

David Osolnik

SCRUM – AGILNA METODA

»Na Scrum gledam kot na metodo timskega dela, s katero se ekipi postavi ohlapen okvir načina dela, vseeno pa v veliki meri pusti proste roke. Metoda vsebuje več elementov, ki dokazano izboljšujejo delovanje timov, vendar kot celota ni dobro raziskana in utemeljena. Zaradi ohlapnosti določil je velik del uspeha odvisen od delovanja članov tima, njihovih znanj, socialnih spretnosti, njihove inovativnosti itd. Poraja se vprašanje, ali ne bi bil tako dober in izurjen kader uspešen tudi pri uporabi katere druge metode dela.«

Scrum spada v širšo skupino agilnih metod, katerih cilj je spremeniti način dela skupin in organizacij na način, da se bodo te lahko hitreje odzivale na številne spremembe, ki jih s seboj prinaša današnji trg. Metoda izhaja iz programerstva, postopoma pa metodo uporabljajo tudi ostale organizacije (Fowler in Highsmith, 2001). Nanjo lahko gledamo kot na okvir za timsko delo manjših ekip, ki se trudijo na koncu omejenega časovnega obdobja (imenovanega šprint) doseči zastavljeni cilj.

Za razumevanje metode Scrum je treba poznati različne timske vloge, dogodke in artefakte, ki jih ta predvideva. Timske vloge zajemajo razvijalce, katerih glavna naloga je delo na projektu, lastnika projekta, ki je nekakšen posrednik med vodstvom in ekipo Scrum, zadolžen za oblikovanje in jasno komunikacijo seznama zahtev projekta, in skrbnika metode, katerega naloga je, da zagotavlja pogoje, v katerih lahko ostali pravilno sledijo metodi. Dogodki Scruma so šprint, ki predstavlja vnaprej določeno časovno omejitev (npr. en mesec). V tem času se zvrstijo vsi ostali dogodki. Drug dogodek Scrum je načrtovanje šprinta, pri katerem se razvijalci in lastnik projekta dogovorijo, kolikšnemu številu zahtev bodo v prihajajočem šprintu lahko ugodili, poleg tega pa določijo

tudi cilj šprinta – kratek dnevni sestanek, na katerem se cela ekipa zbere in odloča o prihodnjih dejanjih. Sledita ocena opravljenega dela, ko tim Scrum vodstvu predstavi svoj napredek in se skupaj z njim dogovori o nadaljnjem delu, ter retrospektiva, na kateri člani tima komentirajo, kaj se jim je zdelo pri načinu njihovega delovanja v preteklem šprintu dobro in kaj slabo, ter se odločijo, kaj bodo v prihodnje spremenili.

Artefakti so seznam zahtev produkta, seznam zahtev šprinta in inkrementi. Seznam zahtev produkta v dogovoru z vodstvom pripravi nosilec projekta in zajema vse lastnosti, za katere si želijo, da bi jih njihov produkt v končni obliki imel. Iz njega nato pri načrtovanju šprinta črpa ekipa Scrum in sestavi seznam zahtev šprinta. To je seznam vseh zahtev, za katere predvidevajo, da jih bodo uspeli razrešiti v tekočem šprintu. Zadnji artefakt so inkrementi. Tudi te tim oblikuje pri načrtovanju šprinta, predstavljajo pa manjše delovne naloge, s katerimi bodo med šprintom člani tima lahko postopno dosegli zadani cilj – ugodili vsem zadanim zahtevam (Schwaber in Sunderland, 2020).

NASTANEK IN RAZVOJ METODE

Uporaba agilnih metod se je začela s prvo znano uporabo inkrementalnega pristopa leta 1957 (Abbas idr., 2008). Čeprav se je kar nekaj metod, ki se še danes uporabljajo, razvilo v 90. letih (npr. Scrum, extreme programming, rapid application development), je agilnost svoj pravi preboj doživela leta 2001, ko je 17 zagovornikov različnih agilnih metod v smučarskem središču zvezne države Utah podpisalo Agilni manifest in ustanovilo prvo Agilno zvezo (Fowler in Highsmith, 2001).

Scrum sta njegova ustvarjalca Schwaber in Sunderland prvič predstavila na konferenci OOPSLA leta 1995 (Schwaber in Sunderland, 2020), leta 2009 pa je bila ustanovljena spletna stran scrum.org, ki skrbi za izobraževanja in dodeljuje certifikate za uporabo metode Scrum (Helping People and Teams Solve Complex Problems, b. d.). Popularnost metode se že od začetka, skupaj z ostalimi agilnimi metodami, vztrajno povečuje. Rezultati raziskave Project management instituta s prek 4000 udeleženci kažejo, da je v letu 2018 skoraj tretjina projektnih managerjev uporabljalo Scrum vedno ali pogosto, podoben delež jih je Scrum uporabljalo včasih,

petina redko, manj kot četrtnina vprašanih managerjev pa Scruma sploh ni uporabljalo (Project Management Institute, 2018).

ZNANSTVENA SPOZNANJA O UPORABNOSTI METODE

Kakovostnih raziskav, ki bi se ukvarjale z učinkovitostjo Scruma, ni veliko. Večina raziskav je zasnovanih kot študija primera ene ekipe Scrum ali organizacije, sestavljene iz nekaj ekip Scrum. Navadno so to organizacije s področja programskih rešitev. Avtorji raziskav pri tem običajno sklepajo na podlagi intervjujev z zaposlenimi, le redko pa uporabijo kakšno drugo tehniko zbiranja podatkov. Veliko raziskav je opravljenih tudi s timi, ki so se s Scrumom šele spoznavali, zato je bila metoda zaradi nepravilne uporabe manj učinkovita.

