

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Žiga Cmerešek

Agilne metodologije razvoja programske opreme s poudarkom na metodologiji Scrum

Diplomsko delo

Ljubljana, 2015

UNIVERZA V LJUBLJANI
FAKULTETA ZA DRUŽBENE VEDE

Žiga Cmerek

Mentor: red. prof. dr. Andrej Mrvar

Agilne metodologije razvoja programske opreme s poudarkom na metodologiji Scrum

Diplomsko delo

Ljubljana, 2015

*Zahvaljujem se prijateljem, sošolcem in sodelavcem,
ki so me podpirali skozi celotno študijsko pot.*

*Posebna zahvala pa gre družini,
ki mi je to pot tudi omogočila in mi na njej stala ob strani.*

Agilne metodologije razvoja programske opreme s poudarkom na metodologiji Scrum

Diplomsko delo vsebuje pregled različnih metodologij razvoja programske opreme, tako klasičnih kot tudi agilne metodologije Scrum. Opredelitvi pojma projekt in opisu projektnega managementa sledi podroben opis nekaj najbolj uporabljenih klasičnih metodologij prejšnjega desetletja. V nadaljevanju naloge se skozi opis vseh prednosti in slabosti omenjenih klasičnih metodologij sprehodimo do agilnosti, kjer sledi podrobnejši opis omenjene metodologije Scrum in vseh njenih značilnosti. Naloga se konča s testnim primerom, pri čemer na osnovni ravni nakažemo vse možnosti prilagajanja, fleksibilnosti in ravnanja z metodologijo Scrum.

Ključne besede: projekt, projektni management, agilnost, Scrum.

Agile software development methodology with an emphasis on methodology Scrum

The thesis contains an overview of the different methodologies of software development, conventional as well as agile methodology Scrum. Through the definition of the noun project and description of the concept project management, this document also contains detailed description of some of the most traditional methodologies used in the preceding decade and all the advantages and disadvantages of it. The second part of the document, talks about agile methodology Scrum and all features and characteristics related to it. The paper concludes with test case, that shows us all of the customization options, flexibility and possibilities of management of the Scrum methodology.

Key words: project, project management, agility, Scrum.

Kazalo

1	Uvod	6
2	Projekt	7
2.1	Projekt kot pojem	7
2.2	Vrste projektov	8
2.3	Udeleženi in njihove vloge.....	9
2.4	Cilji projekta.....	11
3	Projektni management.....	13
3.1	Metodologije projektnega vodenja.....	13
3.1.1	Metodologija PRINCE	15
3.1.2	Microsoftova metodologija MSF	16
3.1.3	Slapovni model	17
4	Projektna agilnost.....	19
5	Metodologija Scrum	21
5.1	Uvod	21
5.2	Scrum ekipa	22
5.2.1	Skrbnik izdelka	22
5.2.2	Razvojna ekipa	23
5.2.3	Scrum master.....	24
5.3	Dogodki.....	24
5.4	Scrum planiranje.....	25
5.4.1	Ocenjevanje uporabniških zgodb	26
5.4.2	Poker planiranje	27
5.5	Opis testnega primera	29
6	Zaključek.....	33
7	Literatura:.....	34

Kazalo slik

Slika 5.1: Scrum proces.....	21
Slika 5.2: Poker planiranje – kartice	28

1 Uvod

V sodobnem poslovnem svetu je trg nepredvidljiv, prav tako so zahteve po spremembah trendov nenehne. Za organizacijo, ki želi uspešno preživeti v takšnem okolju, je ključnega pomena izbira sistema za upravljanja sistemskih procesov znotraj nje. Trg se namreč ne prilagaja, zato sta fleksibilnost in prilagodljivost organizacij vrlini, ki dostikrat veljata za najpomembnejši pri pridobivanju dobičkonosnih poslov.

Tako se v organizacijah, katerih rdeča nit poslovanja je razvoj programske opreme, poraja vprašanje izbire sistemskih procesov, kako v kar najkrajšem možnem času in s kar najmanj porabljenimi sredstvi, tako finančnimi kot tudi fizičnimi, implementirati nov izdelek, produkt ali storitev.

Agilne metode razvoja programske opreme so nastale kot odgovor na iskanje rešitve za visoko stopnjo tveganja pri samem razvoju takšnih projektov, za kršitev dogovorjenih časovnih rokov, nefleksibilnost projektov, netočno načrtovanje izvedbe celotnega razvojnega procesa in prekoračitev stroškov, namenjenih za določen razvoj.

Poleg omenjenih težav pri implementaciji novega izdelka ali storitve na tržišče se je pogosto pojavil tudi problem uvedbe že zastarelega modela izdelka ali storitve, saj je časovno obdobje razvoja trajalo preprosto predolgo, hitro spreminjajoči se trg pa tega ni dopuščal. Tako so klasične metodologije razvoja klicale po novih, fleksibilnejših in predvsem bolj agilnih pristopih razvoja programske opreme, kjer je pogosto spreminjanje zahtev in dodajanje nalog v proces razvoja nekaj povsem normalnega, sprejemljivega in izvedljivega.

Tako bom v nalogi pripravili pregled najbolj priljubljenih klasičnih metodologij razvoja programske opreme ter naštel njihove prednosti in slabosti, v nadaljevanju pa se bom posvetil agilni metodologiji Scrum, ki velja za trenutno najbolj uporabljeno in najučinkovitejšo metodologijo v omenjenem okolju razvijanja programske opreme. Poleg podrobnega pregleda metodologije se bom dotaknili vlog, ki jih definira Scrum, delovnih ciklov pri samem izvajanju razvojnega procesa, načrtovanja in planiranja časovnih okvirjev in Scrum dogodkov, ki jih definira metodologija. Sledil bo podrobnejši opis ene od tehnik načrtovanja, in sicer poker planiranja. Nalogo bom zaključili s pregledom testnega primera, pri čemer bom

na osnovnem primeru prikazali celoten proces razvoja programske opreme po metodologiji Scrum.

2 Projekt

2.1 Projekt kot pojem

Projekt je nedvomno beseda, ki jo pogosto in s pozitivnim prizvokom uporabljamo na različnih življenjskih področjih in med različnimi družbenimi interakcijami. Izhaja iz latinske besede *proiectum*, ki pomeni vreči predse, v našem jeziku pa ima širši in ožji pomen. »V ožjem pomenu besede pod pojmom projektiranje razumemo načrt, skico, predlog za delo, postopek, medtem ko v širšem smislu pomeni vsako delotvorno zamisel oziroma odločitev« (Burke 1995, 1).

Pojem se uporablja na mnogo različnih področjih, zato bi pričakovali, da zanj obstaja jasna in enotna opredelitev, vendar ni tako. Po pregledu izbrane literature lahko ugotovimo, da je vsak avtor nagnjen k nekoliko drugačnemu dojemanju pojma in posledično drugačni interpretaciji. Opredelitve se tako v teoriji vsebinsko razlikujejo, vendar si ne nasprotujejo, bolj bi zadelo, če bi rekli, da imajo opredelitve različne poudarke in različno stopnjo zajemanja značilnosti projektov. Pri preučitvi različnih avtorjev, starejših in novejših, smo naleteli na kopico teorij, ki jih bomo v nadaljevanju predstavili.

Burke (2000, 2) pri projektih opozarja na dve stvari, in sicer začasnost in enkratnost projekta. Prav tako omenja življenjski cikel projekta, porabo sredstev, vključevanje zaposlenih in opredelitev različnih vlog pri projektih.

Po mnenju Locka (2003, 4) je vsak projekt korak v neznanu, poln tveganja in negotovosti. Prav tako poudarja, da je glavni temelj projekta novost, in opozarja na razlike med posameznimi okolji, v katerih se projekt izvaja.

Rozman in Stare (2008, 7) pa ga opredelita kot »podjem (širšo dejavnost, delo) med seboj povezanih zaposlenih, sredstev in aktivnosti, za katerega so značilni neponovljivost projektne procesa in enkratnost proizvoda ali storitve, s tem časovna omejenost celotne dejavnosti in sodelovanje različnih sodelavcev in sredstev v projektu«.

Na podlagi pregleda literature lahko povzamemo, da so skupne značilnosti projekta:

- predlog za delo ali postopek dela, ki vodi k novemu proizvodu ali storitvi;
- proces, ki se ne ponavlja z identičnimi predpostavkami;
- ima vnaprej določene kompleksnosti, cilje, časovni potek in sredstva za realizacijo;
- korak v neznanu, poln tveganja in negotovosti;
- tudi dva projekta med seboj nista povsem enaka;
- uspešno zaključen projekt vodi k izpolnitvi cilja ali namena, ki je vnaprej določen.

2.2 Vrste projektov

Projekte v teoriji in praksi razvrščamo na različne načine, saj zanje ni določenih enotnih kriterijev. Lahko bi rekli, da jih razlikujemo glede na gospodarsko področje, v katerem delujejo in se uporabljajo, glede na zahtevnost in obvladovanje samega projekta ter glede na njegovo ponovljivost.

