelek načrtovalcev grafične zasnove in oddelek, odgovoren za uporabnost). Odgovorni za trženje in prepoznavnost blagovne znamke so morali vstopiti v interaktivni svet spletnih strani, kjer je imela uporabnost pomembno vlogo. Na drugi strani pa so odgovorni za uporabnost spletne strani morali pri njenem načrtovanju upoštevati njihove želje in zahteve. Področje uporabniške izkušnje je tako ponujalo platformo za združevanje interesov vseh skupin, ki so z združenimi močmi spletno stran naredile preprosto za uporabo, dragoceno in učinkovito. To je bil glavni razlog, zakaj se je večina zgodnjih pisanj o uporabniški izkušnji osredotočila ravno na uporabniško izkušnjo spletnih strani.
- napredek v mobilnih in drugih povsod navzočih računalniških tehnologijah, kjer se je interakcija človek–računalnik premaknila na praktično vsa področja človeškega delovanja. To je privedlo do premika s področja načrtovanja uporabnosti na bogatejše področje uporabniške izkušnje, kjer so uporabnikova čustva, korist in motivacija deležna večje pozornosti kot učinkovitost, uspešnost in zadovoljstvo.

Področje uporabniške izkušnje je bilo razvito z namenom, da pokriva celoten vidik uporabnikovih občutkov ob uporabi sistema. Poudarek je bolj na zadovoljstvu in vrednosti izdelka, in ne toliko na dejanski zmogljivosti izdelka. Natančna opredelitev, okvir in elementi uporabniške izkušnje se še vedno razvijajo in se bodo tudi v prihodnje. [8]

2.3 Trenutni in prihajajoči trendi na področju uporabniške izkušnje

Gene Smith, svetovalec na področju oblikovanja vmesnika, informacijske arhitekture in načrtovanja, v svojem blogu navaja pet trendov, ki bodo po njegovem mnenju krojili uporabniško izkušnjo v prihodnosti.

Storitve, kot je programska oprema

Pojav polavtomatskih spletnih storitev testiranja uporabnosti, kot sta Usabilla.com in Usertesting.com, pomeni, da se trg za nekatere vrste storitev testiranja uporabniške izkušnje spreminja. Omenjeni orodji omogočata ustvarjanje različnih testov na izbranih delih delujoče spletne strani ali pa celo na slikah skic in žičnih modelov. Ustvarjene teste in njim pripadajoča vprašanja nato izvajamo na povabljenih uporabnikih kar prek spleta. Na koncu pa nam orodja ponudijo obširno grafično in besedilno statistiko zbranih rezultatov.

Za nekatere vrste projektov bodo omenjena orodja ponujala dovolj dobre, hitre in predvsem poceni rezultate uporabnosti. Rezultati seveda ne bodo enaki kot pri testiranju profesionalnih načrtovalcev uporabniške izkušnje, vendar to za naročnika ne bo pomembno, saj navadno zelo hitro potrebujejo povratne informacije.

S tem pristopom, kjer se uporablja storitve, kot je programska oprema, lahko avtomatiziramo več rutinskih nalog, kot so testiranje uporabnikov, testiranje različne grafike in vsebine.

Analitika uporabniške izkušnje

V zadnjih nekaj letih se je pogovor okrog uporabniške izkušnje močno osredotočil zgolj na oblikovanje in načrtovanje. V naslednjih nekaj letih se bo po mnenju avtorja pogovor bolj osredotočil na merjenje in analitiko. Oblikovanje in načrtovanje seveda ne bosta zanemarjena. Pomembno bo, da bosta omenjeni področji delovali skladno z meritvami in se nanje ustrezno odzivali.

Ideje in tehnike iz trženja in spletne analitike so se začele uveljavljati predvsem med tistimi načrtovalci uporabniške izkušnje, ki jih zanima razumevanje izmerljivih vplivov njihovih odločitev.

Strategija vsebine

Kot pravi avtor, je za mnoge njihove stranke upravljanje z njihovo vsebino stalen izziv. Glede na njegove pogovore s strankami in kolegi v preteklem letu je jasno, da kljub vsem denarjem, ki ga podjetja namenijo sistemom za upravljanje z vsebino (CMS), mnoga še vedno ne obvladajo svoje vsebine. Po njegovem mnenju je rešitev preprosta. Prenehati je treba zapravljati denar za nakup dragih sistemov za upravljanje z vsebino in ga začeti porabljati za

pisanje kakovostne vsebine. V prihodnje bodo podjetja po njegovem mnenju začela resno vlagati v izdelavo vsebinske strategije.

Vračanje spleta v mobilne telefone

Trenutno je razvoj aplikacij za iPhone, Android, Blackberry ter druge mobilne operacijske sisteme zelo razširjena dejavnost in pomeni nekakšen trend. Predvsem prva dva operacijska sistema sta dvignila standard za dostopnost informacij prek mobilne telefonov do točke, ko podjetja, kot so New York Times in podobna, razvijajo lastne aplikacije za distribucijo svojih vsebin na omenjene operacijske sisteme.

Za podjetja, ki razvijajo poslovne mobilne aplikacije, obstajajo tri možnosti:

1. Svoje aplikacije in s tem uporabnike lahko omejijo zgolj na izbrane mobilne operacijske sisteme, kot sta iPhone in Android.
2. Razvijejo lahko ločene aplikacije za vsak operacijski sistem.
3. Razvijejo lahko aplikacije v nevtralnem načinu, z uporabo mobilnega spleta.

Avtor stavi na tretjo možnost, saj bodo razvoj pametnih telefonov, nova pričakovanja ljudi pri dostopnosti informacij prek mobilnih telefonov ter raznolikost in količina operacijskih sistemov, ustvarili popolno zmedo na področju mobilnih spletnih aplikacij. Tako je razvoj aplikacij za mobilni splet najboljša izbira predvsem zato, ker:

- je splet dostopen in bo dostopen po vsakem telefonu (neodvisno od operacijskega sistema);
- nikomur ni treba odobriti spletne aplikacije, kot je to praksa pri prodaji aplikacij za mobilne telefone iPhone;
- bo prihajajoči HTML5 v brskalniku pričaral veliko bogatejšo izkušnjo;
- je splet boljša platforma za razvoj programske opreme (v primerjavi z razvijanjem opreme za vsak sistem posebej);
- bodo mobilni brskalniki začeli pridobivati dostop do strojne opreme mobilnega telefona (kot ima na primer Safari dostop do iPhonovega GPS-sistema).

Namenske aplikacije za različne mobilne operacijske sisteme se bodo še vedno razvijale in obstajale, vendar bodo po mnenju avtorja primernejše za igre in zahtevnejše aplikacije.

Ekonomija resnične izkušnje

Članek v New York Timesu [5] opozarja na nastajajoči trend, ko ljudje dajejo prednost izkušnjam pred otipljivimi oziroma vidnimi stvarmi. Tako porabljajo manj denarja za nepotrebne stvari (drage obleke, igrače, tehnika itd.) in več časa in denarja posvečajo stvarem, ki bogatijo njihovo življenjsko izkušnjo (ogled muzejev, tek, družina, prijatelji itd.).

Kakšne učinke pa bo imel ta trend? Zmanjša se lahko poraba potrošnikov in se s tem še bolj uspava gospodarstvo. Druga možnost je pojav kulture, ki daje resnično vrednost izkušnji. Po telefonu, v kiosku, v spletu, osebno, na delovnem mestu in doma. To je kultura, kjer bodo oblikovalci uporabniške izkušnje zares zaživeli.

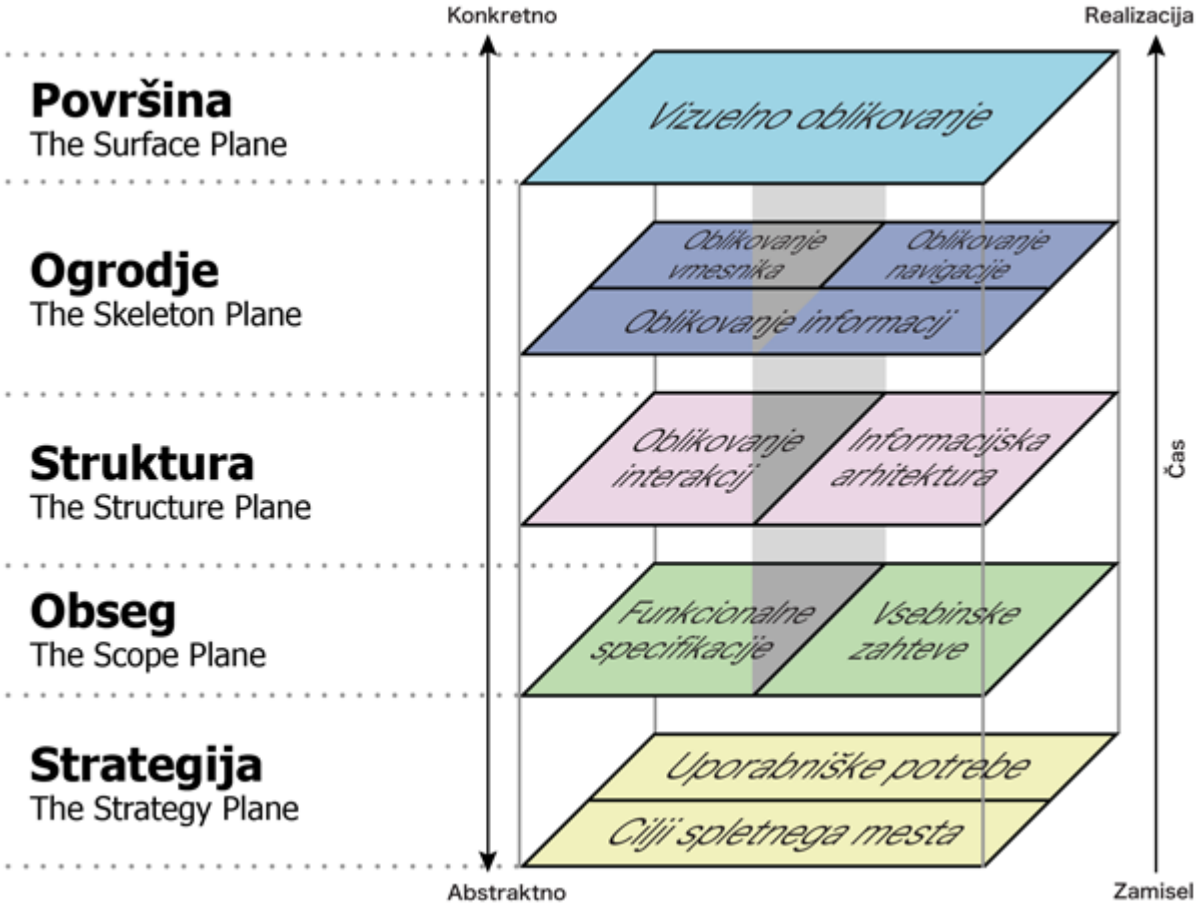
Na koncu avtor navede še nekaj tehnologij in trendov, ki bodo v prihodnosti krojile način, kako uporabniki dojemamo uporabniško izkušnjo:

- geste in večprstni vmesniki *(angl. gestural and multi-touch interfaces)*, na primer Microsoft Surface in podobni,
- družabna omrežja, na primer Facebook, Tweeter, Linked In in podobna,
- tablični računalniki, na primer Apple iPad in podobni,
- vse, kar je povezano z geolokacijo, na primer Google Maps, Facebook Places in podobno.

To so le napovedi, ki so lahko v veliko primerih napačne, vendar jih še vedno radi delamo. [4]

2.4 Področja uporabniške izkušnje

Zaradi različne in nejasne uporabe terminov na področju uporabniške izkušnje in njenega načrtovanja, je Jesse James Garrett leta 2000 izdelal in na svoji spletni strani objavil diagram, s katerim je področje jasno razdelil in definiral njegove elemente. Omenjeni diagram je pozneje, leta 2002, predstavil in podrobneje opisal v knjigi s pojmljivim naslovom »Elementi uporabniške izkušnje« (»The Elements Of User Experience«), kjer uporabniško izkušnjo razdeli na 5 področij oziroma ravni:



Slika 1- Prikaz petih področij oziroma ravni uporabniške izkušnje po avtorju Jesseju Jamesu Garrettu

1. Strategija (»The Strategy Plane«)
2. Obseg (»The Scope Plane«)
3. Struktura (»The Structure Plane«)
4. Ogrodje (»The Skeleton Plane«)
5. Površina (»The Surface Plane«)

Načrtovanje uporabniške izkušnje se po njegovem mnenju razlikuje glede na to, ali gledamo na splet kot na programski vmesnik ali pa ga gledamo kot na hipertekstni sistem povezav. Zato je omenjeni avtor v svoji knjigi uporabniško izkušnjo ločil na omenjena dva dela.

Na najnižji, strateški ravni se ne obremenjujemo s končnim videzom spletne strani ali spletne aplikacije, temveč le z njeno skladnostjo s strategijo. Pri tem moramo biti na vsakem koraku pozorni, da izpolnjuje vse zahteve naših uporabnikov. Na najvišji ravni (površina) pa se osredotočamo zgolj na podrobnosti končnega videza spletne strani ali spletne aplikacije. Vsaka raven je odvisna od ravni pod njo. Zato imajo napačne odločitve na strateški ravni ali pa preskok katere izmed preostalih ravni posledice vse do najvišje ravni.

V nadaljevanju bom predstavil in opisal vseh pet področij oziroma elementov uporabniške izkušnje, ki jih je Jesse James Garrett podrobno opisal v svoji, prej omenjeni knjigi.

2.4.1 Strategija

Najpogostejši krivec neuspešnosti spletnega nastopa ni ne tehnologija v ozadju spletne strani ali spletne aplikacije ne slaba uporabniška izkušnja, temveč dejstvo, da si, preden je bila napisana prva vrstica kode in preden je bila narisana prva črta končnega videza, nihče ni zastavil dveh osnovnih vprašanj:

- Kaj želimo s spletno stranjo ali spletno aplikacijo doseči?
- Kaj uporabnik od nje potrebuje in pričakuje?

Odgovor na prvo vprašanje nam opiše **cilje spletne strani**, odgovor na drugo vprašanje pa nam poda **uporabniške potrebe**, navade in pričakovanja. Cilji spletne strani ter uporabniške potrebe tako skupaj tvorijo strategijo spletne strani, ki je osnova vsake nadaljnje odločitve, ki jo bomo med načrtovanjem uporabniške izkušnje sprejeli.

2.4.1.1 Cilji spletne strani

Pri določanju ciljev je treba te predstaviti in se o njih pogovarjati s celotno ekipo, ki pri projektu deluje. V nasprotnem primeru bo vsak posameznik imel lastno predstavo o ciljih, kar pa seveda ni dobro. Pri določanju ciljev je treba iskati ravnovesje med preveč podrobnimi (»uporabnikom je treba ponuditi javanska komunikacijska orodja«) in preveč splošnimi (»spletna stran naj prinaša čim več denarja«). Glavno je, da so cilji merljivi, saj se lahko le tako prepričamo o uspešnosti njihovega doseganja.

2.4.1.2 Uporabniške potrebe in navade

Pri analiziranju uporabniških potreb in navad je najprej treba sprejeti dejstvo, da spletna stran ni namenjena nam, temveč našim uporabnikom. Najprej jih je treba razumeti in poznati njihove želje, šele tako jim lahko ponudimo in pripravimo spletno stran ali spletno aplikacijo z edinstveno uporabniško izkušnjo, ki jo bodo z veseljem obiskovali in brez velikega napora vedno znova uporabljali.

Proces analiziranja se začne z razdelitvijo uporabnikov s približno enakimi značilnostmi v manjše skupine, kar pripomore k lažjemu razumevanju njihovih potreb. Pri razvrščanju **uporabnike primerjamo** po teh dejavnikih:

- demografske značilnosti (starost, življenjski stan, izobrazba, prihodek),
- psihološke značilnosti (osebnostne značilnosti, vrednote, odnos in zanimanje do raznoraznih stvari, življenjski slog),
- odnos do tehnologije in spleta,
- čas, ki ga preživijo za računalnikom in na spletu,
- ali so računalniki del njihovega življenja,
- ali jim je delo za računalnikom v veselje.

Še posebno pa je treba biti pozoren na to, kako domači so uporabniki na predstavljenem področju. Tako mora biti na primer spletna stran ali spletna aplikacija za trgovanje z vrednostnimi papirji za začetnika veliko preprostejša in mora ponujati veliko manj naprednih možnosti kot pa tista za profesionalnega uporabnika. Seveda je velikokrat z eno spletno stranjo ali spletno aplikacijo nemogoče zadovoljiti celoten spekter uporabnikov. V tem primeru imamo dve možnosti. Ali vso energijo in čas posvetimo eni ciljni skupini ali pa za opravljanje iste naloge pripravimo več ločenih pristopov, primernih za posamezne skupine uporabnikov.

Osnovna potreba vsakega uporabnika pri končnem izdelku je njegova preprostost in uporabnost. Razumevanje drugih uporabniških potreb pa se začne z boljšim poznavanjem naših uporabnikov, pri čemer nam pomaga **področje raziskovanja uporabnika** *(angl. User Research)* z naslednjimi metodami:

- **Tržne raziskave** *(angl. Market research methods)* – kjer z anketami in intervjuji pridobivamo informacije o naših uporabnikih ter njihovem splošnem razumevanju in dojemanju posameznih elementov spletne strani (»Kaj vse uporabnik počne pri uporabi določene funkcije spletne strani?«, »Zakaj uporabnik uporablja določeno funkcijo spletne strani?«)
- **Povezano poizvedovanje** *(angl. Contextual inquiry)* – se navezuje na celotno skupino metod, ki skupaj tvorijo močno orodje, namenjeno razumevanju naših uporabnikov v luči njihovega vsakdanjega življenja. Z njo odkrijemo še tako skrite navade, ki jih z drugimi metodami ne bi odkrili. Slabost te metode je njena časovna in finančna zahtevnosta.
- **Analize naloge** *(angl. Task analysis)* – je metoda, s katero natančno raziščemo posamezne korake, ki jih uporabniki napravijo pri opravljanju določenih nalog. Te raziskave so lahko opravljene prek intervjujev, kjer uporabnik opiše svojo izkušnjo, ali pa z dejanskim opazovanjem uporabnika pri opravljanju nalog.
- **Testiranje uporabnika** *(angl. User testing)* **–** je najpogosteje uporabljena metoda raziskovanja uporabnika in njegovih navad. Tukaj ne gre za testiranje uporabnika samega, ampak za testiranje spletne strani ali spletne aplikacije ob njegovi uporabi.
- **Razvrščanje kartic** *(angl. Card sorting)* – je metoda, s katero raziščemo uporabniški vidik delitve vsebine spletne strani. Uporabniku razdelimo kartice z imenom, opisom in sliko, ki jih mora smiselno in po svoji logiki razvrstiti v skupine. Skupine so lahko pripravljene vnaprej ali pa jih uporabnik določi sam. Analiziranje rezultatov nam pomaga k boljšemu poznavanju uporabnika in njegovega razumevanja predstavljene vsebine.

