



ORGANISAATION MUUTOS SCRUMIIN

Case: Landis+Gyr Enermet

Mikko Marttinen

Opinnäytetyö

Huhtikuu 2008

Liiketalous



**JYVÄSKYLÄN
AMMATTIKORKEAKOULU**

Tekijä(t) MARTTINEN, Mikko	Julkaisun laji Opinnäytetyö	
	Sivumäärä 58	Julkaisun kieli suomi
	Luottamuksellisuus <input type="checkbox"/> Salainen _____saakka	
Työn nimi ORGANISAATION MUUTOS SCRUMIIN Case: Landis+Gyr Enermet		
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma		
Työn ohjaaja(t) BISTER, Timo		
Toimeksiantaja(t) Landis+Gyr Enermet		
Tiivistelmä Tutkimuksen tarkoituksena oli selvittää, kuinka muutos vesiputousmallista ketterään Scrum-menetelmään oli käynnistynyt kohdeyrityksessä Landis+Gyr Enermet. Kohdeyrityksenä toiminut Landis+Gyr Enermet toimi opinnäytetyön toimeksiantajana. Työn teoriaosuus kohdistui ketterään ohjelmistokehitykseen Scrum-menetelmään keskittyen. Scrumin teoriaosuuteen päästiin asteittaisesti vertailemalla ensin eroavaisuuksia vanhan kehitysmenetelmän vesiputousmallin ja ketterän ohjelmistokehityksen välillä. Tämän jälkeen selkeytettiin ketterän ohjelmistokehityksen perusidea, minkä jälkeen voitiin esitellä ketterä Scrum-menetelmä. Scrumin lisäksi käsiteltiin opinnäytetyön toisena teoriaosuutena muutosta. Muutosta käsiteltiin organisaation ja yksilön näkökulmasta. Näiden lisäksi esiteltiin yksilön muutosvastarintaa ja kuuden askeleen Scrum-menetelmän omaksumisprosessi. Tutkimusosuus perustui kohdeyrityksen kahdessa pilottiprojektissa työskennelleiden henkilöiden yksilöhaastatteluista saatujen tulosten arviointiin. Työntekijöitä haastateltiin kahteen otteeseen projektien alku- ja loppuvaiheessa, ja haastattelu toteutettiin avoimin strukturoimattomin kysymyksin. Avoimiin kysymyksiin päädyttiin pienen otannan takia, joka myös määritteli tutkimukselle kvalitatiivisen otteen. Haastatteluista saadut tulokset keskittyvät vesiputousmalliin, Scrum-menetelmään ja muutokseen organisaation sekä yksilön näkökulmasta. Opinnäytetyön tavoitteena oli löytää vastauksia kysymyksiin, oliko muutoksesta Scrum-menetelmään ollut hyötyä organisaatiolle ja kuinka sovelluskehityksen menetelmien kehittämistä voidaan jatkaa. Tutkimuksen tuloksista tuli esiin selkeitä muutoksen mukanaan tuomia hyötyjä, mutta positiivisten asioiden lisäksi myös negatiivisia muutoksesta johtuvia haittavaikutuksia. Tuloksien kautta saatiin lisäksi selvyys yksilöiden ja organisaation sopeutumisesta muutosprosessiin. Yksilöiden tasolla muutos koettiin helpoksi. Organisaatiolle muutos aiheutti kasvukipuja, mutta kivuista ilmeni positiivisiakin piirteitä.		
Avainsanat (asiasanat) Ketterä ohjelmistokehitys, Scrum, Organisaation muutos, Yksilön muutos, Muutosvastarinta		
Muut tiedot Liitteet 5 sivua		

Author(s) MARTTINEN, Mikko	Type of Publication Bachelor´s Thesis	
	Pages 58	Language Finnish
Confidential <input type="checkbox"/> Until _____		
Title ORGANIZATIONAL CHANGE TO SCRUM Case: Landis+Gyr Enermet		
Degree Programme Business Information Systems		
Tutor(s) BISTER, Timo		
Assigned by Landis+Gyr Enermet		
Abstract <p>The aim of this thesis is to define, how the change from waterfall method to agile Scrum method has begun at the subject company, Landis+Gyr Enermet, which was the assigner of the thesis.</p> <p>The theoretical part of this work concentrates on software development focusing on Scrum method. The theoretical part starts by first comparing differences between the old development method, waterfall model, and the new agile software development. After this, the basic idea of Agile software development is clarified followed by a presentation of the agile Scrum method. In addition to Scrum, a second theoretical point of view change is discussed. The change was handled from both organizational and individual viewpoints. In addition to these, individual resistance of change and a six-step imbibe process to Scrum method are presented.</p> <p>The empirical part of this thesis is based on evaluation of the results from individual interviews of two pilot projects, which took place at the subject company. Employees were interviewed twice between the beginning and end phases of the projects with open, not structured questions. The open questions were chosen because of a small sample, which also gave to this research its qualitative touch. Results that were received from interviews concentrated on waterfall model, Scrum method and the change from an organizational and individual point of view.</p> <p>The aim of this thesis was to find answers to questions, whether the change to Scrum has given any advantage to the subject organization and how the development of software developing methods could be continued. The results unfolded clear advantages that had come with the change, but in addition to positive results, change had also unfolded harmful effects. Across the results, transparency of individual and organizational adjustment to change process was received. On individual level, the change was considered to be rather easy. As for the organizational level, change had resulted in some growing pains, but apparent positive features were generated of those pains.</p>		
Keywords Agile software development, Scrum, Organizational change, individual change, Resistance to change		
Miscellaneous Appendices 5 pages		

SISÄLTÖ

1 JOHDANTO	4
2 TUTKIMUSASETELMA	5
2.1 Toimeksiantaja	5
2.2 Tavoitteet ja rajaukset	5
2.3 Tutkimuskysymykset	6
2.4 Tutkimusmenetelmät ja toteutus	7
3 KETTERÄ MENETELMÄ SCRUM	8
3.1 Johdatus vesiputousmallista ketterien menetelmien maailmaan	8
3.2 Ketterä ohjelmistokehitys pähkinänkuoressa	10
3.3 Mikä on Scrum?	11
3.3.1 Arvot	12
3.3.2 Empiirinen prosessin hallinta	14
3.3.3 Avaintoiminnot	16
3.4 Dokumentaatio	17
3.4.1 Tuotteen backlog	17
3.4.2 Sprint backlog.....	18
3.5 Roolit	18
3.5.1 Tuotteen omistaja.....	19
3.5.2 Scrum-mestari.....	19
3.5.3 Tiimi.....	20
3.6 Scrumin prosessimalli	21
3.6.1 Sprint.....	22
3.6.2 Suunnittelupäivä	22
3.6.3 Scrum-palaveri	23

3.6.4 Demo.....	24
3.6.5 Retrospektiivinen katselmus.....	26
3.7 Scrumin hyödyt.....	27
4 MUUTOSPROSESSI.....	30
4.1 Mitä muutoksella tarkoitetaan?	30
4.2 Yksilön muutosvastarinta.....	30
4.3 Siirtyminen Scrumiin.....	31
4.3.1 Organisaation muutos	32
4.3.2 Scrumin omaksumisprosessi	33
5 TUTKIMUKSEN TOTEUTUS JA TULOKSET.....	35
5.1 Tutkimuksen toteutus.....	35
5.2 Tutkimuksen tulokset	37
5.2.1 Aiemmin käytetty vesiputousmalli.....	38
5.2.2 Muutosehdotuksia vesiputousmalliin	39
5.2.3 Odotukset Scrumia kohtaan	40
5.2.4 Haastateltavien tuntemukset muutoksesta Scrumiin	42
5.2.5 Henkilökohtainen sopeutuminen Scrumiin	43
5.2.6 Organisaation sopeutuminen Scrumiin	44
5.2.7 Scrumin tuomien toiminnallisuuksien hyödyllisyys ja hyödyttömyys	45
5.2.8 Organisaation kommunikaatio.....	46
5.2.9 Oma-aloitteisuus projektiryhmissä/tiimissä	48
5.2.10 Muutos vaatimusten hallinnassa.....	49
5.2.11 Onko muutoksesta ollut hyötyä?	51
6 JOHTOPÄÄTÖKSET JA POHDINTA.....	53
6.1 Päätelmiä tuloksista.....	53
6.2 Tutkimusprosessin analysointi.....	55
LÄHTEET	57

LIITTEET	59
Liite 1. Ketterän manifestin käytänteet.	59
Liite 2. Ketterän ohjelmistokehityksen käytänteet.....	60
Liite 3 Aloituskyselyn kysymykset	63
Liite 4 Lopetuskyselyn kysymykset	63

KUVIOT

KUVIO 1. Vesiputousmallin vaihejakomalli	9
KUVIO 2. Projektin monimutkaisuus	15
KUVIO 3 Scrumin prosessimalli	21
KUVIO 4. Alkuhaastattelun henkilömäärä	36
KUVIO 5. Lopetushaastattelun henkilömäärä.....	37

1 JOHDANTO

Ohjelmistokehitysprojekteissa jatkuva epävakaus on arkipäivää. Projektien aikataulut venyvät, budjetit ylittyvät ja kiire kasaantuu projektien loppua kohden. 1970-luvulla kehitetty vesiputousmalli on tarjonnut vuosikymmeniä rungon ohjelmistoprojektien läpiviemiseen, mutta uudet tuulet puhaltavat ohjelmistokehitysalalla. Ketterä ohjelmistokehitys on vuosituhannen vaihteesta saakka kasvattanut suosiotaan kyseenalais- tamalla vanhoja ajatusmalleja ja tarjoamalla uusia näkökulmia sovelluskehitykseen. Vanhoista virheistä on opittu ja nyt tukeudutaan empiiriseen prosessinhallintaan, jossa perusolettamuksena on, että kaikki tehtävät ovat aavistamattomia. Ketterät menetelmät, kuten Scrum, perustuvat vanhoihin, hyviksi todettuihin toimintatapoihin, joita on käytetty jo vuosikymmeniä. Scrum-menetelmää voidaan kutsua yliaktiiviseksi tuottavuustekniikaksi, joka kasaa yhteen kevytrakenteiseen runkoon parhaimmat projektinhallintamenetelmät.

Muutos vesiputousmallista ketterään ohjelmistokehitykseen on pitkäaikainen ja organisaatiota sekä yksilöä ravisuttava muutosprosessi. Muutoksen käynnistyminen vaatii yritykseltä havahtumisen muutostarpeisiin. Muutos tuo mukanaan ongelmia, joita aiemmin ei olisi voitu löytää. Näitä ongelmia organisaatio joutuu kohtaamaan parantaakseen toimintaansa. Yksilötasolla muutokset saattavat aiheuttaa muutosvastarintaa, joka hankaloittaa muutosprosessin etenemistä. Organisaation tasolla muutos saattaa lamaannuttaa koko prosessiketjun hetkellisesti. Kun kasvukivuista on selvitty, päästään vaiheeseen, jossa uusi toimitapa on vakiinnuttanut toimintansa, ja sen mukanaan tuomia hyötyjä voidaan arvioida.

Tämän opinnäytetyön tarkoituksena on tutkia, miten muutos vesiputousmallista Scrumiin on käynnistynyt Landis+Gyr Enermet-yrityksessä. Empiirinen tutkimustyö tehdään haastatteleamalla kahteen Scrum-menetelmää käyttävään pilottiprojektiin kuuluvia henkilöitä. Heidän ajatuksiaan kerätään vanhasta ja uudesta työskentelytavasta sekä muutosprosessista. Tavoitteena on selvittää, onko muutoksesta Scrum-menetelmään ollut hyötyä organisaatiolle ja kuinka sovelluskehityksen menetelmien kehittämistä voidaan jatkaa. Haastatteluiden kautta saadaan lisäksi selvyys yksilöiden ja organisaation sopeutumisesta muutosprosessiin.

2 TUTKIMUSASETELMA

Tutkimusasetelma-luvussa tarkennetaan opinnäytetyön aiheen näkökulmia ja empiirisen tutkimuksen menetelmiä. Aluksi esitellään hyvin lyhyesti opinnäytetyön toimeksiantaja Landis+Gyr Enermet -yritys. Toinen luku selventää opinnäytetyön aiheen rajausta ja tavoitteita sekä esittelee kolme opinnäytetyön tutkimuskysymystä. Lopuksi käsitellään empiirisen tutkimuksen toteutusta ja käytettäviä tutkimusmenetelmiä.

2.1 Toimeksiantaja

Landis+Gyr on vuonna 1896 Sveitsissä perustettu yritys, joka työllistää tällä hetkellä yli 4000 työntekijää 35 maassa. Landis+Gyr on alallaan maailman johtava yritys, joka tarjoaa ylivoimaista mittaustekniikkaa, syvällistä prosessintuntemusta sekä IT-ratkaisuja ja -palveluita mittaustietojen hallinnoinnin automatisointiin. Ydinosaaminen sijoittuu älykkään energiankulutuksen mittaamiseen, jossa yhdistyvät sähköenergiamittarit, lämpö- ja kylmämittarit sekä järjestelmät yhtenäiseksi kokonaisuudeksi. Landis+Gyr Enermet on osa Landis+Gyr organisaatiota. (Tietoa Landis+Gyristä 2006.)

2.2 Tavoitteet ja rajaukset

Tavoitteet

Opinnäytetyön tavoitteena on selvittää, kuinka Landis+Gyr Enermet -yrityksessä alkanut ohjelmistotuotannon muutos Scrum-menetelmään on onnistunut. Tavoitteena on löytää Scrumin hyötyjä ja haittoja sekä vertailla näitä aiempaan työskentelytapaan, vesiputousmalliin. Vertailun pohjalta voidaan löytää toimeksiantajalle sopivia työskentelytapoja tukemaan ohjelmistotuotannon työtapojen kehittämistä tulevaisuudessa ja saadaan tietoa, kuinka organisaatio ja yksilö ovat sopeutuneet muutokseen.

Aiheen rajaus

Opinnäytetyön rajauksen tulisi vastata kysymyksiin, mitä halutaan tietää ja mitä halutaan osoittaa aineistolla (Hirjärvi, Remes & Sajavaara 2004, 75). Ensimmäinen kysymys on, onko Scrum-menetelmän käyttöönotosta ollut hyötyä pilottiprojekteissa. Toinen kysymys on, voidaanko aineistosta löytää ohjelmistotuotannon kehitystä hyödynnäviä asioita. Opinnäytetyön rajauksen ulkopuolelle jäi taloudellisen hyödyn arviointi. Tämän lisäksi aihe rajataan pilottiprojekteihin, joita tutkitaan kokonaisvaltaisesti. Tällöin tutkimuksen painopiste pysyy hyvin laaja-alaisena eikä keskitytä johonkin tiettyyn osa-alueeseen, kuten testaukseen tai kehitykseen.

Aiheen rajaus on vahvasti Scrum-painotteinen. Teoriaosuudessa käsitellään Scrumin menetelmien lisäksi ketterää ohjelmistokehitystä, joka toimii kulmakivenä Scrumin idealle. Ilman ketterää ohjelmistokehitystä ei olisi Scrumia. Teorian toisena pääkohtana on muutos. Muutoksessa painopiste on muutosvastarinnassa sekä organisaatiossa tapahtuvassa muutoksessa siirryttäessä käyttämään Scrum-menetelmää.