Ker je Scrum obsežna, vendar razmeroma ohlapna metoda, je raziskovanje učinkov metode in konstruktov, povezanih z njo, težavno. V nadaljevanju zato predstavljam raziskave, ki se nanašajo na uporabo Scruma, poleg teh pa tudi takšne, ki se ukvarjajo s posameznimi elementi metode.

Uporaba Scruma

V norveški longitudinalni študiji v programerskem podjetju s približno 150 zaposlenimi so raziskovalci primerjali tradicionalni linearno-sekvenčni način dela (angl. *waterfall*) in Scrum (Li idr., 2010). Med načinoma dela niso opazili razlik v količini napak v programih, opazili pa so razlike v predvidljivosti, saj so imeli zaposleni pri uporabi Scruma zaradi rednega vmesnega testiranja in popravljanja napak boljši pregled in nadzor nad projektom. Udeleženci so pri Scrumu poročali tudi o večjem prenosu znanja med zaposlenimi v primerjavi s tradicionalnim načinom dela. Kljub temu se je pokazalo, da je uporaba Scruma povečala pritisk na zaposlene, ki so poročali o višjem stresu, poleg tega pa zaradi časovnega pritiska šprintov tudi niso opravljali nekaterih pomembnih vzdrževalnih del.

Število napak v programih so testirali tudi v podjetju Adobe systems, kjer metodo uporabljajo že dalj časa. Ugotovili so, da se ta negativno povezuje s skupnim rezultatom na vprašalniku Scrum, s katerim so preverjali, v kolikšni meri se timi držijo zahtev metode (Green, 2011). Na drugi

strani so v organizaciji, kjer je bil Scrum vpeljan pred kratkim, razvijalci poročali o slabem pregledu nad projektom, saj so bili posamezniki navajeni osebne avtonomije in so tako pomembne odločitve sprejemali brez posvetovanja z ostalo ekipo. Poleg tega zaradi preveč avtoritarnega skrbnika metode razvijalci na dnevnih sestankih niso delili slabih informacij, pač pa so napake skušali popraviti sami na skrivaj, tako pa se je izgubil smisel timskega dela in samoorganizacije članov. Na koncu projekta so člani znali izpostaviti pozitivne vidike, težave pa so imeli s podajanjem in sprejemanjem kritike, kar je močno otežilo iskanje izboljšav načina dela za prihodnji šprint (Moe in Dingsøyr, 2010). Težav, ki lahko nastopijo pri uporabi metode, je še več. Med desetimi najpogostejšimi, ki jih pri svojem delu opažajo izkušeni uporabniki, jih je nekaj programersko specifičnih, druge pa so: pomanjkljivo ali nepravilno opisane naloge otežujejo načrtovanje in snovanje; težavnost izdelave vedno novih izboljšav zaradi nepopolnega in slabo zasnovanega osrednjega produkta; težave z nizko kakovostjo zaradi hitenja; težavnost razčlenitve zahtev na manjše naloge in medsebojno motenje oz. zaviranje zaradi soodvisnosti različnih nalog, ki se hkrati opravljajo (Miler in Kajdy, 2018).

Scrum tudi ne spodbuja enakomerno vseh vidikov timskega dela (Moe in Dingsøyr, 2008). V eni od ekip z osmimi člani ni prihajalo do delitve mentalnih modelov, saj so bili posamezni člani tima preveč specializirani in so se ukvarjali le s svojim področjem. Prav tako ni bila izboljšana komunikacija zunaj dnevnih sestankov, pa tudi na teh je včasih prihajalo do neposlušanja. Avtorja ocenjujeta, da Scrum ni krepil medosebnega zaupanja članov. Za medsebojni nadzor uspešnosti je bilo po besedah članov z dnevnimi sestanki in retrospektivo poskrbljeno bolje kot pred uporabo metode, tudi za vodenje je bilo s samoorganizacijo in trenerstvom skrbnika metode dobro poskrbljeno. V dobrem timu bi morali biti sodelavci sposobni po potrebi prevzeti delo drugega in ga tako razbremeniti, vendar v tem primeru do tega ni prihajalo. Prav tako se ni izboljšala prilagodljivost na spremembe, ki jo obljublajo agilne metode, saj so člani v veliki meri obdržali način dela, ki so ga bili navajeni od prej. Trpelo je tudi timsko delo samo, saj so bili člani specializirani do te mere, da se med seboj niso dobro razumeli in tako odločitev niso sprejemali skupaj.

Čeprav je Scrum obsežna metoda, vseeno ne zagotavlja podpore pri vseh procesih projektnega managementa. Katayama in Goldman (2011)

sta pri analizi različnih agilnih metod v programerstvu ugotovila, da Scrum ponuja vodila pri določitvi zahtev oz. specifikacij, snovanju, programiranju oz. izdelavi, testiranju delovanja kode, testiranju integracije kode v sistem in testiranju delovanja sistema; izpušča pa fazo zasnove projekta na začetku ter testiranja sprejemljivosti – ali je sistem dovolj dober za lansiranje in vzdrževanje izdelanega sistema po koncu projekta.