Z omenjenimi metodami se pridobi ogromna količina informacij o uporabnikih, ki pa lahko kaj hitro zameglijo realne osebe, ki stojijo za celotno statistiko. Zato je zelo priporočljivo, da za posamezno skupino uporabnikov z istimi navadami ustvarimo izmišljenega predstavnika oziroma osebo, imenovano **persona**. Personi nato pripišemo osnovne osebne podatke, jo na kratko opišemo ter jo zavoljo večje realnosti opremimo s sliko realne osebe. Pri vseh

nadaljnjih odločitvah na višjih ravneh se nato vprašamo: »Kaj bi persona Tanja naredila v tej situaciji?«, ali pa »Kaj bi persona Klemen potrebovala?«. Persone nam tako pomagajo, da na vsakem koraku načrtovanja uporabniške izkušnje mislimo na svoje uporabnike.

2.4.1.3 Strateški dokument

Na koncu cilje spletne strani ter uporabniške potrebe zberemo in predstavimo v strateškem dokumentu. Ta naj bo kratek, jedrnat ter dostopen celotni ekipi, ki deluje pri projektu. Ta je na koncu namreč tista, ki je odgovorna za njeno realizacijo.

Izdelava strategije mora vedno biti prvi korak procesa načrtovanja uporabniške izkušnje, kar pa ne pomeni, da mora biti končana, še preden se lahko projekt nadaljuje. Strategija se lahko v času izdelave razvija in izpopolnjuje, in to je tudi priporočljivo.

2.4.2 Obseg

Na strateški ravni smo se spraševali: »Zakaj izdelujemo spletno stran?«, na tej ravni pa se vprašajmo: »Kaj vse bomo na spletni strani imeli?« Z jasno predstavo o svojih željah in željah naših uporabnikov se na tej ravni osredotočamo na načine, kako na prejšnji ravni določene cilje spletne strani in uporabniške navade spremenimo v dejanske zahteve po vsebini in po funkcionalnostih naše spletne strani. Te zahteve je treba dokumentirati predvsem iz dveh razlogov:

1. **Zato, da vemo, kaj izdelujemo.**
   Če temeljito opišemo in zapišemo, kaj točno je cilj našega delovanja in kaj točno izdelujemo, potem vsi člani ekipe vedo, kaj morajo narediti in kdaj bodo cilji doseženi.
2. **Zato, da vemo, česa ne izdelujemo.**
   V procesu izdelave projekta se nam navadno pojavijo ideje o novih zmožnostih in zahtevah, ki pa ni nujno, da so v skladu z izdelano strategijo projekta. Teh idej nikakor ni treba zavreči, saj se bodo lahko pojavile v prihodnjih različicah in bodo tako postale del načrtov za prihodnost.

Če vsebinskih in funkcionalnih zahtev ne evidentiramo in sproti ne nadzorujemo njihove realizacije, bo projekt najverjetneje postal kot snežna kepa, ki se vali po klancu navzdol, pri tem pa postaja vse večja ter se vse težje ustavi. Čeprav novo dodana zahteva na prvi pogled ne

porabi veliko časa in sredstev, pa ta v kombinaciji z vsemi drugimi zahtevami projekt hitro vrže iz nadzora, podaljša čas izdelave in preseže njegove predvidene stroške.

Na tej ravni se tudi prvič pokaže razlika med spletom kot programskim vmesnikom in spletom kot hipertekstnim sistemom. Na programski strani nas zanimajo predvsem funkcionalnosti spletne aplikacije, ki se navadno zapisujejo v dokument z imenom **funkcionalne zahteve**. Na hipertekstni strani pa nas zanima predvsem sama vsebina spletne strani, ki pa se navadno zapisuje v **urejevalnike spletne vsebine** *(angl. Content Management System).*

Vsebinske zahteve so pogosto tesno povezane s funkcionalnimi zahtevami. Dandanes, ko se večino vsebine na spletnih straneh ureja prek sistemov CMS, morajo ti podpirati vse zahtevane načine vnosa in prikaza vsebine (vnos vsebine v več jezikih in formatih, odobritev vsakega vnosa vsebine nadrejenega itd.). Na drugi strani pa so enako tudi funkcionalne zahteve tesno povezane z vsebinskimi, saj je treba uporabnika z razumljivimi sporočili obveščati o morebitnih napakah v sistemu, mu posredovati navodila za uporabo posameznih funkcij itd.

2.4.2.1 Zbiranje zahtev in njihovo razporejanje

Vprašati uporabnika, kaj dejansko želi, je najboljši način zbiranja vsebinskih in funkcionalnih zahtev. Pri tem si pomagamo z metodami raziskovanja uporabnika *(angl. User Research),* ki sem jih predstavil na prejšnji ravni. Ne glede na to, ali so glavni vir pridobivanja zahtev osebe znotraj podjetja ali končni uporabniki, se bodo zbrane zahteve po zmožnostih spletne strani ali aplikacije na koncu razvrstile v tri glavne kategorije. Zmožnosti, za katere uporabniki trdijo, da jih potrebujejo, zmožnosti, ki jih dejansko potrebujejo, ter zmožnosti, za katere ne vedo, da jih potrebujejo. Skupinski pogovor s predstavniki različnih področij podjetja ter s predstavniki končnih uporabnikov na tako imenovanem sestanku zbiranja idej s tehniko viharjenja možganov *(angl. brainstorming),* je prav tako lahko zelo uspešen, saj pripomore k razmišljanju o možnostih, na katere še nihče ni pomislil.

Zbiranje zahtev je v večini primerih iskanje načina, kako uporabniku, ki je na primer že odločen za nakup izdelka ter le še izbira spletno stran za dejanski nakup, odstraniti čim več ovir in mu karseda olajšati nakup. Pri tem si lahko pomagamo z ustvarjanjem **scenarijev** *(angl. scenario)* v katerih nastopajo persone, ki smo jih z namenom boljšega razumevanja uporabniških potreb definirali na strateški ravni. Scenarij je kratka, preprosta pripoved, ki

opisuje, kako bi persona izpolnila določeno uporabniško potrebo oziroma nalogo. S scenariji lahko pridemo do novih spoznanj o potencialnih vsebinskih in funkcionalnih zahtevah, ki bi lahko še bolje zadovoljile uporabniške potrebe. Seveda pa je vedno treba pregledati tudi konkurenco in njeno delovanje na trgu, saj so skoraj gotovo tudi oni razmišljali o tem, kako najbolje zadovoljiti uporabnika in njegove potrebe.

Zbiranje idej za potencialne vsebinske in funkcionalne zahteve ni težko delo. Vsak posameznik, ki deluje pri projektu, ima vsaj eno idejo za novo zmožnost spletne strani ali spletne aplikacije. Težji del je razporejanje, katere vsebinske in funkcionalne zahteve spletne strani ali aplikacije bodo vključene oziroma definirane v obsegu projekta. Glede na to, da je obseg grajen na podlagi strategije, moramo odločitve o dodajanju novih zahtev sprejemati na podlagi njihovega izpolnjevanja strateških ciljev.

2.4.2.2 Definiranje funkcionalnih zahtev

Nekaj povsem vsakdanjega pri delu s tehnologijo je, da se funkcionalnosti izdelka v času njegove izvedbe spreminjajo, dodajajo in brišejo glede na prvotno zapisane zahteve. Velikokrat se zgodi, da funkcionalnost, za katero smo mislili, da bo delovala, na koncu ne deluje tako, kot smo pričakovali. To pa seveda ni razlog, da opustimo pisanje dokumenta s funkcionalnostnimi zahtevami. Ravno nasprotno, med izvedbo je treba funkcionalnostne zahteve vedno znova posodabljati ter skrbeti, da je dokument usklajen z razvojem.

Velja pa izpostaviti nekaj pravil, ki se jih je treba držati pri pisanju zahtev:

1. Bodimo **pozitivni** in ne navajajmo slabih stvari, ki jih spletna stran ali spletna aplikacija ne sme imeti, temveč navedimo, kaj storiti, da do slabih stvari ne pride.
2. Bodimo **natančni** in pri tem pustimo karseda malo možnosti za različne interpretacije napisanega.
3. **Izogibajmo se subjektivnemu načinu pisanja** ter namesto stavka »Stran mora imeti lep in simpatičen videz« raje uporabimo »Stran mora slediti celostni grafični podobi podjetja«.

2.4.2.3 Definiranje vsebinskih zahtev

Ko govorimo o vsebinskih zahtevah, navadno pomislimo zgolj na besedilo, pozabimo pa, da vsebino sestavljajo tudi slike, avdio in video. Pisanje in izdelava vsebine ni lahko delo in zahteva veliko časa ter porabi veliko sredstev. Zato je treba za lažji nadzor nad njihovo porabo najprej identificirati, katere tipe vsebine potrebujemo. Določene vsebinske zahteve se

morda na prvi pogled zdijo zelo smiselne, dokler nekdo ne postane odgovoren za njihovo pisanje in izdelavo.

Pri definiranju vsebinskih zahtev je zaželeno podati okvirne ocene velikosti posameznega tipa vsebine, ki ga bomo imeli na strani. Tako na primer določimo, koliko besed je primernih za določeno stran, kakšna naj bo dimenzija slik na spletni strani, v kakšnem formatu naj bo objavljen video, kakšne bodo največje velikosti datotek. Prav tako je treba glede na cilje spletne strani ter željo uporabnikov po količini in času vsebinskih sprememb določiti frekvenco posodabljanja vsebine na spletni strani. Pri tem pa je treba iskati ravnovesje med pričakovanji uporabnikov in razpoložljivimi sredstvi.

2.4.3 Struktura

Z vsemi zbranimi vsebinskimi in funkcionalnimi zahtevami imamo zdaj jasno sliko, kaj vse bo vseboval končni izdelek. Ker pa vsebinske in funkcionalne zahteve ne povedo ničesar o tem, kako bodo posamezni deli sestavljeni v delujočo celoto, moramo na tej ravni razviti konceptualno strukturo spletne strani ali spletne aplikacije.

Na podlagi delitve spleta na splet kot programski vmesnik in splet kot hipertekstni sistem ločimo izdelavo strukture na področje **načrtovanja interakcij** *(angl. interaction design)*, ki se pojavlja pri razvoju aplikacij, ter na področje **načrtovanja informacijske strukture** *(angl. information architecture)*, ki pa se uporablja predvsem pri razvoju vsebine spletnih strani. Pri načrtovanju interakcij se osredotočimo na možne načine izvajanja in opravljanja nalog, pri načrtovanju informacijske strukture pa na možne načine izražanja in posredovanja vsebine uporabnikom. Pri obeh pa gre za razumevanje uporabnikov, njihovih načinov delovanja ter njihovega načina razmišljanja. Z vgradnjo tega razumevanja v strukturo spletne strani ali spletne aplikacije omogočimo uporabnikom karseda uspešno uporabniško izkušnjo.

2.4.3.1 Načrtovanje interakcij

Pri načrtovanju interakcij se ukvarjamo predvsem z raziskovanjem možnega vedenja uporabnika in s prilagajanjem sistema njegovemu vedenju. Interakcijo med uporabnikom in računalnikom lahko primerjamo s plesom, ki bo zabaven in prijeten le, ko bosta oba sodelujoča, namesto da bi si med seboj stopala po prstih, znala predvidevati korake drug drugega.

Konceptualni model prikazuje uporabniško dojemanje obnašanja sestavljenih interaktivnih komponent. Primer konceptualnega modela nakupovalne košarice je zaboj, v katerega dajemo in iz katerega jemljemo artikle. Ta model je tako široko prepoznaven in uporabljen v spletu, da ima pripisan status **dogovorjenega** *(angl. convention)*. Z uporabo dogovorjenega konceptualnega modela uporabnikom olajšamo uporabo njim neznane spletne strani.

Velik del vsakega načrtovanja interakcij pa je ravnanje ob **uporabniških napakah**. Kako sistem reagira ob uporabnikovi napaki ter kaj lahko sistem naredi, da prepreči pojavljanje teh napak? Najboljši način obrambe pred napakami je priprava sistema, kjer napake sploh niso možne. Primer take obrambe je avto z avtomatskim menjalnikom. Vžig avta z menjalnikom v prestavi avto sunkovito premakne naprej, pri tem pa se lahko menjalnik tudi poškoduje. To napako so odstranili tako, da se avto vključi samo, ko menjalnik ni v prestavi. Žal pa je malo sistemov, ki se jih lahko načrtuje tako, da je pojavljanje napak nemogoče. Zato je najboljši način, da se karseda oteži pojavljanje napak. Vendar bo tudi tako do nekaterih napak še vedno prišlo in jih bo nemogoče preprečiti. Takrat je naloga sistema, da uporabniku čim preprosteje pomaga pri odpravi nastale napake. Najboljši primer odpravljanja napak, ki jih sistem ne more zaznati in ujeti, je slavna funkcija »razveljavi« *(angl. undo)*. Slabši in v veliko primerih nadležen prijem za preprečevanje napak pa je znano pogovorno okno »Ali ste prepričani?« *(angl. Are you sure?)*.

2.4.3.2 Informacijska arhitektura

S pojmom informacijska arhitektura mislimo predvsem na izdelavo organizacijskih in navigacijskih shem, ki bodo uporabnika učinkovito in preprosto popeljala skozi vsebino spletne strani. Problemi informacijske arhitekture pogosto zahtevajo izdelavo kategorizacijskih shem, ki bodo sovpadale s cilji spletne strani, z uporabnikovimi navadami ter z vsebino na spletni strani. Izdelave kategorizacijskih shem se lahko lotimo na dva načina:

- **od vrha navzdol** – začnemo z najširšimi kategorijami, ki zajemajo potencialne vsebine in funkcionalnosti za doseganje strateških ciljev, nato pa to kategorijo razbijemo na manjše logične podkategorije.
- **od spodaj navzgor** – začnemo z analiziranjem vseh materialov (vsebinskih in funkcionalnih), ki jih najprej združimo v nizkonivojske kategorije, nato pa jih postopoma proti vrhu združujemo v visokonivojske kategorije.

Spletne strani so kot živa bitja, saj zahtevajo stalno skrb in hranjenje. Sčasoma tako postajajo večja in se nenehno spreminjajo. Od učinkovite strukture se tako pričakuje sposobnost prilagajanja rasti spletne strani in njenim spremembam. Število kategorij ne igra nobene vloge, glavno je, da so te smiselno poimenovane, razvrščene ter da izpolnjujejo uporabnikova pričakovanja. Prav tako ni pomembno število korakov za uspešno opravljeno nalogo ali število klikov za prihod na določen del strani. Glavna pokazatelja kakovostne strukture je smiselnost zaporedja korakov oziroma klikov in to, ali je uporabnik korakom sledil naravno in intuitivno.

Osnovna enota informacijske strukture je **vozlišče**, ki pomeni delček ali pa celotno skupino informacij. Tako lahko z vozliščem opišemo posamezno stran ali pa celotno skupino strani. Odvisno od velikosti in obširnosti spletne strani. Vozlišča pa lahko razvrstimo na različne načine:

- **hierarhična (drevesna) struktura** – vozlišča imajo svojega očeta in svojega sina. To je najpogosteje uporabljen strukturni koncept.
- **matrična struktura** – omogoča uporabniku gibanje iz vozlišča v vozlišče v dveh ali več dimenzijah. Matrične strukture se uporabljajo predvsem, kadar je treba zadovoljiti več uporabnikov pri sprehajanju po spletni strani (iskanje izdelkov po barvi, velikosti, obliki …).
- **organska struktura** – ne sledi nobenemu konsistentnemu vzorcu. Primerna so za raziskovanje vsebinskih sklopov, ki med seboj nimajo jasnega odnosa. V primerjavi s prejšnjima dvema strukturama pri organski strukturi uporabnik ne ve točno, kje v strukturi je. Primerna je za spletne strani, kjer je pomembno prosto raziskovanje, na primer pri izobraževalnih straneh ali straneh, namenjenih zabavi.
- **sekvenčna struktura** – je najbolj razširjena na fizičnih medijih, kot so na primer knjige, avdio in video. Sekvenčni tok besed, slike in zvoka je najosnovnejši tip informacijske strukture. V spletu se uporablja pri manjših strukturah, namenjenih individualnim člankom, slikovnim galerijam …

Vozlišča so v informacijski strukturi razvrščena glede na **organizacijska načela**. Ta pomenijo kriterije, s katerimi določamo vozlišča, ki sodijo skupaj, in vozlišča, ki sodijo narazen. Na različnih nivojih strukture spletne strani veljajo različna organizacijska načela. Pri spletni strani velike korporacije bomo na vrhu naše drevesne strukture uporabili organizacijsko načelo, ki bo v ospredje postavilo obiskovalce. Tako bomo imeli kategorije, kot so

»Porabniki«, »Podjetniki«, »Investitorji« itd. Kje drugje pa bodo za organizacijsko načelo uporabljena geografska področja s kategorijami, kot so »Evropa«, »Amerika«, »Azija« itd.

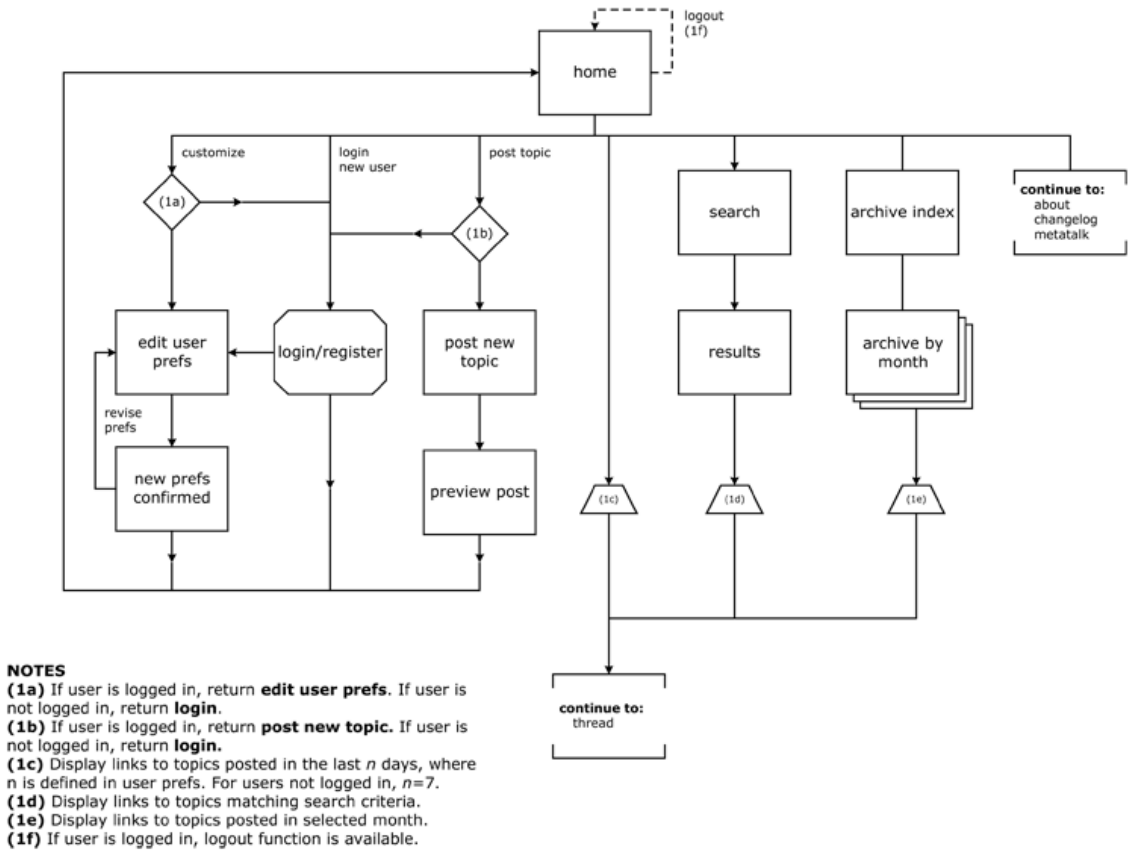
Atribute, ki pomenijo skupek organizacijskih načel, imenujemo **vidiki**. Pri spletni strani z informacijami o avtomobilih lahko uporabimo vidik, ki bo informacije razvrstil po teži avtomobila. Za končnega uporabnika oziroma potencialnega kupca ta razvrstitev ni najboljša, saj ga bolj kot to zanima znamka, model avtomobila itd. Na drugi strani pa je pri uporabnikih, ki se profesionalno ukvarjajo s transportom vozil prek oceana, ta podatek še kako pomemben. Pri uporabi vidikov je treba biti pozoren na izbor pravih vidikov ter prave količine vidikov. Nasičenje spletne stran z ogromno količino možnih atributov lahko zelo slabo vpliva na uporabnika, saj mu tako naložimo ogromno breme pri odločanju, katerega naj uporabi.