2.3 Tutkimuskysymykset

Tutkimuksen lähtökohtana toimii kolme kysymystä, joihin tutkimuksen kautta haetaan vastauksia:

1. Miten yrityksen ohjelmistotuotannon prosessia tulisi kehittää?

Ensimmäinen tutkimuskysymys kuvaa yrityksen tilannetta ennen pilottiprojektien alkamista. Sen tavoitteena on selvittää, mikä aiemmin käytetyssä vesiputousmallissa ei ole sopinut yrityksen toimintaan. Haastatteluiden kautta saadut kehitysideoit tuovat esiin parantamista vaativat kohdat aiemmissa työskentelytavoissa. Tavoitteena on saada kehitysideoita tukemaan Scrum-menetelmää ja parantamaan yrityksen sovelluskehitysprosessia.

2. Kuinka muutosprosessi vanhasta toimintatavasta uuteen on sujunut?

Ensimmäinen tutkimuskysymys tarkastelee aikaa ennen muutosta. Toinen tutkimuskysymys käsittelee muutoksen jälkeistä aikaa. Sen tavoitteena on antaa vastauksia kol-

manteen tutkimuskysymykseen ja tuottaa materiaalia tarkasteltaessa muutosprosessin positiivisia ja negatiivisia puolia.

3. Oliko siirtymisestä Scrum-menetelmän käyttämiseen hyötyä?

Kolmas tutkimuskysymys on tutkimuksen kannalta tärkein kysymys. Vastaukset tähän kysymykseen saadaan toisesta tutkimuskysymyksestä. Tavoitteena on muodostaa selkeä päätelmä Scrumin tuomista hyödyistä sekä haitoista ja vertailla vanhaa tapaa uuteen hyödyn näkökulmasta.

2.4 Tutkimusmenetelmät ja toteutus

Opinnäytetyön empiirinen osio toteutetaan kvalitatiivisena tutkimuksena. Kvalitatiivisen metodologian lähtökohdat sopivat kyseiseen ainutlaatuiseseen tapaukseen paremmin kuin kvantitatiivinen metodologia, koska tutkimuksen otos on rajallinen. Otos rajoittuu toimeksiantajalla marraskuussa 2007 alkaneeseen kahteen Scrumia käyttävään pilottiprojektiin. Kohdejoukko on tarkoituksenmukainen, pilottiprojekteissa mukana olevat henkilöt. Tutkimuksen toteutuksessa on tavoitteena kokonaisvaltainen tiedonhankinta, jossa käytetään induktiivista analyysia.. (Hirjärvi, Remes & Sajavaara 2004, 155.)

Yksilöhaastatteluissa käytetään strukturoimattomia survey-kyselyn omaisia avoimia kysymyksiä, jotka sallivat haastateltavien ilmaista itseään omin sanoin. Avointen kysymysten hyvänä puolena on, että vastauksia ei ehdoteta etukäteen. Vastaukset osoittavat tällöin tietämystä aiheesta ja asiaan liittyvien tunteiden voimakkuutta. Haastateltavalle itselleen tärkeimmät asiat tulevat myös herkemmin esiin. (Hirsjärvi ym. 2004, 190.)

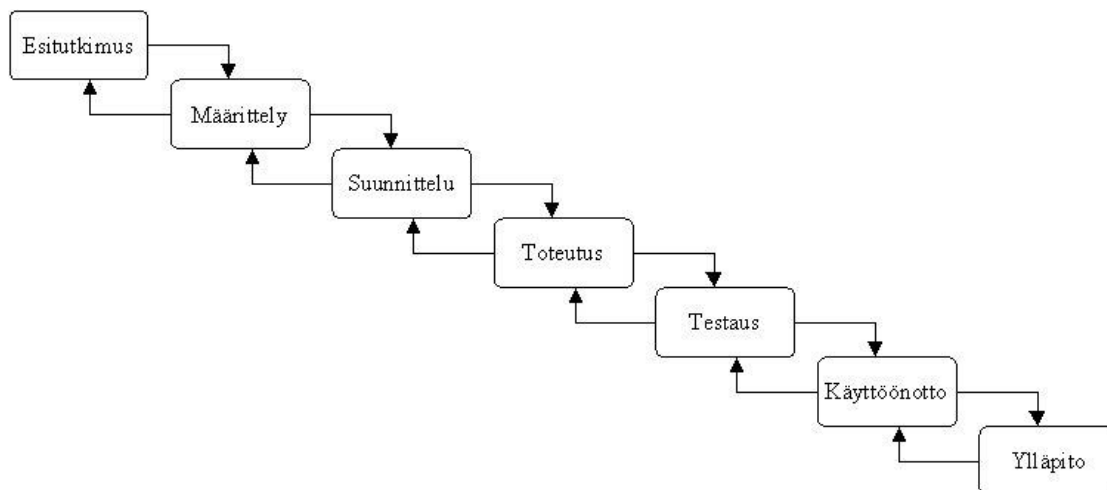
3 KETTERÄ MENETELMÄ SCRUM

Tässä luvussa käydään läpi opinnäytteen laajin teoreettinen pohja, jonka muodostaa ketterä menetelmä nimeltään Scrum. Jotta Scrumin ideologian pystyisi hahmottamaan, tulee ymmärtää perusidea ketterästä ohjelmistokehityksestä (Agile Development).

Luku 3.1 on johdatus vesiputousmallista ketteriin menetelmiin. Ketteristä menetelmistä jatketaan luvussa 3.2, joka pitää sisällään perusidean lisäksi arvoja ja käytänteitä kyseisestä menetelmästä. Luku 3.3 on ensimmäinen, joka käsittelee Scrumia. Kyseisessä luvussa käydään läpi Scrumin historiaa, arvoja ja käytänteitä sekä esitellään empiirinen prosessin hallinta. Scrumin perusidean jälkeen esitellään luvuissa 3.4. - 3.6 Scrumin dokumentaatio, roolit ja prosessimalli. Luku 3.7 päättää Scrumin teoriaosuuden esittelemällä Scrumin hyötyjä yrityksen näkökulmasta.

3.1 Johdatus vesiputousmallista ketterien menetelmien maailmaan

Winston Royce esitteli 1970-luvulla ohjelmistokehityksen vesiputousmallin. Vesiputousmalli on niin sanottu vaihejakomalli, eli se jakaantuu useisiin eri vaiheisiin. Mallista on tehty muunnelmia, joissa painotetaan tekijän mukaan esimerkiksi ylläpidon, testausten tai suunnittelun vaiheita. Yleispätevänä vaihejakona voidaan kuitenkin pitää määrittely-, suunnittelu- ja toteutusmallia. (Leffingwell 2007, 17; Haikala & Märijärvi 2002, 35–37.)



KUVIO 1. Vesiputousmallin vaihejakomalli (Royce 1970, 330)

Winston Roycen poika Walker Royce huomioi, että hänen isänsä kehittämä vesiputousmalli on väärinymmärretty. Roycen mukaan vesiputousmalli tulisi ottaa käyttöön vasta, kun tunnetaan perinpohjaisesti asiakkaan vaatimukset ja ymmärretään teknologian ydin. Winston Royce oli myös painottanut ketteriä käytänteitä, jotka oli jätetty täysin huomiotta. (Leffingwell 2007, 17.)

Vesiputousmallin runko perustuu Leffinwellin (2007) mukaan neljään olettamukseen, joka ovat osoittautuneet virheellisiksi:

1. On olemassa tarpeeksi hyvin määriteltyjä vaatimuksia, mikäli niiden ymmärtämiseen on käytetty aikaa.
2. Muutokset ovat pieniä ja hallittavia.
3. Järjestelmän integraatio onnistuu hyvin.
4. Aikataulut pitävät. (Mts. 20–23.)

Ketterä ohjelmistokehitys perustuu vastakohtaisiin oletuksiin, toisin kuin aiemmin esitellyt vesiputousmallin oletukset. Seuraavassa Leffinwellin (2007) neljä ketterän menetelmän perusolettamusta:

1. Ei oleteta, että yksikään osapuoli ymmärtäisi täysin vaatimuksia.
2. Ei oleteta, että muutokset ovat pieniä ja hallittavia, vaan oletetaan muutostyön olevan jatkuvaa ja toimitettavien lisäyksien pieniä, paremmin jäljitettäviä muutoksia.

3. Oletetaan, että järjestelmän integraatio on riskejä vähentävä tärkeä toiminnallisuus. Siksi integraatio on jatkuvaa ja järjestelmä toimii aina.
4. Ei oleteta, että pystytään kehittämään uutta toiminnallisuutta vanhan toiminnallisuuden päälle kiinteässä aikataulussa. Sen sijaan oletetaan, että voidaan toimittaa tärkeimmät toiminnallisuudet nopeasti. Tällöin palautteen kautta löydetään oikeat ratkaisut ja aikaa säästyy kehittämiseen. (Mts. 26–27.)

3.2 Ketterä ohjelmistokehitys pähkinäkuoressa

Ketterä ohjelmistokehitys on mullistanut lähestymistavan sovelluskehitykseen (Lefingwell 2007, 5). Ketterä ohjelmistokehitys on metodi, jossa luotetaan tiimin jäsenten työhön, heille annetaan tarpeeksi valtuuksia, kannustetaan jatkuvaan palautteeseen ja tunnustetaan muutosten olevan käyttäytymissäntö pikemmin kuin harvinaisuus. Ketterän ohjelmistokehityksen käytänteet eivät ole uusia, vaan ne ovat olleet vuosikymmeniä sovelluskehitysprojektien käytänteitä. Näistä käytänteistä on muodostettu uudellinen lähestymistapa, jossa tavoitteena on käyttää hyväksi todettuja menetelmiä paremman sovelluksen rakentamiseksi. (Schuh 2005, 2.)

Ketterä ohjelmistokehitys luo ytimen useille ketterille menetelmille, jotka ovat nousseet 30 vuoden aikana tyytymättömyydestä sovelluskehitykseen. Ketterä ohjelmistokehitys antaa menetelmille, kuten Scrum ja XP (Extreme Programming), perustavanlaatuisen ytimen hyväksi todettuja toimintoja. Menetelmät toimivat kehikkona ketterän ohjelmistokehityksen yläpuolella. Ketterän ohjelmistokehityksen menetelmät ovat kaikki samankaltaisia, mutta lähtöisin eri näkökulmista. Scrum ja ASD (Adaptive Software Development) keskittyvät hallintaan sekä kommunikaatioon, FDD (Feature Driven Development) tiimin prosessiin ja XP ohjelmointi- ja kehitystapoihin. (Mts. 2–3.)

Arvot ja periaatteet

Vuonna 2001 ketterän ohjelmistokehityksen sovelluskehitysmenetelmien kehittäjät kokoontuivat yhteen ja loivat ketterälle ohjelmistokehitykselle arvot, jotka julkaistiin ”Agile Manifesto”-ssa. Kyseinen manifesti toimii kulmakivenä kaikille ketterän ohjelmistokehityksen menetelmille ja on ollut suurena apuna yhdistettäessä ketteriä me-

netelmiä.(Leffingwell 2007, 9.) Seuraavaksi esitellään vapaasti suomennettuna Agile Manifestin neljä perusarvoa. Nämä perusarvot eivät ole toisiaan poissulkevia, vaan ne pyrkivät määrittelemään ketterän ohjelmistotuotannon painoarvoja.

- “Yksilöitä ja vuorovaikutusta enemmän kuin prosesseja ja työkaluja”
 - ”Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota”
 - “Yhteistyötä asiakkaan kanssa enemmän kuin sopimusneuvotteluita“
 - “Muutoksiin vastaamista enemmän kuin suunnitelman noudattamista””
- (Beck, K. & Beedle, M. & Bennekum, A, Van etc 2001. Agile Manifesto.)

Ketterän manifestin lisäksi määriteltiin myös ketterien menetelmien perustana olevat periaatteet, jotka on esitelty liitteessä 1 (Mts. 9–10).

Käytänteet

Ketterän ohjelmistokehityksen käytänteet on tarkoitettu käyttöön otettaviksi ja tiimeissä muokattaviksi. Ne auttavat tiimiä nopeampaan ja laadullisempaan tuottavuuteen. Parhaiden käytänteiden löytämiseen ei ole yksiselitteistä ohjetta. Tiimin tulee itse arvioida ja tarkkailla, mitkä käytänteet sopivat sen toimintaan. Arvion pohjalta tiimin tulee muokata hyviksi koetut toiminnot paremmiksi ja poistaa huonot toiminnot. Alkutilanteessa voidaan kirjata ylös projektissa käytettävät käytänteet, muokata niitä ja lopulta hyväksyä tai hylätä ne kokemuksen perusteella. (Schuh 2005, 62–65.)

Peter Schuh (2005) on koonnut kirjaansa ketterän ohjelmistokehityksen käytänteiden listan. Nämä käytänteet vaativat aikaa, jotta ne opitaan, niihin sopeudutaan ja ne saadaan viimeistelyä tarpeiden mukaisesti. (Mts. 69–70.) Käytänteet on esitelty liitteessä 2.

3.3 Mikä on Scrum?

Schwaber ja Beedle (2002) ovat sanoneet Scrumista, että se on kevytrakenteinen ketterä projektin hallinnan menetelmä, joka perustuu pienikokoiseen, päätäntävaltaiseen, itsestään organisoituvaan tiimiin (Leffingwell 2007, 41). Scrum voidaan kuvailla myös yliaktiivisena tuottavuustekniikkana, joka jättää kauas taakse suositut ja kalliit

trendit. Terminä Scrum ei ole akronyymi, vaan se viittaa tekniikkaan rugbyssa, jossa pallo saadaan hankalasta tilanteesta takaisin peliin. (Schwaber 2006.)

Scrumin ideana on luoda jokaisen iteraation aikana potentiaalisesti valmista toiminnallisuutta annettavaksi asiakkaalle. Siinä yhdistellään useita erilaisia, hyväksi havaittuja toimintoja tuottavuuden maksimoimiseksi. Sen ominaisuuksina on tunnistaa nopeasti projektia hidastavat esteet, maksimoida yhteistyötä, parantaa kommunikaatiota, ohjailla prosessina kiinnostuksen ja tarpeen välistä kaaosta ja saada ihmiset tuntemaan hyvää mieltä omasta työstään. (Schwaber 2006.)

Scrumin historia

Scrumin filosofiset juuret ovat kaksi vuosikymmentä taaksepäin Japanissa. Takeuchi ja Nonaka olivat ensimmäiset, jotka esittelivät nopeutetun tavan tuotekehityksen prosessin kehittämiseen vastauksena globalisaation aiheuttamaan tarpeeseen kasvattaa kilpailukykyä. (Leffingwell 2007, 42.)

Vuonna 1993 Sutherland ja Schwaber kehittivät suurimman osan Scrumin toiminnoista. Scrum esiteltiin ensimmäisen kerran virallisesti OOPSLA'96 (The ACM Conference on Object-Oriented Programming Systems, Languages and Applications) -tapahtumassa. (Leffingwell 2007, 13–14.) Scrumin virallisen julkistuksen jälkeen vuonna 1996 sen kehittäjät ovat integroineet Scrumin tuhansiin projekteihin ja organisaatioihin ympäri maailmaa. Scrumista on muodostunut vakavasti otettava vaihtoehto vanhoille ohjelmistokehityksen näkökulmille. (Schwaber & Beedle 2002, 19.)

