

■ Ocena prednosti metode Scrum in njenih tipičnih praks

Janez Urevc, Viljan Mahnič

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Tržaška cesta 25, 1000 Ljubljana

janez@janezurevc.name; viljan.mahnic@fri.uni-lj.si

Izvleček

Čeprav je Scrum v praksi najbolj razširjena agilna metoda za razvoj programske opreme, je le malo empiričnih študij o njeni uporabi, prednostih in pomanjkljivostih. Večina poročil temelji na opisu izkušenj, ki niso podprte z empiričnimi podatki. Da bi empirično ovrednotili mnenja uporabnikov o koristih, ki jih prinaša Scrum, in poiskali tiste njegove prakse, ki največ prispevajo k uspehu projektov, so avtorji izvedli anketo med 75 uporabniki te metode v Sloveniji in tujini. Rezultati ankete so potrdili, da se uporabniki v celoti strinjajo s trditvami o prednostih Scruma, ki jih navaja literatura. Med dejavniki, ki najbolj vplivajo na uspeh po Scrumu vodenih projektov, so posebno izpostavili medsebojno komunikacijo znotraj razvojne skupine in komunikacijo med razvijalci in produktnim vodjo. Rezultati lahko služijo podjetjem, ki nameravajo uvesti Scrum v svoj razvojni proces, pri analizi pričakovanih koristi in identifikaciji najpomembnejših praks, ki jih morajo upoštevati, da bi pri tem uspela.

Ključne besede: Scrum, agilne metode, razvoj programske opreme, vodenje projektov.

Abstract

Evaluation of Scrum Benefits and its Typical Practices

In spite of the fact that Scrum is the most widespread agile software development method in industry, empirical evidence about its use, advantages and disadvantages remains scarce. Most reports are based on anecdotal evidence that is not supported by empirical data. In order to empirically evaluate users' opinions of benefits of Scrum and identify those practices that contribute most to the success of a Scrum project, a survey was conducted among 75 Scrum users in Slovenia and abroad. The survey has confirmed that the users fully agree with all advantages of Scrum reported in the literature. Among factors with the greatest impact on the success of Scrum projects, they highlighted communication within the development team and communication between developers and the Product Owner. Results of the survey can serve companies that are planning to introduce Scrum into their development process in analyzing expected benefits of Scrum implementation and identifying the most important practices that need to be considered in order to succeed.

Keywords: Scrum, agile methods, software development, project management.

1 UVOD

Agilne metode za razvoj programske opreme (Williams, 2010) so se začele pojavljati v devetdesetih letih prejšnjega stoletja kot reakcija na tradicionalni disciplinirani pristop, ki zahteva natančno vnaprejšnje načrtovanje in predpisuje točno določen postopek razvoja. V nasprotju s tradicionalnimi pristopi daje *Manifest agilnega razvoja programske opreme*¹ prednost posameznikom in interakciji med njimi pred procesi in orodji, delujoči programski opremi pred obsežno dokumentacijo, sodelovanju z naročnikom pred pogodbenimi pogajanjmi in odzivanju na spremembe pred togim sledenjem načrtu. S tem zmanjšuje pomen nekaterih vrednot, ki so bolj značilne za tradicionalne metode.

Kljub temu da med razvijalci programske opreme in informacijskih sistemov ni enotnega mnenja o primernosti agilnih

metod, vse več podatkov kaže na njihovo uspešnost in naraščajočo popularnost. Tako Ambler (2008) navaja, da uporaba agilnih metod bistveno poveča produktivnost, kakovost in zadovoljstvo vseh udeležencev v razvojnem procesu. Forrester v svojem poročilu (West & Grant, 2010) ugotavlja, da je leta 2009 uporabljalo agilne metode že 35 odstotkov projektov, po Gartnerjevi napovedi (Murphy idr., 2009) pa naj bi leta 2012 kar 80 odstotkov projektov potekalo na agilnem način. Čeprav prevladuje mnenje, da so agilne metode primerne predvsem za manjše projekte (Boehm & Turner, 2004), lahko v literaturi zasledimo poročila o uvajanju teh metod tudi v največjih podjetjih s področja informacijske tehnologije, kot so npr. Microsoft (Begel & Nagappan, 2007), Yahoo (Scotland & Boutin, 2008), Nokia (Laanti idr., 2011) ipd.

Scrum (Schwaber, 2004) je med vsemi agilnimi metodami najbolj razširjen, saj njegov delež po zadnjih podatkih za leto 2011 znaša 52 odstotkov, v kombinaciji z ekstremnim programiranjem pa 66 odstotkov (Version One, 2011). Tudi v Sloveniji se krog uporabnikov te metode vedno bolj širi; skupina Skram.si v omrežju LinkedIn šteje preko tristo članov. Vseeno pa je le malo empiričnih raziskav, ki bi s kvantitativnimi podatki ovrednotile njegove prednosti in tipične prakse. Pregled empiričnih študij o uporabi agilnih metod, ki sta ga izdelala Dybå & Dingsøyr (2008), navaja eno samo študijo, ki obravnava metodo Scrum. Več je bilo narejenega na področju izobraževanja, v okviru katerega tudi v Sloveniji že več let poučujemo to metodo v obliki praktičnega dela na študentskih projektih (Mahnič, 2010; Mahnič, 2012), na podlagi izkušenj pri delu s študenti pa so bila predstavljena tudi priporočila za njeno uvajanje v prakso (Mahnič, 2011).

Da bi natančneje ugotovili, kako uporabniki metode Scrum ocenjujejo njene prednosti in kakšen pomen pripisujejo posameznim tipičnim praksam, ki jih predpisuje Scrum, smo med uporabniki te metode konec leta 2011 izvedli anketo, ki sta jo (poleg uvodnega dela) sestavljala dva vsebinska sklopa vprašanj. Uvodni del je služil za segmentacijo anketirancev in je obsegal vprašanja o njihovi vlogi v projektih, ki so vodeni po metodi Scrum (Scrum Master/Product Owner/Team Member), in izkušnjah (koliko časa že uporabljajo Scrum in na koliko projektih Scrum so že sodelovali). Namen prvega vsebinskega sklopa vprašanj je bil preveriti, v kolikšni meri se anketiranci strinjajo s trditvami o prednostih metode Scrum, ki jih zasledimo v literaturi. V okviru drugega sklopa pa smo želeli ugotoviti, katere tipične prakse, ki jih predpisuje Scrum, po njihovem mnenju največ prispevajo k uspešnosti projektov.

Ideja za izvedbo ankete se je porodila v okviru projekta uvajanja metode Scrum v delo oddelka za spletni razvoj pri časopisnem podjetju Delo (Urevc idr., 2012), za njeno izpolnjevanje pa smo poleg sodelavcev tega oddelka zaprosili še člane slovenske skupine Skram.si, člane skupnosti razvijalcev sistema za upravljanje z vsebinami Drupal² (ker razvoj spletnega portala časopisa Slovenske novi-

ce temelji na uporabi tega sistema) in člane skupine Scrum Practitioners, ki tudi deluje v omrežju LinkedIn.