Tudi če struktura spletne strani v celoti odraža način uporabnikovega razmišljanja o predstavljeni tematiki, se pri uporabi njim nerazumljivih izrazov na spletni strani ne bodo znašli. Zato je še kako pomembno uporabiti izraze in terminologijo, ki jih uporabniki razumejo, in da te dosledno uporabljamo na celotni spletni strani.

2.4.3.3 Arhitekturni diagram

Najpomembnejši dokument za predstavitev informacijske arhitekture ter načrtovanja interakcije je **arhitekturni diagram**. Ni potrebno, da diagram vsebuje vse povezave na vseh straneh, glavno je le, da vsebuje relacije in odnose med kategorijami ter med posameznimi koraki določene interakcije.

Jesse James Garrett je izdelal sistem, ki pripravi diagram strukture spletne strani, imenovan »Visual Vocabulary«. Po prvi objavi tega sistema v spletu, leta 2000, so ga mnogi informacijski arhitekti in načrtovalci interakcij že privzeli in ga s pridom uporabljajo.



Slika 2 — Diagram strukture spletne strani, imenovan »Visual Vocabulary«, katerega avtor je Jesse James Garrett

2.4.4 Ogrodje

Struktura da obliko vsebinskim in funkcionalnim zahtevam, ki izhajajo iz strategije. Na ravni ogrodja pa to strukturo še izpopolnimo in konkretiziramo z njej primernim vmesnikom, navigacijo in informacijskim načrtom. Na tej ravni se tako ukvarjamo predvsem s posameznimi stranmi spletnega mesta in njenimi komponentami.

Pri aplikacijah ogrodje definiramo z **načrtovanjem vmesnika**, katerega področje zajema izbor gumbov, polj in drugih komponent vmesnika. S tem uporabniku omogočimo opravljanje posameznih nalog ter interakcijo s funkcionalnostmi, ki smo jih definirali v zahtevah ter jih strukturirali pri načrtovanju interakcij.

Pri spletnih straneh pa informacijske prostore definiramo z **načrtovanjem navigacije**, ki je specializirana oblika načrtovanja vmesnika. S tem uporabniku omogočamo sprehajanje po

spletni strani ter mu razjasnimo pogled na strukturo, ki smo jo na podlagi vsebinskih zahtev definirali z informacijsko arhitekturo.

Pri **informacijskem načrtovanju** se tako pri spletnih aplikacijah kot tudi pri spletnih straneh ukvarjamo z načini predstavitve informacij za uspešno komuniciranje z obiskovalci oziroma uporabniki. Načrtovanje vmesnika ter načrtovanje navigacije ne moreta biti popolnoma uspešna brez podpore dobrega informacijskega načrta. Ravno zato so vsa tri področja med seboj povezana bolj kot katera koli druga področja ostalih ravni.

Pri načrtovanju vmesnika, navigacije ali česa drugega je zelo zaželena uporaba **dogovorjenih standardov**, izogibati pa se je treba različnim **metaforam**, ki bi lahko zmedle uporabnika. Tako bi na primer pri načrtovanju vmesnika, ki zahteva prikaz številčnice za vpis telefonske številke, uporabili dogovorjen standard prikaza matrike 3 x 4 številke. Pri uporabi nestandardnega prikaza bi uporabnika zmedli in od njega zahtevali vnovično učenje in privajanje na nekaj, česar je že navajen od prej. Tudi z uporabo napačnih metafor, kot je na primer ikona kave za prikaz klepetalnice, lahko uporabniku otežimo sprehod po spletni strani. Namesto da bi se usmeril na vsebino, se tako ukvarja z ugibanjem pomena posameznih napačno uporabljenih metafor.

2.4.4.1 Načrtovanje vmesnika

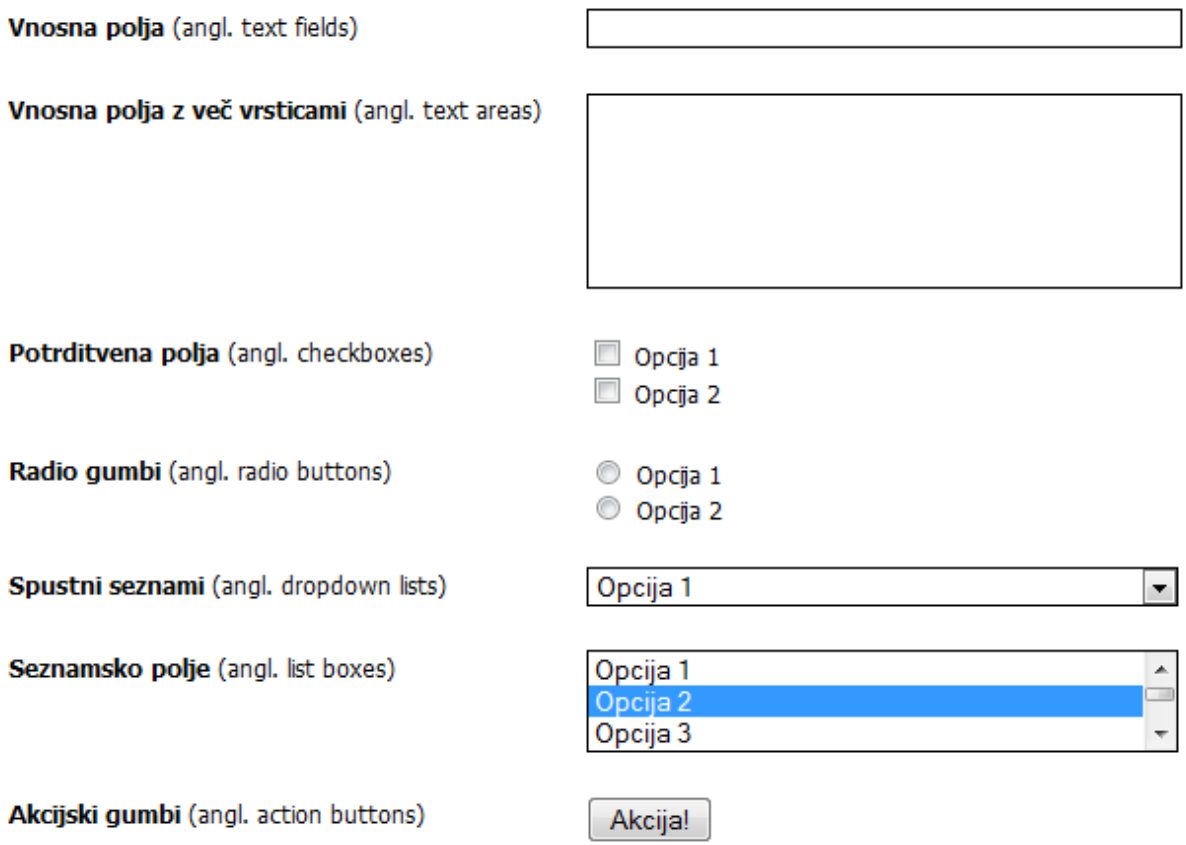
Uspešni vmesniki so tisti, pri katerih uporabniki že na prvi pogled opazijo pomembne stvari. Nepomembni elementi so tako skriti v ozadju ali pa jih sploh ni. Eden največjih izzivov pri načrtovanju vmesnika je ugotoviti, česa uporabnik ne potrebuje.

Če želimo imeti stabilno aplikacijo, je treba pri programiranju te vedno upoštevati ekstremne oziroma robne pogoje *(angl. edgecases)*. Treba je preučiti in upoštevati vse možne načine uporabe določene funkcionalnosti. Če pa želimo imeti uspešen in uporabniku prijazen vmesnik, ne smemo posvečati pozornosti malemu številu uporabnikov, temveč moramo vmesnik prilagoditi mišljenju večine.

Dobro načrtovan vmesnik razume in pozna uporabnikove poti uporabe, zato naredi elemente vmesnika na tej poti karseda dostopne in preproste za uporabo. To pa seveda ne pomeni, da mora biti gumb, za katerega pričakujemo večino klikov, največji element vmesnika. Pri njegovem načrtovanju lahko uporabimo veliko trikov, ki uporabniku pomagajo pri doseganju njihovega cilja. Tako lahko na primer premišljeno označimo privzete možnosti ob prvem prikazu vmesnika ali pa si zapomnimo označene možnosti prejšnjega obiska.

HTML in Flash sta vodilni tehnologiji v spletu, prek katerih se vmesnik uporabniku najpogosteje predstavi. Čeprav je bil HTML v začetku namenjen preprostemu posredovanju informacij, so je hitro spoznali njegovo interaktivna moč. Tako je kmalu po njegovi objavi pridobil osnovne elemente vmesnika:

- **potrditvena polja** *(angl. checkboxes)* omogočajo neodvisen izbor ponujenih možnosti;
- **radio gumbi** *(angl. radio buttons)* omogočajo izbor ene izmed ponujenih možnosti;
- **vnosna polja** *(angl. text fields)* omogočajo vnos besedila;
- **spustni seznami** *(angl. dropdown lists)* imajo enako funkcionalnost kot radio gumbi, le da zavzamejo manj prostora. Navadno se jih uporablja pri hranjenju večjega števila možnosti;
- **seznamska polja** *(angl. list boxes)* imajo enako funkcionalnost kot potrditvena polja, le da zavzamejo manj prostora (na račun drsnika v seznamu). Prav tako se uporabljajo pri hranjenju večjega števila možnosti;
- **akcijski gumbi** *(angl. action buttons)* omogočajo veliko stvari, najpogosteje pa se uporabljajo za pošiljanje informacij, ki jih je uporabnik navedel v zgornjih elementih vmesnika.



Slika 3 – Primeri elementov vmesnika

Flash prav tako ponuja uporabo vseh zgoraj navedenih elementov vmesnika, a ker je v osnovi namenjen animiranju, se lahko vmesnik glede na uporabnika bolje odziva.

Informacijsko načrtovanje ima pomembno vlogo pri problemih načrtovanja vmesnika, predvsem kadar ni treba le zbrati informacij od uporabnika, temveč tudi posredovati informacije do uporabnika. Sporočila o napakah ter posredovanje navodil sta tipična problema informacijskega načrtovanja pri ustvarjanju uspešnega vmesnika.

2.4.4.2 Načrtovanje navigacije

Načrtovanje navigacije mora zadovoljiti tri cilje:

1. Uporabniku mora omogočati prehod od ene do druge točke na spletni strani.
2. Jasno mora ponazarjati relacije med elementi, ki jih vsebuje.
3. Jasno mora ponazarjati relacije med stranjo, kjer je uporabnik trenutno, in drugimi stranmi v navigaciji.

Zelo je pomembno, da vsaka stran spletne strani uporabniku jasno sporoča, kje je in kam vse lahko odide.

Spletne strani ponujajo različne navigacijske sisteme, vsak od njih pa opravi točno določeno nalogo pri sprehajanju uporabnika po spletni strani. Najbolj razširjeni navigacijski sistemi so:

- **globalna navigacija** *(angl. global navigation)* − ni nujno, da se pojavlja na vsaki strani spletne strani. Vsebuje glavne točke, ki bi jih uporabnik potreboval pri sprehodu z enega dela spletne strani na drugega. Uporabnik lahko prek globalne navigacije pride, kamorkoli želi.
- **lokalna navigacija** *(angl. local navigation)* − omogoča uporabniku dostop do bližnjih vsebin v arhitekturi. V hierarhični strukturi to pomeni dostop do starša, sorodnika ali otroka vozlišča, v katerem je. Če arhitektura sledi načinu razmišljanja uporabnikov, bo lokalna navigacija najbolj obiskan navigacijski sistem.
- **pomožna navigacija** *(angl. supplementary navigation)* ponuja bližnjice do vsebinsko povezanih strani, ki niso hitro dostopne prek globalne ali lokalne navigacije.
- **kontekstna navigacija** *(angl. contextual navigation)* je vstavljena v vsebino spletne strani. Če se vrnemo na strateški nivo, bo smiselnost in uspešnost kontekstne navigacije odvisna od razumevanja uporabnikov in njihovih potreb.

- **služnostna navigacija** *(angl. courtesy navigation)* omogoča uporabniku dostop do koristnih vsebin, ki pa jih ne potrebuje pogosto. To so na primer povezave do kontaktnih informacij, formularjev za sporočanje komentarjev, splošnih pogojev uporabe itd.

Oddaljene navigacije so navigacijski sistemi, ki niso vključeni v strukturo spletne strani, ampak stojijo sami, neodvisno od druge vsebine spletne strani. K njim se uporabniki navadno zatekajo, kadar so razočarani nad drugimi navigacijskimi sistemi ter se v njih preprosto ne znajdejo. Med oddaljene navigacijske sisteme štejemo:

- **načrt spletne strani** *(angl. site map)* − omogoča uporabniku celovit pregled nad celotno arhitekturo spletne strani. Načrt spletne strani je navadno hierarhično urejen. Ne vsebuje več kot dve stopnji ter vsebuje povezave do posameznih strani.
- **kazalo** *(angl. index)* je abecedno urejen seznam tem s povezavami do ustreznih strani, podobno kot kazalo v knjigi. Ta navigacijski sistem je najučinkovitejši pri spletnih straneh, ki zajemajo ogromno količino raznolike vsebine. Pri večini drigih spletnih strani pa zadostuje dobro izdelana arhitektura ter načrt spletne strani.

2.4.4.3 Informacijsko načrtovanje

Pri informacijskem načrtovanju iščemo načine, kako predstaviti informacije, da jih bodo uporabniki kar najlažje razumeli in uporabljali.

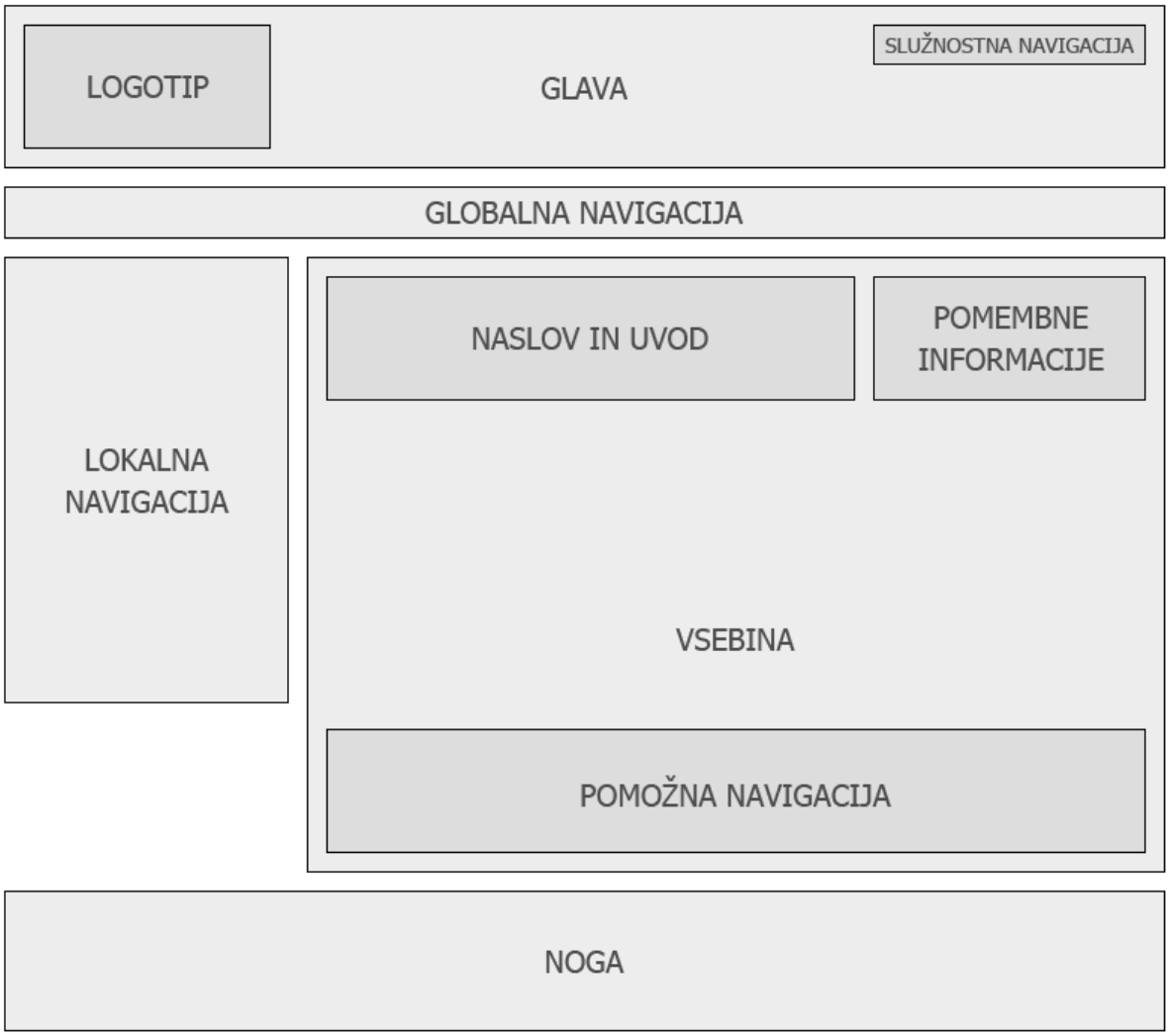
Informacije lahko predstavimo vizualno (s tortnim ali stolpčnim grafom, z uporabo lupe ali daljnogleda za predstavitev iskalnika), lahko jih grupiramo ter razvrščamo v smiselne skupine ali pa jih predstavimo zgolj z besedilom. Cilj je predstaviti informacije tako, da se kar najbolj približamo razmišljanju uporabnikov.

Pri tem je treba izpostaviti funkcijo **označevanja** *(angl. wayfinding)*, ki jo skupaj tvorita informacijsko načrtovanje in načrtovanje navigacije. Ideja prihaja iz realnega sveta, kjer na primer v parkirnih hišah uporabljajo barve za ločitev posameznih parkirnih sektorjev. Dobro označevanje v spletu uporabniku omogoča, da hitro dobi jasno predstavo, kje je, kam lahko gre in katere odločitve ga bodo pripeljale bližje njegovemu cilju. Uporabniku lahko prek različne uporabe barv, ikon in tipografije lažje ponazorimo, kje na spletni strani je.