Solina razlikuje enkratne projekte in projektne procese. Enkratni projekti se pojavljajo le enkrat, medtem ko se projektne procesi v podobnih okoliščinah ponovijo večkrat. To so tipski projekti z enakimi ekonomskimi ali tehnološkimi značilnostmi. Kot primer pri tem navaja inženiring, projekte v gradbeništvu, razvoj programske opreme ipd. (Solina 1997, 65).

Meherman in Wirtz (v Kovač 1995, 149) delita skupine na način in namen njihovega delovanja: investicijski, raziskovalni in razvojni projekti. Kovač (prav tam) loči glavne skupine z dvema determinantama, ki sta pri uspešni izpeljavi posameznega projekta ključnega pomena: časovno obdobje vračanja vloženih sredstev in njihov naročnik. Medtem ko je naročnik pri investicijskih in organizacijskih projektih znan, so pri razvojnih projektih glavni akterji notranji naročniki, kot je marketing ali prodaja. Tako se časovna obdobja vračanja sredstev ločijo glede na vrsto projekta. Pri investicijskih projektih je ta doba krajša, medtem ko se pri ostalih dveh časovno obdobje vračila sredstev nekoliko podaljša, saj je v veliki meri odvisno od dejavnikov, ki se stimulirajo šele po uspešni izvedbi posameznega projekta.

Stare po Turnerju razdeli projekte po velikosti, in sicer na » majhne, srednje, obsežne, velike in večprojektne. Takšna delitev razdeli projekte po obsežnosti, številu interakcij akterjev znotraj tega, številu sodelujočih, po časovnem obdobju, v katerem se izvajajo, finančnem riziku, ki ga projekt predstavlja, in končnem učinku dodane vrednosti določenega projekta« (Turner v Stare, 2008).

Projekti se med seboj v glavnem razlikujejo glede na področje, na katerem delujejo, glede na namen – zakaj se izvajajo, ter glede na način izvedbe – kako bodo izpeljani. V želji uspešne izpeljave določenega projekta se morajo tako naročniki kot tudi izvajalci pred izvedbo projekta striktno dogovoriti o vrsti projekta in na podlagi njegove opredelitve natančno določiti smernice, ki bodo vodile k finančnem in osebnem uspehu.

2.3 Udeleženi in njihove vloge

Na celotni poti, od snovanja projekta pa vse do njegovega zaključka, je načeloma udeleženih več ljudi, ki delujejo na različnih področjih znanja. Tako vsak udeleženec s svojo vlogo doda svoj košček znanja in s tem pripomore k uspešni sestavitvi mozaika. Prav tako se glede na vloge procesa deli odgovornost, ki se prenaša od začetka, prek izvedbe, pa vse do konca posameznega projekta. Jasno dodeljevanje vlog pri določenem projektu v začetni fazi je pri realizaciji ciljev ključnega pomena, saj se kasnejše spreminjanje osebne odgovornosti, pravic, zaslug in dolžnosti za projekt lahko konča tragično.

Udeležence in njihove vloge nekateri avtorji delijo na:

- aktivne udeležence (angl. *key players*), ki so vključeni v projektno organizacijo in formalno sodelujejo pri izvajanju projekta;
- vplivneže projekta (angl. *influencers*), ki posredno vplivajo na doseganje rezultatov projekta s formalno ali prikrito podporo (Stare 2008, 40).

Tako bomo v nadaljevanju po različnih avtorjih (Golob 2001; Kerzner, 2001; Stare, 2008) povzeli opredelitve nekaj najpomembnejših in nepogrešljivih vlog, ko govorimo o projektih in njihovih nalogah ter odgovornostih.

Projektni manager: je oseba, ki je osebno odgovorna za postavitev ciljev, izvajanje ter doseganje učinkovitosti določenega projekta. To naj bi dosegal z ustreznim organiziranjem projekta kot celote, motiviranjem ekipe, ki sodeluje pri projektu, ustreznim planiranjem časovnih okvirjev za izvedbo procesa ter rednim kontroliranjem izvedbe manjših procesov na podlagi celotnega projekta.

Vrhnji management (uprava): glavni namen in naloga uprave v združbi je skrb za usklajenost cilja projekta s strateškimi usmeritvami združbe. Prav tako je odgovorna za dodeljevanje resursov pri podpori projekta (denar, ljudi, opremo ipd.) ter za nadzorovanje projekta skozi celoten življenjski cikel.

Stranka ali naročnik: je razlog, da do snovanja projekta sploh pride. Je oseba ali združba, ki se po zaključku projekta koristi z rezultatom. V prvi vrsti opredeli namen in cilje projekta tako s projektnim managerjem kot tudi z upravo, v nadaljnjih fazah projekta pa potrjuje vmesne rezultate in potrdi končni izdelek projekta.

Projektni tim: je sestavljen iz izvajalcev z vnaprej določenimi aktivnostmi in nalogami, ki vodijo k izdelavi izdelka ali storitve. Ti ljudje so kvalificirani za točno določeno delo in imajo za to potrebno strokovno znanje. Stare po navedbi različnih avtorjev zapiše, da obstajata dva nivoja tima:

ožji tim, ki ga sestavljajo najožji sodelavci managerja projekta, ki so običajno strokovni nosilci posameznih strokovnih področij, ki jih vključuje projekt. Člani ožjega tima, ki projektu posvetijo vsaj 60% (običajno pa 100%) svojega delovnega časa za celoten čas projekta, so največkrat določeni že pred začetkom priprave projekta, lahko na predlog skrbnika, managerja projekta ali sveta projekta. Zaradi strokovnega znanja in izkušenj je zelo pomembna njihova vloga v fazi priprave projekta (strategija izvedbe, potrebne aktivnosti, trajanje aktivnosti in potrebna količina dela, tveganja, ipd.). (Stare 2008, 42);

širši tim sestavljajo ostali izvajalci projektnih aktivnosti, ki delujejo pod neposrednim vodstvom strokovnih nosilcev. Člani širšega tima se določijo v fazi planiranja projekta, nekateri pa celo med izvajanjem projekta (primer: v planu se določi, da se za montažo potrebuje pet monterjev, monterje pa izbere vodja šele tik pred izvedbo). Za člane širšega tima ni nujno, da so na projektu prisotni celoten čas. (Stare 2008, 42).

Poslovno-funkcijski managerji: so odgovorni za pridobitev delovne sile pri projektu in s tem zagotovitev kar najboljših rezultatov, kakovosti in učinkovitosti izpeljanega projekta. Po potrebi sodelujejo tudi kot strokovni svetovalci.

Kot vidimo, so vloge pri projektu zelo različne in zelo jasno določene. Dodelitev in natančna opredelitev vlog v začetni fazi načrtovanja projekta je tako za uspešno izvedbo ključnega pomena.

2.4 Cilji projekta

Cilji projekta so osrednji element pri opredeljevanju projekta. Z njimi določimo rezultate projekta, ki morajo biti doseženi, da zadovoljijo potrebe naročnika. Cilji morajo biti količinsko in kakovostno opredeljeni, vsebovati pa morajo tudi predvidene stroške projekta in časovni okvir (Kovač 1995, 158).

Povzeto po Wirtzu in Mehrmannu (v Kovač 1995, 158) so funkcije ciljev projekta:

- usmerjanje izvajanja projekta;
- selekcija projektnih rešitev;
- osnova za planiranje izvedbe (s pomočjo razgraditve na aktivnosti);
- kontrola izvedbe.

Pri kršenju katerekoli od osnovnih funkcij opredelitve ciljev projekta lahko pride do konfliktnih situacij in sprememb že na samem začetku projekta, ki pa v večini primerov vodijo v pogubno stanje. To se največkrat zgodi zaradi nejasne opredelitve ali drugačnega dojetja cilja projekta različnih akterjev, ki sodelujejo pri projektu. Zato je treba zagotoviti povsem jasno sliko celotnega načrta projekta, skupaj z njegovim ciljem in namenom v zgodnji fazi načrtovanja, in jo predstaviti vsem udeležencem, ožjim in širšim, kajti cilj projekta je tudi osnova za merjenje učinkovitosti izvajanja projekta.

Na tej točki je pomembno omeniti, da pogosto pride do zamenjevanja pojmov namen in cilj, kar zna dodatno zaplesti izvedbo. Ljudje si namreč pojma razlagajo različno, pride pa tudi do zamenjevanja, zato si oglejmo, kako ju razloži angleški slovar.

Oxford Advanced Learner's Dictionary ju opredeljuje takole:

Namen (ang. *Goal*) – nekaj, za kar se nadejaš, da boš dosegel (ang. *Something that you hope to achieve*), dolgoročni cilj. Uporaba (ang.): *to work towards a goal to achieve / attain a goal; You need to set yourself some long-term goals; our ultimate goal must be the preservation of the environment* (Pojmovnik 2010).

Cilj (ang. *Objective*) – za razliko od izraza »goal« je »objective« nekaj, za kar se trudiš, da bi dosegel (ang. *something that you are trying to achieve*). Uporaba (ang.): *the main/primary/principal objective; to meet /achieve your objectives; you must set realistic aims and objectives for yourself; the main objective of this meeting is to give more information on our plans* (Pojmovnik 2010) .