Avtonomija in povratna informacija

Uporaba Scruma na zaposlene prelaga večjo odgovornost, hkrati pa imajo ekipe pri svojem delu več avtonomije. S preučevanjem avtonomije timov se ukvarja večje število raziskovalcev, ki npr. poročajo o šibkem direktnem vplivu avtonomije timov na poslovno uspešnost 25 finskih malih trgovskih podjetij, indirektno pa avtonomija vpliva na uspešnost tudi prek povečane pripadnosti podjetju (Von Bonsdorff idr., 2015). Poleg tega so bolj avtonomni timi v primerjavi z manj avtonomnimi uspešnejši pri delu v nejasnih okoliščinah (Cordery idr., 2010), bolj avtonomni timi pa se bolje odrežejo predvsem takrat, ko je soodvisnost članov visoka in morajo ti za doseg cilja sodelovati (Langfred, 2005). Na drugi strani avtonomnost tima slabo vpliva na uspešnost pri nalogah, kjer posamezniki lahko delajo samostojno. Tam se bolje izkaže visoka avtonomija posameznikov, ne skupine.

Poleg avtonomnosti tima raziskovalci izpostavljajo tudi pomen povratne informacije (Gonzales-Mulé idr., 2016). V raziskavi 110 južno-korejskih timov so namreč odkrili, da so bolj avtonomni timi, ki so hkrati dobivali pogoste in kakovostne povratne informacije o njihovih dosežkih, dosegali boljše rezultate od ostalih timov, medtem ko so se timi z visoko avtonomijo in malo povratnimi informacijami pri nalogah odrezali najslabše. Pozitivne učinke povratne informacije so odkrili tudi na vzorcu 176 zaposlenih, pri katerem je bila negativna povratna informacija povezana z učenjem zaposlenih (Liu in Xiang, 2018). Povezava naj bi stekla prek občutka krivde zaradi negativne povratne informacije, transformacijski vodje pa naj bi bili tisti, ki so najbolj uspešni pri opozarjanju na razkorak med pričakovanji in trenutnimi rezultati ter motivacijo zaposlenih za učenje.

Samoorganizacija in vodenje

Velik del uspešnosti Scruma je odvisen od samoorganizacije članov ekipe, skrbniki metode pa morajo delovati kot uslužni vodje (A Better Way Of Building Products, b. d.). Pri raziskovanju kriznega managementa, kjer so bile ekipe izpostavljene velikim tveganjem, časovnemu pritisku in negotovosti, so se samoorganizirani timi v kriznih situacijah odzivali bolje kot funkcionalni timi (Jobidon idr., 2015). Pri samoorganiziranih timih člani zavzemajo več različnih vlog, medtem ko se člani funkcionalnih timov bolj držijo enakih vlog in v različnih situacijah izkazujejo bolj podobna vedenja. Vseeno naj prevelika variabilnost vlog ne bi bila dobra, saj je znotraj samoorganiziranih timov prišlo do negativne povezave med variabilnostjo vlog in dosežkom. Avtorji ta rezultat pojasnjujejo z nejasnostjo vlog in se sprašujejo, kako lahko določimo mejo med večjo fleksibilnostjo in splošno zmedo.

Pri raziskovanju vodenja sta Holtzhausen in Klerk (2018) ugotovila, da skrbniki metode v veliki meri res prakticirajo uslužno vodenje, poleg tega pa se to tudi pozitivno povezuje z uspešnostjo timov, medtem ko Mroz in sodelavci (2018) poročajo o pozitivnih učinkih participativnega vodenja, saj so udeleženci v njihovi raziskavi participativne vodje na sestankih zaznavali kot bolj sposobne in bolj tople, poleg tega pa so jih imeli tudi raje od direktivnih vodij.

Časovni pritisk

Kot zadnji element, ki ga prinaša uporaba Scruma, navajam časovni pritisk, ustvarjen s šprintom. Ta ima lahko pozitivne in negativne učinke. Pri večjem časovnem pritisku avtonomija slabo vpliva na delovno zavzetost in subjektivno blagostanje oseb z zunanjim lokusom kontrole, medtem ko pozitivno vpliva na tiste z notranjim (Stiglbauer, 2017). Baer in Oldham (2006) na vzorcu 170 zaposlenih in 10 nadrejenih v proizvodni organizaciji ugotavljata negativno povezanost med časovnim pritiskom in kreativnostjo zaposlenih, medtem ko je pri osebah z visoko izraženo osebnostno lastnostjo odprtosti, ki so hkrati poročali o visoki zaznani podpori organizacije pri kreativnosti, v regresijski enačbi prišlo do obrnjene U-krivulje. Iz tega lahko sklepamo, da so ti posamezniki najbolj kreativni pri zmernem časovnem pritisku, medtem ko zelo nizek ali visok časovni pritisk na njihovo kreativnost ne vpliva dobro.

PREDSTAVITEV METODE

KDAJ METODO UPORABIMO?

Scrum (in druge agilne metode) lahko uporabimo takrat, ko želimo doseči boljše sodelovanje in povečati prispevek zaposlenih pri odločanju, ko želimo povečati čas dela na projektu v primerjavi s časom, potrebnim za urejanje dokumentacije, ko želimo tesnejše sodelovanje in soustvarjanje s stranko ter, morda najpomembneje, ko se želimo hitreje odzivati na spremembe v okolju ali hitreje pripeljati svoj produkt od zasnove do izvedbe (Fowler in Highsmith, 2001).