2.4.4.4 Žični model

Pri **načrtu spletne** strani se združijo informacijsko načrtovanje, načrtovanje vmesnika in načrtovanje navigacije ter tako tvorijo enotno in povezano ogrodje. **Načrt spletne** strani torej vsebuje:

- vse izbrane **navigacijske sisteme**, zasnova katerih mora izražati različne poglede na arhitekturo spletne strani;
- vse potrebne **elemente vmesnika** za delovanje vseh funkcionalnosti spletne strani
- **informacijski načrt**, ki podpira navigacijske sisteme, elemente vmesnika ter vsebino same spletne strani.



Slika 4 – Prikaz žičnega modela

Načrt spletne strani je podrobno opisan v dokumentu, imenovanem **žični model** *(angl. wireframe)*. Žični model je oskubljena slika vseh komponent spletne strani in načina njihovega medsebojnega povezovanja. Za manj obsežne spletne strani zadostuje že en žični model, ki služi kot predloga za vse druge strani. Za kompleksnejše spletne strani pa potrebujemo več različnih žičnih modelov, od katerih vsak predstavlja svoj tip strani.

Žični modeli so potreben prvi korak pri procesu nastajanja vizualnega videza spletne strani. Vsi razvijalci, vključeni v proces razvoja projekta, se bodo na določeni točki obrnili nanj. Odgovorni za strategijo, obseg in strukturo se lahko nanj obrnejo za potrditev, da končni izdelek zadovoljuje njihova pričakovanja. Prav tako lahko razvijalci, odgovorni za izdelavo spletne strani ali spletne aplikacije, v žičnih modelih iščejo odgovore o delovanju končnega izdelka.

Glavna vrednost žičnih modelov pa je v njihovi integraciji vseh treh elementov ravni ogrodja:

- **načrtovanje vmesnika** − prek izbora in razvrstitve elementov vmesnika,
- **načrtovanje navigacije** − prek izbora in definiranja osnovnih navigacijskih sistemov,
- **informacijsko načrtovanje** − prek postavitve informacijskih komponent ter določanja njihovih pomembnosti.

Z združitvijo vseh treh elementov v en dokument žični model definira ogrodje, zgrajeno na podlagi definirane strukture, hkrati pa nakazuje pot proti končnemu vizualnemu videzu.

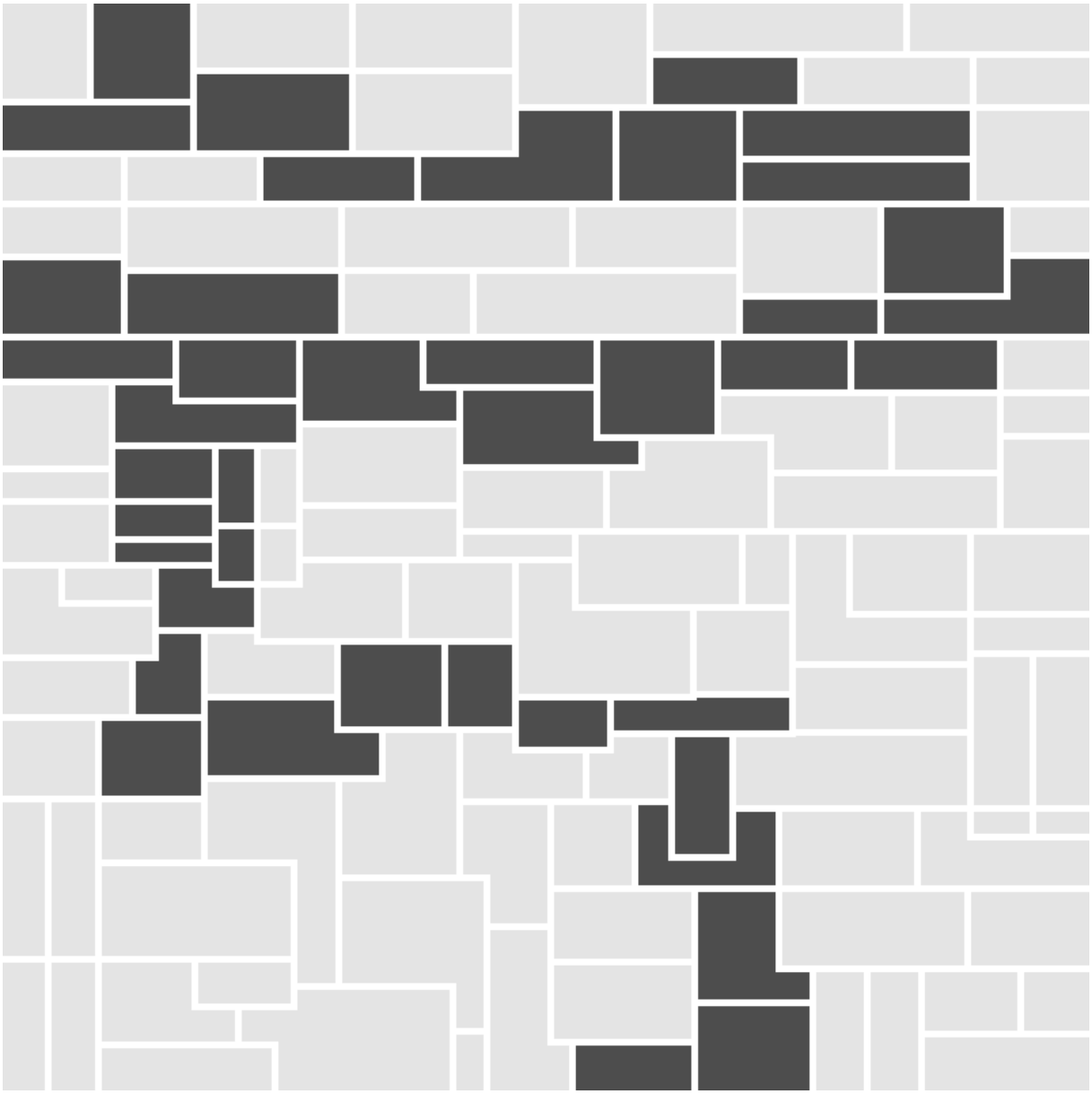
2.4.5 Površina

Na najvišji ravni pozornost posvečamo tistim vidikom spletne strani, ki jih bodo obiskovalci najprej opazili. To je sam videz spletne strani ali spletne aplikacije. Tukaj se združijo vsebina, funkcionalnosti in estetika in tako tvorijo končni videz, ki bo zadovoljil vse cilje, definirane na prejšnjih ravneh.

Pri informacijskem načrtovanju smo določili in razporedili skupine informacij, pri **vizualnem načrtovanju** pa določimo, kako naj bo ta razporeditev predstavljena vizualno. Velja splošno prepričanje, da je vizualno načrtovanje stvar estetike in da ima vsak posameznik drugačen okus. To seveda drži, kar pa ne pomenim da morajo vse odločitve o videzu temeljiti na tem, kaj je vsem sodelujočim videti lepo. Pomembnejše je, kako videz pripomore k uspešnosti določene komponente, kako uspešno podpira cilje, definirane na prejšnjih ravneh, ter ali videz strani uničuje strukturo in otežuje razlikovanje med posameznimi deli spletne strani.

2.4.5.1 Sledenje gibanju očesa

Najpreprostejši način ocene videza spletne strani je sledenje gibanju očesa ob pogledu nanjo. Kateri element pritegne največ pozornosti? Ali največ pozornosti pritegne element, ki je pomemben za doseganje strateških ciljev spletne strani? Pri raziskavah lahko uporabljamo opremo za **sledenje gibanja očesa** *(angl. eyetracking equipment)*, s katero zaznamo, kaj točno uporabnik gleda ter kašni poti sledi njegov pogled. Obstajajo pa tudi cenejši pristopi, ki v primerjavi z analizami očesnega gibanja ne postrežejo z najnatančnejšimi rezultati, vendar v večini primerih zadostujejo. Tako se lahko zanašamo na vprašanja, ki jih postavimo drugim obiskovalcem, ali pa se zanašamo na lasten pogled spletne strani. Pri tem najbolj učinkuje pogled na zaslon z druge strani sobe ali pa neoster pogled z rahlo priprtimi očmi.



Slika 5 - Sledenje gibanju očesa

Nasploh obiskovalci sledijo dokaj konsistentnemu vzorcu gibanja oči, saj je to gibanje nezavedno in instinktivno. Uspešno načrtovan videz spletne strani bo imel pri gibanju oči po spletni strani dve pomembni lastnosti:

- prvič, gibanje mora biti tekoče, brez skakanja gor in dol po različnih izstopajočih elementih,
- drugič, uporabniku mora ponujati neke vrste »voden ogled« vseh možnosti, brez prevelikega poudarka na podrobnostih. Seveda morajo te možnosti podpirati cilje in naloge, ki jih želi uporabnik opraviti.

2.4.5.2 Kontrast in enotnost

Pri vizualnem načrtovanju je glavno orodje za usmerjanje pozornosti **kontrast**. Z njim pritegnemo pozornost na bistvene elemente vmesnika ter pomagamo uporabniku pri razumevanju relacij med navigacijskimi elementi spletne strani. Sprememba kontrasta mora biti dovolj opazna, tako da uporabnik takoj zazna, da je ta namenjen sporočanju nečesa. Ob veliki podobnosti dveh elementov bo uporabnik začel razmišljati: »Zakaj sta si različna? Bi morala biti enaka? Morda je zgolj napaka? Bi moral tukaj kaj opaziti?«.

Slika 6 - Primer slabo uporabljenega kontrasta na oglasu za Aspirin podjetja Bayer

Ohranjati **enotnost** pri videzu je prav tako pomemben element pri zagotavljanju učinkovitega in nedvoumnega komuniciranja z uporabnikom. Pri ohranjanju enotnosti nam pomaga **mrežna ureditev** *(angl. grid-based layout)*. Pri tem osnovno mrežno

ureditev *(angl. master layout)* uporabimo kot predlogo za izdelavo drugih ureditev strani. Vsi elementi, ki bodo postavljeni na to mrežo bodo tako zagotavljali enotnost in konsistentnost skozi vse strani.

2.4.5.3 Notranja in zunanja konsistenca

Ker se spletne strani ali spletne aplikacije pogosto razvijajo po delih, izolirano od drugega razvoja, se zato srečujejo s problemi konsistence pri vizualnem videzu. Ti problemi so predvsem dveh oblik:

- Problemi, povezani z notranjo konsistenco, kjer gre predvsem za uporabo različnih oblikovalskih pristopov pri oblikovanju posameznih delov spletne strani ali spletne aplikacije;
- Problemi, povezani z zunanjo konsistenco, kjer vizualni videz spletne strani ali aplikacije ne odraža vizualnega izražanja, uporabljenega pri drugih izdelkih podjetja.

Predstavitve, ki niso konsistentne z drugim nastopom podjetja na trgu, obiskovalcem dajejo slab vtis o spletni strani kot tudi o celotnem podjetju, ki je očitno še vedno v fazi iskanja samega sebe.

2.4.5.4 Barvna paleta in tipografija

Barve so ene najučinkovitejših načinov sporočanja identitete blagovne znamke oziroma podjetja. Nekatere znamke so tako zelo povezane z barvami, da je težko prebrati njihovo ime, ne da bi v mislih imeli še njihovo značilno barvo. Tako so na primer Coca-Cola, Mercator ter Mobitel tesno povezani s svojo značilno rdečo barvo, Easy Jet z oranžno, Facebook z modro, Tuš z zeleno itd. Seveda to ne pomeni, da omenjena podjetja uporabljajo zgolj te barve. Osnovne barve so po navadi del širše **barvne palete**, ki se uporablja pri vseh materialih podjetja. Barve izbrane v barvni paleti podjetja pa se morajo med seboj dopolnjevati. Izbrane morajo biti tako, da lahko z njimi elemente poudarjamo ali pa jih skrijemo v ozadje. Paziti pa moramo, da so morebitni izbrani odtenki barv med seboj dovolj različni, da jih uporabniki med seboj preprosto in hitro ločijo.

Izbrana **tipografija** tudi služi kot pomemben del izražanja identitete podjetja. Veliko podjetij je zato razvilo svojo tipografijo in tako pri komuniciranju še povečalo zavedanje identitete. Pravila pri uporabi tipografije pa so zelo podobna drugim pravilom vizualnega načrtovanja: ni dobro uporabljati med seboj podobnih pisav, ni dobro uporabljati prevelikega števila različnih pisav itd.

2.4.5.5 Prikaz končnega videza in vodnik po slogih

Prikaz končnega videza *(angl. design composite)* je vizualna predstavitev končnega izdelka, sestavljenega iz vseh izbranih komponent. V njem je prikazano, kako posamezni elementi delujejo v kombinaciji z drugimi pri tvorjenju celotnega videza spletne strani. Lahko pa nas tudi opozori na napake pri združevanju elementov, ki se vizualno ne skladajo.

Dokumentacijo, v kateri zbiramo vse uporabljene odločitve in sloge o končnem videzu spletne strani ali spletne aplikacije, imenujemo **vodnik po slogih** *(angl. style guide)*. Izbrana mrežna ureditev, barvna paleta, tipografija, navodila za uporabo logotipa so osnovne stvari, ki sestavljajo omenjeni dokument. Razlogi za izdelavo takega dokumenta so predvsem ti:

- sčasoma čas pozabimo na sprejete vizualne odločitve;
- odgovorni za vizualno podobo lahko končajo delo v podjetju in tako s seboj odnesejo ogromno znanja, ki je brez ustrezne dokumentacije izgubljeno;
- pomaga pri zagotavljanju konsistence, predvsem pri necentraliziranih podjetjih, kjer so lahko zaposleni na različnih lokacijah.

Celotna vsebina poglavja je bila povzeta po knjigi, omenjeni na začetku. [1]

2.5 Metode uporabniške izkušnje

Področje uporabniške izkušnje krasi (ali ga kazi) širok nabor raziskovalnih metod:

Testiranje A/B *(angl. A/B Testing)*
Je metoda, ki omogoča testiranje dveh (ali več) različic. Vsakemu obiskovalcu spletne strani se prikaže ena izmed različic. Na podlagi dobljenih rezultatov lahko ugotovimo, katera razporeditev elementov je bila uspešnejša, katere uporabljene barve so imele večji vpliv ter katere besede so uporabnikom bližje. [6]

Vzvratno razvijanje *(angl. Backcasting)*
Metoda, s katero v nasprotni smeri iz idealnega scenarija odkrivamo potrebne korake za doseganje tega scenarija. Prihodnost si je navadno lažje predstavljati s koraki nazaj od idealnega scenarija kot pa s koraki naprej od trenutnega stanja proti idealnemu scenariju. [6]

Razvrščanje kartic *(angl. Card sorting)*
Je tehnika, s katero pridobimo uporabniški pogled na strukturo spletne strani. Sodelujoči razvrstijo kartice z imeni v njim smiselne skupine. Razvrščanje je lahko odprto, kjer

sodelujočim pustimo, da sami določijo skupine glede na njihovo dojemanje imen na karticah. Lahko pa je razvrščanje zaprto, kjer sodelujoči kartice razvrstijo v vnaprej definirane kategorije. To metodo najpogosteje uporabljamo pri načrtovanju navigacije. [6]

Skupinski pregled *(angl. Collaborative inspection)*
Skupinski pregled uporabnosti, na katerem so prisotni naročniki, oblikovalci, programerji in končni uporabniki. Ta sestanek navadno zajema pregled vseh nalog in strani ter njihove uporabnosti. Tako lahko spoznamo različne poglede na isti problem. [6]

Konceptualni model *(angl. Concept model)*
Diagram, ki prikazuje relacije med različnimi koncepti. Vozlišča, ki koncepte vsebujejo, so med seboj povezana z usmerjenimi opisanimi puščicami, ki razložijo odnos med vozlišči. [6]

Pogovorno skiciranje *(angl. Conversational sketching)*
To je metoda, kjer zbrana skupina posameznikov skicira njihova videnja in ideje možnih rešitev ter se o razlogih za te rešitve tudi pogovori. Veliko število skic pripomore k raziskovanju različnih prijemov ter omogoča postopno nastajanje skupne rešitve, ki bo ustrezala vsem sodelujočim. [6]

Ocena iznajdljivosti *(angl. Heuristic evaluation)*
Je neke vrste pregled uporabnosti, kjer za to usposobljeni strokovnjaki ocenijo, kako dobro vmesnik sovpada s poznanimi načeli uporabnosti. Navadno dva ali trije strokovnjaki pregledajo sistem ter sporočijo in po pomembnosti razvrstijo njegove morebitne napake. Na ta način hitro in poceni dobimo povratni odziv o uporabnosti sistema. [6]

Diagram opisov strani *(angl. Page description diagram)*
Je obširen seznam vseh oblikovalskih elementov, vsebine ter elementov vmesnika, razvrščenih v tri stolpce po prioriteti (visoka, srednja, nizka). Na ta način dokumentiramo vse elemente posamezne strani, brez opisovanja same ureditve teh elementov. Lahko se uporablja tudi namesto mrežnih modelov. [6]

Skupinsko načrtovanje *(angl. Participatory design angl.)*
Ta metoda v proces načrtovanja aktivno vključuje tudi naročnika. Tako se skupinsko raziskuje probleme, trenutne in idealne izkušnje ter se išče poti, kako doseči idealno. [6]

Persone *(angl. Personas)*
Persone smo že spoznali v prejšnji točki. To so izmišljene osebe, ki jim pripišemo osnovne

osebne podatke, sliko ter dodatne informacije, usmerjene na doseganje ciljev naše spletne strani. [6]

Načrtovanje scenarijev *(angl. Scenario planning)*
Z njimi prek zgodbe opišemo, kašni so možni scenariji uporabe. Rezultat analize scenarijev so različni možni dogodki v prihodnosti. [6]

Tabla za skiciranje *(angl. Sketchboard)*
Je skupinska metoda načrtovanja, s katero postopno od manj podrobnih razvijamo podrobnejše vmesnike. Vse to je predstavljeno na velikem papirju ali tabli. Začne se s pisanjem različnih predlogov, nato pa možne rešitve najprej skiciramo in razvrstimo okrog danega predloga. Na koncu izbrane rešitve prikažemo kot podrobnejše zaslonske maske. Ta metoda nam omogoča načrtovati širšo predstavo spletne strani, brez ukvarjanja s podrobnostmi, kot se to pogosto zgodi pri žičnih modelih. [6]

Potek zgodb *(angl. Storyboards)*
Je metoda, pri kateri z zaporedjem več žičnih modelov, zaslonskih mask, ilustracij ali slik predstavimo posamezne interaktivne dogodke. Ta pristop je prvi uporabil Walt Disney Studio okrog leta 1930. Slike in ilustracije si sekvenčno sledijo ena za drugo in tako tvorijo zgodbo oziroma v njihovem primeru risanko. [6]

Samolepilni listki *(angl. Sticky notes)*
Z uporabo samolepilnih listkov, ki jih lepimo po kakršni koli ravni površini (tabla, stena itd.), lahko elemente, napisane na listkih, preprosto prerazporejamo brez stalnega popravljanja napak na papirju. [6]

Analiza uporabe *(angl. Task analysis)*
Pri tej metodi gre za preučevanje korakov, akcij in poznavanja procesov, potrebnih, da uporabnik opravi dano nalogo. Analiza uporabe pomaga pri razumevanju sistema in njegovega informacijskega toka. Z njo globoko raziščemo korake ter uporabnikove navade za izvedbo naloge. [6]

Test uporabnosti *(angl. Usability testing)*

Se uporablja za testiranje uporabe prototipa ali končnega sistema. Navadno sodeluje 6 do 8 uporabnikov, ki morajo pod nadzorom moderatorjev na glas deliti občutke pri opravljanju določenih nalog. Z njimi dobimo jasno sliko o tem, kateri koraki delujejo dobro in kateri ne. [6]

Spletna analitika *(angl. Web analytics)*

Orodje, ki meri uporabniško vedenje na podlagi dnevnikov *(angl. logs)* dejavnosti na spletni strani. Ta orodja zbirajo informacije o vhodni in izhodni strani, najbolj priljubljenih straneh, poti, skozi katere so uporabniki prišli na stran, iskalne nize, čas obiska stran itd. [6]

Žični tok *(angl. Wireflow)*

S prikazom vseh zaslonskih mask spletne strani ali spletne aplikacije v enem dokumentu ter s povezovanjem sorodnih zaslonskih mask ta metoda vizualizira interakcije znotraj sistema. Z njo lahko preprosto identificiramo ključne interakcije in odnose med posameznimi deli. [6]

Žični model *(angl. Wireframe)*

Je osnoven vizualen vodnik, ki prikaže strukturo spletne strani. Z njim opišemo vsak element in njegovo obnašanje. Usmerjen je na ureditev elementov, naslove, interakcije in povezave z drugimi stranmi celotnega spletnega mesta. Ne prikazuje pa končnega videza s pripadajočimi barvami in slikami. Lahko ga uporabljamo tudi kot prototip za testiranje uporabnosti. [6]

3. Vodenje projektov

3.1 Problemi in trenutno stanje na področju vodenja projektov

Cilj projektnega vodenja je zagotoviti, da je projekt zaključen v predvidenem času, v predvidenem stroškovnem okviru in z vsemi realiziranimi željami naročnika. Uspešnost projekta zagotavlja doseganje teh ciljev, zato vsak projekt zahteva projektnega vodjo, ki je za uspešnost projekta zadolžen in odgovoren.