3.3.1 Arvot

Scrum sisältää viisi perusarvoa, joihin sen toiminnat perustuvat. Nämä perusarvot saavat voimansa Scrum-projektien jäsenistä. Uudet teknologiat ja monimutkaiset vaatimukset saavat tiimit ajautumaan hankaliin tilanteisiin, joista niiden on noustava itse, ilman ulkopuolista apua. Päämäärän saavuttaminen on kompromissien ja umpikujien takana, joten tiimin on oltava päämäärätietoinen ja sitoutunut velvoitteisiinsa. (Schwaber & Beedle 2002, 147.)

Sitoutuminen

Scrum määrittelee toimintoja, jotka tukevat ja kannustavat sitoutumiseen. Tiimeillä on täysi päätösvalta, kuinka vaatimus toteutetaan. Tämä on täysin uudenlainen lähestymistapa useimmissa organisaatioissa ja se ajaa ihmiset entistä luovemmiksi, koska heille ei sanota, mitä tai miten heidän täytyy tehdä työnsä. Kun tiimin jäsenet saavat valtuudet tehdä muutoksia ideoidensa kautta, he alkavat luottaa enemmän itseensä ja johtoon. Tällöin sitoutuminen on helpompaa, koska työssä näkyy oma kädenjälki eikä vain johdon ohjailu. (Mts. 148.)

Avoimuus

Schwaber (2002) painottaa, että kaikkien projektiin liittyvien asioiden tulee olla näkyvillä ja avoimia. Avoimuus ja näkyvissä oleminen ovat perusedellytys Scrumin toimimiselle. Scrum estää kyvyn salata asioita, koska kaikki sen toiminnat ovat avoimia kaikkien tarkasteltaviksi. (Mts. 151.)

Keskittyminen

Epävakaan tekniikan ja epämääräisten vaatimusten kanssa työskentely arvokkaan tuotteen saamiseksi on hankalaa työtä. Se vaatii tekijältään tarkkaavaisuutta ja keskittymistä. Ratkaisujen löytämisen ongelmiin tulee olla jatkuvasti tiimin jäsenten ajatuksissa työn aikana. Häiriötekijät ovat suuri ongelma, koska ne saavat henkilöiden ajatukset pois ongelmien ratkaisemisesta. Scrum tarjoaa keskittymiseen häiriöttömät puitteet, mikä mahdollistaa ongelmien ratkaisemisen. Kun annetaan mahdollisuus häiriöttömään työskentelyyn, ajan myötä keskittyminen pääasiaan pinttyy jäsenten mieleen. (Mts. 149–150.)

Rohkeus

Kun vanhat työskentelymallit ovat yhä näkyvissä, tulee olla rohkea ja toimia uudella tyylillä. Scrumissa rohkeus ei ole sankaruutta, vaan se on rohkeutta tehdä parhaansa. Vaatimusten toteuttamiseen sitoutuminen saattaa vaatia itsepäisyyttä ja sisukkuutta. (Mts. 153.)

Arvonanto

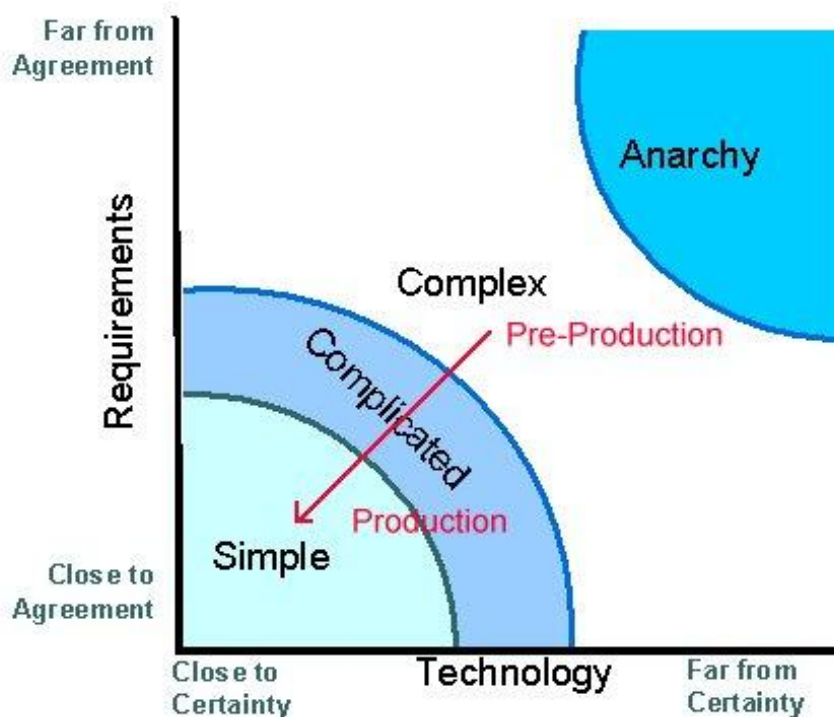
Dynaamisen tiimin voima nousee sen jäsenistä. Johdon järjestämät henkilöt tiimin jäseniksi ovat parhaimmat saatavilla olevat henkilöt, joiden osaaminen taso vaihtelee henkilöittäin. Jokainen ryhmän jäsen on yksilö, ja heillä on omat vahvuutensa ja heik-

koutensa. Scrum antaa tiimin jäsenille valtuuden itse valita, mitä työtä he tekevät oman osaamisensa pohjalta. Täytyy siis oppia arvostamaan muiden tekemistä ja autamaan tarvittaessa, koska vahvuudet ja heikkoudet ovat tiimin yhteisiä. (Mts. 152–153.)

3.3.2 Empiirinen prosessin hallinta

Ohjelmistoala on täysin ennalta arvaamaton ja monimutkainen rakenteeltaan. Muutos on jatkuvaa ja alaa kuvastaa työntekijöiden tietotaitojen suuri vaihtelevuus, vaatimusten muuttuminen kesken projektin, uusien tekniikoiden käyttöön ottaminen sekä järjestelmien rikkoutuminen yhden bitin takia. (Leffingwell 2007, 45.)

Täysin lineaarisessa ohjelmistokehityksessä teknologia, henkilöt ja vaatimukset kulkevat käsi kädessä. Kun vaatimukset ovat projektin toteuttajille selkeät ja heillä on täydellinen kuva siitä, kuinka kehitys tapahtuu käytössä olevien teknologioiden kautta, syntyy vähän virheitä ja uudelleen tekemistä. Kun puolestaan vaatimukset eivät ole selkeitä ja uusia teknologioita käytetään hyväksi kilpailukyvyyn kasvattamiseksi, syntyy enemmän arvaamattomuutta ja työ hankaloituu. Luotettavuus ja varmuus on vaihdettu nopeassa kehityksessä kilpailukyvyyn parantamiseen sekä uusien tekniikoiden hyödyntämiseen, ennen kuin joku toinen ehtii vallata kyseisen idean. Tämä tekijä saa aikaiseksi sen, että kehitystiimit työskentelevät uusien teknologioiden kanssa, vaatimukset muuttuvat jatkuvasti ja aikataulu on tiukka, jolloin tiimillä on hankala keskittyä muutosten kourissa ja edetä työssään eteenpäin. (Schwaber & Beedle 2002, 91–93.)



KUVIO 2. Projektin monimutkaisuus (Clinton 2006)

Kuvio 2, Projektin monimutkaisuus, selvittää, kuinka paljon ongelmatekijöitä ohjelmistokehitysprojekti voi tuottaa. X-akselilla on kuvattu vaatimusten selvyys. Mikäli vaatimukset ovat selkeät, myös kehitys on selkeää. Y-akselilla on kuvattu teknologian tunnettuus projektin jäsenten keskuudessa. Mikäli teknologia on ennestään tuttua, se auttaa ryhmää saavuttamaan projektin tavoitteet. Mikäli käytettävästä teknologiasta ei ole selkeää ymmärrystä ja vaatimukset ovat epämääräisiä, projektit ajautuvat anarkiaan. Kuvioista voidaan ymmärtää, kuinka tärkeä merkitys vaatimuksilla ja käytettävän teknologian tuntemuksella on projektin onnistumisen kannalta. Kun varmuusaste on korkea sekä vaatimusten ja teknologian osalta, voidaan olla lähes varmoja projektin onnistumisesta. (Mts. 93.)

Scrum tukeutuu empiiriseen prosessin hallintaan, jossa perusolettamuksena on, että kaikki tehtävät ovat aavistamattomia. Kyky muuntautua tilanteen mukaan ja jatkuvasti arvioida tilanteita uudestaan korostuu empiirisessä prosessissa. Seuranta ja palaute auttavat tunnistamaan ongelmat. Tällöin projektiryhmä pystyy kykyjensä mukaan toimimaan tilanteen mukaisesti jokaisessa arvaamattomassa tilanteessa. (Mts. 100–101.)

Empiirisessä lähestymisessä tärkeitä kohtia ovat läpinäkyvyys, sopeutuminen ja tarkastelu. Läpinäkyvyys havainnollistaa ongelmat, tarkastelulla arvioidaan projektin tuloksia ja sopeutuminen auttaa muutosten hyväksymiseen läpinäkyvyyden ja tarkastelun kautta. (Leffingwell 2007, 45.)

3.3.3 Avaintoiminnot

Scrumin avaintoiminnot perustuvat Leffingwellin (2007) mukaan yhdeksään pääkohtaan:

1. Sprintin pituus on 30 päivää. Sprintin aikana tuotetaan testattua koodia, joka voidaan antaa asiakkaalle.
2. Scrumissa on määritelty tiukasti aikarajoitteita. Kaikki sen tapahtumat, kuten päivittäinen Scrum-palaveri, demo- ja suunnittelupäivä, ovat ennalta määrätyn pituisia.
3. Päivittäinen 15 minuutin Scrum-palaveri on pääasiallinen kommunikointitapa.
4. Tuotteen omistaja määrittelee vaatimukset. Hän on erottamaton osa tiimiä ja hänen vastuullaan on olla liitoksena tiimin ja asiakkaan välillä.
5. Sprintin aikana toteutettaviin vaatimuksiin eivät saa lisätä toiminnallisuuksia muut kuin tiimin jäsenet.
6. Kaikki työ tehdään annettujen vaatimusten perusteella, jotka määrittellään tuotebacklogissa.
7. Scrum sallii vaatimusten toteuttamisessa tulevien asioiden selkenemisen projektin aikana.
8. Ristikkäistoiminnalliset, enintään kahdeksan hengen tiimin jäsenet kehittävät tuotetta sprintissä.
9. Scrum-mestari ohjaa ja tukee itsestään organisoituvaa tiimiä, joka on vastuussa jokaisen sprintin toimituksesta. (Mts. 44–45.)

3.4 Dokumentaatio

Scrumin dokumentaatio pitää sisällään kaksi dokumenttia, jotka ovat tuotteen backlog (product backlog) ja sprint backlog. Niiden tarkoituksena on toimia projektin seurannan apuvälineenä ja tukevinä tietolähteinä. Projektin statuksen raportointi tapahtuu päivittäisellä jäljellä olevan työn arvioinnilla tunneissa. Schuh on todennut kuitenkin, että tiimin jäsenet eivät ole vastuussa hallinnasta, vaan heidän arviointinsa tapahtuu demossa esitettävien toiminnallisuuksien mukaan. (Schuh 2005, 159–160.)

3.4.1 Tuotteen backlog

Niin kauan kuin tuote on olemassa, on olemassa myös tuotteen backlog. Se on dokumentti, joka pitää sisällään kaikki toiminnallisuudet, ominaisuudet, tekniikat, riippuvuudet ja ongelmien korjaukset, jotka kuvastavat kyseisen tuotteen muutoksia. Nämä vaatimukset ovat saaneet alkunsa tarpeesta kehittää tuotetta. Vaatimuksien lähteinä toimivat asiakkaat, markkinointi, myynti, kehitys ja asiakaspalvelu. (Schwaber & Beedle 2002, 33.)

Tuotteen backlogin kasaamisesta vastaa yksi henkilö, tuotteenomistaja. Hänen tehtävänä on pitää huolta, että dokumentti on ajan tasalla ja priorisoituna tuotteen kehityksen kannalta oikein. Yhden ihmisen asettaminen vastuuseen koko dokumentista on Scrumin periaatteiden mukainen päätös. Tällöin vältetään mahdolliset konfliktit useiden henkilöiden kesken. Organisaation tulee kunnioittaa tämän henkilön päätöksiä, koska tuotteen backlogin vaatimusten priorisoinnit ovat todella näkyviä ja vaikuttavat projektin työskentelyyn ja tuotteen kehitykseen. (Mts. 34.)

Scrumissa tuote-backlogiin lisätään vaatimuksia samaan tapaan kuin toisessa ketterässä menetelmässä nimeltään XP. Tämä tapa on kerätä tuote-backlogiin kaikki tuotteen vaatimukset, joita tuote ei vielä sisällä. (Leffingwell 2007, 220.) Tuote-backlogin tulisi sisältää vaatimusten avaamisen lisäksi yksilöllinen tunniste jokaiselle vaatimukselle, arvio toteutuksen kestosta päivissä, tuotteenomistajan laatima priorisointijärjestys, ohjeistus, se, kuinka vaatimus demotaan, ja muistiinpanoja. (Kniberg 2006, 10.)

Tuotteen omistajan taustasta riippuen vaatimukset ovat tyyliltään business- tai teknillisesti orientoituvia. Kniberg (2006) esittää kirjassaan esimerkin teknillisestä vaati-

muksesta ”Add indexes to the Event table”, joka ei kuvasta itse tavoitetta ”speed up the search event form in the back office”. Indeksit eivät saata olla ratkaiseva vaihtoehto nopeuden kasvattamiseen. Tästä syystä teknillinen osaaminen tulisi jättää tiimille, joka implementoi vaatimuksen, ja tuotteen omistajan tulisi keskittyä business-näkökulmaan. (Mts. 11.)

3.4.2 Sprint backlog

Sprint backlog pitää sisällään vaatimukset, jotka tiimi valitsi toteutettaviksi suunnittelupäivän ensimmäisessä vaiheessa tuotteen omistajan kanssa tuote-backlogista. Jokainen vaatimus arvioidaan pienemmiksi paloiksi tehtäviä, joille arvioidaan kesto neljästä kuuteentoista tuntiin. Tiimin jäsenten tehtävänä on sprintin aikana kirjata sprint backlogiin jäljellä olevat tunnit jokaiseen vaatimukseen. (Schwaber & Beedle 2002, 71.)

Sprint backlog toimii seurannan välineenä tiimille ja Scrum-mestarille. Tämän dokumentin kautta pystytään selvittämään, kuinka paljon on jäljellä toteutettavia kohtia ja kuinka kauan niihin kuluu aikaa. Mikäli arviot ovat olleet liian suuret tai esteet ovat vieneet työajasta sprintin aikana tehokasta työaikaa, huomataan tämä nopeasti sprint backlogista. Seurannan helpottamiseksi on määritelty sprint backlog -diagrammi (burndown chart). Siinä summataan sprint backlogin arvioidun työmäärän kokonaiskesto ja verrataan sitä toteutuneeseen, sen hetkiseen arvioon jäljellä olevasta ajasta graafisesti. (Mts. 71–72.)