Vidimo, da sta pojma različna, prav tako imata v projektih popolnoma različne vloge. Zanimiva je tudi delitev odgovornosti udeležencev projekta pri opredelitvi pojmov namen in cilj. Vodja projekta naj bi s svojo ekipo dosegal cilje projekta in tako po uspešno zaključenem procesu predal nov proizvod ali storitev naročniku, ki pa bi z uporabo tega proizvoda ali storitve poskušal uresničiti namen projekta. Če gremo korak dlje, lahko iz tega izpeljemo, da vodja s svojo ekipo ter izvajalci neposredno ni odgovoren za morebitno nedoseganje namena projekta ali ciljev poslovanja, zaradi katerih je bil projekt izveden, pač pa lahko nedoseganje namena projekta pripisujemo naročniku, ki je strateško načrtoval in snoval določen projekt (Stare 2008, 23).

Namen je torej nekakšen posredni cilj, za katerega pa ni nujno, da ga dosežemo z uspešno izpeljanim projektom. Lahko je le prvi korak do uresnitve namena, ki pa ga na daljši rok uresniči. Rozman pravi, da je »namen projekta največkrat družbeno-ekonomski ali zunanje določen (npr. dobiček)« (Rozman 2008, 8). V mnogo primerih je eden od strateških ciljev združbe dobičkonosnost ali donosnost, lahko pa tudi družbena odgovornost, izražena kot novo ustvarjena ali dodana vrednost na zaposlenega. Cilj projekta pa je izvesti strategijo, ustvariti učinek ali uporabno vrednost (zahtevane kakovosti) na smotrni, učinkoviti način v čim krajšem časovnem obdobju (Stare 2008, 23).

3 Projektni management

Projektni management je dokaj nov pojem, ki se uporablja na več področjih. Tehnična delitev dela, kot jo imenuje stroka, se je pojavila kot odgovor na vedno težje, obsežnejše in zahtevnejše naloge oziroma sklop večjih nalog, projektov. Razčlenitev na manjše naloge in pripis teh specializiranim posameznikom sta olajšala zahtevnost in obseg začetnih projektov. Tako je prišlo do večje storilnosti, zmanjšanja stroškov in navsezadnje tudi do izboljšanja kakovosti izdelkov ali storitev. Vendar pa je takšna razčlemba projekta zahtevala usklajevanje različnih faz delitve tega, ki pa je brez predpisane organizacijske strukture skoraj nemogoča. Tako je prišlo do neuskklajenosti med elementi razčlenjenega procesa dela. Zaradi pomanjkanja takšne discipline se pojavi pojem projektni management, ki v ožjem smislu, če povzamemo razne avtorje, pomeni usklajevanje vnaprej razčlenjenih nalog pri projektu kot takem (Rozman, Kovač, Koletnik v Stare 2008, 46) .

Nekateri avtorji definirajo management tudi kot ustvarjalno reševanje problemov, na primer ugotavljanje vzrokov, opredeljevanje ter izbiro alternativ, načrtovanje in kontrolo – v okviru omenjenih štirih procesov za doseganje ciljev in razvoja. Reševanje problemov je pri tem ključno, ker nastopa v vseh nalogah in dejavnostih managementa, pri čemer je še posebno pomembno odločanje o izbiri najbolj primerne rešitve problema – velikokrat takšne, ki dotlej še ni bila znana (Možina 1994, 16).

Tako se skupaj s projektnim managementom poraja vprašanje, katera metoda oziroma metodologija je najprimernejša, najboljša in najučinkovitejša pri izvedbi določenega projekta. Na to vprašanje bomo poskušali odgovoriti skozi naslednja poglavja.

3.1 Metodologije projektnega vodenja

Ko govorimo o metodologijah, ki se uporabljajo pri vodenju projektov, se moramo zavedati različnosti in nesimetričnosti vsakega projekta, saj – kot smo že omenili – si niti dva projekta med seboj nista enaka. Tako je izbira pravilne metodologije zelo zahtevno delo, saj na to odločitev vplivajo različni dejavniki in smernice posameznega projekta. Vsako podjetje, ki trenutno še ne uporablja ustreznega ogrodja projektne metodologije, mora to najprej identificirati, izbrati, prikrojiti lastnim potrebam in implementirati, preden jo začne

uporabljati. Tako mora vodstvo podjetja pred odločitvijo o vpeljavi projektne metodologije, ki bo zagotavljala uspeh, najprej razmisliti o naslednjih ciljnih (Charvat 2003, 26):

- celotni strategiji podjetja – kako je podjetje konkurenčno na trgu;
- velikosti projektne skupine in/ali obsega projekta;
- prioriteti posameznega projekta;
- kako kritičen je projekt za podjetje;
- kako fleksibilna naj bo metodologija in katere so njene komponente.

Odgovornost odločitve o najprimernejši metodologiji v podjetju pripada projektному vodji, ki pa se odloča na podlagi naslednjih dejavnikov: proračun projekta, velikost projektne skupine, orodja in tehnike, kritičnost projekta in obstoječi procesi (Kerzner 1992, 289).

Proračun projekta: Proračun je ključnega pomena tako pri določanju ciljev pri projektu kot pri izbiri metodologije. Pri projektih, pri katerih ni finančne omejitve, vendar je ključnega pomena časovni okvir, uporabljamo metodologije, ki so dovolj odzivne, da lahko hitro obvladujejo spremembe novih interakcij.

Velikost projektne skupine: Pri velikosti projektnih skupin lahko rečemo, da je glavni pokazatelj zahtevnosti samega projekta. Posledično tako pri manjših skupinah uporabljamo lažje obvladljive metodologije, medtem ko so za večje delovne skupine ali skupine, ki so geografsko razpršene, bolj primerne tako imenovane »težke« oziroma kompleksnejše metodologije.

Orodja in tehnike: Orodja in tehnike uporabe so v večji meri odvisne od zahtevnosti projekta. Lažji in manj zahtevni projekti tako uporabljajo manj orodij ali sploh ne, medtem ko zahtevnejši uporabljajo orodja, kot so uporaba in upravljanje podatkovnih baz, orodja za modeliranje, tehnike vodenja in druga orodja, ki prav tako potrebujejo natančno klasifikacijo pri samem vodstvu določenega projekta.

Kritičnost projekta: Je pojem, na podlagi katerega lahko merimo zahtevnost določenega projekta. Kritičnost projekta namreč pomeni, kolikšna bi bila škoda ob predpostavki, da projekt propade, tako med procesom kakor tudi ob zaključku. Nastala škoda se lahko meri v različnih smereh, tako s finančnega kot tudi s človeškega vidika.

Obstoječi procesi: Na podlagi procesov, ki se v določenem okolju že izvajajo, se je treba naučiti in s tem prilagoditi novim procesom, saj lahko le tako izboljšamo postopek izvajanja določenega projekta.

Skozi čas so se razvile različne tehnike oziroma metodologije izvajanja procesa vodenja. Zaradi boljše preglednosti in razumevanja bomo na kratko omenili nekaj najpogostejših klasičnih metod, ki se pojavljajo pri projektne vodenju, in njihove značilnosti.

3.1.1 Metodologija PRINCE

Najbolj znana klasična metoda vodenja projektov je tako imenovani PRINCE, ki je kratica za *Projects IN Controlled Environments*. Metodologija je bila razvita leta 1989 na Central Computer and Telecommunications Agency (CCTA) v Veliki Britaniji in je postala vladni standard za IT-projekte. Leta 1996 se je začela uporabljati kot splošna metodologija projektne vodenja in je tako postala *de-factor standard* v Veliki Britaniji. Glavni namen razvoja in uvedbe metodologije je bil preseči prakso, v kateri se naloge lansirajo premalo organizirano in določeno. Prav tako je bil namen odpraviti ali vsaj znižati verjetnost glavnih razlogov za neuspeh nekaterih projektov, kot so:

- pomanjkanje koordinacije pri planiranju in uporabi virov;
- pomanjkanje koordinacije pri planiranju in izvajanju aktivnosti;
- pomanjkanje komunikacije med udeleženci pri projektu;
- nepravilno planiranje časa, finančnih in ostalih virov;
- pomanjkanje nadzora nad napredkom projekta;
- pomanjkanje nadzora nad kakovostjo izdelkov projekta;
- pomanjkanje podpore vodstva projektom.

Metodologija PRINCE temelji na predvidevanem planiranju in vodenju z definiranimi aktivnostmi in rezultati, ki potekajo znotraj nadzorovanega okolja, katerega temelj pa je jasno določena organizacijska struktura. Struktura kot temelj celotne metodologije zahteva točno določeno delitev vlog vseh udeležencev pri projektu, njihovih nalog, zadolžitev in odgovornosti ter pravil poročanja in nadziranja projekta.

Tako lahko zapišemo, da metodologija temelji na delitvi odgovornosti med naročnikom in izvajalcem in s tem pridobi skrb za korist projekta. Tako vsi vpleteni skrbijo za uspešno izvajanje projekta, saj čutijo osebno odgovornost do uspešno izpeljanega projekta.