Dobili smo 75 odgovorov, od tega 47 (62,7 %) iz Slovenije in 28 (37,3 %) iz tujine. Med tistimi, ki so odgovorili, je bilo 33 (44,0 %) skrbnikov metodologije (angl. *Scrum Master*), 29 (38,7 %) razvijalcev članov razvojne skupine (angl. *Team Member*) in 13 (17,3 %) produktivnih vodij (angl. *Product Owner*).

Glede na to, da je vsaj v Sloveniji metoda Scrum razmeroma nova, je večina anketirancev (35 ali 46,7 %) imela manj kot šest mesecev izkušenj z njeno uporabo, vendar je bil tudi delež takih, ki metodo uporabljajo več kot 24 mesecev, razmeroma visok (20 anketirancev ali 26,7 %). Od 7 do 12 mesecev izkušenj je imelo 7 anketirancev (9,3 %), od 13 do 24 mesecev pa 11 (14,7 %). Dva anketiranca (2,7 %) še nista imela praktičnih izkušenj, ampak sta metodo poznala samo iz literature.

Večina anketirancev (48 ali 64,0 %) je doslej delala na enem ali dveh projektih, ki so potekali po metodi Scrum. Takih, ki so sodelovali pri 3–5 projektih je bilo 12 (16,0 %), izkušnje z več kot petimi projekti pa je imelo 13 (17,3 %) anketirancev.

Namen tega prispevka je skozi rezultate anketne izpostaviti glavne prednosti metode Scrum in izluščiti tiste tipične prakse, ki po mnenju njenih uporabnikov najbolj vplivajo na uspešnost projektov, vodenih po tej metodi. Pri tem primerjamo mnenja uporabnikov iz Slovenije z mnenji uporabnikov iz tujine ter analiziramo, kako so mnenja odvisna od njihovih izkušenj in vloge, ki jo imajo v metodi. S tem želimo pomagati tistim podjetjem, ki se odločajo za uvedbo metode Scrum, da bi bolje spoznala njene prednosti in se osredinila na tiste dejavnike, ki so po izkušnjah uporabnikov najpomembnejši. Upamo, da bo to pripomoglo k večji uspešnosti projektov in povečanju zadovoljstva ob uvajanju agilnih metod. Obenem pričakujemo, da bodo podatki koristni tudi za podjetja, ki že delujejo na podlagi Scruma, saj jim bodo omogočili optimizacijo procesov na podlagi izkušenj drugih.

Da bi bralec bolje razumel pomen posameznih anketnih vprašanj, bomo v nadaljevanju najprej na kratko predstavili metodo Scrum. Nato bomo podrobneje opisali odgovore na oba vsebinska sklopa vprašanj, v sklepu pa povzeli najpomembnejše rezultate. Ker uporablja Scrum specifično terminologijo, ki jo je v nekaterih primerih nemogoče dobesedno

¹ <http://agilemanifesto.org> (angleška verzija); <http://agilemanifesto.org/iso/sl> (slovenska verzija).

² <http://drupal.org>.

prevesti v slovenščino, navajamo v opisu metode poleg (po našem mnenju najprimernejšega) slovenskega prevoda tudi izvorni izraz v angleščini.

2 METODA SCRUM

Scrum izhaja iz premise, da je razvoj programske opreme preveč zapleten in nepredvidljiv proces, da bi ga lahko natančno načrtovali vnaprej. Namesto tega je treba vzpostaviti empiričen nadzor, ki omogoča sproten vpogled in prilagajanje trenutnemu stanju. To lahko dosežemo z iterativnim in inkrementalnim pristopom.

Vloge v metodi Scrum

Razvoj programske opreme po metodi Scrum predvideva tri vloge: produktni vodja (angl. *Product Owner*), razvojna skupina (angl. *Scrum Team*) in skrbnik metodologije (angl. *Scrum Master*).

Produktni vodja je predstavnik naročnika in za stopa vse, ki so zainteresirani za rezultate projekta. Njegova glavna naloga je upravljanje seznama zahtev (angl. *Product Backlog*), določanje njihove prioritete in grupiranje zahtev v posamezne izdaje (angl. *Release*). Seznam zahtev ni nikoli dokončen, ampak se lahko dopolnjuje ves čas trajanja projekta. Vsaka zahteva je praviloma opisana v obliki uporabniške zgodbe (angl. *User Story*), ki vsebuje besedilo zgodbe in seznam sprejemnih testov (Cohn, 2004). Ker gre za zelo ohlapen opis zahteve, mora biti produktni vodja stalno na voljo razvijalcem za pojasnila glede podrobnosti, povezanih z realizacijo. Po vsaki iteraciji pa mora preveriti, ali je zgodba realizirana ustrezno ali ne. Ta vloga je ključnega pomena za uspešnost projekta, zato mora imeti produktni vodja jasno vizijo o tem, kaj je treba narediti, in skladno s tem usmerjati razvojni proces.

Razvojna skupina je zadolžena za implementacijo zahtevane funkcionalnosti. Sestavljena mora biti tako, da pokriva vsa področja, ki so pomembna za izvedbo projekta. Skupina deluje po načelu samoorganizacije in sama določa, kako na najboljši način realizirati posamezne zahteve. Člani razvojne skupine so kolektivno odgovorni za uspeh posameznih iteracij in projekta kot celote.

Skrbnik metodologije ima na videz enako vlogo, kot jo ima pri tradicionalnem pristopu vodja projekta, vendar so njegove zadolžitve v resnici drugačne. Njegova naloga je skrbeti za nemoten potek dela, tako da vsi, ki delajo na projektu, upoštevajo pravila

metode Scrum in ustrezno izvajajo predpisane aktivnosti. Pri tem mora paziti, da se Scrum vklaplja v kulturo organizacije tako, da daje najboljše rezultate. Skrbnik metodologije v veliki meri deluje kot varuh, ki varuje razvojno skupino pred škodljivimi zunanjimi vplivi, odpravlja morebitne ovire in s tem zagotavlja optimalne pogoje za delo.

Potek razvojnega procesa po metodi Scrum

Na začetku projekta produktni vodja izdela seznam zahtev, ki vsebuje vse do takrat znane uporabniške zgodbe. Zgodbe razvrsti po prioriteti in jih razdeli v predvidene izdaje. Razvojna skupina oceni zahtevnost vsake zgodbe z ustreznim številom točk (angl. *Story Points*) in določi predvideno hitrost razvoja (angl. *Velocity*), tj. število točk, ki jih lahko realizira v eni iteraciji. V skladu s tem nato izdela načrt izdaje (angl. *Release Plan*), tako da v skladu s prioriteto razporedi zgodbe po posameznih iteracijah. Seštevek točk vseh zgodb v neki iteraciji ne sme preseči predvidene hitrosti razvoja.