Toisin kuin product backlog, sprint backlog on avoin tiimin muokattavaksi. Tiimi on määritellyt, mitä toimenpiteitä sen tarvitsee tehdä toteuttaakseen tuotteen omistajan haluaman vaatimuksen. Tiimillä on täysi oikeus muuttaa työmääräarvioitaan omaloitteisesti. (Mts. 71–72.)

3.5 Roolit

Scrumissa jaotellaan roolit kolmeen eri ryhmään: tuotteen omistaja, Scrum-mestari ja -tiimi. Nämä kolme ryhmää jaotellaan yleisesti myös leikkisesti ”kanoihin ja sikoihin”. Sikojen eli Scrum-mestarin ja -tiimin tehtävänä on tuottaa toimituskelpoinen ratkaisu sprintin aikana. Scrum-prosessissa tulee ottaa kanojen, esimerkiksi asiakkaan,

osakkeenomistajan tai arkkitehdin, tarpeet, ideat tai vaikutteet huomioon. Kanat tulee pitää sivussa itse projektin aikana toteuttavasta osapuolesta. Tuotteen omistajan tehtävänä on olla kanojen ja sikojen välissä. Hän tuo esiin kanojen tarpeet ja toimii toiseen suuntaan sikana tuoden toteuttavalta osapuolelta tietoa eteenpäin. (Bird 2006.) Luvuissa 3.5.1–3.5.3 esitellään jokaisen kolmen roolin lähtökohtia alkaen tuotteen omistajasta, jonka jälkeen seuraa Scrum-mestarin ja viimeisenä muttei vähäisimpänä tiimin esittely.

3.5.1 Tuotteen omistaja

Tuotteen omistaja on henkilö, joka on virallisesti vastuussa koko projektista (Schwaber 2002, 34). Hän on vastuussa pääoman tuottoasteesta investoinnissa (ROI). Kyseisen investointi on kehittää tuotetta. Tuotteen omistajan tavoitteena on saada projektista toimituskelpoinen, suurempaa arvoa tuottava tuote, kuin mitä alkuinvestointi on ollut. (Bird 2006.)

Tuotteen omistaja on yhteydessä eri tuotteeseen vaikuttaviin tahoihin päättääkseen tuotteen kehittämissuunnan. Kyseinen henkilö kasaa yksin vaatimukset tuotteen backlogiin, jotka hän on saanut kehitysideoiden kautta esimerkiksi asiakkailta tai osakkailta. (Leffingwell 2007, 107.) Tuotteen omistajan tehtävänä on esittää eri tahoilta saadut vaatimukset tuotteen backlogin muodossa tiimille, joka suorittaa vaatimusten tuotteistamisen. Hän on ainut henkilö, joka pystyy muuttamaan tuotteen backlogin sisältöä ja joka on vastuussa vaatimusten priorisoinnista sekä tuotteen backlogin saatavuudesta. Tuotteen omistajan tehtävä on hyvin näkyvä, koska prioriteetin muutokset vaikuttavat suoraan tiimin työskentelyyn. (Schwaber & Beedle 2002, 34.) Leffingwell määrittelee esiteltyjen kohtien lisäksi tuotteen omistajalle kaksi erillistä tehtävää. Tuotteen omistajan tulee olla jatkuvasti saatavilla tarkentamassa vaatimuksia. Toisena kohtana Leffingwell määrittelee, että tuotteen omistajan tulee olla läsnä toiminnallisuuden katselmoinnissa. (Leffingwell 2007, 107.)

3.5.2 Scrum-mestari

Scrum-mestari johtaa projektitiimejä, eli hän toimii laajennetun merkityksen saavan projektipäällikön roolissa. Kyseinen termi on uusi johdon rooli, joka esiteltiin Scrumissa. Scrum-mestari toimii johdon, asiakkaan sekä itseohjautuvan tiimin välillä.

Ryhmien välillä toimiessaan Scrum-mestari kiinnittää ja varmistaa ryhmien toimintaa sekä vaalii Scrumin sääntöjä ja arvoja. (Schuh 2005, 24; Schwaber & Beedle 2002, 31–32.)

Scrum-mestari edustaa johtoa ja tiimiä. Johdon puolelta tarkasteltuna Scrum-mestari seuraa päivittäin projektin etenemistä ja raportoi eteenpäin johtoryhmälle. Tiimipuolella hän arvioi projektin nopeutta, reagoi nopeuden hidastuessa, selvittää, mikä hidastuttaa tiimin työskentelyä ja poistaa esiin tulleet esteet. Näiden roolien yhteensovittaminen voi olla hankalaa, koska hänen työnsä on todella näkyvää organisaatiossa. Projektin nopeutta hidastavien esteiden, ”impedimenttien”, esiin tuominen tuo esille organisaation menettelytavat, rakenteen ja välineet, jotka vahingoittavat tiimin tuottavuutta. Ongelmien tunnistamisella pystytään vaikuttamaan ja poistamaan esteet. (Schwaber & Beedle 2002, 31–32.)

Scrum-mestarin roolina on järjestää päivittäinen Scrum-palaveri, jossa hän voi tehdä nopeita päätöksiä vähäisilläkin tiedoilla. Tällä tavalla toimittaessa tiimi voi jatkaa työskentelyään päätökseen nojautuen, vaikka päätös myöhemmin peruttaisiinkin. Muita Scrum-mestarin työtehtäviä ovat tuotteen omistajan ja tiimin kanssa suunnittelupäivän läpiviennissä toimiminen sekä demon ja retrospektiivisen kokouksen järjestäminen sprintin lopuksi. (Mts. 32.)

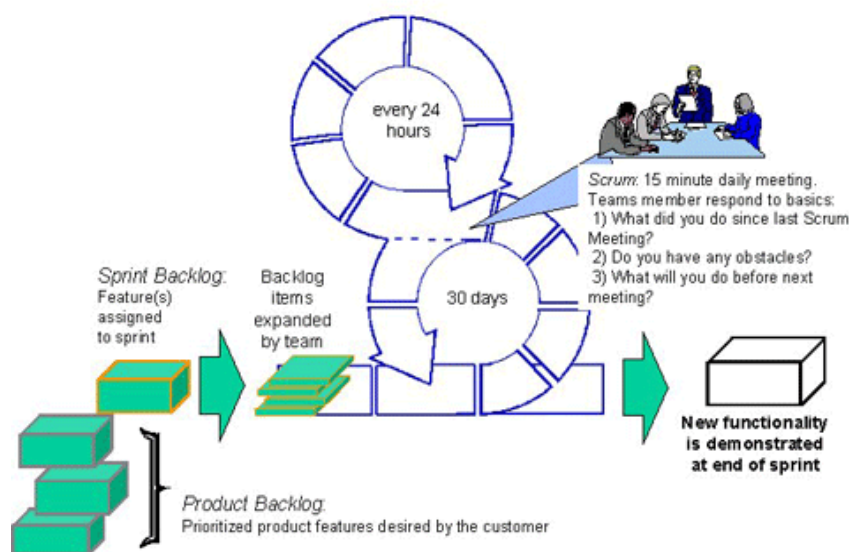
3.5.3 Tiimi

Tiimin rooli on korvaamaton, koska Scrum ohjaa projektin onnistumisen tai epäonnistumisen täysin tiimin harteille. Tiimiin kuuluvat henkilöt, jotka toteuttavat toiminnallisuudet alusta loppuun saakka. Yksittäisiä työnimikkeitä voivat olla esimerkiksi kehittäjä, testaaja, analysoija, tekninen kirjoittaja ja kaikki muut henkilöt, jotka ovat mukana kehityksessä. Tiimin jäsenten määrä tulisi olla suosituksen mukaan 5–9 jäsentä per tiimi. (Schuh 2005, 23; Kniberg 2006, 101.) Scrum-tiimin jäsen on kommunikoiava, avoin ja yhteistyökykyinen henkilö, joka ei pelkää ottaa vastaan haasteita ja on valmis sitoutumaan tehtäväänsä yhteisen projektin onnistumiseksi. Yksin työskentelevät hiljaiset henkilöt eivät sovellu Scrumiin, koska prosessi saa voimansa keskustelusta, tiedon jakamisesta, halusta oppia uusia asioita ja sopeutua uusiin tilanteisiin. (Schuh 2005, 9.)

Scrum-tiimin tärkein arvo syntyy tiimin organisoitumisesta ja hallinnoinnista, roolien ylittämistä sekä itsensä ylittämistä. Organisoitumisella ja hallinnoinnilla tarkoitetaan tässä tapauksessa vapautta valita omat tehtävänsä ilman hallinnon puuttumista asiaan. Roolien ylittäminen tarkoittaa sitä, että tiimit tunnistavat itse jäsenistään ominaisuudet, joiden kautta se muodostaa järjestyksen toimintaansa. Jäsen, joka osaa parhaiten tehdä jonkin asian, ottaa sen omalle vastuulleen. Viimeisenä arvona on itsensä ylittäminen, joka tarkoittaa tiimin jatkuvaa ponnistelua päämääränsä saavuttamiseksi ja oman toimintansa parantamiseksi. (Leffingwell 2007, 42–43.)

3.6 Scrumin prosessimalli

Scrumin prosessimalli koostuu 30 päivän syklistä eli sprintistä ja siihen liittyvistä aikajaksotetuista ominaisuuksista (Schuh 2005, 24). Vasemmalta oikealle päin selostettuna sprint alkaa suunnittelupäivällä (planning day), jossa käydään läpi tuotteen backlog sekä valitaan sprint backlogiin sprintin aikana toteuttavat toiminnallisuudet. Toiminnallisuudet toteutetaan 30 päivän aikana sprintissä, jona aikana järjestetään päivittäin 15 minuutin pituinen Scrum-palaveri. Sprintin loppuessa järjestetään demotilaisuus ja retrospektiivinen katselmus (retrospective). Kyseinen sykli tuottaa uusia toiminnallisuuksia, jotka voidaan mahdollisesti antaa suoraan asiakkaalle sprintin jälkeen.



KUVIO 3 Scrumin prosessimalli (Schwaber 2006)

Luku 3.6, Scrumin prosessimalli, esittelee edellä mainittuja Scrumiin tiiviisti kuuluvia aikajaksotettuja ominaisuuksia. Luvussa 3.6.1 esitellään sprintti. Luvut 3.6.2–3.6.5 esittelevät prosessimallin mukaisesti sprintin alusta loppuun sijoittuvat ominaisuudet alkaen suunnittelupäivästä ja loppuen retrospektiiviseen katselmukseen.

3.6.1 Sprint

Scrumissa sanalla sprint tarkoitetaan keskimäärin 30 päivän pituista, suunniteltua aikajaksoa. Sprintin aikana toteutetaan vaatimuksissa määritellyt toiminnallisuudet alusta loppuun saakka potentiaalisesti annettaviksi asiakkaalle. Sprintin etuna on uusien ominaisuuksien nopein mahdollinen valmistuminen. Pituus on ohjeellinen, ja se juontaa juurensa nykyisestä 90:n tai 120 päivän julkaisusyklistä, jolloin yhden julkaisun aikana ehditään suorittaa keskimäärin kolme sprinttiä. Päivien määrä tulee kuitenkin määrittellä kokemusten perusteella ja toteuttaa parhaaksi havaittua aikaväliä kolmesta viiteen viikkoon. (Schuh 2005, 226–227; Leffingwell 2007, 45.) Sprintin etuna oleva nopea sykli uusien toiminnallisuuksien valmistumisessa on väärin käytettynä sen kompastuskivi. Uusien ominaisuuksien julkistaminen nopeassa tahdissa on liiketoiminnan kannalta kriittinen toimenpide, mutta tuotteen laadun takia täytyy olla vaihtoehto siirtää toiminnallisuuden julkistamista seuraavaan kertaan tarpeen mukaisesti. (Schuh 2005, 206–227.)

3.6.2 Suunnittelupäivä

Suunnittelupäivä on Scrumin kriittisin ja tärkein tapahtuma. Mikäli suunnittelupäivä ei onnistu, se voi sotkea koko sprintin toiminnan. Suunnittelupäivän tavoitteena on saada kokoon tiimi ja tuotteenomistaja suunnittelemaan sprintti. Päivän aikana, kahdessa peräkkäisessä tapaamisessa, annetaan tiimille tarpeeksi tietoa, jotta se pystyy työskentelemään sprintin ajan rauhassa. Konkreettiset tuotokset päivän ajalta ovat sprintin tavoitteiden määrittäminen, sprint backlog, demon päivämäärä, päivittäisen Scrum-palaverin aika ja paikka sekä tiimin jäsenien nimeäminen. (Kniberg 2006, 15; Schwaber & Beedle 2002, 47.)

Suunnittelupäivän ensimmäisessä osassa määritellään sprintin tavoite sekä demon paikka ja aika. Tuotteen omistaja esittelee tuotteen backlokin vaatimukset ja lisäksi määrittellään vaatimusten tekemiseen kuluva aika karkeasti päivien tarkkuudella.

- Sprintin tavoitteen selvittäminen antaa vastauksen kysymykseen: ”Miksi me teemme tämän sprintin?”. Tavoitteen selkeytyminen antaa tiimille sprintin aikana selvyuden, mihin suuntaan toiminnallisuuksia tulee kehittää.
- Demon paikan ja ajan päättäminen määrittelee sprintin aikavälin pituuden. Aikavälin pituutta ei ole rajoitettu, joten optimaalinen pituus tulee itse määritellä ja pyytää hyväksi todetussa aikarajassa.
- Tuotteen omistajan esittelemät vaatimukset tuotteen backlogista selvittävät tiimille, mitä vaatimukset tarkoittavat. Tämä on velvoite myös tuotteen omistajalla, jonka tulee tietää, miksi kyseinen vaatimus on backlogissa ja kuinka se demotaan.
- Tuotteen backlogin vaatimusten työstämiseen kuluvan ajan arviointi vaatimus kerrallaan määrittelee, kuinka kauan vaatimusten toteuttaminen kestää työpäivinä. Tässä arvioinnissa voi käyttää esimerkiksi seuraavanlaista asteikkoa: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100 ja ?. (Kniberg 2006, 13–16, 19–20, 32–33; Schwaber & Beedle 2002, 47–48.)

Suunnittelupäivän toisessa osassa tiimi ja Scrum-mestari suunnittelevat, mitkä vaatimukset toteutetaan ja kuinka ne toteutetaan (Schwaber & Beedle 2002, 47). Ensimmäisenä tehtävänä on määritellä käytössä olevan henkilötyömäärän raamit eli se, kuinka monta päivää tiimin jäsenillä on aikaa käytettäväksi sprinttiin. Tämän summan perusteella valitaan tuotebacklogista tärkeysjärjestyksessä vaatimuksia sprint backlogiin, mutta vaatimusten työmääräarviot eivät saa ylittää tiimin arviota. (Kniberg 2006, 24–27.) Kun vaatimukset ovat valittu, tarkennetaan kunkin vaatimuksen työmääriä tuntitasolla neljästä kuuteentoista tunnin tehtäviksi. Kyseinen lista tehtävistä esittää tiimin parasta arviota ja ymmärrystä sprintin aikana toteutettavasta työstä. Tehtävät muuttuvat ja tarkentuvat tiimin työskentelyn aikana. Mikäli arvioidun tuntimäärän kokonaiskesto ylittää vaatimusten alkuperäisen arvioinnin määrän, tulee tuotteen omistajan kanssa tarkastella, mitä vaatimuksia pystytään poistamaan. Jos aika riittää, voidaan tuotebacklogista valita lisää tehtäviä sprinttiin mukaan. (Schuh 2005, 228.)