Oglejmo si prednosti metodologije PRINCE (Bentley 1997):

- zahteva natančno opredeljevanje vlog, zadolžitev, odgovornosti in pravil pri projektu;
- zahteva natančno pripravo na projekt in njegovo izvajanje;
- predvideva razvoj storitev ali izdelkov;
- lahko jo uporabimo pri vseh vrstah projektov, saj je prilagodljiva;
- predvideva nadzor nad dejavniki tveganja;
- zagotavlja vključevanje uporabnika, izvajalca in naročnika;
- je popolnoma usmerjena v izdelke projekta.

Po pregledu literature lahko zapišemo, da so bistvene prednosti pri vpeljavi metodologije PRINCE povečanje učinkovitosti projekta, poslovna utemeljitev projekta in stalen nadzor nad kadri, ki delujejo v posamezni organizaciji.

3.1.2 Microsoftova metodologija MSF

MSF je metodologija, ki ga je razvilo podjetje Microsoft in podobno kot metodologija PRINCE temelji na razdeljevanju vlog ter dodeljevanju nalog in odgovornosti vsem udeležencem. Loči šest vlog, ki so med seboj enakovredne: produktno, programsko, razvojno, uporabniško, testno in logistično. Neodvisno od teh vlog in hkrati najpomembnejšo dodaja vlogo vodje projekta, ki tako nosi odgovornost za uspeh ali neuspeh projekta. Pri tej metodologiji je razvoj implementiran skozi štiri faze: definicije zahtev, načrtovanja, izdelave in stabilizacije. Izdelek ali storitev, ki se pridobi pri vsaki uspešno končani fazi, je ločen z različnimi verzijami. Metodologija zahteva jasno opredeljen časovni okvir zaključka projekta, ki pa se lahko ob predpostavki, da gre za večji projekt, deli na podprojekte, pri katerih se natančno določijo krajši časovni intervali, v katerih je zajeta določena funkcionalnost. Te

funkcionalnosti imenujemo verzije izdelka ali storitve, ki skupaj, ob uspešnem zaključku projekta, tvorijo cilj oziroma namen projekta. Takšen način razvoja prav tako zagotavlja sprotno testiranje in hitro odpravljanje napak, kar pa posredno boljša kakovost izdelka oziroma storitve.

Prednosti metodologije MSF so predvsem v:

- motivaciji in predanosti udeležencev pri projektu – zaradi kratkoročno videnih rezultatov;
- opredelitvi kratkoročnih ciljev;
- nižanju stroškov projekta – zaradi hitrejšega zaznavanja in odpravljanja odstopanj od prvotnega načrtovanja ciljev.

Slabost metodologije MSF se kaže pri dodeljevanju vlog udeležencem pri projektu, saj zahteva šest enakovrednih vlog, ki si prav tako delijo enako mero odgovornosti pri projektu, kar pa lahko privede do težav pri odločanju v spornih in kritičnih situacijah.

3.1.3 Slapovni model

Kot že ime pove, slapovna metodologija predstavlja zaporedje faz razvoja po principu izlivanja slapa v reko. Glavna značilnost te metodologije je, da ko se prvotna faza prelevi v naslednjo fazo razvoja, vračanje v predhodno fazo ni mogoče, kar pa pomeni, da ne dovoljuje napak. Zato je takšna metodologija močno omejena. Kot odgovor na potrebo vse pogostejših sprememb med samim postopkom razvoja so metodologijo nadgradili in vanjo vključili povratno informacijo v obliki testiranja, ki predstavlja preverjanje kakovosti izdelka oziroma storitve. Tako je metodologija primerna predvsem za kompleksne projekte, katerih struktura je zelo statična in pri katerih lahko zagotovimo, da skozi faze razvoja ne bo prišlo do kakršnihkoli sprememb.

Prednosti metodologije:

Zmanjševanje količinsko režijskega dela, ki pa samo po sebi ni v neposredni povezavi s procesom izvajanja projekta, saj je celotno načrtovanje mogoče izvesti vnaprej.

Slabosti metodologije:

- nefleksibilnost;
- okrnjenost pri vračanju v predhodne faze projekta;
- omejenost, saj zahteva zaključek nekega postopka pred prehodom na naslednjega;
- vse do zadnje faze razvoja lahko predstavlja sistemsko tveganje neustreznosti zahtevam.

Krajši pregled nekaj najpogostejših in najbolj znanih klasičnih metodologij projektne managementa nam potrjuje kompleksnost izbire najprimernejše metodologije pri določenem projektu. Kot smo že omenili, metodologija, ki bi ustrezala vsem vrstam projektov, ne obstaja, vendar se lahko s primernimi znanji približamo okolju našega projekta in posledično izberemo najprimernejšo in s tem zmanjšamo tveganje na minimum. Nekatera podjetja se odločajo za razvoj svojih metodologij, spet druga za nakup komercialnih. Pri odločanju so izredno pomembni naslednji dejavniki:

- natančno poznavanje vrste okolja, v katerem se projekt odvija;
- delovna skupina, ki je odgovorna za izvajanje nalog in zahtev;
- natančno razumevanje vnaprej določenega cilja oziroma namena projekta.

Zaradi vse bolj dinamičnega poslovnega okolja, hitro spreminjajočih se trendov in dnevnega povečevanja konkurence v informacijskih krogih se določene klasične metodologije niso izkazale za najbolj uspešne. Hiter razvoj in nenehna konkurenčnost na trgu sta klicala po uvedbi sprememb in razvoju novih, sodobnejših in predvsem bolj dinamičnih metodologij, katerih glavna značilnost je možnost prilagajanja izdelka oziroma storitve že v samem procesu njegovega/njenega razvoja. Tako so se v luči sodobnega projektne managementa rodile agilne metode projektne vodenja, katerih glavna značilnost je agilnost, ki je odgovor na nenehno spreminjajoči se trg in trende, ki mu sledijo.

4 Projektna agilnost

Agilen pristop k projektu pomeni sposobnost hitre reakcije v procesu izvedbe projekta. Bistvene značilnosti takšne metodologije so:

- postavljanje prioritete na razvoj izdelka;
- delujoča programska koda;
- enakovredna delitev odgovornosti na celotno razvojno skupino in zavedanje o dinamičnosti okolja;
- stalna ozaveščenost o morebitnih spremembah končnega izdelka ali storitve.

Na podlagi teh elementov so se metodologije izkazale za uspešnejše in primernejše od tradicionalnih, klasičnih metodologij. Zamisel o razvoju agilnih metodologij se je rodila v devetdesetih letih prejšnjega stoletja, leta 2001 pa je bil postavljen manifest agilnosti.

Manifest agilnosti je skupek vrednot, ki so ključnega pomena pri uporabi takšnih metodologij. Kot že omenjeno, metodologija temelji na štirih bistvenih značilnostih, na katerih se gradi proces razvoja nekega izdelka ali storitve. Manifest je predstavilo 17 strokovnjakov na področju agilnega razvoja in ga lahko opredelimo z naslednjimi lastnostmi:

- bistvena je sposobnost zmanjševanja nepotrebnega dela in s tem poenostavljanje procesa razvoja, saj se s tem izognemo nepotrebni težavam in zapravljanju dragocenega časa;
- težnja k odličnosti prek snovanja, načrtovanja pa vse do izvedbe procesa razvoja izboljša agilnost;
- najvišja prioriteta pri metodi je zadovoljitev stranke in striktno upoštevanje postavljenih časovnih okvirjev;
- omogočeno je sprejemanje zakasnelih ali celo poznih zahtev v fazi razvoja, kar je ena izmed glavnih prednosti razvoja v sodobnem času;
- delujočo verzijo izdelka ali storitve izdajamo vsaj enkrat mesečno, saj nam pogosta preverba kakovosti in konkurenčnosti trga nakaže možnost sprememb razvojnih smernic;

- agilni procesi promovirajo trajnostni razvoj; sponzorji, razvijalci in uporabniki morajo biti zmožni konstantnega tempa za nedoločen čas;
- načelo samoorganiziranosti v razvojnem okolju je ključ do najboljših arhitektur, planiranj in načrtovanj;
- predvidevanje preteklih napak in možnost reagiranja nanj v prihodnje je sposobnost, skozi katero organizacija »raste«; s konstantnim prilagajanjem se spoprijema z metodologijami, ki so v določenem času in okolju zanjo najučinkovitejše;
- navsezadnje je zelo pomemben tudi človeški dejavnik; vsi udeleženci, tako vodstvo kot razvijalci in naročniki, morajo skozi celoten proces dnevno ali vsaj tedensko sodelovati; pomembna je zagotovitev primerne delovnega okolja, podpore in zadostne mere zaupanja ter motivacije.