Ko odločitve v projektih postanejo napačne in se njihova učinkovitost zmanjša, pri čemer mislim predvsem na prekoračitev rokov in stroškov ter zmanjšanje kakovosti končnega izdelka, takrat se začnemo spraševati o vzrokih, ki tako stanje povzročajo, in iskati rešitve, ki bodo v prihodnosti povečale uspešnost izvajanja projektov. Kljub veliki količini člankov, nasvetov, navodil in različnih metodologij za uspešno vodenje posameznega projekta se danes projektni vodje vedno znova srečujejo s podobnimi problemi in težavami.

Eden izmed problemov, ki ga je vredno izpostaviti, je določitev roka za končanja projekta, s tem pa je pogosto povezano zamujanje dogovorjenih rokov. Pogosto se nov projekt ne začne v predvidenem roku in je dodan v vrsto drugim, že sprejetim projektom, ki so v različnih fazah izdelave. Hkrati pa se od vseh pričakuje, da bodo končani v predvidenem roku. Zato je zelo pomembno, da se vsak projekt začne, tudi če samo z določanjem odgovorne ekipe in nalog, za katere so posamezni člani ekipe odgovorni. Projektni vodja mora tako vsakič, ko je sprejet nov projekt, oceniti in določiti prioritetne naloge posameznih projektov in s tem zagotoviti, da se vsak projekt začne in konča v predvidenem roku.

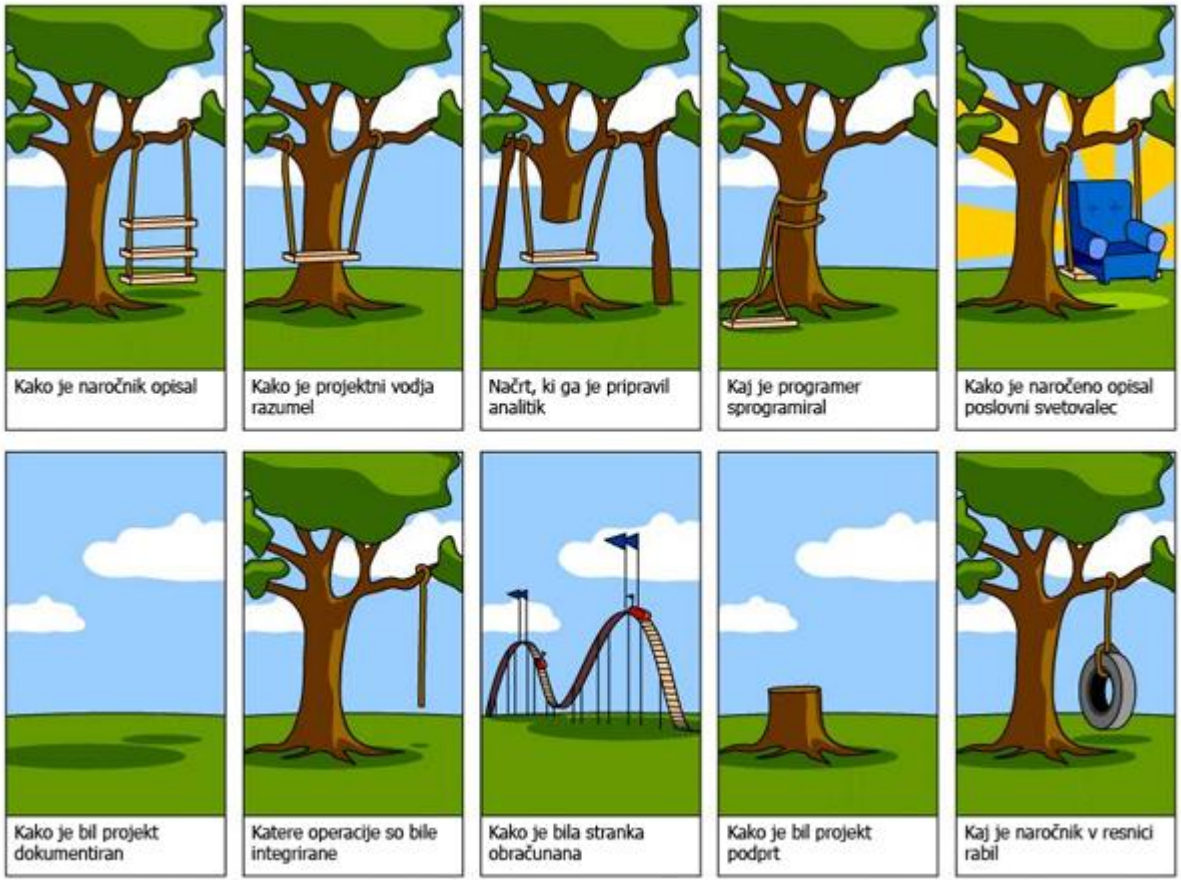
Velikokrat se v času izdelave projekta pojavljajo nove zadolžitve in zahteve, ki jih je bilo v fazi določanja roka nemogoče predvideti, kar pa lahko prav tako povzroči zamudo pri končanju projekta. Takrat je naloga vodje projekta, da preuči smiselnost in nujnost dodatnih del ter zavoljo končanja projekta v dogovorjenem roku doda dodatno delovno silo, prerazporedi delo ter prilagodi prioritete.

Pogosto je razlog za nedokončan projekt tudi v tem, da se projekt sploh ne bi smel začeti. Velikokrat se projekt sprejme zaradi razlogov, ki so v nasprotju s strategijo in vizijo podjetja.

Treba je vedeti, da obstaja smo ena stvar, ki je pomembnejša od pravilnega vodenja projekta, in to je vodenje pravega projekta.

Preseganje predvidenih stroškov projekta *(angl. budget overrun)* štejemo med enega najpogostejših problemov pri vodenju projektov. Do omenjenega problema nas lahko pripeljejo nepravilno razumevanje naročnikovih želja, slaba ocena zahtevnosti projekta ter nenehno spreminjanje že določenih načrtov in zahtev. Pogosto je krivec prekoračitve predvidenih stroškov način dela, ko vodja projektov posveča preveč pozornosti pripravi poročil o stanju projekta ter organizaciji rednih sestankov, pozabi pa na spremljanje stroškov projekta. Vsak član projektne ekipe želi ustvariti čim več produktivnih ur, in če projektni vodja ne nadzira ustreznosti njihove porabe, se kaj hitro zgodi, da je predviden strošek presežen. Zato je zelo pomembno, da vodja karseda pogosto preverja dejansko in pričakovano porabo ur svojih članov projektne ekipe. Preseganje predvidenih stroškov lahko negativno vpliva na naročnika, saj ta zaradi slabega vodenja projekta plača več. V najslabšem primeru, ko naročnik ni pripravljen kriti nastalih dodatnih stroškov, pa lahko negativno vpliva na celotno podjetje. Odgovornost za nastalo škodo pa nosi nihče drug kot vodja projektov.

Komunikacijski šumi na relaciji naročnik, projektni vodja, ekipa ter nejasno postavljeni cilji projekta prav tako spadajo med pogostejše vzroke za zamude in višje stroške. Delo pri



Slika 7- Prikaz posledice napačnega razumevanja naročnikovih želja in zahtev

zadolžitvah, ki so se spremenile, vendar zaradi slabe komunikacije niso bile pravočasno posredovane odgovorni osebi, je lahko zelo moteče in ima negativne posledice na motivacijo celotne ekipe. O jasnosti postavljenih ciljev projekta pa govorimo takrat, ko so ti merljivi, saj lahko le tako določimo uspešnost njihovega doseganja.

Seveda pa ne smemo spregledati dejstva, da je uspešnost projekta odvisna tudi od celotne ekipe in njenih posameznikov. Projektni vodja je tisti, ki mora sestaviti in motivirati projektno skupino, v kateri bo vladalo pozitivno vzdušje. Uspešnost projekta pa nikakor ne sme biti odvisna od enega samega člana ekipe, Kajti če ta iz kakršnih koli razlogov zapusti podjetje ali zboli, je tako ogrožena uspešnost celotnega projekta. Zato mora biti ekipa sestavljena tako, da lahko vsak njen član nadomesti oziroma se hitro izuči ter zamenja ključnega člana ekipe.

Danes, ko smo priča poplavi različnih spletnih strani in spletnih aplikacij, do katerih dostopa iz dneva v dan več uporabnikov, postaja uporabniška izkušnja še kako pomemben del teh. Razlika med dobro in slabo spletno stranjo ali spletno aplikacijo ni več odvisna od njenega programskega ozadja oziroma uporabljene tehnologije, temveč predvsem od načina, kako komunicira z uporabnikom. Nekatera podjetja uporabniško izkušnjo pomikajo daleč v ozadje, čeprav je ravno to lahko jeziček na tehtnici, ki spletno stran ali spletno aplikacijo naredi uspešno. [3]

3.2 Metodologija vodenja projektov Waterfall

Če hočemo imeti izdelek, ki ga bodo uporabniki preprosto in z veseljem uporabljali, moramo pri skoraj vseh fazah razvoja projekta (zbiranje zahtev, načrtovanje, videz, izvedba itd.) misliti na uporabnike, razumeti njihove navade ter marsikdaj tudi čudne načine uporabe. Iz izkušenj vemo, da se večini projektov v času njihove izdelave dodaja, odvzema ter spreminja prvotno definirane funkcionalne in vsebinske zahteve. Tako se lahko skoraj pri koncu razvoja projekta pojavi odlična ideja, ki pa lahko za seboj prinese spremembo v začetku definiranega načrta. Ker je načrtovanje uporabniške izkušnje področje, ki se z razvojem in izobraževanjem uporabnikov hitro spreminja, je treba imeti metodologijo vodenja, ki podpira in dopušča izvajanje sprememb v vseh fazah projekta. Utopično je namreč pričakovati, da bodo zahteve in načrti na začetku razvoja projektov pripravljeni tako celovito in korektno, da bodo ob zaključku ostali nespremenjeni. Velikokrat se ravno najboljše ideje in rešitve pojavijo pozneje v razvoju in se zaradi nepotrebnih omejitev te integrirajo šele v prihodnjih različicah, do katerih pa lahko zaradi neuspeha prve različice sploh ne pride.

Primer metodologije, ki pomeni osnovo in iz katere je izpeljana večina načinov vodenja projektov v podjetjih, ki se ukvarjajo z razvojem spletnih strani in spletnih aplikacij, je tako imenovana **metodologija Waterfall**. Tako ime ima zaradi toka dela, ki vedno teče po poti navzdol. Pot poteka enosmerno, od zbiranja zahtev navzdol do izvedbe, testiranja ter vzdrževanja. Najprej se zbere vse zahteve ter se na podlagi zbranih informacij izdela načrt. Ko je le-ta dokončan in potrjen iz strani naročnika, se ga preda programerjem kot glavno vodilo pri izvedbi. Ko je pisanje kode zaključeno, se končan izdelek testira ter se ga na koncu skupaj z vzdrževalno pogodbo preda naročniku. Spreminjanje osnovnih zahtev, načrta ali videza v fazi izvedbe in programiranja ni mogoče. Opravlja se lahko le manjše spremembe ter

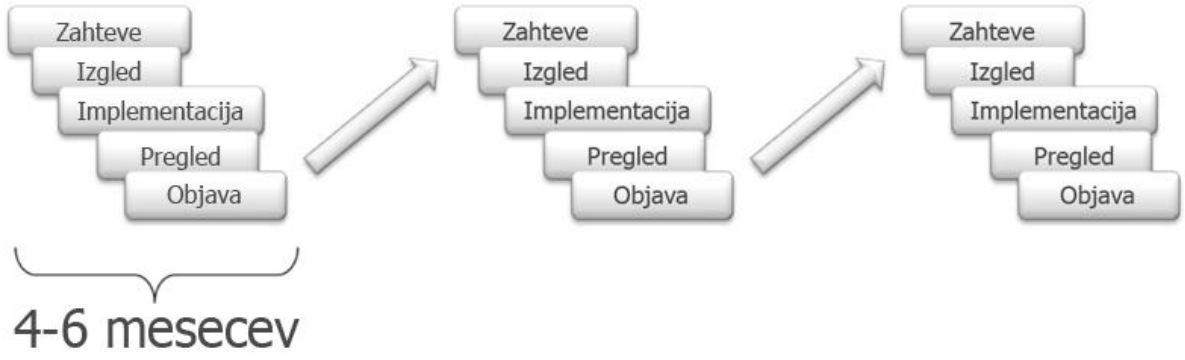
Slika 8 - Metodologija waterfall

odpravlja morebitne napake.

Ta zaporedni pristop naj bi v teoriji zagotavljal, v nasprotju z iterativnim *(angl. iterative)* ali prirastnim *(angl. incremental)*, da ni izpuščena nobena podrobnost, preden se projekt ne premakne v naslednjo fazo. Imeti načrt in videz pripravljen vnaprej tako pomeni, da bo na koncu manj predelav in naknadnih sprememb (le-te zahtevajo dodaten čas in posledično denar). Realnost pa je žal drugačna. Zaradi časa, ki mine med načrtovanjem in izvedbo ter zaradi pomanjkanja v začetku jasno definiranih zahtev, navadno tekom razvoja pride do

večjih predelav in naknadnih sprememb ali pa do slabega končnega izdelka. Pogosto se namreč problemi, ki jih nismo mogli predvideti v začetku, pojavijo v poznejših fazah in zahtevajo spremembe v načrtu ali pa videzu spletne strani oziroma spletne aplikacije.

Metodologija Waterfall in njene izpeljanke včasih formalno ali neformalno vsebujejo elemente iterativne ali pa prirastne metodologije. **Prirasten razvoj** *(angl. incremental development)* projektov je pristop, pri katerem se posamezni deli sistema razvijajo ločeno in ob različnih časih. Na koncu, ko je posamezen del končan in preverjen, pa se ta integrira oziroma doda v celoten sistem. Posamezni deli kode, od katerih vsak pomeni delujoč del celote, se izdajajo v štiri- do šestmesečnih ciklih. Vsak cikel vsebuje enake faze, kot jih srečamo pri metodologiji Waterfall. Slaba stran tega pristopa je, da so deli sistema velikokrat med seboj povezani, zato sprememba v enem delu sistema za seboj prinese tudi spremembe v kakšnem drugem delu, kar pa na koncu pripelje do predelave celotnega sistema. Obstaja pa tudi bojazen, da bo za vsak del sistema težko pripraviti ustrezen načrt že v prvem poskusu. Za take primere pa tudi ta pristop ne vsebuje prostora za vračanje v prejšnje faze.



Slika 9 - Prirastni razvoj

Ironično je, da je Winston W. Royce leta 1970 metodologijo Waterfall prvič opisal zgolj zaradi izpostavitve njenih napak pri predlaganju in predstavitvi boljše **iterativne metodologije**. Pri iterativnem razvoju projektov se posebej nameni čas pregledovanju in izboljševanju posameznih delov sistema. To je navadno cikel, ki ga načrtovalci uporabniške izkušnje najbolj potrebujejo, saj jim omogoča izboljšave izdelka s pridobljenim znanjem iz prejšnjih različic. Iterativne metodologije tako upoštevajo dejstvo, da razvojne ekipe težko načrtujejo in izvedejo vse zahteve pravilno že v prvem poskusu. Zato so morebitne predelave

in možnost naknadnega spreminjanja integrirane v metodologijo. Slaba lastnost tega pristopa pa je, da je težko ugotoviti, kdaj je izdelek dovolj končan in primeren za objavo.

Pri prirastnih metodologijah gre torej bolj za možnost izboljšanja razvojnega procesa ter spreminjanja in dodajanja zmožnosti izdelka med posameznimi izdajami. Pri iterativnih metodologijah pa je poudarek na izboljšanju kakovosti končnega izdelka.

Slika 10 - Iterativni razvoj

Zakaj sploh definirati vse te metodologije in pristope, če je realnost veliko bolj neurejena, kot pa kažejo zgoraj predstavljeni preprosti diagrami? Primerjava predstavljenih metodologij (z njihovimi pozitivnimi in negativnimi lastnostmi) z **agilnimi metodologijami** nam pomaga pri odkrivanju in spoznavanju največjih izzivov, s katerimi se bodo morali srečevati načrtovalci uporabniške izkušnje pri agilnem vodenju projektov. [2]

3.3 Agilno vodenje projektov kot odgovor drugim metodologijam

V primerjavi z metodologijo Waterfall, ki gradi zid med razvijalci in načrtovalci uporabniške izkušnje, skuša agilno vodenje projektov omenjeni področji povezati. Agilno vodenje projektov so razvili posamezniki iz izkušenj in najboljših primerov uporabe. Njihov glavni namen je bil razviti metodologijo, ki karseda hitro prikaže dodano vrednost projekta med njegovim razvojem, hkrati pa omogoča imeti boljši nadzor in fleksibilnost pri načrtovanju projekta. Potreba po čim pogostejših testiranjih in meritvah ter po hitri odzivnosti pri odkritju napak in slabosti je vodila k nastanku termina **agilno vodenje projektov** za opis procesa, ki združuje najboljše primere uporabe. Najpogostejše metodologije, ki jih štejemo med agilne, so: Scrum, Crystal, Lean, DSDM (Dynamic Systems Development Method), FDD (Feature Driven Development), RUP/AUP (Rational/Agile Unified Process), XP (Extreme Programming).