Nadaljnji razvoj poteka v iteracijah (angl. *Sprint*), ki v originalni verziji metode Scrum trajajo 30 dni, danes pa največ uporabljajo iteracije, ki trajajo dva ali štiri tedne. Vsaka iteracija se začne s sestankom za načrtovanje iteracije (angl. *Sprint Planning Meeting*), na katerem se produktni vodja in razvojna skupina dogovorita, katere zgodbe bodo realizirane v naslednji iteraciji.

Na podlagi tega razvojna skupina izdela seznam nalog (angl. *Sprint Backlog*), ki vsebuje vse naloge, potrebne za realizacijo dogovorjene funkcionalnosti do konca iteracije. Med iteracijo ta seznam še dopolnjujejo, obsežnejše naloge pa razdelijo na več manjših, tako da vsaka zahteva 4 do 16 ur dela.

Člani razvojne skupine se vsak dan zberejo na petnajstminutnem sestanku (angl. *Daily Scrum Meeting*), na katerem vsak izmed njih odgovori na tri vprašanja: »Kaj si naredil od prejšnjega sestanka? Kaj boš delal do naslednjega sestanka? Ali imaš pri delu kakšne težave?« Namen tega sestanka je sinhronizirati delo vseh članov razvojne skupine in sproti identificirati morebitne probleme.

Na koncu vsake iteracije je predpisan sestanek za pregled rezultatov (angl. *Sprint Review Meeting*), na katerem razvojna skupina predstavi rezultate svojega dela produktnemu vodji in vsem zainteresiranim uporabnikom. Metoda striktno zahteva, da razvijalci spoštujejo koncept »Done«, kar pomeni, da mora biti

vsaka zgodba realizirana, testirana in dokumentirana v celoti, tako da jo je moč neposredno predati v produkcijo. Produktni vodja sme sprejeti samo tiste zgodbe, ki v celoti ustrezajo omenjenemu konceptu, in samo seštevek točk teh zgodb se lahko upošteva kot dejanska hitrost razvoja (angl. *Actual Velocity*) razvojne skupine.

Po tem sestanku (in pred začetkom naslednje iteracije) skrbnik metodologije organizira še sestanek za oceno kakovosti razvojnega procesa (angl. *Sprint Retrospective Meeting*), katerega namen je poiskati možnosti za izboljšave razvojnega procesa, tako da bi bil ta še bolj učinkovit v naslednjih iteracijah.

3 OCENA PREDNOSTI METODE SCRUM

V prvem sklopu anketnih vprašanj smo anketirance spraševali, v kolikšni meri po njihovem mnenju držijo trditve o metodi Scrum, objavljene v literaturi. Za izhodišče smo vzeli prednosti, ki jih navajata Rising & Janoff (2000). Anketiranci so veljavnost posameznih trditev ocenjevali s pomočjo petstopenjske Likertove lestvice, pri čemer je ocena 1 (najslabša ocena) pomenila, da je trditev v celoti napačna, ocena 5 (najboljša ocena) pa, da trditev v celoti drži (tj. da se anketiranec z njo v celoti strinja). Spraševali smo o spodnjih trditvah.

1. Izdelek (projekt) lahko razdelimo na zaporedje obvladljivih delov.

Ta trditev izhaja iz iterativne in inkrementalne narave metode Scrum, ki zagotavlja, da smo v določenem trenutku osredinjeni na zahteve trenutne iteracije, ki (zaradi doslednega upoštevanja prioritete posameznih zgodb) prinašajo naročniku največjo korist. To zahteva od produktnega vodje (razvijalci pa mu morajo pomagati pri tem), da načrtuje razvoj v obliki večjega števila izdaj, ki jih je moč postopoma in v čim krajših časovnih intervalih predajati v produkcijo. Ključno vprašanje je, ali je takšna razdelitev mogoča pri vseh projektih, saj se lahko zgodi, da so vse komponente nekega izdelka preveč prepletene med seboj.

2. Projekt napreduje tudi, ko zahteve niso stabilne.

Seznam zahtev lahko dopolnjujemo z novimi uporabniškimi zgodbami ves čas, dokler projekt ni končan. Pri tem je pomembno, da produktni vodja vzdržuje prioriteto posameznih zgodb in s tem zagotavlja, da so zgodbe vedno razvrščene glede na poslovno vrednost, ki jo prinašajo naročniku. To

poenostavi načrtovanje iteracij (vsakokrat izberemo za realizacijo tiste zgodbe, ki so trenutno na vrhu seznama), obenem pa zagotavlja, da naročnik vedno dobiva rešitev, ki je zanj v tistem trenutku najbolj koristna. Pri tem je treba poudariti, da se potem, ko je vsebina iteracije že dogovorjena, produktni vodja ne sme vmešavati v delo razvojne skupine. Med samo iteracijo mora imeti razvojna skupina mir, da se lahko v celoti posveti realizaciji dogovorjene funkcionalnosti. Na začetku naslednje iteracije spet sledi dogovarjanje o realizaciji najpomembnejših uporabniških zgodb iz seznama zahtev, ki se je medtem lahko spremenil.

3. Vsi udeleženci imajo popoln vpogled v potek dela.

To je zagotovljeno z rednimi vsakodnevnimi sestanki, na katerih se člani razvojne skupine medsebojno informirajo o poteku dela in morebitnih težavah, na katere naletijo. Uporabniki pa imajo možnost, da na koncu vsake iteracije pregledajo realizirano funkcionalnost in podajo svoje pripombe.

4. Izboljša se komunikacija med člani razvojne skupine.

Ta trditev se neposredno navezuje na prejšnjo. Naj poudarimo, da komunikacijo izboljšuje tudi t. i. pristop »*face-to-face*«. Glede na to, da vse podrobnosti projekta produktni vodja in člani razvojne skupine razčiščujejo sproti, se komunikacija vsekakor poveča v primerjavi z metodami, pri katerih razvijalci delajo na podlagi pisno podanih zahtev.

5. Člani razvojne skupine so zaslužni za uspeh v času razvoja in ob koncu projekta.

Metoda omogoča članom razvojne skupine, da sami ocenjujejo zahtevnost posameznih zgodb in predvideno hitrost razvoja, zato praviloma niso (ali vsaj ne bi smeli biti) izpostavljeni pritiskom zaradi nerazumno postavljenih rokov in nesprejemljivih zahtev produktnega vodje. Pred tem jih ščiti tudi skrbnik metodologije. Obenem delujejo po načelu samoorganizacije in sami izbirajo način, kako bodo realizirali posamezne zahteve. Zato so pri svojem delu bolj sproščeni. Vse to prispeva k večji odgovornosti in zavzetosti razvojne skupine, da tisto, kar je obljubila na začetku vsake iteracije, tudi uresniči. S tem pa se poveča tudi zavedanje o kolektivnih zaslugah za uspešen potek projekta.