ZA KOGA JE METODA PRIMERNA?

Scrum je namenjen manjšim samostojnim ekipam (navadno do 10 članov) s skupnim ciljem, ki ga je možno doseči v obdobju enega meseca ali manj (Schwaber in Sunderland, 2020). Ekipe so lahko multidisciplinarne (Rigby idr., 2018), vendar je pomembno, da delo posameznih članov ni preveč specializirano, saj sta tako sodelovanje in pregled vseh članov nad potekom projekta otežena (Moe in Dingsøyr, 2008).

Scrum je najbolj primeren za projektno delo in inoviranje, ko ekipe rešujejo kompleksne in odprte probleme (Rigby idr., 2016). Lahko pa se ga uporabi tudi pri kateremkoli delu, ki ga skupina zaposlenih opravlja avtonomno in v več korakih – inkrementalno (Strode, 2006). Ker se pri metodi poudarja tesno sodelovanje z naročnikom (Fowler in Highsmith, 2001), je lahko posebej primerna za različno agencijsko delo, manj primerna pa je za rutinska dela, npr. vzdrževanje ali računovodstvo (Rigby idr., 2016).

Uporaba Scruma je manj primerna tudi v primerih, ko so razmere na trgu stabilne in predvidljive (ni sprememb v potrebah strank), ko so ciljne skupine za organizacijo nedosegljive in je tako sodelovanje z njimi onemogočeno, ko so rešitve problemov znane že v naprej in jih je treba le implementirati ter ko je testiranje sestavnih delov končnega izdelka nemogoče ali nesmiselno (Rigby idr., 2016). Poleg omenjenih okoliščin je treba pred uporabo oceniti tveganje, ki ga prinaša lansiranje nepopolnega produkta. V splošnem so agilne metode zaradi svoje iteracijske narave (izdelava produkta v več različicah, ki se navadno nadgrajujejo)

manj primerne takrat, ko je pomembno, da je produkt že v začetku popoln (npr. izdaja novega iPhona), saj bi v nasprotnem primeru prišlo do velikih izgub (Boehm, 2002).

KAKO METODO IZVEDEMO?

Uporaba Scruma se v različnih organizacijah razlikuje, saj si ga večina vsaj deloma prilagodi ali pa uporablja le določene elemente, drugih pa ne. V raziskavi uporabe Scruma v nemških podjetjih raziskovalci tako ugotavljajo, da so ekipe ponekod sestavljene le iz dveh članov, drugje pa tudi iz 25 članov (Diebold idr., 2015). Nekatere ekipe imajo skrbnika metodologije in lastnika produkta, druge samo enega od njiju ali nobenega, ponekod pa si v različnih situacijah vlogo skrbnika deli tudi več oseb hkrati. Prav tako se pri ekipah razlikuje trajanje šprintov ter trajanje in pogostost sestankov. Šprinti so trajali od enega do štirih tednov, sestanki pa so trajali od 15 minut do pol ure. V nekaterih podjetjih so bili sestanki dnevni, drugje so potekali enkrat na dva dni ali pa so bili sklicani po potrebi. Poleg naštetih razlik so ekipe variirale tudi v tem, koliko časa so porabile za načrtovanje šprinta, koliko časa so posvetile ocenjevanju dela in retrospektivam, kdo vse je moral biti ali lahko bil na sestankih, kako so reševale problem nedokončanega dela ob koncu šprinta itd. (Diebold idr., 2015).

Zaradi velikega števila različnih dobrih in slabih praks, prilagoditev okoliščinam v organizacijah in združitvev z drugimi agilnimi metodami (zelo popularen je npr. t. i. Scrumban) včasih težko sodimo, kaj je še Scrum in kaj to več ni. V želji po čim manjši zmedbi bom v nadaljevanju predstavil Scrum, kot ga v novoizdanem vodniku Scrum zagovarjata začetnika metode Schwaber in Sunderland (2020).

Za uspešno izvedbo metode je treba poznati sestavo tima Scrum ter dogodke in artefakte Scruma. Tim zajema manjše število ljudi (priporočeno največ 10), ki v določenem časovnem obdobju (imenovanem šprint) skupaj rešujejo zadani problem. Tim nima določenega vodje, ampak se samoorganizira in samonadzoruje, odgovornost za dobro ali slabo opravljeno delo pa nosi celotna ekipa. V primerjavami s tradicionalnimi načini dela imajo člani tima veliko večji vpliv in svobodo odločanja znotraj posameznega šprinta, saj se v timu sami odločajo, na kakšen način bodo dosegli zastavljene cilje (Schwaber in Sunderland, 2020).

Tim Scrum

Znotraj tima lahko posamezniki zasedajo tri različne vloge. Večina je navadnih članov, zaradi programerskih korenin Scruma imenovanih razvijalci. Njihova glavna naloga je delo na zadanem projektu, prisotni in soodgovorni pa so tudi pri načrtovanju šprinta.

Ena oseba v timu ima vlogo lastnika projekta (angl. *product owner*). Lastnik projekta je nekakšen posrednik oz. vmesni člen med vodstvom in ekipo Scrum. Odgovoren je za oblikovanje seznama zahtev (angl. *product backlog*). Ta mora biti jasen, uporabniške zgodbe pa morajo biti urejene po prioriteti, ki jo lastnik projekta določi skupaj z vodstvom. Naloga lastnika projekta je tudi ta, da je vedno na voljo za dodatna pojasnila in opis zahtev, drugače pa je z ekipo najmanj v stiku, saj se pogosto zgodi, da je lastnik projekta hkrati tudi lastnik več drugih projektov.