Leta 2001 je skupina projektnih načrtovalcev in upravljavcev podjetij objavila »**Agilni manifest**« *(angl. Agile Manifesto).* V njem so zapisali: »Odkrivamo in vam pomagamo odkrivati nove, boljše načine razvoja programske opreme. Pri delu smo spoznali, da imajo večjo vrednost:

- **posamezniki in interakcije** v primerjavi s procesi in orodji
- **delujoča programska oprema** v primerjavi z obširno dokumentacijo
- **sodelovanje z naročnikom** v primerjavi s pogajanji z njim
- **odzivanje na spremembe** v primerjavi s striktnim držanjem načrta

Čeprav imajo tudi elementi na desni strani zgornjega seznama svojo vrednost, poveličujemo tiste na levi strani.«. Ta kratka izjava povzame glavne cilje agilne metodologije vodenja projektov, ki so: s sodelovanjem med razvijalci in naročnikom že zgodaj v procesu razvoja izdelati delujočo programsko opremo ter se hkrati zavedati, da se stvari lahko med razvojem projekta spremenijo in da se najverjetneje tudi bodo.

Agilno vodenje projektov ponese prirastni način razvoja, predstavljen v prejšnjem poglavju, do ekstrema. Razvojne iteracije so navadno dolge od dva do štiri tedne. Na koncu vsake iteracije je cilj imeti delujočo in testirano kodo, ki jo je mogoče izdati oziroma objaviti. V

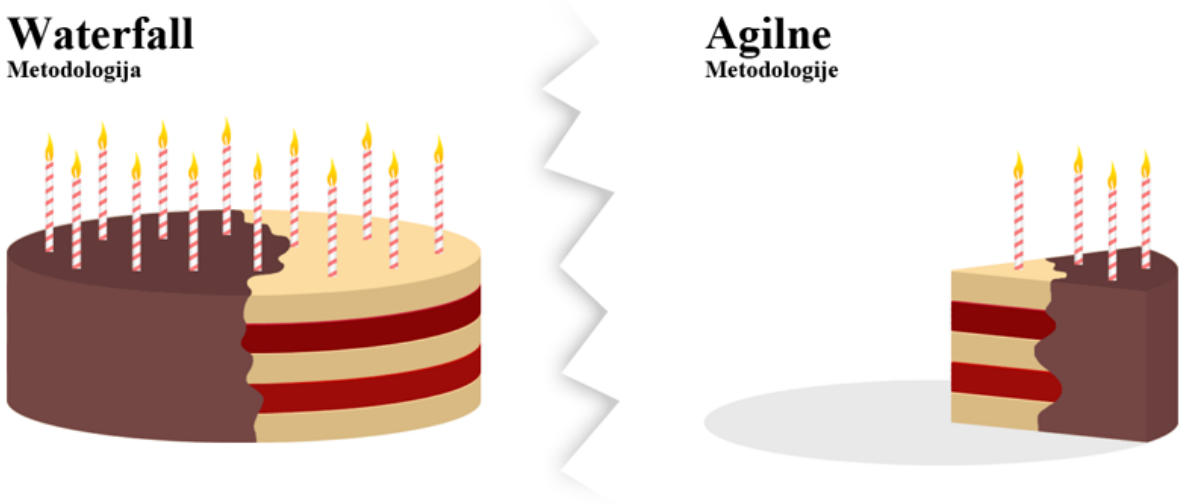
praksi so navadno posamezne izdaje pri več iteracijah združene kot večja delujoča celota, primerna za izdajo oziroma objavo.

Agilni pristop pa je hkrati znotraj posamezne iteracije podoben iterativnemu. Proces programiranja in razvijanja zahteva preizkušanje in izboljševanje različnih idej. Če koda ne prestane testiranja, se ta skozi iteracije izboljšuje, dokler ni primerna za izdajo oziroma objavo.

Iz povedanega lahko agilno vodenje projektov opredelimo kot prirastno serijo iterativnih korakov. Hitrejše obračanje manjših elementov dela daje v primerjavi z drugimi metodologijami več priložnosti za preusmeritev in spreminjanje projekta, kar pripomore k večji učinkovitosti in spretnosti celotne ekipe. Prav tako je manj možnosti, da bo porabljeno veliko časa in dela na zahtevah, ki so bile pozneje ugotovljene za slabe oziroma potrebne sprememb, s čimer se zmanjšuje potreba po predelavi in naknadnih spremembah.

Vse to seveda ne pomeni, da z uporabo agilnih metodologij projekt končamo hitreje kot pri drugih metodologijah. Pomeni predvsem to, da lahko začne projekt z izdajanjem delujočih kod v krajšem časovnem razmiku hitreje prinašati koristi podjetju. Ekstremno hitri časovni razmiki med posameznimi iteracijami tudi ne pomenijo, da je celoten proces hiter, neurejen in brez organizacije. Dobro organizirane agilne ekipe imajo navadno jasno začrtane cilje in sledijo jasnim načrtom. Agilno vodenje projektov temelji na nenehnem prilagajanju in kontroliranju ter uporablja filozofijo vodenja, ki spodbuja timsko delo, samoorganizacijo in odgovornost. Gre za niz najboljših inženirskih praks, ki omogočajo hiter razvoj kakovostnih spletnih in programskih rešitev, ter poslovni pristop, ki omogoča večjo integracijo naročnikovih in uporabnikovih potreb in ciljev. Čeprav se ta metodologija v večini primerov uporablja pri manjših, centraliziranih ekipah, raziskave in poročila iz prakse kažejo, da je lahko uporabna tudi pri večjih skupinah.

Najboljši način primerjave agilnih metodologij z metodologijo Waterfall je priprava torte. Po metodologiji Waterfall torto pripravimo tako, da najprej pripravimo plast testa, na to plast testa nato dodamo novo plast jagod, na plast jagod damo nato novo plast testa itd. Na koncu celotno torto prelijemo s čokolado in na vrh vstavimo svečke. Torto lahko začnemo jesti takrat, ko je le ta v celoti končana in prelita s čokolado. Pri agilnih metodologijah pa izdelamo vsak košček torte posebej ter ga, ko je končan, postavimo na pladenj. Tako se lahko začnemo sladkati že veliko prej, preden je celotna torta končana.



Slika 11 - Grafični prikaz razlike med metodologijo Waterfall in agilnimi metodologijami

V nadaljevanju se bom osredotočil predvsem na agilno metodologijo vodenja projektov, imenovano Scrum, ki se najpogosteje uporablja pri integraciji načrtovalcev uporabniške izkušnje v proces vodenja projektov. [2]

3.3.1 Glavni elementi metodologije Scrum

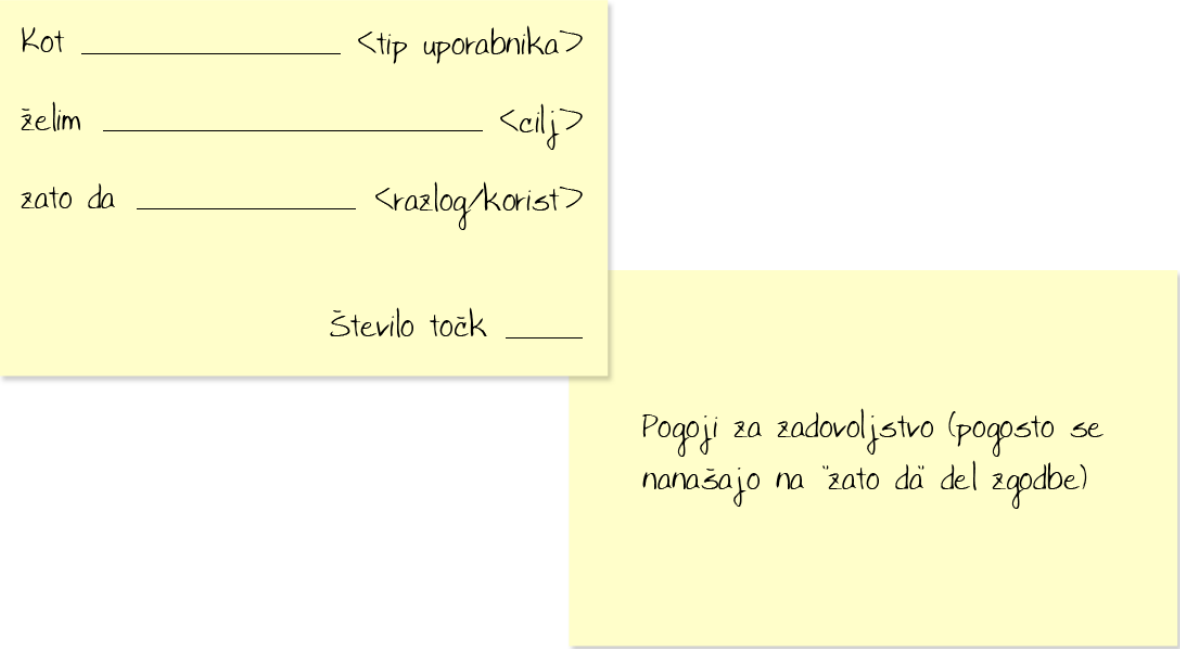
3.3.1.1 Zgodbe

Glavni elementi metodologije Scrum so zgodbe *(angl. stories)*. Z njimi v obliki uporabniških nalog opišemo glavne elemente sistema, ki ga razvijamo. Zgodbe so vedno strukturirane tako, da obsegajo minimalno delovno enoto, s čimer zagotovimo, da je vsaka zgodba vodljiva in primerna za izdelavo.

Zgodbe navadno pišejo naročniki oziroma predstavniki naročnikov, projektni vodje, načrtovalci uporabniške izkušnje ter končni uporabniki. Čeprav bo večina zgodb napisanih vnaprej, je glavno načelo agilnega pristopa, da je mogoče zgodbe kadarkoli, med razvojem, dodati v **zalogo opravil** *(angl. backlog)*. Novo dodana zgodba lahko tako povzroči vnovično določanje prioritet celotne zaloge opravil in vpliva na potek delovnega plana. To načelo je tu zato, ker je malo verjetno, da bodo vse zahteve prepoznane in zbrane vnaprej. Navadno se bodo nove zahteve in nove spremembe začele pojavljati, ko bodo uporabniki začeli uporabljati sistem.

Zgodbe napišemo na **kartotečne kartice** *(angl. index cards)*, navadno velikosti 8 x 13 cm (ali 13 x 18 cm). Takšna velikost je zaželena in določena z namenom prisiliti pisce zgodb, da so te kratke in ne vsebujejo nepotrebnih informacij o izvedbi. Zgodbe morajo biti napisane na poseben in vnaprej določen način: »Kot ___ želim ___« (»Kot mizar želim pregledovati izdelke glede na tip lesa«). Tako avtorje zgodb usmerimo, da razmišljajo o tem, katerega uporabnika sistema predstavljajo oziroma želijo zadovoljiti ter katero nalogo uporabnik izvaja. Pogosto se tem stavkom doda še »Kot ___ želim ___ zato da ___«, s čimer izpostavimo dodano vrednost, ki jo tako prinesemo podjetju, ali pa razlog za izvedbo naloge.

Tako napisanim zgodbam nato dodamo **točke**. Točke so merilo za trud, ki je potreben za izdelavo sistema, ki bo omogočal izvajanje v zgodbi predstavljene funkcionalnosti. Čeprav so so točke poljubno izbrano merilo, morajo biti smiselno razporejene glede na potreben trud. Tako mora biti zgodba, ki ji pripišemo dve točki, za polovico lažje izvedljiva od tiste, ki ji pripišemo štiri točke.



Slika 12 - Prikaz kartotečnega listka, na katerem so napisane zgodbe, točke in pogoji za zadovoljstvo.

Veliko ekip ob pisanju zgodb na zadnjo stran kartotečne karte napiše tudi **pogoj za zadovoljstvo** (kriterij sprejemljivosti, kriterij uspešnosti, kriterij za zagotavljanje kakovosti – *angl. QA criteria*). Te pogoje navadno napiše naročnik in tako da ekipi vedeti, kako razume napisano zgodbo in njej pripadajoče funkcionalnosti ter kdaj in pod katerimi pogoji bo zgodba zaključena. Še bolje pa je, da te pogoje napiše naročnik v sodelovanju z načrtovalcem uporabniške izkušnje, saj bo tako upoštevan tudi uporabnik.

Agilne ekipe so zgodbam naklonjene predvsem zato, ker:

- v fazi načrtovanja in določanja prioritet delu, spodbujajo verbalno komunikacijo in skupinsko načrtovanje;
- je v njih uporabljen jezik, ki ga razume uporabnik, torej kdorkoli;
- ponujajo zadostno količino informacij za ocenitev in načrtovanje dela;
- podpirajo iterativni razvoj, ki omogoča spremembe med razvojem projekta (spremembe v načrtu, videzu itd.).

[2]

3.3.1.2 Zaloga opravil

Vse napisane zgodbe o projektu nato shranimo v **zalogo opravil**. To zalogo nato naročnik in razvijalci pregledajo ter zgodbam iz zaloge opravil določijo prioritete. Tako vsi sodelujoči pri projektu vedo, kakšna je pomembnost posameznih zgodb in s tem elementov končnega produkta. Prioritete se zgodbam v zalogi opravil nenehno spreminjajo, saj se med razvojem projekta dodajajo, brišejo ter spreminjajo.

Pri projektih, ki imajo fiksno ceno ali čas izdelave, v zalogi opravil določimo črto, ki loči zgodbe, ki bodo upoštevane pri razvoju končnega produkta in zgodbe, ki ne bodo upoštevanje. Izberemo tiste, ki po številu točk, ustrezajo časovnemu ali finančnemu okvirju. V času razvoja pa se bo omenjena črta premikala gor in dol, saj se bodo v zalogi opravil zgodbe dodajale in odstranjevale, prav tako pa se lahko tem spremenimo prioritete in točkovne ocene. Zaradi tega dejstva se tej črti pogosto reče **plavajoča črta** *(angl. waterline).*

Navadno je zaloga opravil fizična lokacija, kamor so kartotečne kartice pripete – stena, tabla, itd. To omogoča manipulacijo in neposreden vpogled v pomembne zgodbe, v prihajajoče dogajanje in v že opravljene naloge. Nekateri za hranjenje zgodb uporabljajo temu primerno

programsko opremo, ki omogoča hitrejše prilagajanje na spremembe v razvoju projekta, vendar hkrati zameglijo celoten pogled na zalogo opravil. [2]

3.3.1.3 Sprint

Dva oziroma štiri tedne dolga razvojna iteracija se v metodologiji Scrum imenuje **sprint** *(angl. sprint)*. Tako ime imajo predvsem zaradi vnaprej določene kratke poti z jasno določenim ciljem, in ne zaradi same hitrosti dela. Namen ni hiteti z delom in čim hitreje prispeti na cilj, tako kot je navadno značilno za sprinte na kratke proge.

V času načrtovanja posameznega sprinta ekipa iz zaloge opravil izbere tiste zgodbe, ki bodo v sprintu razvite. Naloga se tako opredeli, ko razvijalec oceni potreben trud in količino ur ter tako zgodbi določi število točk. Posamezna naloga mora biti koristna, preverljiva oziroma testirana in dovolj majhna, da se jo lahko v celoti opravi v enem sprintu. Vsak razvijalec ima v enem sprintu na voljo določeno število točk, ki določajo količino dela, ki ga lahko sprejme. Zgodbe se iz zaloge izbirajo glede na določeno prioriteto ter glede na to, kako so med seboj povezane.

Vsak sprint se tako konča s potencialno objavljivim prirastkom h končnemu produktu. Čeprav nova koda ne bo objavljena in izdana takoj, mora biti sprogramirana, testirana in dana uporabnikom v uporabo. Ta zahteva preprečuje pojavljanje napak ter zmanjša potrebo po naknadnem pregledovanju kode, tudi po končanem sprintu. [2]

3.3.1.4 Sestanek Scrum

Scrum je dnevni sestanek, ki pomeni način, kako člani ekipe komunicirajo med seboj in kako s tem zagotovijo, da delo poteka po načrtu. Vsi člani ekipe se sestanejo na jutranjem sestanku, kjer vsak odgovori na tri vprašanja:

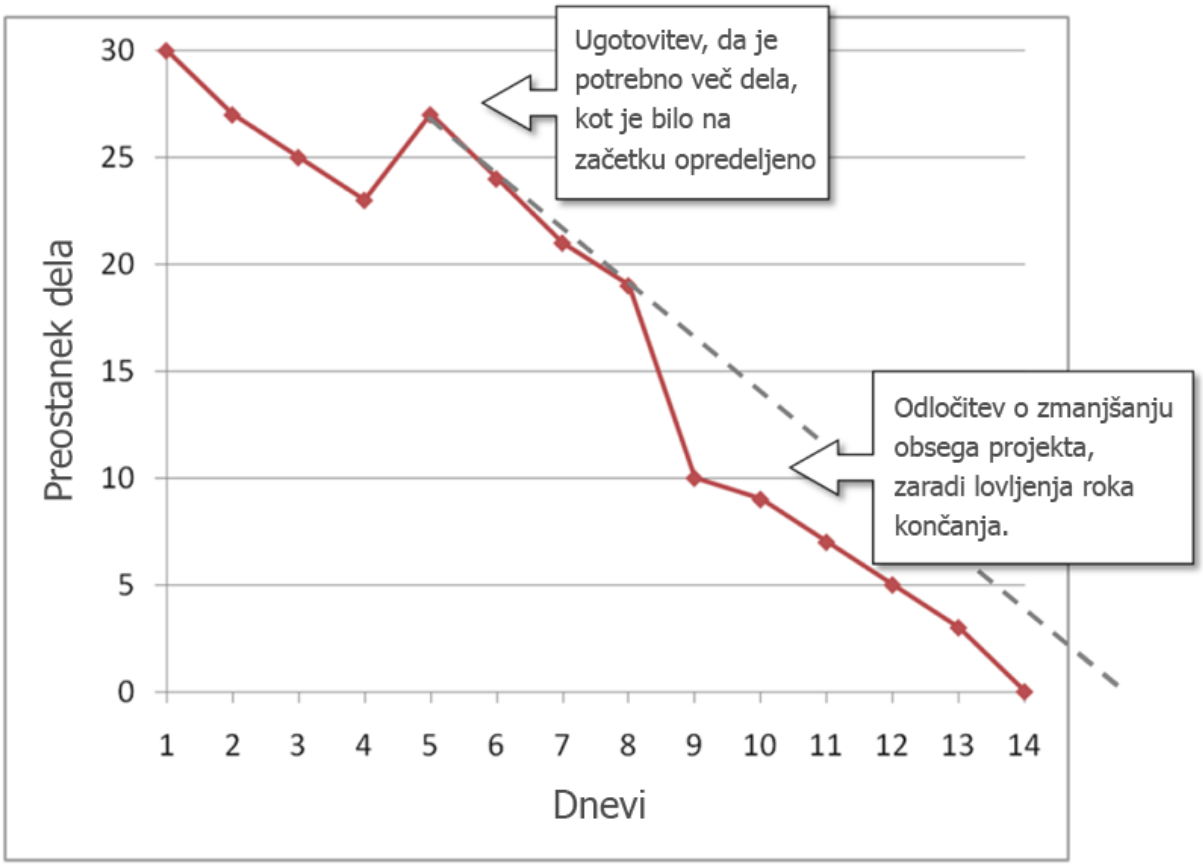
- **Kaj si opravil včeraj?**
- **Kaj boš opravil danes?**
- **Kje so težave?**

Z odgovori na ta vprašanja drugi člani ekipe razumejo nastale težave, jih poskušajo skupaj rešiti ter si dnevno določajo novo delo. Sestanki Scrum tako omogočajo tekoče posodabljanje grafičnega prikaza napredka dela na pogorigramu posameznega sprinta *(angl. burndown chart)*. Celotni ekipi se tako omogoči nemoteno delo, ne glede na to, ali so v zaostanku ali ne. [2]

3.3.1.5 Pogorigram

Pogorigram pomaga slediti opravljenemu delu z zgodbami znotraj posameznega sprinta in tudi vseh sprintov. Število točk (ki jih pripišemo zgodbam), ki se skozi določen čas spremenijo v dejansko kodo, imenujemo **hitrost dela** *(angl. velocity)*. Načrtovanje hitrosti dela skozi celoten (s fiksno dolžino opredeljen) projekt, pove ekipi, koliko opravil iz zaloge opravil lahko sprogramirajo. S tem se določi tudi plavajočo črto *(angl. waterline)*.