6. Naročnik dobiva novo funkcionalnost v dogovorjenih časovnih intervalih.

To je zagotovljeno z iterativnim in inkrementalnim postopkom razvoja. Vsaka iteracija se konča s sestankom, na katerem razvojna skupina predstavi rezultate svojega dela. Rezultat vsake iteracije mora biti nova funkcionalnost, ki je neposredno uporabna. Dolžina iteracij je dogovorjena vnaprej, tako da naročnik točno ve, v kakšnih intervalih bo dobival predvidene rezultate. Na podlagi sprotnega vzdrževanja načrta izdaje pa lahko vedno oceni, kdaj bo izdaja končana v celoti oziroma kolikšen obseg funkcionalnosti bo lahko realiziran do določenega roka.

7. Naročnik (uporabniki) lahko sproti preverja(jo), kako izdelana programska oprema deluje v resnici.

Koncept »Done« zahteva, da je rezultat vsake iteracije delujoča programska koda, ki je v celoti preverjena in integrirana v trenutno rešitev. Zato lahko naročnik po vsaki iteraciji preveri, kako bo načrtovani sistem deloval v resnici. Pogosto se dogaja, da so uporabniki šele takrat, ko v živo preizkusijo neko rešitev, sposobni pravilno formulirati svoje zahteve. Sprotno preverjanje izdelane programske opreme tako zagotavlja, da uporabnik dobi tako rešitev, kot jo v resnici potrebuje. Po drugi strani pa odpravlja obsežne integracijske teste na koncu projekta.

8. Metoda prispeva k izgradnji boljših odnosov med naročnikom (uporabniki) in razvijalci: poveča se medsebojno zaupanje in presek skupnega znanja.

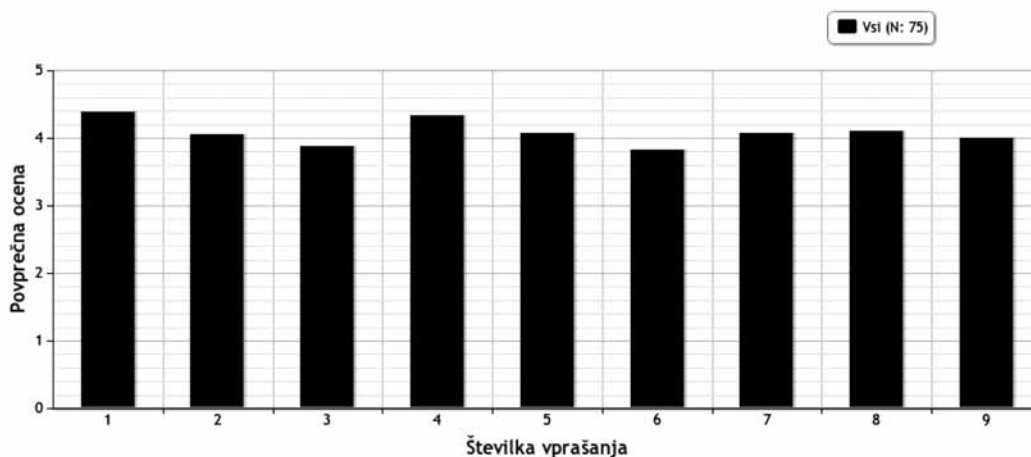
Metoda predvideva sodelovanje predstavnikov naročnika (produktne vodje) ves čas trajanja projekta, ne samo na začetku in na koncu, kot je praviloma pri tradicionalnem pristopu. To vpliva na boljše medsebojno (s)poznavanje vseh sodelujočih, kar se odraža v izboljšanju odnosov in povečanju zaupanja. Razumeti je treba, da v projektih razvoja programske opreme pogosto sodelujejo tudi strokovnjaki z netehničnih področij. Pogosta težava v tako heterogenih skupinah se izkaže prav v velikih interesnih razlikah in majhnem preseku skupnega znanja. Dobra in sprotna komunikacija odpravlja te težave in povečuje presek skupnega znanja.

9. Metoda prispeva k izgradnji pozitivnega vzdušja, v katerem vsi pričakujejo uspeh projekta.

Ta točka je zelo povezana s predhodno. V okolju, v katerem vladajo pozitivni odnosi in medsebojno zaupanje, bo vladalo tudi pozitivno vzdušje, to pa se bo seveda odražalo tudi v pozitivnih pričakovanjih.

Rezultati

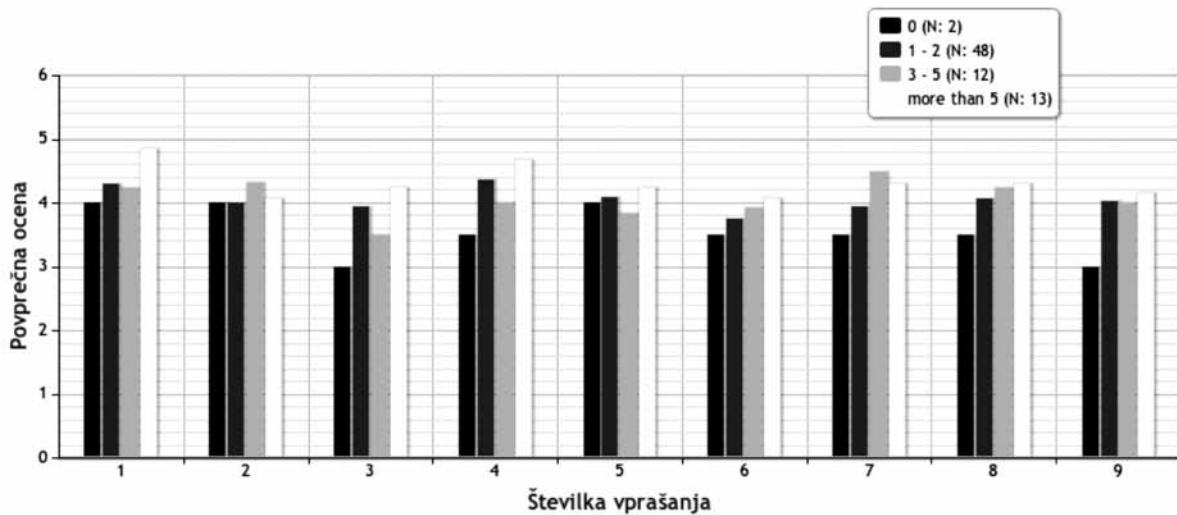
Povprečne ocene, izračunane na podlagi odgovorov vseh 75 anketirancev, so prikazane na sliki 1. Iz slike



Slika 1: Povprečne vrednosti odgovorov na prvi sklop vprašanj

je razvidno, da se uporabniki metode Scrum strinjajo z vsemi trditvami, saj se vse ocene nahajajo na intervalu med 3,8 in 4,4. Tudi standardni odklon, ki smo ga izračunali, je sorazmerno nizek in je za vse trditve manjši od 1,0.