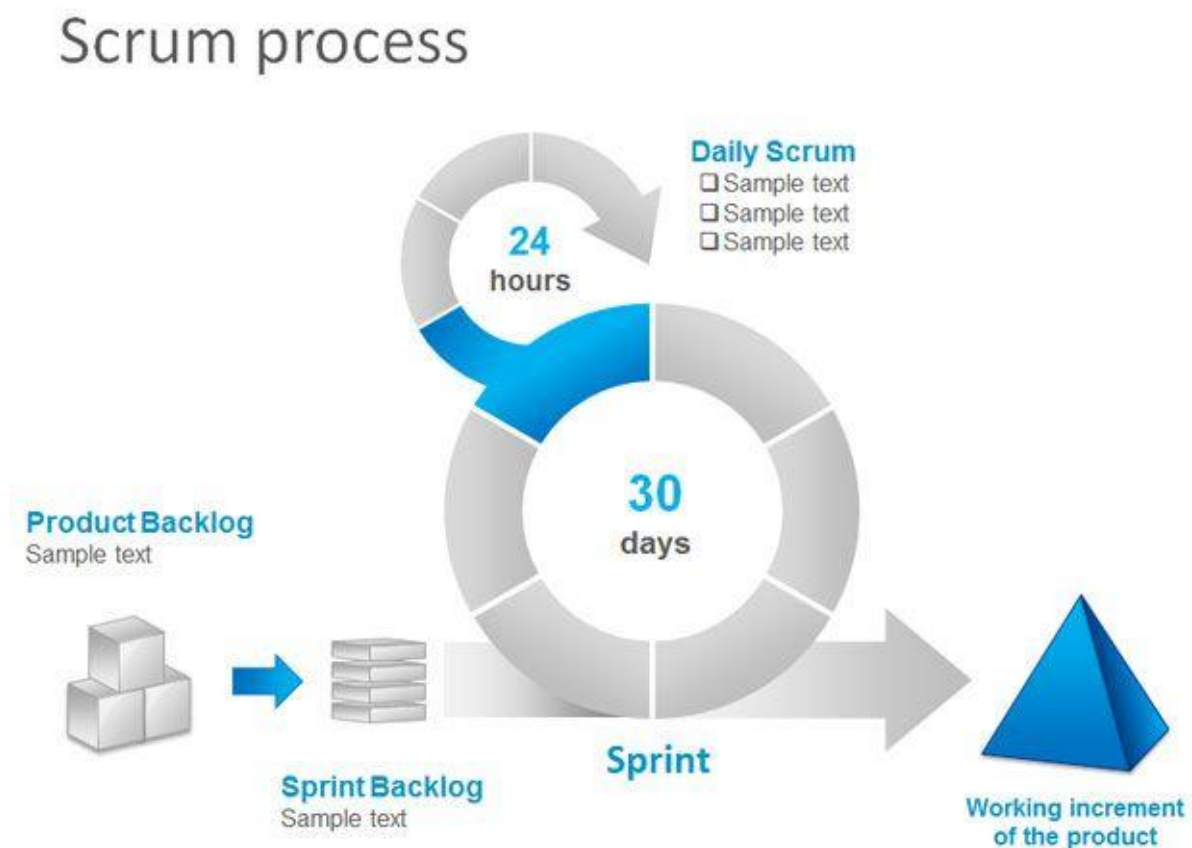
V sklopu zakonitosti, ki jih določajo agilne metodologije, je prišlo do njihove razširitve in začela se je množična uporaba agilne oziroma tako imenovane »prilagojevalne« metodologije, saj se je izkazala za uspešnejšo in učinkovitejšo od tradicionalnih »napovedovalnih« metodologij. Glavna razlika med omenjeno klasifikacijo metodologij je teža same metodologije in zahtevnost projekta. Za »prilagojevalne« metode je značilno, da je podrobneje planirana le naslednja interakcija, medtem ko je pri »napovedovalnih« metodah značilno načrtovanje celotnega izdelka ali storitve. Posledično je takšna metoda s finančnega in časovnega vidika veliko zahtevnejša od prej omenjene »prilagojevalne«.

5 Metodologija Scrum

5.1 Uvod

Metodologija Scrum je agilna metoda, ki je na področju projektnega vodenja razvoja programske opreme najbolj priljubljena in trenutno največkrat uporabljena metodologija. Metodologijo je leta 1993 poimenoval strokovnjak agilnega vodenja, J. Sutherland. Uradna pravila te metode so se pojavila dve leti kasneje, izpod peresa K. Schwaberja in omenjenega J. Sutherlanda.

Slika 5.1: Scrum proces



Celoten proces in njegovo logiko lahko vidimo na sliki 5.1. Glavna značilnost oziroma temelj metodologije Scrum je iteracija, imenovana »Sprint«. Iteracija pomeni časovni okvir, v katerem se bo opravila določena količina dela oziroma število delovnih nalog, ki so predvidene pri določenem projektu. Iteracije so navadno razdeljene v serijah 30 dni, vendar se

vse pogosteje pojavljajo tudi 14-dnevni »Sprinti«, ki nudijo možnost še bolj dinamičnega razvoja. Tako je »Sprint« glavni element same metodologije, znotraj katerega se odvija razvoj.

V »Product Backlogu« je spisek zgodb, ki predstavljajo opis funkcionalnosti izdelka, skupek teh pa predstavlja celoto produkta ali storitve, ki jo pripravi naročnik in zaradi katere se izvaja celoten proces razvoja. Zgodbe oziroma zahtevki nalog, ki so v »Product Backlogu«, se po prioritetenem vrstnem redu prestavijo v »Sprint Backlog«, od koder se določene zahteve zapišejo v »Sprint« na podlagi treh dejavnikov: prioritete naloge, težnje vrednosti naloge in težavnosti posamezne naloge, v smislu časovnega obdobja, ki je določen za uspešno izvedbo te naloge. Zahteve, ki izpadejo iz omenjenih kategorizacij, ostanejo zapisane v »Sprint Backlogu« in se ponovno razvrstijo po prioritetenem vrstnem redu, kjer počakajo na naslednji cikel Sprinta.

Metodologijo Scrum sestavlja tudi »Daily Scrum Meeting«, Scrum sestanek, ki se odvija na dnevni ravni in je nekakšen povzetek opisa dela, morebitnih težav in ugotovitev prejšnjega dne ter načrtovanja aktualnega dne. Sestanek služi tudi kot posvet med razvijalci in skrbnikom metodologije (»Scrum Mastrom«) in je ključnega pomena pri agilnosti izvedbe projekta. Rezultat uspešno zaključenega projekta je seveda nov izdelek, program ali storitev.

5.2 Scrum ekipa

Ekipa je najpomembnejši del celotnega mozaika metodologije Scrum. Temeljiti mora na zaupanju, transparentnosti, sodelovanju in medsebojnem komuniciranju. Uspešnost projektov je v največji meri odvisna od zgoraj naštetih dejavnikov in pomanjkanje katere od teh skoraj zagotovo vodi v neuspeh. »Ekipni model metodologije Scrum je zasnovan tako, da optimizira fleksibilnost, kreativnost in produktivnost« (ScrumGuides.org 2015).

5.2.1 Skrbnik izdelka

Skrbnik izdelka je oseba, odgovorna za začetno snovanje ideje in postavitev ciljev nekega projekta. Uspešno doseganje ciljev je v prvi vrsti vodilo skrbnika izdelka, saj se ob nedoseganju oziroma propadanju katerega od teh odgovornost pripiše prav njemu. Če

pogledamo skozi metodologijo Scrum, se njegove zahteve oziroma uporabniške zgodbe pojavijo v »Product Backlogu«, tj. seznamu zahtev, kjer se jim določi prioritete, nato pa se pomikajo v sam Sprint, kjer se izvedejo. Upravljanje seznama zahtev vključuje:

- jasno opredelitev nalog in zahtev;
- transparentnost med skrbnikom in ostalimi udeleženci;
- zagotavljanje natančnega opisa nalog in zahtev in s tem popolno razumevanje seznama zahtev s strani razvojne ekipe;
- razvrstitev nalog in zahtev po vrstnem redu, ki nam kar najbolj pomaga pri realizaciji postavljenega cilja;
- za uspeh skrbnika izdelka je nadvse pomembno, da tako organizacija kot tudi njeni udeleženci spoštujejo njegove odločitve in cilje.

5.2.2 Razvojna ekipa

Razvojna ekipa je odgovorna za sam razvoj nove funkcionalnosti ali storitve. Njena naloga je implementacija zahtev oziroma nalog v končni izdelek ali v novo delujočo verzijo nekega produkta. Optimalno je, da je ekipa dovolj majhna, da se je sposobna obnašati agilno in da je dovolj velika, da je sposobna opraviti tudi večjo količino dela. V praksi se optimalnost v razvoju kaže med 4 in 8 osebami. Pri večjem številu se največkrat problem pojavi že pri koordinaciji in kompleksnosti opravljanja empiričnega procesa, medtem ko se pri manjših ekipah kaže omejenost v razponu znanja.

Sinergija razvojne ekipe in porazdelitev znanja znotraj nje je tako ključ uspešnosti razvoja. Seveda je velikost ekipe zelo odvisna od specifikacije dela, ki ga opravlja, in od organizacije, v kateri deluje. Prav tako kot znanje naj bi se delile tudi odgovornost in zadolžitve. V metodologiji Scrum namreč velja pravilo o enakopravnosti vseh članov razvojne skupine in s tem porazdelitev odgovornosti. Prav tako metodologija prepoveduje povečevanje ali zmanjševanje razvojne skupine, bodisi po koncu razvojnega cikla bodisi med samim ciklom.