Zadnja vloga se imenuje skrbnik metodologije (angl. *scrum master*). To je poznavalec Scruma, katerega glavna naloga je, da zagotavlja pogoje, v katerih metoda lahko pravilno poteka. Njegove naloge zajemajo učenje učinkovitega samoupravljanja, sklicevanje in vodstvo dnevnih sestankov, pomoč pri načrtovanju šprinta, organizacijo retrospektiv in predvsem pomoč pri razreševanju morebitnih težav, na katere naletijo razvijalci (Schwaber in Sunderland, 2020). Bistvo je, da skrbnik metodologije ni tisti, ki ukazuje in dodeljuje naloge, ampak ekipi pomaga in ji daje zagon (Holtzhausen in Klerk, 2018). Tako kot lastnik projekta tudi skrbnik metodologije lahko sodeluje v več timih hkrati ali pa poleg svojega dela prevzame tudi vlogo razvijalca (Schwaber in Sunderland, 2020).

Dogodki Scrum

Pri Scrumu poznamo pet različnih dogodkov: šprint, načrtovanje šprinta, dnevni sestanek (angl. *daily scrum*), ocena opravljenega dela (angl. *sprint review*) in retrospektiva. Šprint je časovno vnaprej določeno obdobje, navadno v trajanju enega meseca, v katerem se odvijejo vsi ostali dogodki in opravi delo. Ko se šprint zaključi, se takoj začne nov šprint. Med šprintom se cilj ne spreminja, lahko pa se po potrebi spremenijo zahteve. V nepredvidljivih situacijah, ki delo otežijo, se lahko tim z lastnikom projekta dogovori o novem obsegu projekta, preostanek pa se nato prenese v naslednje šprinte.

Osnovni obseg dela se določi na načrtovanju šprinta, kjer so prisotni vsi člani tima, lahko pa so prisotni tudi različni svetovalci. Načrtovanje traja največ osem ur, na njem pa udeleženci odgovorijo na vprašanja:

- Kaj se da narediti v tem šprintu? – Določijo število zahtev, ki jih bodo v šprintu razreševali.
- Zakaj je ta šprint pomemben? – Določijo seznam zahtev šprinta in cilj šprinta, ki najbolje poveže zahteve.
- Kako se bo delo opravilo? – Razdelijo zahteve na več manjših zadolžitvev (inkrementov), ki so lahko opravljeni v dnevu ali manj.

Ko se načrtovanje zaključi, steče delo. Razvijalci se med seboj dogovarjajo in opravljajo zastavljene zadolžitve. Vsak dan se ob določeni uri zberejo na dnevnem sestanku. Ta traja 15 minut, njegov namen pa je, da se preveri napredek in ugotovi, ali je morda zaradi novih informacij treba spremeniti zastavljeni načrt. Ko se šprint približuje koncu in je delo opravljeno, sledita še ocena dela in retrospektiva. Ocena opravljenega dela je največ štiri ure trajajoč dogodek, na katerem tim vodstvu predstavi svoj napredek in nove informacije, ki so jih o problemu pridobili med šprintom. Nato sledi pogovor o tem, katere potrebe so še nezadovoljene in kakšne so priložnosti, ki bi se jih lahko izkoristilo v naslednjih šprintih.

Retrospektiva je dogodek, na katerem se člani tima ne osredotočajo na rezultate, pač pa na potek preteklega dela. Poskušajo ugotoviti, kaj je bilo pri posameznikih, komunikaciji, procesih, orodjih ... dobro in kaj slabo. Pregledajo probleme, ki so se pojavili med delom, in kako so jih ali kako jih niso rešili, na koncu pa oblikujejo nekaj predlogov, kako lahko svoje delovanje spremenijo tako, da bodo v prihodnje še bolj uspešni. Retrospektiva traja največ tri ure in predstavlja zaključni dogodek tekočega šprinta (Schwaber in Sunderland, 2020).

Artefakti Scrum

Za razumevanje metode je treba poznati tudi tri artefakte Scrum: seznam zahtev produkta (angl. *product backlog*), seznam zahtev šprinta (angl. *sprint backlog*) in inkremente. Seznam zahtev produkta je po prioritarnem vrstnem redu urejen seznam vseh stvari, ki jih je treba pri produktu doseči. Z njegovo pomočjo tim določi cilj produkta, ki služi kot

usmeritev oz. dolgoročni cilj, h kateremu tim stremi, določi pa tudi seznam zahtev šprinta. Ta seznam nastane pri načrtovanju šprinta in je sestavljen iz zahtev, za katere tim pričakuje, da jih lahko razreši v trenutnem šprintu. S pomočjo seznama zahtev šprinta tim oblikuje cilj šprinta, ki te zahteve najbolje povezuje. Ta mora biti dosežen do konca šprinta. Tudi inkremente določi tim pri načrtovanju šprinta. To so manjše delovne naloge, s katerimi člani tima postopno razrešujejo zadane zahteve in se približujejo cilju šprinta. Pri opisu naloge morajo biti člani posebej pozorni na to, da določijo tudi točne kriterije za to, kdaj se določena naloga šteje kot uspešno opravljena (Schwaber in Sunderland, 2020).