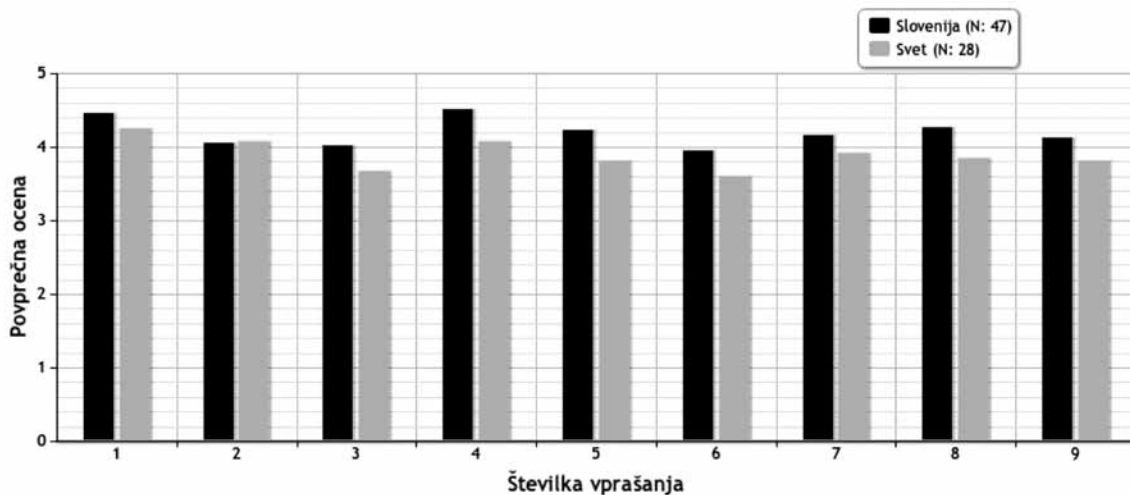
Anketiranci so se najbolj strinjali s trditvama št. 1 (projekt lahko razdelimo na zaporedje obvladljivih delov) in št. 4 (izboljša se komunikacija med člani razvojne skupine). Po drugi strani sta se za najmanj sprejeti trditvi izkazali št. 3 (vsi udeleženci imajo popoln vpogled v potek dela) in št. 6 (naročnik dobiva novo funkcionalnost v dogovorjenih časovnih intervalih).



Slika 2: Povprečne vrednosti odgovorov na prvi sklop vprašanj v odvisnosti od izkušenosti anketiranca

Odgovore smo dodatno analizirali glede na izkušnje anketirancev (slika 2), tako da smo anketirance razdelili na štiri skupine, upoštevajoč število projektov, vodenih po metodi Scrum, pri katerih so že sodelovali (0 projektov, 1–2 projekta, 3–5 projektov in več kot 5 projektov). Pokazalo se je, da se s trditvami opazno bolj strinjajo tisti, ki imajo s Scrumom več izkušenj.

Še posebno pa je zanimivo, da se pri večini trditev povprečne ocene povečujejo sorazmerno s številom projektov, pri katerih je sodeloval anketiranec. Na podlagi te ugotovitve bi lahko trdili, da z več izkušnjami raste tudi zaupanje v metodo Scrum. Videti je, da so novi uporabniki najprej nekoliko bolj previdni, a se z izkušnjami začnejo bolj zavedati njenih prednosti.



Slika 3: Povprečne vrednosti odgovorov na prvi sklop vprašanj v odvisnosti od lokacije anketiranca

Zanimiva je tudi primerjava odgovorov slovenskih in tujih strokovnjakov, ki jo prikazuje slika 3. Prav z vsemi trditvami se slovenski uporabniki metode Scrum strinjajo nekoliko bolj kot njihovi kolegi iz tujine.

4 FAKTORJI, KI VPLIVAJO NA USPEH PROJEKTA

V drugem sklopu vprašanj nas je zanimalo, katere tipične prakse, ki jih predpisuje metoda Scrum, po njihovem mnenju najbolj vplivajo na uspeh projekta. Anketiranci so z ocenami od 1 do 5 ocenjevali spodnje dejavnike.

1. Jasno zastavljen seznam vseh zahtev

Seznam zahtev služi kot podlaga za ocenjevanje zahtevnosti posameznih zgodb, izdelavo načrta izdaje in načrtovanje posameznih iteracij. Sprejemni testi, ki jih mora vsebovati vsaka uporabniška zgodba, omogočajo preverjanje, ali je zgodba realizirana v skladu s potrebami uporabnikov.

2. Sprotna komunikacija s produktnim vodjo

Ker so uporabniške zgodbe samo ohlapen opis vsake zahteve, je za razjasnitev vseh podrobnosti potrebna sprotna komunikacija s produktnim vodjo. Če je ta komunikacija slaba ali ne poteka sproti in se nejasnosti ne razrešijo, lahko pride do situacije, ko razvojna skupina ne more nadaljevati z delom ali pa realizira nekaj, kar ne zadovolji pričakovanj naročnika.

3. Dober skrbnik metodologije

Skrbnik metodologije vsakodnevno bdi nad uporabo metodologije in potekom projekta. Zagotavljati mora take pogoje za delo, da je razvojna skupina čim bolj produktivna. Dober skrbnik metodologije bo zelo zgodaj identificiral odstopanja od začrtanega načrta in nastajajoče težave. Z usklajenim delovanjem vseh sodelujočih bo zagotovil, da bodo takšne okoliščine razrešene čim hitreje.

4. Dobra komunikacija med člani razvojne skupine

Dobra komunikacija med člani razvojne skupine se odraža tudi v dobrih in pozitivnih odnosih, pri čemer vsi člani delujejo kooperativno. Omogoča tudi sproten, celovit in transparenten vpogled v potek dela na projektu ter medsebojno pomoč v primeru težav.

5. Pravilne ocene zahtevnosti posameznih zgodb.

Pravilne ocene zahtevnosti so prvi pogoj za izdelavo zanesljivega načrta izdaje in pravilno načrtovanje posameznih iteracij. Na podlagi teh ocen in predvidene hitrosti razvoja razvrščamo zgodbe v posamezne iteracije. Če ocene niso vsaj približno pravilne, se nam lahko zgodi, da v neko iteracijo uvrstimo preveč ali premalo zgodb.

6. Začetna razporeditev zgodb po iteracijah – načrt izdaje

Načrt izdaje omogoča določitev rokov za izvedbo in časovnih okvirov, v katerih bomo sposobni posamezne dele projekta predati naročniku. Načrt izdaje daje celovit vpogled v časovne okvire projekta kot celote.