5.2.3 Scrum master

Scrum master ali skrbnik metodologije Scrum je vmesni člen med skrbnikom izdelka in razvojno skupino. Njegova naloga je predvsem delovanje v razvojni skupini, odstranjevanje ovir med zahtevki skrbnika izdelka in razvojno ekipo in koordinacija znotraj te. Njegova odgovornost je poučevati vse udeležence o omenjeni metodologiji in skrbeti za njeno dosledno upoštevanje. Skrbnik metodologije opravlja naloge, podobne projektne vodji, glavna razlika je v tem, da je zadolžen predvsem za usklajevanje same razvojne ekipe in interakcije med razvojem in lastnikom. Prav tako ima nalogo organizacije in vodenja Scrum sestankov in dogodkov, kot to zahtevajo metodologija, razumevanje dolgoročnega planiranja v določenem empiričnem okolju in predstavitev ter pomoč pri razumevanju metodologije s strani razvojne ekipe. Tako je za celotno razumevanje metodologije in pravilno ravnanje tako udeležencev kot celotne organizacije odgovoren skrbnik metodologije, ki na podlagi dobrega razumevanja določenega cilja in namena lastnika izdelka koordinira, usklajuje in vodi metodologijo znotraj razvojne skupine skozi celoten proces razvoja. Pri pregledu literature se pojavi fraza, da je skrbnik metodologije v službi razvojne skupine.

5.3 Dogodki

Časovno obdobje procesa ali cikel (ang. *Sprint*) v metodologiji predstavlja določeno dokončano iteracijo, katere rezultat je delujoča verzija izdelka ali storitev. Za uspešno končan cikel so potrebni uspešno načrtovani in izpeljani vmesni koraki. Metoda Scrum predpisuje naslednje sestavne dele cikla:

- *Sprint Planning Meeting* ali načrtovalni sestanek. Na njem se zberejo vsi člani razvojne skupine, skupaj s produktnim vodjem, ki predstavi zalogo uporabniških zgodb in zahtev, predvidenih pri celotnem projektu. Nato skupaj s člani celotne ekipe izberejo podmnožico uporabniških zgodb oziroma zahtev, ki se jih bo realiziralo v slednji iteraciji. Sestanek se tako deli na dva dela: v prvem delu se ekipa sprašuje »Kaj?«, pri čemer lastnik izdelka predstavi prioritete zahteve, ki so zapisane v »Product Backlogu«, v drugem delu sestanka pa se ekipa sprašuje »Kako?«, pri čemer razvojna skupina na podlagi načrtovanja oceni zahtevnost realizacije posamezne uporabniške zgodbe, zahteve oziroma naloge in sestavi seznam nalog, ki jim bo kos v

določeni dobi iteracije. Načrtovanje oziroma ocenitev zahtevnosti se v praksi pojavi kot trd oreh, zato obstajajo različne tehnike spoprijemanja z njimi, ki pa jih bomo opisali nekoliko kasneje.

- *Daily Scrum Meeting* ali dnevni Scrum sestanek. Na njem se dnevno sestane razvojna ekipa skupaj s skrbnikom metodologije, čigar prisotnost pa ni nujna. Sestanek poteka vsak dan ob isti uri približno enako časovno obdobje (15 min). Na sestanku posamezni člani odgovarjajo na tri vprašanja: »Kaj sem naredil od prejšnjega Scrum sestanka?«, »Kaj bom naredil do naslednjega?«, »S katerimi težavami sem se srečeval in kako sem jih reševal?«. Na podlagi odgovorov na omenjena vprašanja se v razvojni ekipi zagotovi transparentnost, možnost usklajevanja in skupnega reševanja določenih problemov.
- *Sprint Review Meeting* ali sestanek za pregled iteracij se odvije po koncu posamezne iteracije, tj. ko v razvoju pride do delujoče verzije izdelka ali storitve. Sestanek je namenjen pregledu do sedaj narejenega dela, seznanitvi s težavami, ki so se pri razvoju pojavile, in predlogom za izogibanje teh v prihodnje. Na sestanku je prisotna razvojna ekipa skupaj s skrbnikom metodologije in lastnikom izdelka. Demonstracija funkcionalnosti nove verzije izdelka ali storitve, pregled težav, ki so se pojavile, in predvidevanje teh v prihodnje so osrednje teme dogodka kot takega.
- *Sprint Retrospective Meeting* ali sestanek za oceno kakovosti razvojnega procesa je sestanek, ki se zgodi po sestanku za pregled iteracij. V glavnem je sestanek namenjen načrtovanju izboljševanja razvojnega procesa in vsega, kar spada k temu (komunikacija, koordinacija, transparentnost, uporaba orodij in tehnologij, razvojnočasovne ovire ipd.).

5.4 Scrum planiranje

Planiranje oziroma načrtovanje je ključnega pomena pri realizaciji projekta. Dobra napoved izpolnitve zahtev omogoča dobre pogoje v interakciji z lastnikom izdelka in nam s tem znatno poveča pogajalske zahteve pri dogovarjanju za izvedbo nekega projekta. Upoštevanje predvidenih rokov določene organizacije pri razvoju novega izdelka velja na trgu za eno od močnejših vrlin in je tudi za naročnike pomembnejši kriterij, tudi od finančnega. Večini

lastnikov izdelkov je namreč pomembneje dobiti novo verzijo, izdelek ali storitev na dogovorjeni rok, pa čeprav je posledična cena tega produkta dvakrat višja. Višina resursov, ki je porabljen, je seveda odvisna od okolja, v katerem organizacija deluje, od samega izdelka in od prvotnega namena izvedbe projekta.

Največkrat se problem pojavi med selekcioniranjem poslovnih in razvojnih ciljev. Lastniki izdelkov, naročniki in ostali poslovni interesi vidijo prioritete predvsem v finančnem smislu in izkupičku po uspešno končanem projektu, medtem ko je z vidika razvojne ekipe zgodba drugačna. Razvojni sektor namreč vidi seznam zahtev kot množico nalog, ki se večrazredno delijo na podnaloge, medtem ko je lastnik izdelka največkrat usmerjen k eni uporabniški zgodbi, ki se šele v nadaljevanju razbije na podnaloge. Teh je v praksi mnogo manj kot pri razvojni ekipi. Natančna definicija uporabniških zgodb, zahtev in nalog je tako ključnega pomena pri sestavi načrta planiranja in organizacije projekta.

5.4.1 Ocenjevanje uporabniških zgodb

Uporabniška zgodba pomeni skupek nalog in zahtev, ki ob uspešni izvedbi predstavlja novo verzijo izdelka, produkta ali storitve. Opisuje novo funkcionalnost in ima za lastnika izdelka neko uporabno vrednost. Celoten projekt je sestavljen iz določenega števila uporabniških zgodb, vsaka zgodba pa se naprej deli na različne naloge. Velikost oziroma obsežnost uporabniških zgodb ocenjujemo s številom točk (Story Points), pri čemer večje število pomeni večjo zahtevnost uporabniške zgodbe. Tukaj ni pomembna absolutna vrednost, ki jo ima določena zgodba, pač pa njena relativna vrednost. Se pravi, če je Z1 (zgodba 1) vredna 5 točk, Z2 (zgodba 2) pa 15, pomeni, da je za uspešno izvedbo Z2 potrebno trikrat več vloženega dela kot za uspešno izvedbo Z1. Za dodeljevanje točk ni pravila, razvojna ekipa največkrat oceni po občutku in po izkušnjah preteklih ciklov. V praksi se za najbolj primerno ocenjevanje pokaže ocenjevanje po Fibonaccijevi lestvici: 1, 2, 3, 5, 8, 13, 21 ..., saj se lahko po tej lestvici pregledno orientiramo po relativnih vrednostih. Če pride do razvoja popolnoma nove tehnologije, pri kateri se uporabljajo zgodbe, katerih naloge in zahteve so razvojni skupini neznane, se zgodba, za katero ekipa meni, da potrebuje najmanj vloženega dela, oceni z 1. Nato se ostale zgodbe primerja s to in povečuje njihovo vrednost glede na večkratnik vloženega dela.

Na podlagi ocenitve vseh uporabniških zgodb lahko izračunamo hitrost razvoja in s tem podamo predvideni čas izvedbe projekta. Hitrost je torej število vseh uporabniških zgodb, ki jih je razvojna skupina sposobna dokončati v posamezni iteraciji. Če je tako v iteraciji uspešno zaključenih 5 uporabniških zgodb z oceno 10, je hitrost razvoja v tej iteraciji 50, kar pomeni, da lahko v naslednji iteraciji pričakujemo in planiramo vsaj tolikšno vrednost, če ne celo nekoliko večje. Ker so vrednosti ocen približne in se jih ne da popolnoma definirati, je lahko rezultat enačbe tudi napačen, vendar se z različnimi tehnikami poskušamo kar najbolj približati dejanskim napovedim hitrosti razvoja. Obstaja več načinov, kako se sistemsko lotiti planiranja oziroma načrtovanja, najpogostejše in najbolj priljubljeno pa je poker planiranje (*Poker Planning*), ki ga bom opisal v nadaljevanju.