3.6.3 Scrum-palaveri

Jokaisena työpäivänä sprintin aikana pidetään samaan kellonaikaan samassa paikassa päivittäinen, tyypillisesti 5–15 minuuttia kestävä tapaaminen, päivittäinen Scrum-palaveri. Päivittäisen Scrum-palaverin järjestää Scrum-mestari, jonka tehtävänä on

pitää tapaaminen annettujen aikarajojen sisällä. Jotta tapaamiset pysyisivät lyhyinä, ne pidetään seisoen, jotta turhat puheenaiheet karsiintuisivat pois. (Schuh 2005, 272–277.)

Kaikkien tiimin jäsenten tulee olla mukana tapaamisessa. Mikäli fyysinen läsnäolo ei ole mahdollista, tulee joko ilmoittaa kerrottavat asiat jollekin toiselle ryhmän jäsenelle tai hyödyntää tekniikkaa, esimerkiksi kaiutinpuhelinta. (Schuh 2005, 278.) Asiat, jotka päivittäisessä Scrum-palaverissa tulee jokaisen tiimin henkilön kertoa, ovat seuraavat:

- Mitä minä tein eilen?
- Mitä minä teen tänään?
- Onko minulla esteitä? Olenko estetty tekemästä työtäni? (Leffingwell 2007, 133.)

Päivittäisen Scrum-palaverin tarkoituksena on karsia muut tapaamiset, tunnistaa ja poistaa esteitä projektilta, parantaa kommunikointia ryhmän kesken, jakaa tietoisuutta sekä tukea nopeaa päätöksentekoa. Scrumin periaatteita mukaillen avoimuutta ja lineaarisuutta korostetaan myös päivittäisessä Scrumissa. Kuka tahansa voi tulla tapaamiseen, oli hän sitten yrityksen johtokunnasta, tuotteen omistaja tai muu tiimin ulkopuolinen henkilö. Ainoana sääntönä muiden henkilöiden osallistumiselle on, että he eivät saa osallistua keskusteluun. Kuuntelemalla ulkopuoliset osanottajat saavat käsityksen projektin tilasta. Osallistuminen päivittäiseen Scrum-palaveriin voidaan nähdä informatiivisempänä tapahtumana kuin raporttien lukeminen. (Schwaber & Beedle 2002, 40.)

Knibergin mukaan päivittäiseen Scrum-palaveriin kuuluu olennaisena osana ”tehtävätaulun” ylläpitäminen. Tehtävätaululle on kirjoitettu suunnittelupäivän aikana valitut vaatimukset ja arviot työmääristä. Näiden paikkoja ja arvioita tiimin jäsenet tai Scrum-mestari päivittävät tarpeen mukaan päivittäisen Scrum-palaverin aikana tai tiimin ohjeiden mukaisesti ennen tapaamista. (Kniberg 2006, 61.)

3.6.4 Demo

30 päivän välein, neljässä tunnissa, on esitettävä, mitä sprintin aikana on saatu tehtyä. Kyseinen tapahtuma kasaa samaan tilaan tiimin, johdon, käyttäjät, asiakkaat ja tuotteen omistajan tarkistamaan sprintin tulokset. Vaikka määritelmä kuulostaa virallisel-

le, kyseessä on hyvin epävirallinen demoaminen, enemmänkin työskentelytapaaminen, jonka tavoitteena on määrittellä paras suunta seuraaville sprinteille. (Schwaber & Beedle 2002, 54–56.)

Tilaisuuden kulku on rakenteeltaan hyvin yksinkertainen. Mikäli paikalla on henkilöitä, jotka eivät tunne aihetta, tulee Scrum-mestarin esitellä lyhyt tiivistelmä aihealueesta. Tämän jälkeen verrataan sprintin tavoitteita ja tuotteen backlockia sprintin lopputuloksiin. Keskustelun ja kommentoinnin avulla tulisi löytyä parannusideoita ja suunta seuraaville sprinteille. (Mts. 55–56.)

Kniberg on ohjeistanut seuraavanlaisen muistutuslistan sprintin demosta:

- Demossa tulee keskittyä vain olennaiseen, koska liiallinen valmistautuminen antaa kuvan, ettei tiimillä ollut tarpeeksi valmista esiteltävää.
- Sprintin tavoitteet tulee esittää selkeästi läsnä oleville, business-orientoituvaan tyyliin, ei teknisten yksityiskohtien mukaan.
- Mikäli mahdollista, demoamisen tulisi tehdä henkilön, joka ei ole ollut sprintiä tekemässä.
- Jotta keskityttäisiin business-arvoltaan olennaisiin kohtiin, tulee vähäpätöiset korjaukset vain todeta suoritetuiksi. (Kniberg 2006, 66.)

Demosta on kaikille läsnä oleville henkilöille hyötyä. Kyseinen sosiaalinen tapahtuma luo laaja-alaista vuorovaikutusta, josta jokainen läsnäolija hyötyy omalla tavallaan. (Mts. 65.) Tuotteen omistaja saa selville, kuinka paljon toiminnallisuuksia on saatu aikaiseksi. Johdolle selviää, mitä tiimi on annetuilla resursseilla saanut aikaiseksi. Asiakas saa ensimmäisen vilkaisun tuotteeseen ja antaa palautetta. Kehittäjät ja insinöörit saavat ideoita teknisistä ratkaisuista. (Schwaber & Beedle 2002, 55.)

Yksi tärkeimmistä demon anneista on, että demon jälkeen tiimillä on 99 % valmista julkaistavaa materiaalia, joka on yksi Scrumin tavoitteista. Myös tiimin saama palaute antaa heille tilanteesta riippuen joko hyvän mielen tehdystä työstä tai ”happaman maukuista lääkettä”, joka saa heidät jatkamaan työtä entistä sisukkaammin. (Kniberg 2006, 65.)

3.6.5 Retrospektiivinen katselmus

'At regular intervals, the team reflects on how to become more effective, then tunes and adjust its behavior accordingly.' (Agile Manifesto 2001)

Scrum toteuttaa Agile Manifeston periaatteen 12 mukaista toimintaa sprintin retrospektiivisen katselmuksen kautta (Leffingwell 2007, 179). Kyseisessä, yhdestä kolmeen tuntia kestävässä kokoontumisessa kasataan palautetta kaikilta osanottajilta ja muutetaan ne toimintapisteiksi. Kokoontumiselle tulee valita sihteeri, joka pitää kirjaa määritellyistä toimintapisteistä. Läsnä tulee olla tiimin, tuotteen omistajan, Scrum-mestarin sekä mahdollisesti asiakkaan. Tilaisuus tulee pitää häiriöttömässä tilassa, jossa keskustelu voi olla vapaata ja ihmisten keskittyminen ei pääse alenemaan. (Kniberg 2006, 67–68; Schuh 2005, 264.)

Schuh esittää neljän askeleen läpiviennin retrospektiivisen katselmuksen asialistaksi:

1. Läsnäolijat kertovat sprintin aikana esiin tulleista hyvistä asioista ja antavat parannusehdotuksia seuraavaan sprinttiin.
2. Ryhmä käy läpi kaikki esiin tulleet kohdat ja priorisoi ne tärkeysjärjestykseen.
3. Esiin tulleista kohdista keskustellaan ryhmässä prioriteettien asettamassa järjestyksessä, kunnes ne loppuvat tai aika loppuu.
4. Toimintapisteet kirjataan ylös. (Schuh 2005, 264.)

Kniberg esittää kirjassaan, että retrospektiivinen katselmus on Scrumin toiseksi tärkein tapahtuma, koska se on tiimin mahdollisuus edistää asioita organisaatiossa. Kun yhden henkilön ideasta muodostuu keskustelun kautta koko tiimin yhteinen ehdotus, parantuvat mahdollisuudet idean läpiviennille. Kniberg painottaa, että ongelmat tulee saattaa muiden tiimien tietoisuuteen arvokkaan tiedon takia. Tiedon välittäjänä voivat toimia wikisivustot tai henkilö, joka toimii ”tietosiltana” eri tiimien välillä. (Kniberg 2006, 67, 69.)

3.7 Scrumin hyödyt

Scrumista on nähtävissä teorian osalta selkeitä hyötyjä. Nämä hyödyt tulevat esiin toimintojen ja asetettujen arvojen kautta.

Läpinäkyvyys

Schwaberin mukaan Scrumin tärkein hyöty on avoimen ja rehellisen asiakassuhteen muodostaminen. Luvussa 3.2.3, Empiirinen prosessinhallinta, kerrotaan, että ohjelmistokehitys on täysin ennalta arvaamatonta. Asiakkaan ja tuottavan osapuolen välinen neuvottelu sovelluksesta, jonka asiakas maksaa, on jatkuvaa kädenvääntöä rahan, toiminnallisuuksien, ajan sekä laadun välillä. Scrumin läpinäkyvyys luo mahdolliseksi peittää esiin tulevia asioita, jolloin ainoa mahdollisuus on toimia asiakkaan kanssa rehellisesti. Tällöin asiakas osallistuu suoraan kehitystyöhön ja saadaan enemmän tärkeää palautetta, joka auttaa muokkaamaan tulevaisuuden prioriteetteja. (Schwaber & Beedle 2002, 82–83; Leffingwell 2007, 117.)

Ongelmakohtien löytäminen

Scrumille luonteenomainen iteratiivisuus tuo esiin ongelmakohdat ja riskit, jolloin toimimattomat asiat huomataan nopeasti, koska niiden vaikutus voimistuu lyhyen aikavälin kuluessa. Tällöin ongelmakohdat tunnistetaan helpommin ja niihin voidaan reagoida nopeammin. (Schwaber 2007, 26–27.)

Hallinto

Ohjelmistoalan työ pitää sisällään yllätyksellisiä vivahteita, eikä kehitys ole yksiselitteistä. Tulee tilanteita, joissa työntekijät joutuvat etsimään ratkaisuja kysymykseen, mikä on paras työskentelytapa. Jos työntekijöille on asetettu rajoitteita työskentelytapoihin, he eivät saavuta parhaimpaa mahdollista lopputulosta, koska he eivät ole vapaita selvittämään, kuinka tehdä oma työ parhaiten. Tuottavuus kaksinkertaistuu, kun työntekijät saavat työskennellä vapaasti ilman heitä rajoittavia ohjeistuksia, jotka kertovat, kuinka asioita tulisi tehdä. Tuottavuus nelinkertaisuus, kun hallinnoijat voivat keskittyä omaan työhönsä työntekijöinä. (Mts. 23–24.)

Käyttöönottavuus

Scrumin integroiminen yrityksen olemassa oleviin käytäntöihin ja projekteihin onnistuu ilman suurempia muutoksia. Sen kevyt rakenne mahdollistaa vähäiset muutokset

päivittäiseen työskentelyyn, jolloin käyttöön ottaminen ei ole vaikeaa. Suurin muutos tapahtuu asiakkaan suhteen. Asiakas ei saa puuttua tiimin työskentelyyn sprintin aikana ja hänen tulee ymmärtää, että tiimi voi muuttaa sprintin tavoitteita. (Schuh 2005, 25.)

Oppiminen

Jotta jatkuvaan muutokseen voidaan reagoida nopeasti, tulee tiimin olla läheisessä yhteydessä ulkopuolisten informaatiolähteiden kanssa. Tämä näkökulma voidaan nähdä jatkuvana yksilöiden, ryhmän ja organisaation oppimisena. Opin välittämistä tapahtuu kommunikaation kautta sekä tiimin kesken että myös sen ulkopuolelle. (Schwaber & Beedle 2002, 108.)

Sitoutuminen

Sitoutuminen työhön, jota ei ole mahdollista toteuttaa tietyssä ajassa, ei ole mahdollista. Vesiputousmallissa tämä tilanne pystyy toteutumaan, koska tiettyjen tehtävien tulee valmistua tiettyyn hetkeen mennessä. Scrumin vaihtoehtona on selvittää, mikä on mahdollista toteuttaa tietyssä ajassa, jolloin motivaatio pysyy korkealla. (Schwaber 2007, 24–25, 91.)

Asiakaslähtöisyys

Jokaisen sprintin tuotoksena tulee olla asiakkaalle mahdollisesti annettava toiminnallisuus. Kun asiakas on tilannut toiminnallisuuden, hän pääsee tarkastelemaan tuloksia jo ensimmäisen sprintin jälkeen. Mikäli toiminnallisuus on hankalasti toteutettava, siitä saadaan ensimmäisen sprintin jälkeen jo esimakua. Kun asiakas näkee valmiin toiminnallisuuden, hän saattaa tehdä päätöksen lisäinvestoinneista toiminnallisuuden syventämiseksi tai tyytyä saatavilla olevaan. Asiakkaan ei tällöin tarvitse investoida kuin hänelle tarpeelliseen toiminnallisuuteen. Hän voi itse valikoida, mitä hän haluaa, ja tilata räätälöidysti, kunnes on vakuuttunut tilatuista toiminnallisuuksistaan. Toiminnallisuuksia kehittänyt yritysään ei tällöin joudu ylläpitämään ominaisuuksia, joita asiakas harvoin käyttää tai ei käytä laisinkaan, vaan voi keskittyä oleellisen business-arvoltaan tärkeämpään toiminnallisuuteen. (Schwaber 2007, 130–131.)

Itsestään organisoituvat tiimit

Sovelluskehitystehtävät ovat dynaamisia, joten staattisia henkilökeskeisiä vastuualueita ei voida asettaa. Tiimissä työtehtävät jäsenten välillä löytyvät osaamisen kautta.

Tehtävät ovat dynaamisia, joten kuka tahansa tiimin jäsen voi tehdä tehtäviä, joihin heillä on parhaimmat edellytykset osaamisensa puolesta. Tiimin dynaaminen rakenne on usein muutaman henkilön yhteistoimintaa. Tällöin dynaamista sovellusta kehitetään vuorovaikutuksen avulla. (Schwaber & Beedle 2002, 117.)

4 MUUTOSPROSESSI

Opinnäytetyön neljäs luku käsittelee muutosta Scrumiin. Ensimmäisessä alaluvussa käydään läpi perusidea siitä, mitä muutoksella tarkoitetaan. Toinen alaluku käsittelee muutoksen mukanaan tuomaa oiretta eli muutosvastarintaa. Kolmas luku, muutos Scrumiin, käsittelee Scrum-menetelmän käyttöönottoa organisaatiossa.

4.1 Mitä muutoksella tarkoitetaan?

Kirjallisuus esittää organisatorisen muutoksen kolmivaiheisena prosessina. Ensimmäisenä kohtana on yrityksen havahtuminen muutostarpeisiin, jotka ovat ilmenneet sisäisistä tai ulkoisista tekijöistä, ja käytettävien menetelmien toteaminen vanhentuneiksi. Toisena kohtana yritys luo vision tavoitteellisesta tilasta, määrittelee toimenpiteet tavoitteen saavuttamiseksi ja toteuttaa ne. Lopuksi yritys vakiinnuttaa uuden toiminnan, jolloin käytänteet ovat osa yrityksen normaalia toimintaa. (Hannus 1994, 121.)