7. Pravilna ocena hitrosti razvoja na začetku vsake iteracije

Hitrost razvoja nam pove, kakšno količino dela je razvojna skupina sposobna opraviti v eni iteraciji. Na začetku jo ocenimo glede na različna merila (velikost razvojne skupine, izkušnost itn.), kasneje pa jo prilagajamo glede na dejansko doseženo število točk v predhodnih iteracijah. Ta faktor je zelo povezan s tistim, ki smo ga omenili pod točko 5. Samo kombinacija dobrih ocen zahtevnosti uporabniških zgodb in dobre ocene hitrosti razvoja bo omogočila korektno načrtovanje posameznih iteracij.

8. Dosledno izvajanje sestankov za načrtovanje iteracije

Rezultat tega sestanka je seznam vseh uporabniških zgodb, ki jih moramo realizirati v iteraciji, in dekompozicija teh zgodb na posamezne naloge. Ker ta predstavlja načrt dela posamezne iteracije, bo neustrezno izvajanje tega sestanka lahko pripeljalo do slabo vodene in/ali izpeljane iteracije.

9. Dosledno izvajanje vsakodnevnih sestankov

Vsakodnevni sestanki omogočajo sproten vpogled v stanje projekta. Omogočajo, da morebitne težave odkrijemo takoj, ko se pojavijo, in jih odpravimo sproti. Paziti pa je treba, da ne postanejo predolgi, saj s tem zmanjšamo čas, ki ga lahko izkoristimo za razvoj. Ravno zato so vsakodnevni sestanki časovno omejeni.

10. Dosledno upoštevanje koncepta »Done«

Scrum zahteva, da je na koncu iteracije vsaka funkcionalnost realizirana tako, da jo je moč neposredno predati v uporabo. S tem odpade obsežno testiranje na koncu projekta, ampak imamo vedno na razpolago delujočo verzijo izdelka. Dosledno upoštevanje koncepta »Done« omogoča, da naročnik sproti preizkuša novo funkcionalnost in na podlagi tega precizneje oblikuje svoje zahteve.

11. Dosledno izvajanje sestankov za predstavitev rezultatov iteracije

V okviru teh sestankov produktnemu vodji in zainteresiranim uporabnikom predstavimo rezultate vsake iteracije. Služijo tako za pregled in potrjevanje realizirane funkcionalnosti kot za razpravo o morebitnih pomanjkljivostih in nadaljnjih usmeritvah. S tem sproti preprečujemo morebitne odklone in zagotavljamo, da izdelana funkcionalnost res ustreza potrebam naročnika.

12. Dosledno izvajanje sestankov za oceno kakovosti razvojnega procesa

Ti sestanki služijo za stalno izboljševanje razvojnega procesa. Razvijalcem dajejo možnost, da na podlagi

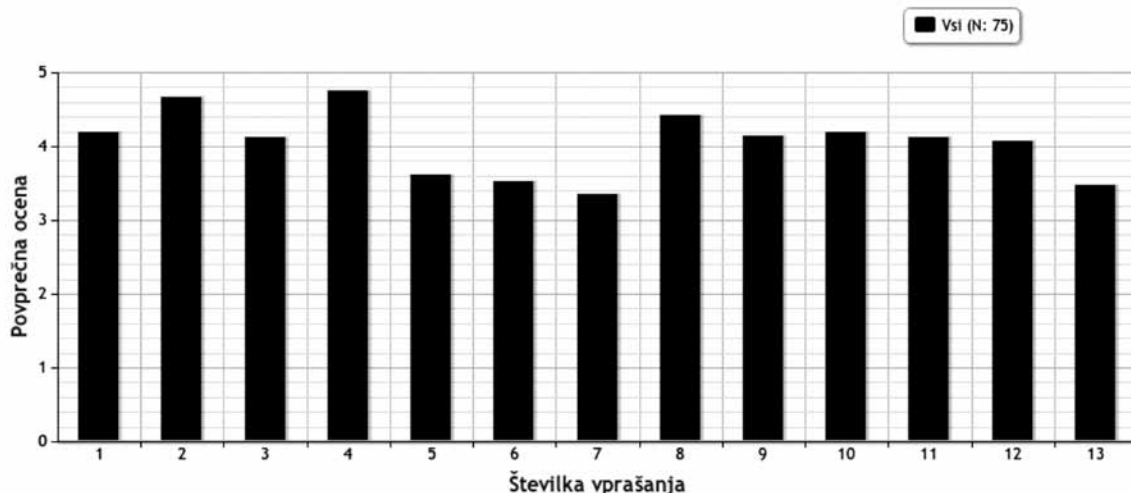
analize pozitivnih in negativnih izkušenj iz prejšnje iteracije definirajo izboljšave, ki jih bodo uvedli v naslednji iteraciji. Seveda je treba poudariti, da so ti sestanki koristni samo, če se dogovorjeni ukrepi potem tudi izvajajo.

13. Dobro orodje za spremljanje poteka del na projektu

Čeprav Scrum ne zahteva obsežne dokumentacije, je koristno imeti orodje, ki nam olajša spremljanje poteka del na projektu. Tovrstna orodja omogočajo vzdrževanje seznama zahtev, seznamov nalog za posamezne iteracije, beleženje podatkov o vložnem in preostalem delu, grafične prikaze napredka ipd.

Rezultati

Iz rezultatov, ki so prikazani na sliki 4, je razvidno, da anketiranci pripisujejo največji pomen dobri komunikaciji med člani razvojne skupine (vprašanje 4) in dobri komunikaciji s produktnim vodjo (vprašanje 2). To je tudi razumljivo, saj za Scrum (tako kot za vse agilne metode) velja, da stalna in kakovostna komunikacija nadomešča pisanje obsežne in zahtevne dokumentacije.