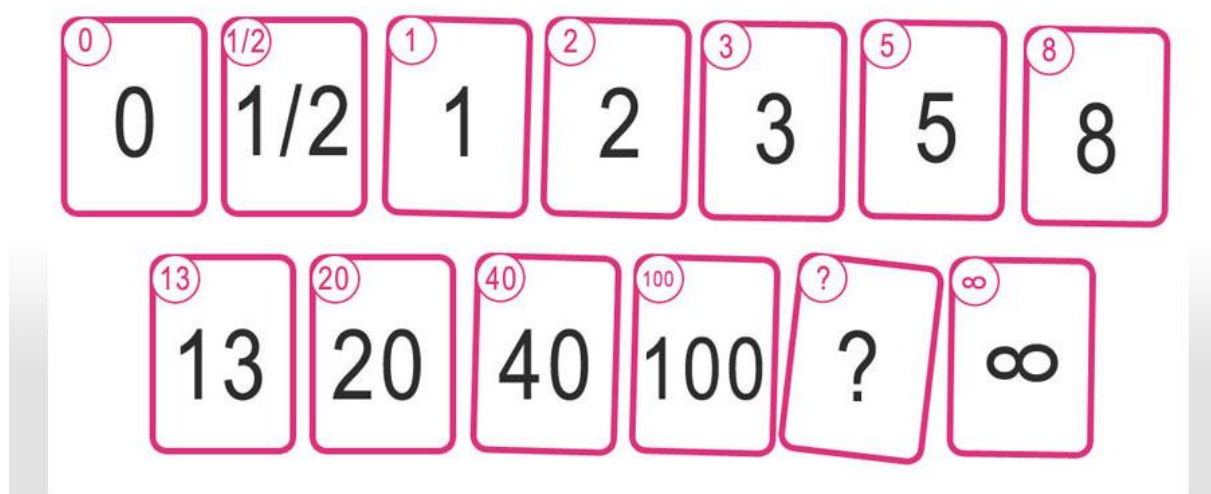
5.4.2 Poker planiranje

Poker planiranje je tehnika, ki služi za določanje ocen določenim uporabniškim zgodbam in nam posledično lahko dokaj natančno napove časovno obdobje procesa nekega projekta. Tehnika velja za eno najučinkovitejših in najbolj natančnih pri omenjeni metodologiji Scrum. Poleg omenjene funkcionalnosti o napovedovanju časovnega obdobja nam služi tudi kot dobro orodje za transparentnost med udeleženci projekta, dodeljevanje nalog različnim razvijalcem, sodelovanje interdisciplinarne ekipe in enotno razumevanje določenih zgodb za vse udeležence.

Pri poker planiranju sodelujejo vsi udeleženci projekta, tako razvijalci in skrbnik metodologije kot tudi lastnik izdelka. Vsak član razvojne ekipe prejme trinajst kart, na katerih so napisane številke od 0 do 100 (*Story Points*); najbolje da so razvrščene po Fibonaccievi lestvici, saj s tem najbolje prikažemo relativno sorazmerje med uporabniškimi zgodbami. Nato s seznama zgodb, ki ga pripravi skrbnik metodologije, obravnavamo vsako zgodbo posebej, in sicer tako, da jo oceni vsak član razvojne ekipe. Pri tem uporabi kartico s številko, za katero meni, da najbolje odraža zahtevnost in zalogo potrebnega vloženega dela za uspešno izvedbo. Vsi člani imajo najprej kratek čas za premislek, nato pa sočasno razkrijejo svojo karto, saj mora odločitev temeljiti na osebni odločitvi in ne vplivu drugih. Ob predpostavki, da se dve oceni med seboj zelo razlikujeta, sta razvijalca, ki sta oceni podala, dolžna svojo odločitev argumentirati. Tako se preostali del ekipe seznanijo z razmišljanjem obeh in pripomore h končni odločitvi. Cilj je kar najbolj poenoten rezultat, vendar pa ni nujna

absolutna točnost. Ko se vsi člani razvojne ekipe strinjajo s povprečno oceno uporabniške zgodbe, ki jo predlaga skrbnik metodologije, dobi ta uporabniška zgodba svojo vrednost zahtevnosti, s tem pa lahko napovemo časovno obdobje realizacije. Tako poker planiranje nadaljujemo, dokler ne ocenimo vseh uporabniških zgodb, seštevek povprečne vrednosti vsake od teh pa nam poda približno oceno dolžine obdobja celotnega procesa izvedbe projekta. Ta ocena se ne odraža v urah in minutah, pač pa je podana v merski enoti Story Points. Enačba za pretvorbo v časovni okvir je odvisna od velikosti ekipe razvijalcev, števila delovnih ur, ki jih dnevno opravijo, in vrednosti ocenitve posamezne uporabniške zgodbe. Ta je odvisna od okolja organizacije in storitve, ki jo organizacija razvija. Če šteje razvojna ekipa pet članov, ki dnevno opravijo 8 učinkovitih delovnih ur, je smiselno nastaviti 1 Story Point na en delovni dan, se pravi 8 ur učinkovitega dela. Nastavitev je odvisna od zahtevnosti najlažjega zahtevka, ki ga imamo v posamezni uporabniški zgodbi. Če menimo, da lahko določeno nalogo opravimo v polovici delovnega dne, potem lahko zmanjšamo ocenitev na $\frac{1}{2}$ točke ali pa povečamo vrednost delovnega dne na 2 točki. Za določanje teh podatkov je v osnovi odgovoren skrbnik metodologije.

Slika 5.2: Poker planiranje – kartice



Na sliki vidimo kartice z različnimi vrednostmi, po katerih razvijalci ocenjujejo posamezne uporabniške zgodbe. Prikazane so tudi kartice z vrednostjo 0, ? in znakom za neskončno. Z

vrednostjo 0 tako označimo uporabniške zgodbe, za katere menimo, da so skoraj že dokončane ali pa je treba v njih vložiti minimalno truda. Kartica z znakom »?»« pomeni neopredeljenost pri ocenjevanju. Kartica pride v poštev, ko se nekdo iz razvojne ekipe znajde v slepi ulici in se ne more odločiti, kako bi ocenil določeno zgodbo. V takšnem primeru se največkrat nasloni na odločitev preostalih iz razvojne ekipe. Znak za neskončno stoji za uporabniško zgodbo, za katero razvijalec meni, da jo je nemogoče dokončati. V takšnem primeru sledijo poglobljanja in reševanje problema s strani lastnika izdelka.

5.5 Opis testnega primera

Za lažjo predstavitev in boljše razumevanje metodologije bomo v nadaljevanju opisali testni primer uporabe metodologije Scrum. Primer je zasnovan na osnovni razvijalski ravni in nima uporabne funkcionalnosti, saj je kar najbolj prilagojen enostavnosti razumevanja in demonstracije.

Tako smo za testni primer izbrali razvoj spletne strani nekega novega izdelka, s katerim želimo prodreti na trg. Namen in funkcionalnost izdelka niti nista pomembna, zato ju ne definiramo. Pomembna pa je sestava spletne strani, ki nam bo omogočila uspešno predstavitev in prodajo izdelka. Namen in cilj sta torej jasna, treba je le kar najbolje realizirati razvoj.

Lastnik izdelka poda seznam uporabniških zgodb, ki jim daje prednost glede na pomembnost implementacije v samem razvoju.

Na *Sprint Planning Meetingu* lastnik izdelka poda seznam vseh uporabniških zgodb in predstavi namen in cilj projekta celotni ekipi. Pri projektu bo sodelovalo 5 ljudi, in sicer štirje razvijalci in skrbnik metodologije.

Seznam uporabniških zgodb:

- Z1: Uporabnik ima na vstopni strani možnost samodejnega pregleda galerije izdelka, prav tako lahko po fotografijah brska ročno s pomočjo kurzorja.
- Z2-: Na strani je podroben opis izdelka, njegova zgradba, možnost uporabe in vse splošne informacije o njem.

- Z3: Stran vsebuje segment »How to use it«, kjer je natančno opisan postopek uporabe izdelka, skupaj s slikovnim materialom.
- Z4: Uporabnik ima možnost ogleda videopredstavitve uporabe izdelka in možnost delitve istega videa na socialna omrežja.
- Z5: Uporabnik ima možnost pošiljanja povpraševanja za izdelek prek vpisne e-poštne oblike, kamor lahko vpiše vsa vprašanja in podatke za morebitno zanimanje za nakup izdelka.

Prioritete: pri čemer imajo uporabniške zgodbe z oceno 1 najvišjo prioriteto in tiste s 5 najnižjo.