PRIMER UPORABE METODE

V slaščičarni Proustove magdalenice so slaščičarji dobili idejo za recepturo nove torte in jo predstavili vodstvu. Torta se je vodjem zdela okusna, niso pa bili prepričani, da bi se dobro prodajala, saj ima slaščičarna v ponudbi že več podobnih tort. Po pregledu obstoječe ponudbe in analizi trendov so oblikovali seznam zahtev produkta (slika 1). Sestavili so ekipo Scrum, pri čemer je nekdo iz vodstva prevzel nalogo lastnika projekta. Za skrbnika metode so določili nekoga iz kadrovske službe, ki je do zdaj izpeljal že nekaj projektov Scrum, vlogo razvijalcev pa so zasedli trije tržniki, dve osebi iz nabave in eden od slaščičarjev.

Ekipa si je čas šprinta postavila na en delovni teden, tako da se bodo vsi prihodnji šprinti začeli v ponedeljek zjutraj in končali v petek popoldan. Pri načrtovanju šprinta so ugotovili, da bodo v prvem tednu verjetno lahko vsaj delno razrešili zahteve, vezane na personalizacijo, umetniško delo in lokalne sestavine, ostale zahteve pa bodo obravnavali na prihodnjih šprintih. Zahteve produkta so zapisali na seznam zahtev šprinta in jih začeli deliti na inkremente (slika 1). Ko so to naredili, jih je lastnik projekta zapustil, saj je imel veliko drugega dela, zagotovil pa jim je, da bo po telefonu vedno na voljo, če bo treba bolj podrobno razložiti katero od zahtev produkta ali se dogovoriti za nov obseg dela.

Razvijalci in skrbnik metode so se nato sami dogovorili, kako se bodo lotili dela, česa se bodo lotili najprej in kako bodo med seboj komunicirali. Ker skrbnik metode ni bil skrbnik metode v nobeni drugi ekipi, je poleg svoje vloge prevzel še vlogo razvijalca in pomagal tudi na ta način.

Po vrstnem redu, ki so ga določili, so nato razvijalci opravljali naloge, zapisane v inkrementih. Vsakemu članu je bila določena ena barva, s katero je obarval besedilo inkrementa, ki ga je opravljal.

Skrbnik metode je vsak dan ob 10.00 sklical in vodil 15-minutni stojič sestanek, na katerem so razvijalci poročali o svojem napredku in morebitnih težavah, sprejemali pomembne odločitve ter se dogovarjali, če je bilo zaradi spremenjenih okoliščin treba spremeniti plan ali način dela. Poleg tega je skrbnik metode člane spodbujal k sodelovanju, preverjal, ali metoda poteka brez napak, in pri morebitnih sporih nastopal kot mediator med razvijalci.

V petek ob 12.00 so razvijalci in skrbnik metode vodstvu, lastniku projekta in trem glavnim slaščičarjem predstavili delo, ki so ga uspeli opraviti v tekočem šprintu. Uspeli so se dogovoriti za dobro in izvedljivo obliko personalizacije torte z imenom, ki je vidno ob prerezu. Za motiv umetniškega dela so kot simbol lokalnega in slovenskega izbrali čebelo, za dobavo pa so se dogovorili tudi z nekaterimi poslovnimi partnerji. Vseeno jim zadano delo ni povsem uspelo, saj skic niso prejeli še od nobenega izbranega umetnika, prav tako pa še niso uspeli raziskati vseh možnosti dobave. Po izmenjavi mnenj in pogovoru z vsemi udeleženi so se odločili, da neopravljeno delo prenesejo v naslednji šprint, poleg tega pa v naslednji šprint vključijo še zahtevo po pravični trgovini.

Po zaključku predstavitve je skrbnik metode organiziral kratko retrospektivo, pri kateri so člani tima na mizo lepili zelene listke z dobrimi vidiki načina dela v preteklem šprintu in rdeče listke s slabimi vidiki načina dela. Ko so z lepljenjem listkov končali, so jih skupaj uredili v skupine. Pri skupinah rdečih listkov so nato poiskali in na tablo napisali predloge, s katerimi bodo lahko v naslednjem šprintu negativne vidike čim bolj zmanjšali, pri zelenih listkih pa so se pogovorili, kako dobre prakse izvajati še naprej. Po koncu retrospektive se je šprint zaključil, člani tima pa so odšli domov in uživali v zasluženem vikendu. V ponedeljek so se ponovno zbrali na novem načrtovanju šprinta.



Slika 1. Artefakti Scrum

VLOGA PSIHologa

Psiholog lahko pri izvajanju Scruma prevzema tri različne vloge. Zaradi svojih znanj o odnosih, komunikaciji, usmerjanju (glej tudi poglavje Coaching) in mediaciji (glej tudi poglavje Mediacija) je lahko odličen skrbnik metode, uslužni vodja, ki zna ekipo motivirati in podpreti tam, kjer je potrebno. Poleg tega lahko psiholog prevzame vlogo zunanje podpore. V tej vlogi lahko vskoči, ko je treba zgladiti morebitne spore, lahko izvaja različna izobraževanja in delavnice na temo kakovostne komunikacije, organizacije dela, postavljanja ciljev, upravljanja s stresom, podajanja konstruktivne informacije itd.