Slika 4: Povprečne vrednosti odgovorov na drugi sklop vprašanj

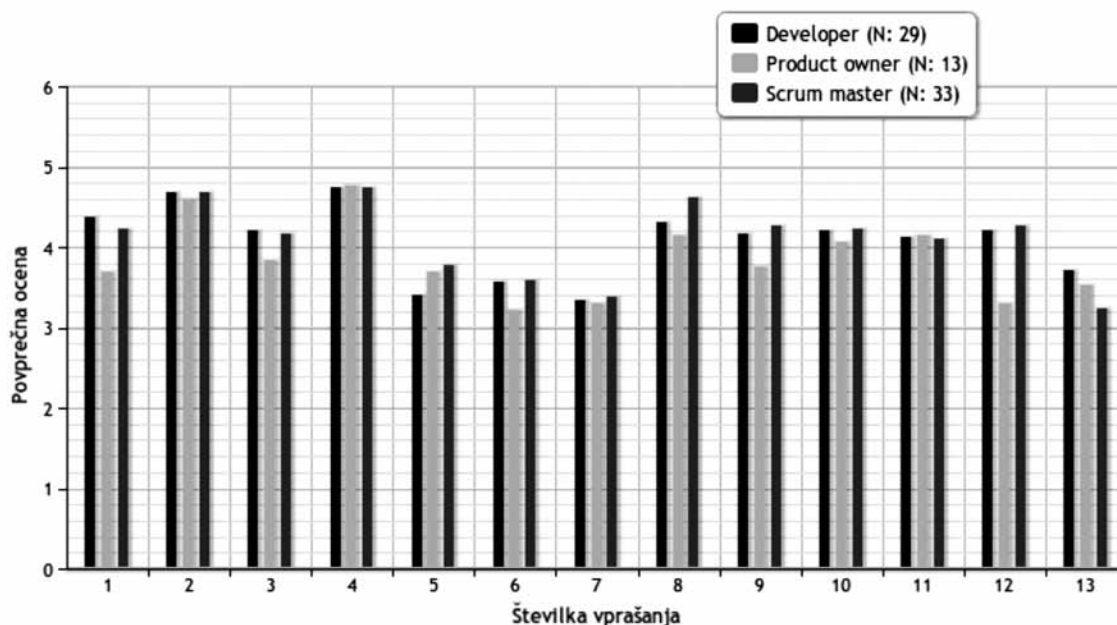
Po drugi strani pa vidimo, da so bile najnižje ocenjene tiste prakse, ki se nanašajo na ocenjevanje hitrosti razvoja (vprašanje 7), načrtovanje izdaje (vprašanje 6) in ocenjevanje zahtevnosti posameznih zgodb (vprašanje 5). V določeni meri je to presene-

tljivo, še posebno zato, ker so ocena hitrosti razvoja in ocene posameznih zgodb podlaga za načrtovanje posameznih iteracij, načrt izdaje pa zagotavlja pregled nad projektom kot celoto. Po drugi strani pa je takšen rezultat razumljiv, saj Scrum večinoma upora-

bljamo v agilnih okoljih, v katerih je v ospredju prilaganje spremembam v zahtevah, medtem ko sta natančnost načrtovanja in sledenje načrtu drugotnega pomena. Sorazmerno nizko je bila ocenjena tudi potreba po ustreznem orodju za spremljanje poteka del, kar pa potrjuje, da je metoda Scrum razmeroma preprosta za uporabo, mnogi uporabniki pa namesto računalniške podpore raje uporabljajo fizične kartice in tablo, na kateri s prestavljanjem kartic prikazujejo, kako napreduje delo na projektu.

Analiza odgovorov glede na vlogo anketiranca je prikazana na sliki 5. Čeprav pri večini vprašanj med odgovori ni bistvenih razlik, velja opozoriti na nekatera odstopanja v odgovorih produktnih vodij, ki imajo pri nekaj vprašanjih opazno nižjo vrednost od ostalih. Ta razlika je najbolj očitna pri odgovorih na vprašanje 12, ki govori o doslednem izvajanju

sestankov za oceno kakovosti razvojnega procesa (angl. *Sprint Retrospective Meetings*). To se da preprosto razložiti z dejstvom, da pravila metode Scrum ne zahtevajo obvezne prisotnosti produktnega vodje na teh sestankih, saj so izboljšave razvojnega procesa predvsem v domeni razvijalcev, za njihovo uvajanje pa mora skrbeti skrbnik metodologije. Zato so ti sestanki za produktnega vodjo pogosto manj opazni in posledično izpadejo manj pomembni. Seveda pa ne bi bilo nič narobe, če bi se tudi produktni vodja bolj zavedal prave vrednosti teh sestankov. S tem ko bolje teče delovni proces, namreč veliko pridobi tudi on: prej dobi svoj izdelek, ta pa je tudi bolj kakovosten. Zato predlagamo, da se ta vidik projekta še posebno predstavi produktnemu vodji, katerega – če je to mogoče – povabimo k sodelovanju na omenjenih sestankih.



Slika 5: Povprečne vrednosti odgovorov na drugi sklop vprašanj glede na vlogo anketiranca v metodi

V primerjavi z ostalimi so produktni vodje nižje ocenili tudi pomen jasno zastavljenega seznama vseh zahtev (vprašanje 1) in načrta izdaje (vprašanje 6). Pričakovali bi, da bo produktni vodja tem vidikom projekta pripisoval večjo vlogo, saj sta tako eden kot drugi tesno povezana z njegovim delom. Prav produktni vodja je odgovoren za izdelavo in upravljanje seznama zahtev, dober seznam zahtev pa je prvi pogoj za izdelavo načrta izdaje, ki produktnemu vodji daje informacijo o časovnem poteku projekta in mu

omogoča, da pravilno načrtuje dinamiko uvajanja nove programske opreme v svoji organizaciji.

Nižje vrednosti odgovorov na vprašanje 1 lahko razložimo s predpostavko, da se zdijo zahteve produktnemu vodji jasne in samoumevne, zato upravljanju seznama zahtev ne pripisuje takega pomena kot drugi udeleženci v razvojnem procesu. Pri tem se očitno ne zaveda, da je seznam zahtev gonilo vsega razvoja in vpliva na vse druge aktivnosti. Zato je nujno, da produktnega vodjo na to še posebej opozo-

rimo. Težje pa je razložiti nižje vrednosti odgovorov na vprašanje 6. Razlagi za takšen rezultat bi lahko bili dve. Mogoče je, da pri t. i. projektih »On-going« in »Start-up«, pri katerih zaradi nejasnih in spremenljivih zahtev največ uporabljamo agilne metode, produktne vodje časovnemu vidiku projekta ne dajejo takšnega poudarka, kot to navadno pričakujemo. Pogoste spremembe v zahtevah lahko tudi povzročijo, da postanejo načrti izdaj (če jih ne sproti vzdržujemo) hitro neuporabni, kar še dodatno zniža oceno o njihovi koristnosti. Drugi razlog bi lahko bil v preslabem poznavanju metodologije in nerazumevanju pomena, ki ga ima načrt izdaje za potek projekta. Na to opozarja tudi Cohn (2005), ki pravi, da načrt izdaje služi kot kašipot k cilju, h kateremu naj napreduje projekt, brez njega pa se razvijalci le brezkončno pomikajo od iteracije do iteracije. Vsekakor bi bilo zanimivo, če bi to vprašanje podrobneje raziskali v kateri od prihodnjih podobnih raziskav.

Za razliko od produktnih vodij pa je videti, da skrbniki metodologije dajejo načrtovanju večji poudarek. To se odraža v odgovorih na vprašanji 5 in 8, pri katerih so ocene skrbnikov metodologije višje kot pri drugih vlogah. Omenjeni vprašanji se nanašata na točnost ocen zahtevnosti posameznih zgodb in dosledno izvajanje sestankov za načrtovanje iteracije. Oba dejavnika sta povezana s tekočim potekom projekta in vplivata na vsebino in obseg dela v posameznih iteracijah. Ker so skrbniki metodologije odgovorni za nemoten potek dela, je povsem razumljiva njihova skrb za ta faktorja. Delno pa je lahko povečana skrb za pravilno ocenjevanje zgodb in načrtovanje posameznih iteracij tudi posledica zgodovinske obremenjenosti nekaterih skrbnikov metodologije s klasičnim vodenjem projektov, ki (včasih tudi nehoti) enači vlogo skrbnika metodologije z vlogo vodje projekta, čeprav metoda Scrum temu nasprotuje in zahteva, da razvojna skupina deluje po načelu samoorganizacije, skrbnik metodologije pa skrbi za njeno pravilno uporabo in zagotavljanje optimalnih pogojev za delo.