Uporaba točk kot merske enote za opredelitev zahtevnosti zgodb omogoča neposredno



Slika 13 - V pogorigramu so točke prikazane na y-osi. S tem, ko se zgodbe zaključujejo in pretvarjajo v delujočo kodo, se njihova točkovna vrednost odmika oziroma briše iz preostalega dela (y-osi). Pri 4. dnevu se je tako dodala nova zgodba ali pa je razvijalec ocenil, da bo za določeno zgodbo porabil več časa in ji pripisal večjo točkovno vrednost. Ker je zato hitrost dela, ki je predstavljena kot število točk/število dni, prekoračila dogovorjeni rok, so se odgovorni 8. dan odločili, da zaradi lovljenja roka zmanjšajo obseg projekta. Brez dnevnih sestankov Scrum ta stopnja prilagodljivosti in gibčnosti pri s ukvarjanju z nastalimi težavami ne bi bila mogoča.

povezavo z zalogo opravil. Če bi uporabljali čas, denar ali drugo mersko enoto, bi zgubili povezavo med opravljenim delom in delom, ki je še na seznamu za izdelavo. Tako bi ekipa zgubila občutek, ali je v zaostanku z delom ali ni. [2]

3.3.1.6 Zagotavljanje kakovosti (angl. Quality Assurance − QA)

Pogoji zadovoljstva s **kriteriji sprejemljivosti** *(angl. acceptance criteria)*, napisanimi na zadnji strani kartotečne kartice, postanejo med sprintom izhodni kriteriji končanja posamezne zgodbe. Ekipa predstavljene pogoje zadovoljstva spremeni v dejanske teste za končni izdelek. Ko so ti testi uspešno prestani, je izdelek pripravljen za objavo oziroma izdajo.

Celotna ekipa se zaveda vseh kriterijev končanja, še preden se začneta razvoj in delo. Tako si pri razvoju prizadevajo zadovoljiti te kriterije, in ne kakšnih drugih, ki temeljijo zgolj na domnevah. Zaradi usmerjenosti na točno določene izhodne kriterije končanja, razvijalci v teoriji napišejo minimalno število vrstic kode, s katero bodo zadovoljili te kriterije. Tako bo minimiran potreben napor, povečana pa bo dodana vrednost projekta.

Celoten proces testiranja in doseganja pogojev zadovoljstva zahteva, da so ti pogoji jasno in vnaprej definirani. Navadno je najbolje, da so predstavljeni s konkretnimi številkami. Stavek »končni izdelek mora biti uporaben«, ni najprimernejši za opis pogoja zadovoljstva. Bolje je uporabljati stavke, kot je »uporabnik lahko brez napak v manj kot dveh minutah opravi nakup majhne količine izdelkov (5 ali manj)«, s čimer jasno definiramo kriterij zadovoljstva. [2]

3.3.1.7 Konica

Konica *(angl. spike)* je delovanje, s katerim razvijalci pridobijo ravno dovolj znanja o kakšni novi, neznani stvari (na primer o novi tehnologiji). Z novim znanjem lahko nato podajo približno oceno o zahtevnosti zgodbe. Tako kot zgodbam se tudi konicam določajo točke ter imajo svoje mesto v sprintu. Tako učenje novih stvari omejimo i definiramo največjo količino časa, ki ga bomo temu namenili.

Potreba po konicah se navadno pojavi ob načrtovanju sprintov. Če članom ekipe primanjkuje potrebnega znanja za izvedbo zgodbe, se tega ne da pridobiti iz drugih virov ali pa je področje povsem novo in neraziskano, se dodatno delo in izobraževanje odobri ter se določi število potrebnih točk za izvedbo. Konice se nato obravnava kot vse druge zgodbe ter se jih skupaj s pripadajočimi točkami upošteva tudi pri risanju pogorigrama. Navsezadnje je izobraževanje tudi delo, ki je potrebno za končanje izdelka. [2]

3.3.2 Kaj metodologija Scrum prinaša načrtovalcem uporabniške izkušnje

3.3.2.1 Prednosti

Načrtovalec uporabniške izkušnje je zelo pomemben član ekipe, ki uporablja agilno vodenje projektov po metodologiji Scrum. S svojim delovanjem lahko ogromno pripomore k uspešnosti končnega izdelka. S sprejemanjem **ključnih filozofskih točk metodologije Scrum** se pojavi pomembna vloga vsakogar, ki lahko ekipi pomaga z realnimi podatki o uporabnikih.

- Napisane **zgodbe** so lahko razširjene in obogatene z integracijo person in raziskav uporabniških zahtev. Pogoji zadovoljstva zgodb, napisani na zadnji strani kartotečnih kartic lahko vsebujejo zahtevo po določeni stopnji uporabniškega vedenja.
- Določanje in spreminjanje prioritet zgodbam v **zalogi opravil** je ponavljajoča se naloga. Pri tem določanju nam pomagajo persone, uporabniški scenariji, potek zgodb in žični modeli.
- Koncept **sprintov** omogoča usmerjanje na specifično skupino uporabniških nalog, primernih za testiranje, ki pomagajo pri vzdrževanju pozornosti razvijalcev glede ključnih zmožnosti končnega izdelka.
- **Pogorigram** lahko razumemo tudi kot sredstvo, ki nam sporoča, kako blizu smo zadovoljitvi ciljev naših uporabnikov.
- Testi za **zagotavljanje kakovosti** lahko marsikaj pridobijo, če mednje vključimo tudi teste uporabnosti. Uspešno prestani uporabniški testi veliko prispevajo k zadovoljevanju izhodnih kriterijev končanja.
- **Konice** bodo pogosto potrebovale vključevanje načrtovalcev uporabniške izkušnje v proces raziskovanja smiselnosti vpletanja novih tehnologij v izdelek. Načrtovalci uporabniške izkušnje bodo tako s svojim delovanjem ocenili vpliv novih tehnologij na uporabnika.

Z integracijo obstoječih metod uporabniške izkušnje in s sprejemanjem nekaterih novih tehnik sta lahko uporabniška izkušnja in njeno načrtovanje gonilna sila pri razvoju in vodenju projektov. [2]

3.3.2.2 Izzivi

Izkušeni načrtovalci uporabniške izkušnje pri delovanju znotraj ekip, ki delujejo po metodologiji Scrum, bolj ali manj izpostavljajo podobne izzive. V anketi, ki so jo opravili pri skupini Nielsen Norman Group in v kateri je sodelovalo 174 profesionalnih načrtovalcev in razvijalcev (večinoma iz Združenih Držav Aamerike, Anglije ter Avstralije), so ti kot največje izzive pri agilnem vodenju projektov po metodologiji Scrum izpostavili spodnje:

- Zahteve niso nikoli popolno zbrane in definirane v prvem poskusu, zato je veliko časa zapravljenega za nenačrtovane iteracije, namenjene testiranju → pridobivanju uporabniških odzivov *(angl. user feedback)* → popravljanju.
- Raziskave, analize in načrtovanje niso narejeni vnaprej, kar pogosto povzroča težave pri doseganju rokov za dostavljanje dela razvijalcem.
- Težko je favorizirati funkcionalen videz pri marketinško usmerjenih ekipah.
- Ohranjati je treba pozornost naročnika pri razvojnih nalogah.
- Pomanjkanje časa za čakanje na odziv uporabnikov, zato večino ocen izdelka napravi kar ekipa sama.
- Ekipi je težko pripraviti jasno vizijo o uporabnikih izdelka.
- Pomanjkanje časa med konceptom in prototipom za testiranje pri uporabnikih. Časovni načrti so namreč zelo tesni za kakovostno testiranje.
- Pomanjkanje časa za dokumentiranje načrtov in predlog za vse različne bralce.
- Premalo spoštovanja in pozornosti se posveča pravi uporabnosti končnega izdelka. Problemi uporabnosti se rešujejo ločeno, znotraj vsakega sprinta, kar pa ni najbolje, saj je uporabnost izdelka stvar, ki jo je treba reševati celostno.
- Težave z razvijalci, ki imajo pogosto odpor do nenehnih sprememb zahtev glede končnega izdelka.
- Pomanjkljive zahteve povečajo pogostost napak, zaradi česar iteracije, namenjene izboljševanju in odpravljanju napak, izpodrivajo razvojne iteracije.
- Obstaja krivulja učenja, saj je za vse to nov proces.
- Pogosto primanjkuje pozornosti in koncentracije.

Vse predstavljene težave načrtovalcev uporabniške izkušnje so utemeljene, izpostavljeni problemi pa realni. A to še ne pomeni, da pomenijo razloge za neuspeh. Marsikateri načrtovalci uporabniške izkušnje delujejo zelo uspešno v podobnih okoliščinah, saj je

navsezadnje od njih odvisno, da si izborijo mesto v ekipi. Avtorji ankete in njej pripadajočega obširnega poročila predlagajo za odpravo težav naslednje predloge. [2]

3.3.2.3 Predlogi

Vnaprejšnjem načrtovanju ni potrebno posvetiti veliko časa

Pri bolj linearnih metodologijah razvoja je vnaprejšnje načrtovanje zelo pomembno, saj je med razvojem manj možnosti za spremembe. Pri metodologiji Scrum pa je lahko zapravljanje časa pri načrtovanju sistema, še preden imajo uporabniki in naročnik možnost razumeti vpliv sprememb nanj, zelo nevarno. Pri zgodnjem razvoju je pogosto bolje uporabiti realno kodo za predstavitev zmožnosti sistema in šele nato izvajati spremembe, ki temeljijo na odzivu uporabnikov.

Seveda to ne pomeni, da je čas, namenjen vnaprejšnjemu načrtovanju, izgubljen, saj imajo očitno zgodnje raziskave kar nekaj prednosti. Trik je v tem, da se čas, namenjen vnaprejšnjemu razvoju, pametno in koristno uporabi. Obstaja tudi veliko tehnik (terenski obiski, potek zgodb, prototipi žičnega modela itd.) za hitro zbiranje uporabniških zahtev ter za prikaz širše slike uporabniške izkušnje ekipi.

Težko je govoriti o uporabnikih, če so ti slabo definirani

Preveč je usmerjanja na stranko, ki pa pogosto ni uporabnik. Oseba, ki podpiše račun ali potrdi naročilo, je stranka, oseba z miško v roki pa je uporabnik. Za spletne trgovine, kjer gre za poslovanje s končnim uporabnikom *(angl. business-to-customer ali na kratko B2C),* v omenjeni definiciji opisujeta isto osebo. V podjetniških okoliščinah ali pri poslovanju s podjetji *(angl. business-to-business ali na kratko B2B)* pa sta ti dve osebi pogosto drugačni.

Ta razlika lahko pripelje do težav, predvsem če v ekipi ni nikogar, ki bi zares predstavljal končnega uporabnika sistema v razvoju. Uporabniške potrebe morajo biti razvojni ekipi predstavljene ob hitrih dnevnih sestankih scrum. Zato je malo časa za kompleksne in obširne dokumente o uporabniških zahtevah ter kupe uporabniških primerov.

Pri delu v agilnem okolju je treba biti gibčen in spreten pri uporabi različnih metod. Uporaba person v celotnem projektu pomaga ekipi pri enotnem razumevanju potreb in sposobnosti ciljnih uporabniških skupin. Včasih je potrebno, da ekipi rahlo spodleti, saj le tako vidijo potrebo po večji količini podatkov ali pa po povsem drugačnih, novih podatkih. Biti v

pripravljenosti z vsemi podatki, ko se začnejo debate ali izbiranje zmožnosti končnega izdelka, je lahko odličen začetek.

Pri metodlogiji Scrum je razvijalec središče

Razvijalci se hitro odločijo in sprejmejo določeno pot za edino pravo. Tisti, ki nočejo spremeniti svojega načina delovanja znotraj agilne ekipe, so enako trmasti kot tisti, ki nikakor nočejo sprejeti agilnega pristopa. Na koncu to pomeni, da mora načrtovalec uporabniške izkušnje pristopati k metodologiji Scrum enako kot pri kateri koli drugi metodologiji upravljanja. Delovati znotraj omejitev določene metodologije, dokler niso dokazane prednosti, ki bi jih prinesla sprememba, je najlažji način za spremembo načina razmišljanja in dela razvijalcev. Takrat sta pogostost pregledovanja ter možnost preprostega vračanja na prejšnje korake razvoja glavni prednosti, s katerima lahko načrtovalec uporabniške izkušnje spremeni delo razvijalcev v pozitivno smer.

Cilj je imeti boljšo spletno stran ali aplikacijo, ne pa zgolj opravljati teste

Uporabniško testiranje je osnovna metoda pri mnogih načrtovalcih uporabniške izkušnje. Pomeni enega izmed načinov ugotavljanja uporabnostnih napak ali težav, ki se pojavljajo pri predlaganem ali končnem vmesniku.

Tradicionalni testi (kamere, ki snemajo vsak gib; kompleksne matrice nalog) imajo malo možnosti za uspeh. Obstaja veliko drugih tehnik, ki porabijo manj časa ter postrežejo z odličnimi rezultati veliko prej, čeprav niso tako natančne ter temeljite kot tradicionalne. Tesen urnik dela, značilen za metodologijo Scrum, spodbuja iskanje drugih tehnik, ki so lahko hitrejše in ustreznejše za določene probleme oziroma določena vprašanja. Treba je vedeti, da je cilj imeti boljšo spletno stran ali aplikacijo, ne pa zgolj opravljati teste.

Metodologija Scrum ni ugodna za centralizirane ekipe načrtovalcev uporabniške izkušnje

Pravilen pristop načrtovalcev uporabniške izkušnje pri metodologiji Scrum v začetku spoznavanja tega procesa pogosto zahteva celoten razpoložljivi delovni čas. Odvisno od potrebe po njihovem vpletanju v posamezen proces razvoja se lahko načrtovalci uporabniške izkušnje premikajo iz ekipe v ekipo. Možnost integracije uporabniške izkušnje in njenih načrtovalcev globlje v delo posamezne ekipe je lahko tudi prekletstvo, še zlasti če ni na voljo dovolj sredstev.

Stik in komunikacija med vsemi načrtovalci uporabniške izkušnje znotraj podjetja sta ključ do uspeha, pri čemer si lahko pomagamo s centralizacijo celotne ekipe. Še vedno pa mora biti glavna odgovornost načrtovalca uporabniške izkušnje pripadnost ekipi, znotraj katere deluje. Zato je bolje, da vsak načrtovalec deluje znotraj ekipe, s katero trenutno sodeluje, kot pa da so vsi združeni na enem mestu. Če v podjetju take organizacije ni mogoče imeti, to še ne pomeni propada. Še vedno je mogoče podpirati agilen pristop k vodenju projektov po metodologiji Scrum, čeprav ima podjetje centralizirano ekipo uporabniških načrtovalcev.

Metodologije Scrum se moramo lotiti tako, kot je napisano v knjigah, a uporabniške izkušnje se nikjer ne omenja

Pri novih in negotovih ekipah ima oprijemanje navodil pogosto prednost pred zdravo pametjo. Prednost pravilno delujočih agilnih ekip, ki delujejo po metodologiji Scrum je v tem, da lahko vodja ekipe, imenovan gospodar scruma *(angl. scrum master)*, oziroma katerikoli član, hitro odstrani vse ovire za uspešnost projekta. Spreminjanje delovanja ekipe je del procesa. Sposoben načrtovalec uporabniške izkušnje bo hitro spoznal in identificiral, kje lahko tehnike in orodja uporabniške izkušnje pripomorejo k uspešnosti ekipe.

Če ekipa nenehno ne analizira in ne izboljšuje preteklih pristopov in procesov, potem ne gre za ekipo z agilnim pristopom. To je eden izmed razlogov, zakaj nekatere knjige vrste »kako naj« *(angl. how to books)* gurujev agilnega vodenja projektov zvenijo bolj spiritualno kot konkretno. Osnova agilnega vodenja projektov in s tem tudi metodologije Scrum je zgolj skupek navodil in priporočil, resnična vrednost metodologije se pojavi takrat, ko začne ekipa iskati svoje poti, ki ji najbolj ustrezajo.

Ena glavnih odlik izkušenega načrtovalca uporabniške izkušnje je sposobnost izbora pravega orodja iz celotnega nabora orodij, ki bo kar najbolje opravil zahtevano nalogo. Obstaja na stotine metod testiranja uporabnosti ter uporabnika, in če izbrana metoda ne deluje najbolje, je na voljo veliko drugih, od katerih bo ena zagotovo delovala. [2]

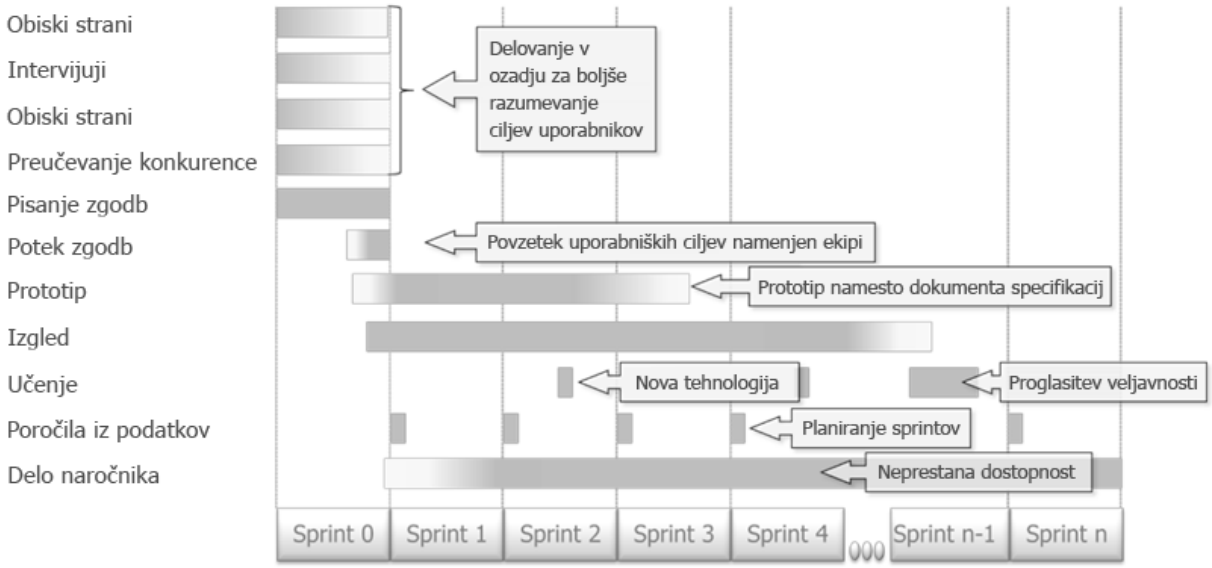
3.3.3 Integracija uporabniške izkušnje v metodologijo Scrum

Po mnenju skupine Nielsen Norman Group obstaja pet glavnih pokazateljev, ki demonstrirajo dobro integracijo uporabniške izkušnje v ekipe, ki uporabljajo agilno vodenje projektov po metodologiji Scrum.