Muutoksen riskit ovat sitä suurempia, mitä suuremmasta muutoksesta on kyse. Sen suuruuteen vaikuttaa poikkeama aiempaan tilanteeseen, kun tarkastellaan organisaatiota, valtarakenteita ja kulttuuria. Haasteellisempänä kohtana muutoksessa on uuden ajattelumallin omaksuminen ja sitä vastustavan muutosvastarinnan lieventäminen. Loppujen lopuksi ratkaisevin asia muutoksen onnistumiseksi on yksilöiden kyky ja halu muutokseen. (Mts. 115–117.)

4.2 Yksilön muutosvastarinta

Muutos ja kehitys vaativat sen kohteena olevalta henkilöltä oppimista. Oppiminen vaikuttaa henkilön tietoihin ja taitoihin, mutta myös häneen itseensä, ihmissuhteisiin ja minäkuvaan. Muutos voi tehdä kipeää, koska se saattaa huojuttaa henkilön luomia elämänrakenteita ja tätä kipua henkilö pyrkii torjumaan muutosvastarinnalla. (Petäjä, Koponen, Lautamatti & Bauer 2000, 51.)

Vanhojen kaavojen tunnistaminen sekä tahdon herättäminen ja tukeminen uuteen auttaa henkilöä ponnistamaan uusiin menetelmiin. Toisaalta vanhojen kaavojen olemassaololla on suojaava merkitys. Kun henkilö huomaa itse, kuinka merkityksellinen muutos on, hän kykenee sopeutumaan uuteen tilanteeseen paremmin. Täten voidaan sanoa, että muutosvastarinta on terve ilmiö. (Mts. 51.)

Muutosvastarintaa esiintyy kolmella tasolla, jotka esitellään seuraavassa:

1. Ajattelun tasolla muutosvastarinnalla yritetään etsiä selitystä sille, että muutos on vaikea tai epätarkoituksenmukainen. Tällöin vastavoimana on tukeutua faktoihin ja järkipäiseen ajatteluun. Osallistumisen kautta, oman aktiivisuuden sekä tutkivan asenteen avulla, ajattelun muuttuminen väistää vanhat ajatusmallit uusien tilalta.
2. Tunnetason vastarinta jakaantuu positiivisiin ja negatiivisiin tunteisiin tiettyä asiaa kohtaan. Tunnetasoja pystytään sulattelemaan aktiivisen kuuntelun, kierrosten, taiteellisten harjoitteiden, keuharjoitteiden sekä päiväkirjan avulla. Tavoitteena on, että henkilö pystyy kohtaamaan itselleen vaikeat tunteet ja etenemään niiden kanssa, jolloin saavutetaan vapautumisen ilo sekä sisäistä kypsymistä.
3. Tahdon tasolla olevaa muutosvastarintaa esiintyy pelon tunteena tulevaisuutta kohtaan. Tämä voi ilmetä lamaautumisena, kyynisyytenä, välinpitämättömyytenä tai aktiivisena vastaiskuna muutosta kohtaan. Apukeinoja tahdon tason muutosvastarintaa kohtaan ovat rohkaisevat esimerkit, selkeät suunnitelmat, sopimusten pitäminen, tukea antavat keskustelut, aikataulujen ja tietojen luotettavuus ja sitoutuminen. (Mts. 51–52.)

4.3 Siirtyminen Scrumiin

Kuten luvussa 4.1 mainittiin, muutostarpeen huomaaminen on lähtökohtana muutoksen toteuttamiseen. Nykyään on huomattavissa, että laadukkaan sovelluksen kehittäminen on todella hankalaa nopeassa julkistamissyklissä. Syiden etsiminen ei ole pääasia, vaan on ymmärrettävä, että muutos sovelluskehitykseen on raskasta työtä, joka vaatii muutosta sekä organisaatioon että sovelluskehitykseen. (Leffingwell 2007, 289–290.) Kyseinen luku käy läpi muutosta organisaation ja sovelluskehityksen nä-

kökulmasta ja esittää viiden askeleen ”pelin” organisaation siirtymisestä käyttämään Scrumia.

4.3.1 Organisaation muutos

Schwaber esittelee organisaation muutoksen kolmena kohtana. Ensimmäinen kohta on empiirisen prosessin hallinnan hyväksyminen. On parempi tietää joka päivä, missä tilanteessa projekti on menossa kuin huomata paljon myöhemmin tarkan suunnitelman olevan täysin väärin. Toinen kohta on tiimin työtä hidastavien esteiden, ”impedimenttien”, poistaminen organisaatiotasolla. Itsestään organisoituva ja itsestään johdettu tiimi on vastuullinen sekä valtuutettu hyödykkeen toteuttamisesta. Scrum eliminoi lähtökohdat perinteiselle hierarkkiselle mallille, jossa johto kertoo tiimille, kuinka heidän tulisi tehdä työnsä. Johdon tehtävänä on tukea tiimin itsenäistä työtä antamalla heille mahdollisuudet työskennellä ilman esteitä. Kolmantena kohtana on toteamus: Pienempi ennustettavuus on parempi kuin väärät uskot. Scrum perustuu uskomukseen, että työskenneltäessä hyvin epävakaa, teknillisessä sovelluskehitysympäristössä, kukaan ei pysty arvioimaan, mitä tiimi pystyy toteuttamaan ja kuinka paljon syntyy kustannuksia. Projektin aikana tiimi pystyy antamaan arvioita sekä neuvottelemaan suunnitelmasta ottaen huomioon riskit. Suunnitelma on, että tiimi tuottaa parhaan mahdollisen lopputuloksen annetuista lähtökohdista. (Schwaber, Leffinwell & Smits, 2005, 5–6.)

Scrum vaatii johdon panostusta esteiden poistamiseksi sekä prosessin tukemista. Organisaatiosta esteitä voidaan löytää neljältä alueelta: Scrum-prosessista, tuotteen kehittämisen prosessista, organisatorisesta toiminnasta tai ihmisten toiminnasta. Kun esteitä poistetaan näiltä alueilta, osallistutaan jatkuvaan organisaation kehittämiseen, joka tähtää laadun ja tuottavuuden paranemiseen tiimeissä. (Mts. 9, 15.)

Muutos on kovaa työtä. Mikäli muutoksesta johtuvia ongelmia syyllistetään muutoksen takana olleeseen henkilöön, voi muutos pysähtyä kokonaan. Pahimmillaan pysähtyy koko yrityksen muutos kehittää parempaa sovellusta. Kun muutos ei onnistu kivuttomasti, tulee ymmärtää sen olevan osa muutosta ja nähdä se mahdollisuutena ratkaista ongelmia yhdessä. Useat esteet voivat olla niin syvänjuurtuneita toimintoja, että niiden poistaminen kivuttomasti on mahdotonta. (Mts. 9.)

4.3.2 Scrumin omaksumisprosessi

Tilannearvio ja pilottiprojektin suunnittelu

Aloitustilanteessa tulee tehdä tilannearvio nykyisestä organisaatiosta. Scrumin tuomat muutokset tulee esittää johdolle, jolloin selviää, ollaanko valmiita muutokseen. Jotta ympäristö saadaan kuntoon, tulee selvittää, ollaanko valmiita ketterälle muutokselle, kouluttaa pilottiprojektien henkilöstö sekä luoda tuote backlog-pilottiprojekteille. (Schwaber, Leffinwell & Smits 2005, 10; Leffinwell 2007, 304–305.)

Pilottiprojektit

Jotta Scrumin vaikutuksia voidaan näyttää toteen, tulee järjestää pilottiprojekteja. Kokemuksien kautta selviää esiin tulevia esteitä organisaatiossa sekä esteitä Scrumin käyttöön ottamiseen. Pilottiprojektien tulisi kestää kahdesta kolmeen kuukautta, jona aikana suunnitellaan ja sopeudutaan vastaan tuleviin tilanteisiin. (Mts. 11, 305–306.)

Laajentaminen

Onnistuneiden pilottiprojektien perusteella voidaan aloittaa Scrumin käyttöönoton laajentaminen organisaatiossa. Pilottiprojekteista saadun tiedon avulla voidaan poistaa esteitä laajemman käyttöönoton tieltä sekä aloittaa kouluttaa laajempaa osajoukkoa. Yleiselle keskustelulla saada lisää näkyvyyttä Scrumiin organisaatiossa sekä lievittää muutostavastarinnan syntyä, jolloin edistetään Scrumin hyväksymistä. (Mts. 11–12, 306–307.)

Vaikutuksen saavuttaminen

Pilottiprojektien vaikutus on loppujen lopuksi vain pintaraapaisu todellisen vaikutuksen saavuttamiselle. Merkityksellisemmän vaikutuksen saavuttamiseksi on laajennettava Scrumin hyödyntämistä entistä suuremmissa projekteissa. Hyödyntämällä aiemmin kerättyä tietoa Scrumin käyttöön ottamisesta voidaan olla varmoja hyvistä lopputuloksista. (Mts. 12–13, 307–308.)

Arvioi, mukaudu ja mittaa

Kun organisaatiossa on huomattava osuus projekteista omaksunut muutoksen Scrumiin, laajenemisen jatkumiseksi tulee määritellä kattavampi mittauskriteeristö ja arvioida organisaatiossa tapahtunutta muutosta. Scrum sekä ketterät menetelmät ovat täysin laskettavissa ja mitattavissa. Tällöin saadaan laadullista ja määrällistä palautetta

sekä prosessi- että projektitasolla. Tulokset arvioinnista tulisi saattaa kaikille esiin ja analysoida. (Mts. 13, 308.)

Laajenna voittaaksesi

Organisaatiossa tapahtuneiden muutosten ja arviointimenetelmien selvittämisen jälkeen on aika laajentaa Scrumin käyttöön ottaminen koko organisaation tasolle. Vasta viimeisessä vaiheessa voidaan muokata Scrumin tiukkoja toimintaohjeita. Kouluttamista ja arviointia tulisi jatkaa ja asiakas voidaan tiukemmin sisällyttää projekteihin. Vaihe jatkuu, kunnes muutos on käynyt läpi kaikki projektit. (Mts. 14, 309.)

5 TUTKIMUKSEN TOTEUTUS JA TULOKSET

Luku 5 jakaantuu kahteen lukuun, jotka käsittelevät opinnäytetyön empiirisen osion. Empiiriseen osioon kuuluu ensiksi tutkimuksen toteutuksesta kertova luku 5.1. Kyseisessä luvussa esitellään tutkimuksessa käytetty menetelmä ja kerrotaan toteutuksesta. Empiirisen osion toinen vaihe on tutkimuksen tulosten avaaminen. Haastatteluiden tulokset ovat avattuna kysymys kerrallaan luvussa 5.2. Haastatteluiden pohjana toimineet kysymykset on sisällytetty liitteisiin 3 ja 4.

5.1 Tutkimuksen toteutus

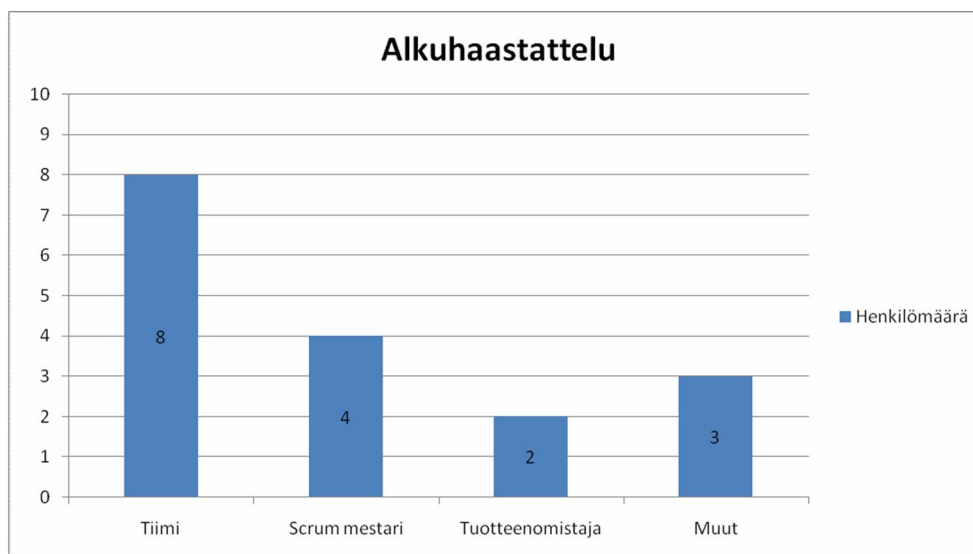
Tutkimuksen toteutuksessa ei ole voitu käyttää kvantitatiivista eli laskennallista lähtökohtaa, vaan tutkimus on toteutettu laadullisesti eli kvalitatiivisesti. Syy laadulliseen tutkimukseen on pieni otoskoko, jolloin tulosten virhemarginaali nousee liian suureksi käytettäessä laskennallista menetelmää. Kvalitatiivisuus tuo mukanaan menettelytapoja, jotka sopivat kyseisen opinnäytteen aiheeseen paremmin. Kysymykset muutoksesta ovat yksilöllisiä ja tunteita herättäviä, jolloin avoin vastausmahdollisuus tuo esiin haastateltavan asiantuntevuutta sekä tuntemuksia aiheeseen liittyen. On arvokkaampaa saada pienemmältä otokselta kehitysideoita kuin tilastoida virhemarginaaliltaan suuria arvoja.

Haastattelut ovat toteutettu avoimin strukturoimattomin kysymyksin, yksilöhaastatteluina. Tällöin haastateltava voi vastata kysymyksiin ilman eri vaihtoehtoja avoimesti. Yksilökohtaiset haastattelut loivat mahdollisuuden lausua nimettömästi suljetun oven takana mielipiteitä muutoksesta, jolloin vastauksen kaunisteluun ei ollut tarvetta. Rehellisesti kerrottu tieto on tärkeää tutkimuksen kannalta toimeksiantajan jatkokehitystä silmällä pitäen.