Lastnikom projektov lahko pomaga pri oblikovanju jasnega seznama zahtev in sodeluje pri načrtovanju šprinta, pri čemer skrbi, da si ekipa ne naloži preveč dela, ki bi pozneje lahko vodilo do povečanega stresa in izgorelosti. Lahko nastopi tudi v vlogi svetovalca, ko organizaciji svetuje, v katere procese je smiselno vpeljati metodologijo Scrum in v katere ne ter ali je morda smiselno vpeljati le posamezne elemente metode. Skrbi lahko tudi za oblikovanje timov, ki bodo skupaj dobro delovali in pri delu prevzemali različne timske vloge.

Zaradi svojih znanj o merjenju lahko vzpostavi ustrezen sistem ocenjevanja uspešnosti dela timov in posameznikov. Predvsem pri retrospektivah lahko svetuje tudi posameznim timom, kako izboljšati svoje delovanje v prihodnjih šprintih, saj ima od ostalih več znanja na področju komunikacije, odnosov, vodenja in timske dinamike ter tako lahko zazna pomanjkljivosti, na katere člani tima niti niso pozorni.

PREVIDNOST IN OMEJITVE PRI UPORABI METODE

Scrum sicer lahko izvaja katerakoli ekipa, vendar je pomembno, da jo vsi člani že v naprej vsaj bežno poznajo. Skrbnik metode mora Scrum zelo dobro poznati in je tisti, ki lastnika produkta in razvijalce vpelje v proces ter jih sprotno uči. Pri vpeljevanju metode tako organizacije pogosto najamejo zunanjega skrbnika metode, ki izобрази vse ostale v organizaciji. Spletna stran scrum.org kot uradna spletna stran metode Scrum skrbi za izobraževanja in dodeljuje različne certifikate, vendar ti za izvajanje niso nujno potrebni.

Kljub potencialnim prednostim, ki jih metoda obljublja, številni avtorji poročajo o težavah pri njenem vpeljevanju v organizacijo (npr. Moe in Dingsøyr, 2008; Cohn in Ford, 2003), saj je pri njej potrebna sprememba organizacijske kulture, predvsem glede hierarhičnih odnosov. Če skrbnik metode npr. namesto podpore in spodbude ekipi dodeljuje naloge in je kritičen do dela razvijalcev, to lahko zavre samoorganizacijo in aktivno soodločanje ekipe, ki ju želimo doseči (Moe in Dingsøyr, 2008). Metoda tudi ne zajema vseh stopenj projektnega managementa, saj izpušča zasnovo projekta, oceno sprejemljivosti izdelka za lansiranje in vzdrževanje oz. podporne storitve (Katayama in Goldman, 2011). Poleg tega zaradi časovnega pritiska, ki ga ustvari šprint, lahko prihaja do večjega števila napak in posledično nižje kakovosti izdelka (Miler in Kajdy, 2018), vendar to nekatera izkušena podjetja že uspešno rešujejo (Green, 2011).

LITERATURA

Abbas, N., Gravell, A. M. in Wills, G. B. (2008). Historical roots of agile methods: Where did “agile thinking” come from? V P. Abrahamsson, R. Baskerville, K. Conway, B. Fitzgerald, L. Morga in X. Wang (ur.), *Agile processes in software engineering*

- and extreme programming. XP 2008. *Lecture notes in business information processing* (zv. 9, str. 94–103). Springer. https://doi.org/10.1007/978-3-540-68255-4_10
- A better way of building products. (b. d.) <https://www.scrum.org/resources/what-is-scrum>
- Baer, M. in Oldham, G. R. (2006). The curvilinear relation between experienced creative time pressure and creativity: Moderating effects of openness to experience and support for creativity. *Journal of Applied Psychology*, 91(4), 963–970. <https://doi.org/10.1037/0021-9010.91.4.963>
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64–69.
- Cohn, M. in Ford, D. (2003). Introducing an agile process to an organization. *IEEE Computer*, 36(6), 74–78. <https://doi.org/10.1109/MC.2003.1204378>
- Cordery, J. L., Morrison, D., Wright, B. M. in Wall, T. D. (2010). The impact of autonomy and task uncertainty on team performance: A longitudinal field study. *Journal of Organizational Behavior*, 31(2/3), 240–258. <https://doi.org/10.1002/job.657>
- Diebold, P., Ostberg, J. P., Wagner, S. in Zendler, U. (2015). What do practitioners vary in using Scrum? V C. Lassenius, T. Dingsøy in M. Paasivaara (ur.), *Agile processes in software engineering and extreme programming. XP 2015. Lecture notes in business information processing* (zv. 212, str. 40–51). Springer. https://doi.org/10.1007/978-3-319-18612-2_4
- Fowler, M. in Highsmith, J. (2001). *The agile manifesto*. <https://agilemanifesto.org/>
- Gonzalez-Mule, E., Courtright, S. H., DeGeest, D. Seong, Y. in Hong, D. (2016). Channeled autonomy: The joint effects of autonomy and feedback on team performance through organizational goal clarity. *Journal of Management*, 42(7), 2018–2033. <https://doi.org/10.1177/0149206314535443>
- Green, P. (2011). Measuring the impact of Scrum on product development at Adobe systems. V *Proceedings of the 44th Hawaii International Conference on System Sciences*. IEEE Computer Society.
- Helping people and teams solve complex problems*. (b. d.) <https://www.scrum.org/about>
- Holtzhausen, N. in de Klerk, J. J. (2018). Servant leadership and the Scrum team's effectiveness. *Leadership & Organization Development Journal*, 39(7), 873–882. <https://doi.org/10.1108/LODJ-05-2018-0193>
- Jobidon, M., Turcotte, I., Aubé, C., Labrecque, A., Kelsey, S. in Tremblay, S. (2017). Role variability in self-organizing teams working in crisis management. *Small Group Research*, 48(1), 62–92. <https://doi.org/10.1177/1046496416676892>
- Katayama, E. T. in Goldman, A. (2011). From manufacture to software development: A comparative review. V A. Sillitti, O. Hazzan, E. Bache in X. Albaladejo (ur.), *Agile processes in software engineering and extreme programming. XP 2011. Lecture notes in business information processing* (zv. 77, str. 88–101). Springer. https://doi.org/10.1007/978-3-642-20677-1_7