- **Načrtovalci uporabniške izkušnje so most med posameznimi člani ekipe,** saj so nenehno v interakciji z naročnikom oziroma njegovim predstavnikom ter z ekipo razvijalcev. Komunikacija (med člani ekipe, med ekipo in naročnikom, med ekipo in končnimi uporabniki) je največkrat izpostavljen problem vsake ekipe, zato se metodologija Scrum, kot tudi druge agilne metodologije še posebej trudijo za odpravo tega problema. Od načrtovalcev uporabniške izkušnje se pričakuje komunikativnost ter sposobnost govorjenja ter razumevanja jezika razvijalcev, naročnikov ter končnih uporabnikov. Seveda to ne pomeni, da lahko načrtovalec uporabniške izkušnje s svojimi sposobnostmi reši vse komunikacijske probleme. Lahko pa postopoma izboljšuje komunikacijo med posameznimi skupinami znotraj ekipe in si tako pribori mesto v zgodnjih delih načrtovanja, kar pa pomeni več možnosti za pomemben vpliv na uporabnost končnega izdelka. Hitro pa se lahko zgodi, da vsa ta novo nabrana moč načrtovalcu uporabniške izkušnje stopi v glavo. S čimer bo znotraj ekipe postal nepriljubljen. Zato mora razvijalce in ekipo zbliževati in vključevati vse prisotne v procese odločanja.
- **Delo načrtovalcev uporabniške izkušnje mora biti fleksibilno in se mora začeti zgodaj v procesu razvoja projekta.** Raziskave (obiskanost spletne strani, intervjuji z uporabniki, preučevanje ozadja uporabnikov ter konkurence) lahko pri zgodnjem procesu priprave in načrtovanja projekta postrežejo z veliko dragocenimi podatki. Načrtovalec uporabniške izkušnje si mora izboriti čim več časa za vnaprejšnje načrtovanje, vendar mora biti pri njegovi porabi še kako pazljiv in preudaren. Razvijalce mora spodbujati, naj počakajo z začetkom razvoja, dokler ne bodo definirani in razumljeni vsi pomembni cilji. Ali pa naj, dokler cilji še niso določeni, razvijajo ozadje sistema, ki ga uporabnik ne vidi.
- **Grobi prototipi pomenijo stalen vir informacij kot nadomestilo za dokument s specifikacijami.** Eden izmed elementov, ki ga je nujno treba imeti, je prototip. Skupek grobih žičnih modelov, ki se združijo v prikazu poteka zgodb, lahko nadomesti tradicionalen dokument s specifikacijami. Namen tega dokumenta je pri tradicionalnih metodologijah predstaviti razvijalcem, kaj je treba sprogramirati ter kako je naj končen izdelek videti. Metodologija Scrum se zaveda dejstva, da taki dokumenti s specifikacijami navadno zastarajo v trenutku, ko so napisani. Glavno mesto za pisanje specifikacij pri metodologiji Scrum je na zadnji strani kartotečne kartice z zgodbo v obliki pogojev zadovoljstva. Brez dokumenta s specifikacijami tako žični modeli in

poteki zgodb v obliki prototipa postanejo ključni elementi za sporočanje konceptov in interaktivnih zahtev končnega izdelka celotni ekipi. Načrtovalec uporabniške izkušnje pa je tako kot lastnik teh prototipov še kako pomemben člen pri odločanju osnovnih oblikovnih in načrtovalskih odločitev.

- **Delo načrtovalcev uporabniške izkušnje poteka vzporedno z drugim razvojem.** Ena glavnih idej metodologije Scrum je, da ekipa skupaj deluje z istimi zgodbami znotraj istega sprinta. Načrtovalci uporabniške izkušnje morajo stati ob strani razvijalcem, ko ti razvijajo vmesnik. Hkrati pa morajo gledati tudi na prihajajoče sprinte in zagotoviti, da bodo vsi dvomi odpravljeni, preden razvijalci začnejo delati z njimi. Zaradi tega dejstva morajo načrtovalci uporabniške izkušnje hkrati, ob zgodbah trenutnega sprinta, na vzporednem tiru obdelovati tudi druge zgodbe, ki bodo prišle v poštev pri naslednjem ali naslednjih dveh sprintih.



Slika 14 Prikaz vzporednega dela

Celotna vsebina tega poglavja je povzeta po obširnem poročilu podjetja Nielsen Norman Group imenovanem »Agilna uporabnost: Najboljši primeri za integracijo uporabniške izkušnje v projekte z agilnim načinom razvoja« *(angl. Agile Usability: Best Practices for User Experience on Agile Development Projects).* [2]

4. Rezultati opravljene ankete

V anketi je sodelovalo 10 slovenskih podjetij – nekatera sodijo v vrh slovenske spletne produkcije. Anketa je vsebovala 20 vprašanj s področja uporabniške izkušnje ter načina vodenja projektov. Na podlagi rezultatov ankete ter lastnih izkušenj pri sodelovanju z različnimi podjetji sem prišel do sklepov, ki so predstavljeni spodaj.

4.1 Zavedanje o uporabniški izkušnji je prisotno

Vsa v anketi sodelujoča podjetja so odgovorila, da imajo v podjetju koga, ki je odgovoren za uporabniško izkušnjo končnega izdelka. Seveda pa se je treba vprašati, kako te odgovorne osebe znotraj podjetja delujejo:

- So prisotne v celotnem razvoju projekta?
- Se v času razvoja pogovarjajo in sodelujejo s končnimi uporabniki?
- Ali pred začetkom načrtovanja projekta analizirajo ciljne skupine uporabnikov in njihove navade?
- Ali izvajajo različna testiranja med izdelavo projekta?
- Ali končan izdelek testirajo pri končnih uporabnikih?
- Ali izdelujejo prototipe, ki jih testirajo pri uporabnikih?

Rezultati kažejo, da večja podjetja, ki si to lahko privoščijo, omenjene pristope in teste tudi izvajajo. Težava nastane pri manjših podjetjih, kjer je vodja projektov ali pa kdo drug v podjetju hkrati tudi načrtovalec uporabniške izkušnje. Tako je pogosto uporabniška izkušnja potisnjena na stran in je opravljena zgolj površinsko, brez večjega truda.

Prav tako problem nastane pri projektih, ki imajo manjši proračun. Uporabniška izkušnja pri takih projektih poveča strošek izdelave, ki pa ga naročnik ni pripravljen sprejeti, zato so taki izdelki na koncu pogosto narejeni na podlagi predvidevanj in primerov uporabe z drugih spletnih strani. Seveda s tem ni nič narobe, dokler so cilji spletne strani ter ciljna skupina uporabnikov, ki jih nagovarjamo, vsaj približno enaki.

4.2 Kako se podjetja lotijo projektnega vodenja

Pri lotevanju projektnega vodenja so nekatera podjetja bližje agilnim pristopom in metodologiji Scrum, spet druga pa bližje tradicionalnemu pristopu Waterfall. Nobeno izmed podjetij pa se strogo ne drži pravil ene ali druge metodologije. Vsako je našlo svojo pot in svoj pristop, ki jim najbolj ustreza, kar agilne metodologije tudi poudarjajo in svetujejo. Tako se nekateri vodje projektov na začetku sami sestanejo z naročnikom, kjer zberejo vse funkcionalne in vsebinske zahteve ter jih združijo in napišejo v dokument s specifikacijo projekta. Spet drugi se s celotno ekipo, ki bo sodelovala pri projektu (vodja projektov, oblikovalci, načrtovalci, programerji itd.) z naročnikom sestanejo in skupaj pišejo funkcionalnosti in zahteve končnega izdelka.

Malo podjetij vsak dan sklicuje tako imenovane sestanke scrum s celotno ekipo, kjer si izmenjajo informacije o že narejenih elementih končnega izdelka ter se dogovorijo o prihodnjem delu, ki ga bo posamezen član ekipe opravil v prihodnje.

Dejstvo, da se večina podjetij sestaja z naročnikom večkrat med celotnim potekom projekta, priča o tem, da je z nekaj sestanki pred začetkom izdelave projekta nemogoče priti do končne rešitve. Večina podjetij med razvojem naročniku kažejo posamezne delujoče dele izdelka, kar prav tako lahko vpliva na večjo pogostost sestankovanja z naročnikom. Podjetja pa se na morebitne naknadne spremembe in popravke med razvojem projektov odzivajo različno. Nekatera zahtevo integrirajo, druga jo zapišejo ter integrirajo v naslednji različici spletne strani ali spletne aplikacije. Nobeno podjetje pa ni odgovorilo, da zahtevo zavržejo, ker so z razvojem projekta predaleč. Iz povedanega lahko sklepamo, da so vsa podjetja našla način, pa naj bo ta preprost, zahteven, poceni ali drag, kako se na take naknadne spremembe odzvati.

Sodeč po anketah se podjetja v procesu vodenja projektov največkrat srečujejo s težavami (od najpogostejših do najmanj pogostih) pri potrditvi videza, pri določanju cen, pri komunikaciji z naročnikom in znotraj ekipe, pri upoštevanju rokov, pri vzpostavljanju avtoritete vodje projekta, pri odpravljanju napak končnega izdelka in na koncu pri izdelavi uporabniku prijaznega izdelka.

5. Sklepne ugotovitve

Zakaj pa je uporabniška izkušnja tako pomembna? Vzemimo za primer iPhone. Njegova uspešnost ne tiči v procesorski moči, superkameri, uporabljeni tehnologiji itd. Njegova uspešnost je povezana predvsem z uporabniško izkušnjo. IPhone preprosto deluje in opravlja funkcije na intuitiven način, ki je blizu uporabniškega načina razmišljanja.

Ali je zavoljo vključevanja načrtovalcev uporabniške izkušnje v proces vodenja projektov smiselno spreminjati celotno metodologijo vodenja projektov, ki smo je že navajeni? Seveda ne. Glavno je le, da se najde način, kako pred začetkom in med razvojem v proces vključiti načrtovalce uporabniške izkušnje in upoštevati njihove ugotovitve. Predstavljena agilna metodologija Scrum uporablja prijeme, ki so pisani na kožo načrtovalcem uporabniške izkušnje, zato je njihovo vključevanje lažje. Pri metodologijah, ki težko sprejmejo spremembe globoko v razvoju izdelka, bodo načrtovalci uporabniške izkušnje težko delovali. Njihove želje in ugotovitve raziskav uporabniških odzivov na testiranje med razvojem bodo tako hitro naletela na gluha ušesa, saj bodo spremembe s seboj prinesle preveč stroškov in bo tako projekt presegel dogovorjeni rok končanja.

Zaposlenim preprosto ne sme biti vseeno. Dobra uporabniška izkušnja izdelka ne bo nastala po naključju. V podjetju je treba imeti vsaj enega izkušenega načrtovalca uporabniške izkušnje, in ne zgolj oblikovalca, ki je tudi strokovnjak za načrtovanje uporabniških vmesnikov *(angl. UI design expert)*. Imeti je treba nekoga, ki bo veliko testiral, uporabljal prototipe, predstavljal rezultate itd. Dobra uporabniška izkušnja bo s takim načinom dela prišla sama od sebe.

6. Priloge

Dodatek A

Vprašalnik o vodenju projektov ter uporabniški izkušnji

Sem absolvent fakultete za Računalništvo in informatiko. S spletom ter spletnimi tehnologijami se ukvarjam že skozi celotno srednjo šolo in fakulteto, kar je močno pripomoglo k moji odločitvi, da sem za svojo diplomsko nalogo izbral temo: Uporabniška izkušnja pri vodenju spletnih projektov. Ker diplomsko delo vsebuje tudi pregled stanja na področju vodenja spletnih projektov in uporabniške izkušnje v slovenskih podjetjih, sem za vas pripravil nekaj vprašanj. Hvala za vaš čas in odgovore.

1. **Ime podjetja: _____**

2. **Vaš položaj v podjetju**
   - Direktor
   - Vodja projektov
   - Načrtovalec uporabniške izkušnje
   - Oblikovalec
   - Programer
   - Drugo: _____

3. **Kako je sestavljena vaša ekipa?**
   Število vodij projektov, načrtovalcev uporabniške izkušnje, oblikovalcev, programerjev ...

4. **Z naročnikom se**
   - sestanemo v povprečju enkrat pred začetkom projekta ter na koncu ob predaji projekta. Vmesni sestanki so sklicani redko oziroma takrat ko je to res potrebno.
   - sestajamo večkrat (na 2–4 tedne), med celotnim potekom projekta.
   - Drugo: _____

5. **Vsak delovni dan v podjetju začnemo s sestankom z ekipo (vodja projekta, programerji, oblikovalci, načrtovalci ...), ki sodeluje pri projektu?**
   - Da
   - Ne
   - Drugo: _____

6. **Na začetku projekta**
   - se vodja projekta sestane z naročnikom, s katerim skupaj zbereta funkcionalne in vsebinske zahteve končnega izdelka.
   - se skupaj sestanemo z naročnikom in celotno ekipo, ki sodeluje pri projektu. Na tem sestanku vsakemu od udeležencev razdelimo lističe, kamor vsak (v preprostem in razumljivem jeziku) napiše, kaj kot predstavnik določene skupine uporabnikov od končnega izdelka pričakuje.
   - Drugo: _____

7. **Naročniku pokažemo izdelek**
   - Ko je ta končan.
   - Postopoma, vsakič ko imamo delujoč posamezni del končnega izdelka.
   - Drugo: _____

8. **Testiranja se lotimo**
   - na koncu, ko je izdelek v celoti končan, preden ga predamo naročniku.
   - tako, da posamezne delujoče dele testiramo med razvojem končnega izdelka.
   - Drugo: _____

9. **Kako se odzivate na naknadne spremembe, tekom razvoja projekta?**
   Ko je že definiran načrt, ko je že pripravljen videz ter je že napisano 3/4 kode, se pojavi nova naročnikova zahteva. Kaj storite?
   - Zahtevo zavržemo, saj smo predaleč z razvojem.
   - Zahtevo integriramo.
   - Zahtevo zapišemo ter jo integriramo v naslednjo različico spletne strani.
   - Drugo: _____

10. Kje v procesu vodenja projektov imate največ težav?

(Možnih je več odgovorov)

- Pri upoštevanju rokov
- Pri potrditvi dizajna
- Pri določanju cene
- Pri komunikaciji z naročnikom
- Pri komunikaciji znotraj ekipe
- Pri vzpostavljanju avtoritete vodje projekta
- Pri predaji končnega izdelka naročniku
- Pri odpravljanju napak končnega izdelka
- Pri izdelavi uporabniku prijaznega izdelka

11. Katero aplikacijo uporabljate za nadzor nad projekti?

12. Sta vam pojma agilno vodenje projektov in metodologija Scrum znana?

- Da
- Ne

13. Imate v podjetju koga, ki je odgovoren za uporabniško izkušnjo (UX)?

- Da
- Ne

14. Je načrtovalec uporabniške izkušnje prisoten med celotnim razvojem projekta?

- Da
- Ne, zgolj na začetku, pri načrtovanju
- Nimamo načrtovalca uporabniške izkušnje

15. Ali se med razvojem pogovarjate in sodelujete s končnimi uporabniki?

- Da
- Ne

16. Sproti testirate posamezne elemente na uporabnikih?

- Da

- Ne

17. Ali pred začetkom načrtovanja projekta analizirate ciljne uporabnike in njihove navade?

- Da
- Ne

18. So vaše spletne strani/spletne aplikacije po vašem mnenju uporabniku prijazne in preproste za uporabo?

- Da
- Ne
- Ne vem

19. Ali končano spletno stran ali spletno aplikacijo testirate?

User testing, Usability testing, Task Analysis, ...?

- Da
- Ne
- Ne vem, kaj je to.

20. Ali izdelujete delujoče prototipe, ki jih testirate pri uporabnikih?

- Da
- Ne

21. Izvajate testiranja A/B?

- Da
- Ne
- Ne vem, kaj je to.

22. Kako v vašem podjetju združujete uporabniško izkušnjo in vodenje projektov?

Prostor za vaše izkušnje pri združevanju teh dveh pojmo v...

23. Kako bi ocenili spodnja področja v vašem podjetju:

1 – Slabo … 5 – Odlično

- Vodenje projektov
- Analiziranje in načrtovanje
- Uporabniška izkušnja končnega izdelka
- Oblikovanje
- Programiranje in razvoj

7. Slike

Slika 1 – Prikaz petih področij oziroma ravni uporabniške izkušnje 10
Slika 2 – Diagram strukture spletne strani, imenovan »Visual Vocabulary«..... 21
Slika 3 – Primeri elementov vmesnika 23
Slika 4 – Prikaz žičnega modela 26
Slika 5 – Sledenje gibanju očesa 28
Slika 6 – Primer slabo uporabljenega kontrasta na oglasu za Aspirin podjetja Bayer 29
Slika 7 – Prikaz posledice napačnega razumevanja naročnikovih želja in zahtev 36
Slika 8 – Metodologija waterfall 38
Slika 9 – Prirastni razvoj 39
Slika 10 – Iterativni razvoj 40
Slika 11 – Grafični prikaz razlike med metodologijo Waterfall in agilnimi metodologijami .. 43
Slika 12 – Prikaz kartotečnega listka 44
Slika 13 – Prikaz pogorigrama 47
Slika 14 – Prikaz vzporednega dela 55

8. Viri

[1] Jesse James Garret, The Elements Of User Experience, AIGA –- American Institute of Graphic Arts in New Riders, 2003

[2] Chris Nodder in Jakob Nielsen, Agile Usability: Best Practices for User Experience on Agile Development Projects, Nielsen Norman Group, 2010

[3] (2001) Artemis Solves Top 10 Project Management Problems. Dostopno na: http://dssresources.com/papers/features/yager10032001.htm

[4] (2010) Five User Experience Trends I'll be Watching in 2010. Dostopno na: http://nform.ca/blog/2010/01/five-user-experience-trends-il

[5] (2010) Recession’s Emerging Credo – Do More, Buy Less – NYTimes.com. Dostopno na: http://www.nytimes.com/2010/01/03/business/economy/03experience.html?_r=1&hp

[6] (2010) Rick Barron: User Experience Testing Methods. Dostopno na: http://www.slideshare.net/visionary/user-experience-testing-methods

[7] User Experience: Our Definition - Nielsen Norman Group. Dostopno na: http://www.nngroup.com/about/userexperience.html

[8] (2010) User experience - Wikipedia, the free encyclopedia. Dostopno na: http://en.wikipedia.org/wiki/User_experience