Ennen haastatteluiden alkamista kerättiin lista henkilöistä, jotka kuuluivat pilottiprojektien henkilöstöön. Listan henkilöt jaettiin neljään kategoriaan tehtäviensä mukaan. Scrumissa työtehtävät jaetaan kolmeen kategoriaan: tiimi, Scrum-mestari ja tuotteen

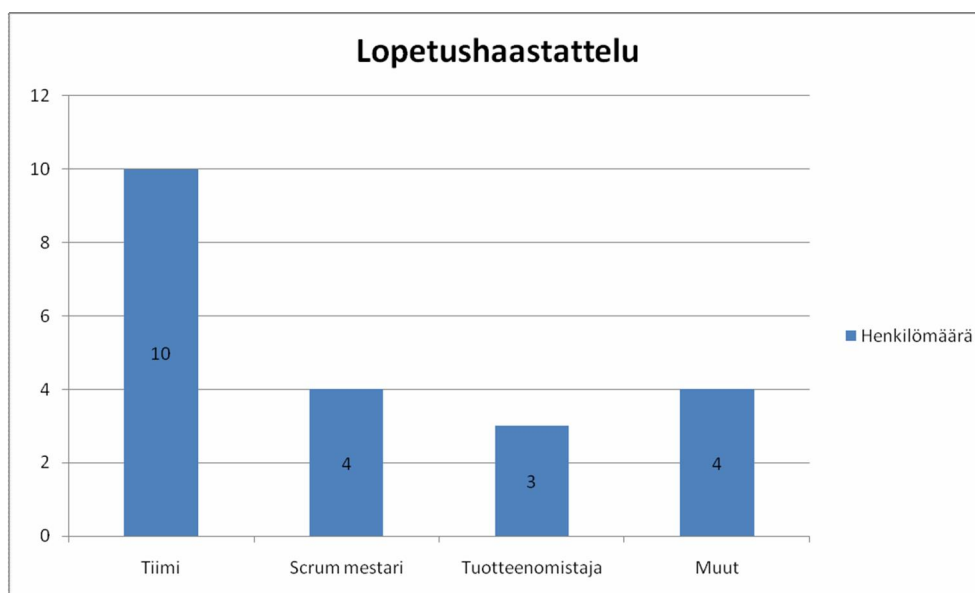
omistaja. Haastateltaviin kuului myös henkilöitä, joita ei voitu kategorisoida tämän jaon mukaan, joten muodostettiin Muut-osio. Muut-osioon kuului henkilöitä tuotteen tuesta ja yksiköiden johdosta. Heidän roolinsa ei ollut suoranaisesti pilottiprojekteihin liittyvä, mutta laajemmassa mittakaavassa he kuuluivat selkeästi kohdehenkilöihin. Kysymyksissä selkeä jaottelu tuli esiin vain yhdessä kysymyksessä vaatimuksiin liittyen. Kyseisessä kysymyksessä oli selkeästi huomattavissa jaottelu henkilöihin, jotka arvioivat vaatimuksiin käytettävää aikaa ja vaatimusten priorisointia.

Aloitushaastattelu toteutettiin loka-marraskuussa 2007 toimeksiantajan tiloissa. Haastatteluun osallistui 17 haastateltavaa neljästä kategoriasta. Haastateltavat valittiin heidän saatavuutensa mukaan 29.10.–8.11.2007. Aikaväli jätettiin mahdollisimman suppeaksi, koska Scrumin käyttöönotto oli jo alkanut haastatteluiden loppuvaiheilla, joten mielipiteet olisivat voineet alkaa muuttua pidemmällä aikavälillä.



KUVIO 4. Alkuhaastattelun henkilömäärä (n=17)

Lopetushaastattelu toteutettiin maaliskuussa 2008 toimeksiantajan tiloissa. Haastatteluun osallistui 21 haastateltavaa neljästä kategoriasta. Haastateltavat valittiin heidän saatavuutensa mukaan 3.3.–14.3.2007. Lopetushaastattelun ajankohta oli noin neljä kuukautta projektien alkamisesta. Tänä aikana pilottiprojektit ehtivät suorittaa keskimäärin kolme sprinttiä, yhden julkaisusyklin.



KUVIO 5. Lopetushaastattelun henkilömäärä (n=21)

5.2 Tutkimuksen tulokset

Tutkimuksen tulokset esitellään haastatteluiden kysymysten pohjalta. Ensimmäiset kolme lukua, luvut 5.2.1–5.2.3, ovat ensimmäisen haastattelun vastausten tiivistelmiä. Kysymykset koskevat vanhaa vesiputousmallia ja odotuksia tulevaa muutosta kohtaan. Luvut 5.2.4–5.2.7 ovat toisen haastattelun vastausten tiivistelmiä, jolloin muutos Scrumiin oli tapahtunut. Luvut 5.2.8–5.2.10 ovat kummassakin haastattelussa kysytyjä kysymyksiä. Näiden kysymysten aihealueet koskettavat kommunikaatiota, vaatimuksia ja oma-aloitteisuutta. Viimeisessä luvussa käsitellään tavoitteiden kannalta tärkein, aiemmat kysymykset yhteen kokoava, toisessa haastattelussa esitetty kysymys Scrumin hyödyllisyydestä organisaatiossa.

5.2.1 Aiemmin käytetty vesiputousmalli

Toimeksiantajan aiemmin käyttämä, 1970-luvulta lähtöisin oleva vesiputousmalli jakoi mielipiteitä. Haastateltavilta kysyttiin positiivisia ja negatiivisia puolia kyseisestä mallista. Vesiputousmalli nähtiin teoriassa hyvänä mallina, mutta odotusten mukaisesti negatiivisten ominaisuuksien osuus positiivisiin verrattuna oli määrällisesti kaksinkertainen. Vastauksissa oli myös huomattavissa ristiriitaisia mielipiteitä esimerkiksi ennustettavuuden suhteen, joka nähdään positiivisena ja negatiivisena asiana.

Positiiviset puolet

Vaikka kritiikki vesiputousmallia kohtaan on hyvinkin voimakasta, sitä ei nähdä kaikista huonoimpana mallina. Malli nähdään selkeänä sekä aukottomana kokonaisuutena, joka toimii teoriassa oikein mainiosti. Optimaalisena toimintaympäristönä mallille nähdään pienet projektit, ei niinkään isot pitkäkestoiset projektit. Vesiputousmalli on helppo omaksua, koska se on periaatteiltaan yksinkertainen. Se pakottaa tekemään dokumentaatiota ja niiden katselmointia, jotka nähdään erittäin hyvänä ja toimivana asiana.

Staattiseen tuotantoon saadaan ennustettavuutta Vesiputousmallin vaiheistuksella. Ennustettavuus helpottaa projektin suunnittelua, jolloin pystytään tunnistamaan edes jollakin tasolla projektin aikataulua ja budjetoitua. Vaiheiden kautta saadaan esiin selkeät kontekstiriippuvaiset tehtävät. Tällöin työntekijät voidaan lokeroida työskentelemään tietylle osa-alueelle, mikä helpottaa johtamista, ja työntekijät ymmärtävät paremmin, mitä heidän tulee tehdä. Eri vaiheiden lopussa olevat virstanpylväät toimivat myös mainioina laadun mittareina, jotka mittaavat projektin onnistumista.

Negatiiviset puolet

Vesiputousmalli nähdään raskaana ja jäykkänä mallina, jonka ominaispiirteenä on kiireen kasautuminen loppua kohden. Koska joustoon ei ole varaa aikataulun suhteen, työtehtävät ruuhkautuvat. Kiire konkretisoituu testaukseen, jonne tulee ruuhka testattavista toiminnallisuuksista. Ruuhka saa aikaiseksi sovelluksen laadun huononemisen ja julkaisun epävakauden. Vaaraksi muodostuu tällöin ongelmien ilmeneminen asiakkaalla, joka puolestaan aiheuttaa kustannuksia tukipalveluihin. Testauksessa oman työkuorman hallinta muodostuu kiireen takia äärimmäisen hankalaksi, koska joustoa

omien tehtävien suhteen ei ole. Vastakohtaisesti toteuttava porras joutuu odottelemaan projektin loppuvaiheessa.

Luvussa 3.2.3, Empiirinen prosessin hallinta, todettiin, että vaatimuksien ollessa epäselviä työskentely on vaikeaa. Tilanne konkretisoituu Vesiputousmallissa vaatimusten muutoksena projektin aikana. Uusia ominaisuuksia voi syntyä projektin loppuvaiheilta, mikä laajentaa työmäärää, mutta aikataulu pysyy samana. Koska projektin tulokset tulevat esiin vasta loppuvaiheessa, valmistumissykli on hidas ja mahdollisuus uusien ominaisuuksien löytymiseen on suuri. Asiakastahon seurannan puutteen vuoksi on mahdollista, että tehty tuote ei vastaa odotuksia, jolloin kiire kasvaa entisestään.

Käytännössä Vesiputousmalli on mahdoton toteuttaa sellaisenaan. Se aiheuttaa kiirettä loppua kohden, mutta toisaalta odottelua ja turhautumista liialliseen suunnitteluun sekä dokumentaation tekemiseen. Seurannan rajoittuminen ylipitkiin, liian harvoin pidettäviin kokouksiin mahdollistaa muutosten tapahtumisen aikataulun suhteen myöhäisessä vaiheessa. Työkuorman jatkuva kasvaminen kaikilla osa-alueilla syö ennustettavuutta kustannusten ja aikataulutuksen suhteen. Kyseinen epäjärjestelmällinen malli ei sovellu täten laisinkaan ylläpitoon ja pieniin muutoksiin jäykkyytensä ja rasakan prosessinsa puolesta.

5.2.2 Muutosehdotuksia vesiputousmalliin

Muutosehdotukset jakautuivat kolmen pääkategorian alle, jotka ovat testaus, vaatimukset ja kommunikointi. Tuloksina saatiin myös useita muita muutosehdotuksia pääkategorioiden aiheiden ulkopuolelle.

Testaus nähdään ongelmallisena Vesiputousmallissa. Vaikka testaus nähdäänkin ongelmana, ratkaisuvaihtoehtoja ei tullut esiin kuin muutama. Tärkeimpänä kohtana on testauksen aloittaminen aikaisemmin ja julkaisun sulkeminen aiemmassa vaiheessa. Testauksen aikana ei tulisi päästää kehittäjää enää kiinni koodiin, jolloin muutostenhallinta hankaloituu huomattavasti.

Vaatimukset ovat lähtökohtana koko projektille, joten niihin tulisi kiinnittää erityistä huomiota. Vaatimusten pohjana tulisi olla business-tarpeen mukaiset, asiakkaalle luvatut toiminnallisuudet. Tarkkojen spesifikaatioiden kirjoittamisen sijaan tulisi suun-

nitella yhdessä eri osapuolten kanssa ja keskittyä varsinkin käytettävyyteen. Jotta muutokset eivät vaihtelisi projektin aikana aiheuttaen ongelmia aikataulun kanssa, vaatimukset tulisi jähdyttää aikaisessa vaiheessa. Muutostenhallintaan tulisi saada lisää joustavuutta, mutta muutoksia kohtaan tulisi olla kriittinen. Tärkeintä vaatimuksissa loppujen lopuksi on asiakkaalle tehtyjen lupauksen pitäminen.

Kommunikointia tulisi lisätä ja sen laatua parantaa. Lyhyillä säännöllisillä palavereilla saataisiin tiedonkulku paremmaksi eri osa-alueiden välillä ja työntekijät tietäisivät, mitä muut tekevät. Kehitykseen päin tulisi saada tietoa aikatauluista enemmän ja välituloksia tekemisistä tulisi saada selkeämmin esille. Kommunikoinnin tulisi myös lisääntyä eri osapuolen välillä, jotta seuranta saataisiin tiiviimmäksi.

Muita muutosehdotuksia löytyi useita. Tulisi saada kehitykseen lisää iteratiivisuutta, joka karsisi isot, pitkään kestävät tehtävät ja lyhentäisi projekteja, jolloin aikataulut pitäisivät. Iteratiivisuus mahdollistaisi myös välietapit, jolloin nähtäisiin, mitä on saatu aikaiseksi, ja toisaalta näkyvä laatu paranisi tätä kautta. Tuotehallinto, organisaatio, jossa tuotteen omistajat työskentelevät, tulisi saada enemmän mukaan projekteihin. Lisäksi tulisi tasoittaa eri osapuolten, varsinkin testauksen, mukana olemista koko projektin ajalle. Projektin henkilöt tulisi kiinnittää vain yhteen projektiin ja ryhmien välille tulisi tehdä erotus kehityksen ja ylläpidon välille.

5.2.3 Odotukset Scrumia kohtaan

Odotukset Scrumia kohtaan olivat ristiriitaisia. Haastatteluista tuli esiin positiivisia puolia, hyötyjä, toiveita, pelkoja/riskejä ja Scrumin mukanaan tuomia ominaisuuksia odotuksina. Varsinaisia negatiivisia puolia ei tullut esiin, vaan ne naamioitiin peloiksi ja riskeiksi, jolloin muutos ei saanut negatiivista kuvaa ihmisten mielessä. Odotukset Scrumia kohtaan olivat hyvin korkealla ja todella positiiviset. Haastateltavat olivat avoimin mielin, osin varauksellisesti, suuntaamassa eteenpäin kohtia uusia haasteita. Varsinaisia negatiivisia miitteitä ei Scrumia kohtaan ollut ja oltiin erittäin tyytyväisiä, että muutos tapahtuu.

Odotukset tulevista hyödyistä tukivat positiivisia odotuksia. Vesiputousmalliin verrattuna suurimmat odotukset saatavista hyödyistä nähtiin kiireen muuttumisesta jatkuvaan tasaiseen työkuormaan. Töiden jouhean etenemisen uskotaan lisäävän työtehoa,

jolloin laatu paranee ja hyötykuorma kasvaa. Testaamisen alkaminen jo projektin alkuvaiheessa nähtiin äärimmäisen tärkeänä hyötynä, jonka tulisi vaikuttaa valmistumiseen ja julkaisun sisällön arviointiin positiivisesti. Muutos nähtiin myös kannustavana ja uuden motivaation lähteenä. Kannustavia kohtia olivat oman työn suunnitteluun vaikuttaminen, kommunikaation lisääntyminen, fokuosoinnin helpottuminen tuleviin töihin ja muutostenhallinnan helpottuminen. Tietämys Scrumista mahdollisti hyvinkin tarkkojen hyötöyodotusten kertomisen ja erityisenä hyötynä on varmasti ollut Scrum-koulutus ennen haastatteluita.

Haastateltavat esittävät toiveita Scrumia kohtaa. Scrumin hyvien puolien huomaaminen ja muutostenhallinnan paraneminen nähtiin tärkeänä jatkon kannalta. Toiveena oli, että Scrumin myötä ei tulisi tilanteita, joissa ongelmat huomataan vasta asiakkaan ympäristössä. Myös vaatimuksiin asetettiin toiveita, että ne saataisiin pienikokoisiksi, jolloin niiden ymmärtäminen helpottuisi.

Scrumin mukanaan tuomat ominaisuudet olivat selkeästi jo ihmisten mielessä haastatteluiden aikana. Haastateltavat mainitsivat useita Scrumin ”kulmakiviä” odotuksina, kuten läpinäkyvyys, rehellisyys, ongelmien esiin tuleminen ja päivittäiset palaverit. Muita asioita, joita Scrumin uskotaan tuovat mukanaan, olivat kommunikaation lisääntyminen, seurannan parantuminen, ennustettavuuden helpottuminen ja joustavuuden kasvaminen muuttuvassa maailmassa.

Haastatteluissa tuli esiin pelkoja ja selkeitä riskitekijöitä Scrumin suhteen. Eniten pelkoa tuottivat asiakkaan mukana oleminen ja dokumentaation määrä. Asiakkaan mukana oleminen nähtiin pelottavana tekijänä, koska se oli selkeästi uusi asia tietyille osapuolille. Työmäärän uskottiin kasvavan joissakin kohdin paljon nykyisestä ja pelkona oli pienten vesiputousmallien syntyminen lyhyellä aikavälillä, jolloin testaus jäisi pulonkaulaksi. Resurssien suunnittelun vaikeutumisen uskottiin myös tuovan ylimääräistä työtä. Työtehtäviin sitoutuminen nähtiin ongelmakohtana ja kysymykseksi nousikin, kuinka sitouttaa henkilöt paremmin työhönsä ja mitä mahdollisuuksia alihankkijoilla on sitoutumiseen? Mahdollisina ongelmakohtina mainittiin myös ongelmiin reagoiminen, ihmisten fyysinen sijainti eri paikoissa, ihmisten kiinnisaaminen, useassa projektissa työskenteleminen, yhteistyö projektien kesken ja laatuprosessin kiinnittäminen Scrumiin. Haastateltavat tahtoivat saada myös selville, kuinka tiukasti Scrum-

ohjeistusta noudatetaan, osataanko oikeasti jättää vaatimuksia tekemättä, jos ei ehditä, ja kuinka kauan organisaatiolta kestää sopeutua muutokseen.