5 SKLEP

Naša raziskava potrjuje, da se uporabniki metode Scrum v celoti strinjajo s trditvami o prednostih te metode, ki jih navaja literatura, še zlasti s tem, da Scrum omogoča razbitje projekta na zaporedje manjših obvladljivih delov in izboljša komunikacijo med člani razvojne skupine. Prav komunikacija pa je

po mnenju anketirancev najpomembnejši dejavnik, ki vpliva na uspešnost po Scrumu vodenih projektov. To vključuje tako komunikacijo med razvojno skupino in produktnim vodjo, kot tudi komunikacijo znotraj razvojne skupine in s skrbnikom metodologije. Posebno je pomembno, da stopnja strinjanja s prednostmi metode Scrum narašča sorazmerno z večanjem praktičnih izkušenj njenih uporabnikov.

Druga, nekoliko presenetljiva ugotovitev govori o tem, da uporabniki metode Scrum ne pripisujejo tako velike vloge ocenjevanju in načrtovanju časovnega poteka projekta, kot bi lahko pričakovali. To lahko razložimo z naravo projektov, pri katerih uporabljamo Scrum. Pogosto gre za spletne projekte in projekte »Start-up«, pri katerih sledimo principu »Release early, release often«. Pri tem principu velikokrat pravzaprav ne moremo govoriti o roku izvedbe, saj se izdelek neprestano izboljšuje in dodeluje, izdaje pa se dogajajo pogosto in inkrementalno. To bi lahko potrdil tudi komentar enega od anketirancev: »Težko je reči, da naročnik dobi dogovorjeno funkcionalnost v roku – ta je namreč fleksibilna in razširljiva ter se sproti prilagaja razmeram na projektu.«

6 VIRI IN LITERATURA

- [1] Ambler, S. W. (2008). Has agile peaked? Let's look at the numbers. *Dr. Dobb's Journal*. <http://www.ddj.com/architect/207600615?pgno=1>.
- [2] Begel, A. & Nagappan, N. (2007). Usage and preceptions of agile software development in an industrial context: an exploratory study. Zbornik prispevkov: *First International Symposium on Empirical Software Engineering and Measurement*. Madrid, Španija.
- [3] Boehm, B. & Turner, R. (2004). *Balancing Agility and Discipline*. Boston, MA: Addison-Wesley.
- [4] Cohn, M. (2004). *User stories applied for agile software development*. Boston, MA: Addison-Wesley.
- [5] Cohn, M. (2005). *Agile Estimating and Planning*. Upper Saddle River, NJ: Prentice Hall PTR.
- [6] Dybå, T. & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50, 9–10, 833–859.
- [7] Laanti, M., Salo, O. & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology*, 53, 3, 276–290.
- [8] Mahnič, V. (2010). Teaching Scrum through team-project work: students' perceptions and teacher's observations. *International Journal of Engineering Education*, 26, 1, 96–110.
- [9] Mahnič, V. (2011). Problemi in rešitve pri uvajanju metode Scrum v proces razvoja programske opreme. Zbornik prispevkov: *18. konferenca Dnevi slovenske informatike*. Portorož, Slovenija.
- [10] Mahnič, V. (2012). A Capstone Course on Agile Software Development Using Scrum. *IEEE Transactions on Education*, 55, 1, 99–106.

- [11] Murphy, T., Duggan, J., Norton, D., Prentice, B., Plummer, D. & Landry, S. (2009). Predicts 2010: Agile and Cloud Impact Application Development Directions, Gartner. <http://www.gartner.com/DisplayDocument?id=1244514>.
- [12] Rising, L. & Janoff, N. S. (2000). The Scrum software development process for small teams, *IEEE Software*. 17, 4, 26–32.
- [13] Schwaber, K. (2004). *Agile Project Management with Scrum*, Redmond, WA: Microsoft Press.
- [14] Scotland, K. & Boutin, A. (2008). Integrating Scrum with the process framework at Yahoo! Europe, Zbornik prispevkov: *Agile 2008*. Toronto, Canada.
- [15] Urevc, J., Štebe, R. & Mahnič, V. (2012). Izkušnje z uvajanjem metode Scrum v časopisni hiši Delo, Zbornik prispevkov: *19. konferenca Dnevi slovenske informatike*. Portorož, Slovenija.
- [16] Version One (2011). State of Agile Survey. http://www.versionone.com/pdf/2011_State_of_Agile_Development_Survey_Results.pdf.
- [17] West, D. & Grant, T. (2010). Agile Development: Mainstream Adoption Has Changed Agility. Forrester research. http://www.forrester.com/rb/Research/agile_development_mainstream_adoption_has_changed_agility/q/id/56100/t/2.
- [18] Williams, L. (2010). Agile software development methodologies and practices. *Advances in Computers*. 80, 1–44.

■

Janez Urevc je diplomiral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. V svojem diplomskem delu se je ukvarjal z uvedbo agilne metode Scrum v oddelek spletnega razvoja časopisne hiše Delo, pri tem pa posebno pozornost namenil ovrednotenju njenih prednosti, merjenju učinkovitosti razvojne skupine in analizi točnosti ocenjevanja uporabniških zgodb. Poleg agilnih metod ga zanimajo spletne in mobilne tehnologije, predvsem v povezavi s prsto programske opremo. V času študija je dve leti deloval kot vodja MMC Kiperpipa, sedaj pa deluje kot samostojni razvijalec ter največ časa posveča raziskovanju najaktualnejših tehnologij na spletu in mobilnih platformah. Svoje prispevke objavlja v različnih poljudnih in strokovnih publikacijah ter konferencah.

■

Viljan Mahnič je izredni profesor in predstojnik Laboratorija za tehnologijo programske opreme na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Pri svojem pedagoškem in raziskovalnem delu namenja posebno pozornost agilnim metodam za razvoj programske opreme in informacijskih sistemov, metrikam v programske opreme ter zagotavljanju učinkovitosti in kakovosti razvojnega procesa. Rezultate svojih raziskav je objavil v več uglednih znanstvenih revijah, redno pa sodeluje tudi na domačih in mednarodnih konferencah. Doslej je vodil več projektov s področja razvoja informacijskih sistemov, še zlasti na področju študijske informatike, pomaga pa tudi pri uvajanju metode Scrum v slovenska podjetja.