Z1 = 5

Z2 = 2

Z3 = 3

Z4 = 2

Z5 = 1

Po predstavitvi cilja projekta in njihovih uporabniških zgodb na sestanku dobimo približno oceno obsežnosti projekta in napovemo približno obdobje trajanja projekta. Prej omenjeni štirje razvijalci bodo delali po 8 ur dnevno, 1 delovni dan posameznega razvijalca pa bo vreden 1 točko (*Story Point*). Tako je količina dela, ki ga celotna ekipa lahko opravi v enem delovnem dnevu, 4 točke (*Story Points*). Odločili smo, da se bomo zaradi večje dinamike in agilnosti posluževali tedenskih ciklov, se pravi da bo vsak novi Sprint trajal natanko en teden, tj. 5 delovnih dni, kar pomeni, da je hitrost ene iteracije v tej razvojni ekipi 20. Do te številke pridemo, če pomnožimo dnevno hitrost ekipe (4) s številom delovnih dni, ki jih vsebuje en cikel (5). Pri projektu bomo uporabili že prej omenjeno tehniko poker planiranja, pri čemer bomo dobili ocene zahtevnosti vsake od uporabniških zgodb.

Uporabniške zgodbe	Točke zahtevnosti	Prioriteta
Z1	10	5
Z2	5	2
Z3	21	3
Z4	8	2
Z5	13	1

Tabela 5.5.1 Ocenjevanje zgodb s strani razvojne ekipe

Na sestanku je razvojna ekipa podala predlog, da se zaradi obsežnosti zgodbe Z3 razdeli na dve zgodbi, in sicer na opis navodil za uporabo in galerijo slikovnega materiala. Prva ima prioriteto 2, druga pa 3. S tem ukrepom se prav tako porazdeli njihova zahtevnost. Tabela je po sestanku in sprejetju sprememb s strani lastnika izdelka videti takole:

Uporabniške zgodbe	Točke zahtevnosti	Prioriteta
Z1	10	5
Z2	5	2
Z3a	8	2
Z3b	13	3
Z4	8	2
Z5	13	1

Tabela 5.5.2: Tabela po natančni interpretaciji uporabniških zgodb

Po skupnem pregledu tabele se uporabniške zgodbe uvrstijo v Sprinte, se pravi v iteracije, kjer bo prišlo do njihove realizacije.

CIKEL	PREDVIDENA HITROST	ZGODBE	POTREBNA HITROST
Sprint 1	20	Z5, Z3a	21
Sprint 2	15	Z4, Z2,	13
Sprint 3	21	Z3b, Z1	23

Tabela 5.5.3: Začetni načrt izvajanja ciklov

Na začetku nastavimo predvideno hitrost, kot smo jo v začetku izračunali (20) na podlagi zmožnosti razvojne ekipe. V Sprinte poskušamo vpeljevati uporabniške zgodbe glede na prioriteto, ki jo imajo za implementacijo.

Tako začnemo z zgodbo z najvišjo prioriteto, tj. Z5, ki meri 13 točk. Poleg nje se odločamo med tremi zgodbami s prioriteto 2. Zaradi začetne motivacije in možnosti izkazovanja razvojne ekipe tako načrtujemo nekoliko bolj optimistično. Zato se odločimo za zgodbo Z3a, ki meri 8 točk. Tako je potrebna hitrost za dokončanje prvega cikla 21, kar je za 1 točko več od naše zmožnosti razvojne ekipe, a to na začetku ne pomeni rdečega alarma, saj bomo temu primerno reagirali v ciklu 2.

Zaradi nekoliko večjega zalogaja dela v ciklu 1 smo predvidevali in napovedali, da nam lahko ostane še kakšna naloga tega cikla tudi v naslednjem ciklu, se pravi ciklu 2. Zato smo tukaj napoved nekoliko zmanjšali in se s tem prilagodili morebitnim zapletom nalog v ciklu 1.

Uporabniške zgodbe z najnižjo prioriteto pridejo na plano v zadnjem ciklu, pri katerem načrtujemo, da bomo obravnavali zgodbi Z3b in Z1. Prav tako bo tukaj potrebna nekoliko večja hitrost, vendar lahko ob predpostavki, da cikel 2 zaključimo pravočasno in pri tem opravimo vse potrebne naloge, cikel 3 razdelimo in obravnavamo zgodbi Z3b in Z1 posebej, se pravi vsako v svojem Sprintu. V tem primeru bi dodali Sprint 4, kjer bi realizirali posamezno uporabniško zgodbo, v tem primeru samo Z1.

CIKEL	PREDVIDENA HITROST	ZGODBE	POTREBNA HITROST
Sprint 1	20	Z5, Z3a	21
Sprint 2	15	Z4, Z2,	13
Sprint 3	17	Z3b	13
Sprint 4	15	Z1	10

Tabela 5.1.4: Dejanski načrt izvajanja ciklov

V zadnjem ciklu svojih napovedi nismo nastavili pretirano optimistično, saj se je v ciklu 1 izkazalo, da so uporabniške zgodbe nekoliko bolj kompleksne, kot smo sprva načrtovali. Zato smo v nadaljnje temu primerno reagirali in si v naslednjih ciklih zmanjšali predvideno hitrost opravljenega dela. Čeprav je primer povsem enostaven, lahko na njem lepo vidimo možnost prilagoditve razvoja in agilnega reagiranja, tako na opisano delitev dela znotraj razvojne skupine kot tudi reagiranje na nenadne spremembe direkcije s strani lastnika izdelka, ki so posledica sodobnega, hitro spreminjajočega se trga.

6 Zaključek

Agilne metode vsekakor predstavljajo nove smernice in ponujajo nove priložnosti na področju razvoja programske opreme. Tako se od klasičnih razlikujejo predvsem v tem, da je med izvedbo agilne metodologije med razvojem mogoče dodati, spremeniti ali celo odstraniti posamezen zahtevek ali pa kar celoten cikel. Takšna filozofija je složna z današnjim sodobnim trgom programske opreme, na katerem je sprememba trendov skorajda vsakdanje pravilo. Našteti dejavniki so glavni razlog za klic organizacij k agilnim metodologijam. Metodologija Scrum velja trenutno za najbolj priljubljeno, saj njena intuitivnost, prilagodljivost in enostavnost največkrat zadovoljuje različne tipe organizacij, ki delujejo v različnih okoljih. Edini problem, ki se pojavlja pri sprejemanju in odločitvi organizacije za uporabljeno metodo, je preklon med klasičnimi in agilnimi metodami. Preklon v organizaciji zahteva manjšo reorganizacijo in skoraj popolno spremembo filozofije razvojnega mišljenja, zato se nekoliko konservativnejša vodstva za te ukrepe še vedno ne odločajo. Dejstvo pa je, da je pri implementaciji novega izdelka na trg pridobivanje finančnih resursov za izvedbo projekta bistveno manjše če ne poznamo hitrostnega koeficienta razvojne ekipe, tj. število delovnih dni, potrebnih za uspešno izvedbo celotnega projekta. Prav tako gre v teh vodah največkrat za projekte, ki predstavljajo negotovost ob izidu in je njihov uspeh zaradi hitro spreminjajočega se trga težko napovedati. Zato se že na samem začetku izvedbe srečujemo s tveganim kapitalom, ki pa se ob predpostavki neznane hitrosti razvoja razvojne ekipe in ob posledičnem kršenju časovnih okvirjev izvedbe projekta le še zviša. Izbira metodologije je v veliki meri odvisna od smeri razvoja in trga, na katerem organizacija deluje, zato univerzalno idealna metodologija za zdaj še ne obstaja. Vendar pa se je v prihodnosti lahko nadejamo, če upoštevamo hitrost razvoja programske opreme, kot jo imamo danes.

7 Literatura:

1. Bentley, Colin. 1997. *Practical Handbook*. Ljubljana: BEGEN založba d.o.o.
2. Burke, Rory. 1999. *Project Managementn*, Wiley, 3. izdaja. Ljubljana: BEGEN založba d.o.o.
3. Burke, Rory. 2003. *Project management: planning and control techniques*. Chichester: John Wiley & Sons.
4. Charvat, Jason. 2002. *Project management nation: tools, techniques and goals for the new and practicing IT project manager*. New York: John Wiley & Sons.
5. Golob, Renato. 2002. *Direktor se odloči: uvajanje projektnega managementa v praksi*. Ljubljana: GV založba.
6. Kerzner, Harold. 2001. *Strategic planning for project management: using a project management maturity model*. New York: John Wiley & Sons.
7. Kovač, Jure. 1995. *Uresnicevanje strategije podjetja po projektnem načinu*, doktorska disertacija. Maribor: Ekonomsko poslovna fakulteta.
8. Lock, Dennis. 2003. *Project management*. (8th ed.) Aldershot: Gower.
9. Možina, Stane. 1994. *Management*. Radovljica: Didakta.
10. Rozman, Rudi. 2008. Namen in cilji projekta. *Projektna mreža Slovenije* 11 (2): 4–9.
11. Rozman, Rudi, Stare, Aljaž. 2008. *Projektni management ali ravnateljjevanje projekta*. Ljubljana: Ekonomska fakulteta.
12. Schwaber, Ken in Sutherland Jeff. 2013. *Scrum Vodič*. Dostopno prek: <http://www.scrumguides.org/scrum-guide.html> (1.september 2015)
13. Solina, Franc. 1997. *Projektno vodenje razvoja programske opreme*. Ljubljana: FE in FRI.
14. Stare, Aljaž. 2008. *Obvladovanje sprememb v izvedbi projekta*, Doktorska disertacija. Ljubljana: Ekonomska fakulteta.