- Langfred, C. W. (2005). Autonomy and performance in teams: The multilevel moderating effect of task interdependence. *Journal of Management*, 31(4), 513–529. <https://doi.org/10.1177/0149206304272190>
- Li, J., Moe, N. B. in Dybå, T. (2010). *Transition from a plan-driven process to Scrum: A longitudinal case study on software quality* [prispevek na konferenci]. V *ESEM 2010 – Proceedings of the 2010 ACM – IEEE International Symposium on Empirical Software Engineering and Measurement*. Association for Computing Machinery, Bolzano. <https://doi.org/10.1145/1852786.1852804>
- Liu, W. in Xiang, S. (2018). The positive impact of guilt: How and when feedback affect employee learning in the workplace. *Leadership & Organization Development Journal*, 39(7), 883–898. <https://doi.org/10.1108/LODJ-10-2017-0296>
- Miler, J. in Kajdy, K. (2018). Problems and solutions of software design in Scrum projects. *Annals of Computer Science and Information Systems*, 15, 975–978. <https://doi.org/10.15439/2018f151>
- Moe, N. B. in Dingsøyr, T. (2008). Scrum and team effectiveness: Theory and practice. V P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan in X. Wang (ur.), *Agile processes in software engineering and extreme programming. XP 2008. Lecture notes in business information processing*, (zv. 9, str. 11–20). Springer. https://doi.org/10.1007/978-3-540-68255-4_2
- Mroz, J. E., Yoerger, M. in Allen, J. A. (2018). Leadership in workplace meetings: The intersection of leadership styles and follower gender. *Journal of Leadership & Organizational Studies*, 25(3), 309–322. <https://doi.org/10.1177/1548051817750542>
- Project Management Institute (27. 6. 2018). *Success in disruptive times: Expanding the value delivery landscape to address the high cost of low performance*. https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2018.pdf?sc_lang=temp=en
- Rigby, D. K., Sutherland, J. in Noble, A. (2018). Agile at scale: How to go from a few teams to hundreds. *Harvard Business Review* 96(3), 88–96.
- Rigby, D. K., Sutherland, J. in Takeuchi, H. (2016). Embracing agile: How to master the process that's transforming management. *Harvard Business Review*, 94(5), 40–50.
- Schwaber, K. in Sutherland, J. (2020). *The Scrum guide. The definitive guide to Scrum: The rules of the game*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>
- Stiglbauer, B. (2017). Under what conditions does job control moderate the relationship between time pressure and employee well-being? Investigating the role of match and personal control beliefs. *Journal of Organizational Behavior*, 38(5), 730–748. <https://doi.org/10.1002/job.2165>
- Strode, D. (2006). *Agile methods: A comparative analysis* [prispevek na konferenci]. 19th Annual Conference of the National Advisory Committee on Computing Qualifications (NACCQ 2006), Wellington.

von Bonsdorff, M. E., Janhonenb, M., Zhouc, Z. E. in Vanhalad, S. (2015). Team autonomy, organizational commitment and company performance – a study in the retail trade. *The International Journal of Human Resource Management*, 26(8), 1098–1109. <http://doi.org/10.1080/09585192.2014.934881>

AGILE SRCUM METHODOLOGY

Scrum belongs to a wider group of agile methods aimed at changing the way groups and organizations work so that they can respond more quickly to the many changes that today's market brings. It can be seen as a framework for smaller teams that are trying to achieve a set goal in a predetermined time frame (called a sprint). To understand the scrum method, it is necessary to know all the team roles, events and artefacts that it provides. Team roles include developers whose main task is to work on the project, the project owner, who acts as an intermediary between management and the scrum team and is responsible for creating and clearly communicating the project requirements list, and the method administrator.

Scrum events include a sprint, a sprint planning session in which the developers and project owner agree on the sprint goal and how many requests they will be able to meet in the upcoming sprint, a short daily meeting where the whole team gathers and decides on future actions, an assessment of the work done when the scrum team presents its progress to the management team and agrees on any further work that needs to be done, and a retrospective where team members comment on what they thought was good or bad about their performance in the previous sprint, and decide what they will change in the future. Artefacts are a list of product requirements, a list of sprint requirements, and increments. The list of product requirements is prepared by the business owner in agreement with the management, and includes all the features they want their product to have in the final form. The scrum team then draws from this and compiles a list of requirements when planning a sprint. These are also designed by the team itself when planning a sprint, and represent smaller work tasks that will allow the team members to gradually achieve the set goal during the sprint, i.e. to meet all the set requirements.