5.2.4 Haastateltavien tuntemukset muutoksesta Scrumiin

Haastateltavien tuntemukset muutoksesta Scrumiin on ensimmäinen luku, jossa käsitellään jälkimmäisen haastattelukerran kysymystä. Vastaukset kyseiseen kysymykseen antoivat laajan, yksilöiden kertoman yleiskuvan tärkeimmistä kohdista, joita on huomioitu Scrumin myötä. Vastauksista tuli esiin sekä positiivisia että negatiivisia huomioita, mutta myös siltä väliltä.

Positiiviset puolet

Muutos Scrumiin on ollut tarpeellinen ja kaivattu ravistus organisaatiolle ja se on tehnyt hyvää. Uuden menetelmän käyttöön ottaminen nähdään positiivisena asiana ja muutos on saatu vauhdikkaasti käyntiin oikeaan suuntaan. Organisaatiota ravistuttanut muutos vie yritystä eteenpäin, mutta se ei ole ollut pelkästään ruusuilla tanssimista. Scrum on paljastanut paljon ongelmakohtia, joita ei aiemmin olisi voitu huomata ja tämä on saanut joiltakin osin turbulenssia organisaatioon. Muutosprosessi nähdään kuitenkin suhteellisen lievänä yritystä eteenpäin vievänä asiana. Muutosvastarintaa ei ole ollut havaittavissa lähes ollenkaan, ja vain todella pientä vastustamista on ollut huomattavissa ryhmissä. Ihmiset ovat omaksuneet yleensäkin muutoksen todella hyvin, mistä kerrotaan enemmän seuraavissa luvuissa.

Yleisesti ottaen muutoksen Scrumiin katsotaan olleen erittäin positiivinen. Kommunikaation kasvaminen nähdään johtavan parempiin lopputuloksiin ja kasvattavan työntekijöiden motivaatiota. Kommunikaatio vaikuttaa myös tiedonkulkuun eri tahoille, joka on tehostunut, jolloin voidaan ennakoida tilanteita paremmin, tiedetään koko ajan, missä mennään, mitä on valmiina ja mitä muut tekevät. Näiden ominaisuuksien vuoksi myös hallinta on aiempaa helpompaa ja läpinäkyvyys pysyy koko ajan hyvänä. Laaja vuorovaikuttaminen saa tiimeihin yhteenkuuluvuutta, jota parantaa lisäksi yhdessä suunnitteleminen. Iteratiivisuuden ansiosta tehtävät ovat aiempaa pienempiä, jolloin omat työtehtävät on helpompi ajoittaa. Tehtäviin käytettävä aika on myös saatu rauhoitettua onnistuneesti. Positiivisena ominaisuutena nähtiin myös tehtävien mahdollinen poistaminen, mikäli niitä ei ehditä tekemään, mikä tuo joustavuutta. Viimei-

senä hyvin tärkeänä positiivisena kohtana nähtiin tuotehallinnan tiiviimpi ja lähempi kontakti asiakkaaseen päin.

Negatiiviset puolet

Vaikka organisaation ravistus on tuonut positiivisia puolia esiin muutoksesta, on se saanut pakan sekaisin ja tuonut esiin kasan negatiivisia asioita. Testauksessa on yhä jatkuva kiire, koska Scrum on muodostanut testaukseen minivesiputouksia, jolloin työt kallistuvat yhä loppua kohden. Kiireen takia ei ehditä keskittyä tarpeeksi toteutettaviin vaatimuksiin, mikä huonontaa tuotteen laatua. Useaan yhtäaikaiseen projektiin kuuluminen vähentää entisestään aikaa keskittymiseen ja aiheeseen syventymiseen. Sprintin aikainen kiire pakottaa vähentämään muita aikaa vieviä tehtäviä, kuten alkuvaiheen dokumentaatiota ja vaatimusmäärittelyjä. Tämä puolestaan saa aikaan liian löyhiä vaatimuksia tuotteen omistajalta, mikä taas vaikeuttaa tiimin toimintaa, varsinkin kun oma-aloitteisuudelle ei ole tarpeeksi tilaa. Jotta kiirettä saataisiin vähennettyä, on vähennetty sprintin aikana toteutettavien vaatimusten määrää. Parannus laadussa ei kuitenkaan saisi tapahtua muiden parannusten kustannuksella. Haastatteluista tuli esiin kritiikkiä tiimin toimintaa kohtaan kiireen purkamisessa. Sitoutumisen asteen nähtiin laskeneen tiimin kesken huomattavasti verrattuna aiempaan työskentelyyn. Tällä on vaikutusta myös aikatauluun sitoutumiseen.

Muutos vaatii vielä kehittämistä sekä työtavoissa että prosesseissa. Parantamisen varaa on varsinkin ketterän ohjelmistokehityksen menetelmien kehittämisessä, tiimin vastuussa ja oma-aloitteisuudessa. Jatkossa vastauksia kaivataan kysymyksiin, kuinka työkuorma pidetään tasaisena koko sprintin ajan ja millä tavoin prosessia tulisi räätälöidä organisaatioon sopivaksi.

5.2.5 Henkilökohtainen sopeutuminen Scrumiin

Työntekijöiden henkilökohtainen sopeutuminen Scrumiin on ollut enimmäkseen erittäin helppoa. Töitä on jouduttu tekemään muutoksen takia, mutta oppiminen on helpottanut muutokseen sopeutumista ja hyväksymistä sekä motivoinut muutokseen. Oppimisen kautta on ymmärretty, että malli on toimiva, eikä ole haluja takaisin vanhaan malliin. Vanhat tavat ovat mielessä vaistonvaraisesti, mutta ne eivät aiheita muutosvastarintaa. Muutosvastarintaa on ollut todella vähäistä ja hiljaista. Tilanteissa, joissa

ei ole ollut itselle sopivia tehtäviä, on alkanut mieli palata vanhaan. Tiettyjen toimintojen tarpeellisuutta on arvosteltu.

Sopeutumisessa on ollut huomattavissa projektikohtaisuutta. Oma osaamistaso on ollut pääasiallinen kriteeri sille, kuinka hyvin on pystynyt sopeutumaan muutokseen. Mikäli projekti on ollut hankala, ei muutos ole ollut henkilökohtaisesti aivan yhtä helppo kuin vastaavasti helposti ymmärrettävässä projektissa. Tällöin muiden ihmisten tuki on ollut tärkeää asioiden ymmärtämisessä ja helpottanut muutosta.

Muutos on ollut muutamille henkilöille helppoa, koska aiempaa pohjaa Vesiputousmallista ei ole ollut, joten on voitu aloittaa puhtaalta pöydältä. Haastatteluissa tuli myös vastaan henkilöitä, jotka eivät koe minkäänlaista muutosta tapahtuneen työssään. Oma työrytmi on pysynyt aivan samana, joten erikoista sopeutumista muutokseen ei ole edes tarvittu.

5.2.6 Organisaation sopeutuminen Scrumiin

Organisaation sopeutuminen Scrumin herätti laajan skaalan mielipiteitä. Mielipiteet vaihtelivat pelottavan helposta kohtuulliseen, huonosta täydelliseen käytännön sekamelskaan. Annettiin kuitenkin ymmärtää, että organisaation muutos ei ole ollut aivan kivuton ja se vaatii vielä paljon oppimista, parantamista ja palasien yhteen kokoamista. Takaisinpäin ei kuitenkaan haikaile enää kukaan.

Mitkä ovat syyt organisaation oireilemiseen? Vastauksista tulee esiin useita ns. lukko-kohtia, joita on huomattu Scrumin myötä. Tällaisia lukkoja ovat esimerkiksi riittämättömät resurssit, ketterän ohjelmistokehityksen käytänteiden valmiusasteet, henkilöiden roolit/työnjako ja vaatimusten hallinnan kehittäminen. Oireilua ei nähdä kuitenkaan pelkästään huonona asiana, vaan ravistus on tehnyt hyvää organisaatiolle, ja positiivisia asioita on kuulunut myös henkilöiltä, joilta ei niitä odottaisi kuulevan. Koska organisaatio elää prosessiensa kautta, tuntuu, että prosessissa olevat uudet irtopalaset saavat aikaan turbulenssin, joka saa organisaation kadoksiin. Pieniä sopeutumista nakertavia ongelmia on siis nähtävissä koko ajan, mutta innostus on korkealla ja sopeutumisen vauhti kovaa hybridivaiheessa Vesiputousmallin ja Scrumin välillä.

Tavoitteiden saavuttamisessa on vielä paljon työtä jäljellä, jotta sopeutuminen saataisiin organisaation osalta valmiiksi. Vanhat projektikäytänteet ovat vielä elossa, kaikki tasot eivät ole vielä mukana muutoksessa ja tavoitteisiin ei ole päästy. Tuotehallinto ei ole päässyt irtaantumaan tarpeeksi asiakkaan suuntaan tuotekehityksestä. Ei ole myöskään muodostunut Scrumin mukaisia tiimejä ja vaaditaan paljon enemmän valmistelua ennen projekteja esimerkiksi vaatimusten ja testauksen osalta. Vastausten pohjalta on huomattavissa selkeä aloitus organisaation hitaalle siirtymävaiheelle, mutta organisaatio ei ole vielä täysin valmis muutokseen. Sopeutumisen ja järkeistyksen kautta organisaation muutosta pidetään kuitenkin mahdollisena.

5.2.7 Scrumin tuomien toiminnallisuuden hyödyllisyys ja hyödyttömyys

Scrumin prosessimalli tuo mukanaan organisaatiolle neljä uutta toiminnallisuutta, jotka on esitelty luvussa 3.5. Nämä neljä toiminnallisuutta, suunnittelupäivä, Scrum-palaveri, demo ja retrospektiivinen katselmus, saivat positiivista palautetta hyödyllisinä prosessin vaiheina, mutta myös kritiikkiä ja parannusehdotuksia. Yhtäkään toiminnallisuutta ei nähty turhana tai haluta poistaa, mutta järkeistystä kaivattiin jokaiseen toiminnallisuuteen. Seuraavassa on esitelty jokainen toiminnallisuus.

Suunnittelupäivästä on muodostunut projektien alkuun tärkeä peruskivi, joka kasaa yhteen tuotteen omistajan sekä tiimin. On erittäin tärkeää, että keskustellaan yhdessä monipuolisesti vaatimuksista ja suunnitellaan sprinttiä. Suunnittelupäivä nähdään myös oppimistilaisuutena. Kritiikkiä suunnittelupäivä saa pituudestaan. Ensimmäiset suunnittelupäivät olivat liian pitkiä, mutta aikataulun järkeistäminen on ollut todella hyvä asia. Lisäksi kritiikkiä nostattaa kaksinkertainen työmäärän arviointi. Ensin vaatimukset arvioidaan päivätasolla ja sen jälkeen tuntitasolla. Ajan suhteen kritiikkiä on esittänyt varsinkin tuotehallinto. Tuotteen omistajalta kuluu yhä liikaa aikaa istua useilla suunnittelupäivillä ja demotilaisuuksissa. Haastateltavien toivoivat yhä enemmän järkeistämistä suunnittelupäivään ja vaatimusten kehittämistä enemmän ennen suunnittelupäivää.

Päivittäisen Scrum-palaverin hyödyllisyys nähtiin vaihtelevana. Tiedonkulku ja kommunikaatio ovat parantuneet huomattavasti ja päivittäinen tapaaminen on tuonut läpinäkyvyyttä prosessiin. Isot pitkäkestoiset työtehtävät ovat saaneet aikaan päivittäisen samojen asioiden uudelleen kertomisen, joka on turhauttanut ja saanut kiinnostuk-

sen asteen laskemaan muiden henkilöiden tehtäviä kohtaan. Vaikka Scrum-palaveri nähdään hyödyllisenä ja tärkeänä toiminnallisuutena, sitä ei kuitenkaan koeta välttämättömäksi, mikäli ei ole mitään uutta kerrottavaa.

Demotilaisuus sai, kuten aiemmin esitelty Scrum-palaveri, sekä negatiivista että positiivista palautetta. Demo nähtiin mahtavana tilaisuutena oppimisen suhteen ja muiden töiden näkeminen koettiin hyväksi asiaksi. Demotilaisuudet ovat venyneet liian pitkiksi ja asiakkaan mukaan tuleminen on saanut tilaisuuden ilmapiirin muuttumaan myyntihenkisemmäksi. Tavoitteena on arvioida ja kehittää tuotetta asiakkaan läsnä ollessa, mutta avointa keskustelua ei ole vielä syntynyt. Ajan säästämiseksi tulisi arvioida, onko tarvetta järjestää demoa, mikäli ei ole mitään esitettävää. Demotilaisuudesta löytyy vielä kehitettävää varsinkin asiakkaan läsnäolon suhteen. Tulisi tunnistaa myös entistä paremmin henkilöt, joille on hyötyä osallistumisesta ristikkäisissä demotilaisuuksissa.

Demon jälkeen pidettävä, sprintin päättävä retrospektiivinen katselmus nähtiin uutena tärkeänä toiminnallisuutena. Tilaisuuden suurimpana hyötynä pidettiin palautteen antamista. Palautteen avulla saadaan materiaalia menetelmien kehittämiseen ja muodostuu läpinäkyvyyttä tiimiin ja organisaatioon. Palaute tulisi saada paremmin esille, koska toiminnot ongelmien selvittämiseksi kasvattaisivat motivaatiota. Retrospektiivisessä katselmuksessa on ollut ajankäytön suhteen ongelmia. Kuten aiemminkin, myös kyseinen tapaaminen koetaan liian pitkänä, jopa turhana toiminnallisuutena. On huomioitava kuitenkin, että jokainen toiminnallisuus koetaan tärkeänä osana sprinttiä, eikä niitä tahdota poista, vaikka niihin kuluukin yhä liian paljon tehokasta työaika.

5.2.8 Organisaation kommunikaatio

Organisaation kommunikaatiota arvioitiin alku- ja loppuhaastattelussa. Alkuhaastattelun perusteella kommunikaatio koettiin kohtuullisen toimivaksi, mutta myös haasteelliseksi ja parantamisen varaa oli varsinkin tuotehallinnon suuntaan.

Ennen prosessin muuttamista Scrum-malliin kommunikaatio koettiin kohtalaiseksi ja hyväksi, mikäli kohdehenkilö sattui olemaan paikalla. Avaimena toimivaan kommunikointiin nähtiin henkilösuhteet, mutta ilmaistiin selkeästi, että henkilökemiaongelmia ei esiinny jarruttamassa kommunikaatiota. Kuitenkin kommunikaatiota pidettiin

haastavana ja vain ajoittain toimivana. Haastavaksi kommunikaatio koettiin erityisesti tuotehallinnan suuntaan. Tuotehallintoon päin menevät viestit ovat kadonneet matkan varrella, avainhenkilöt ovat liian kiireisiä antamaan vastauksia ja kommunikointi tuotehallintoon päin on välillä täysin olematonta, mikä laskee motivaatiota. Muutostenhallintaprosessi koettiin vaikeaksi, koska se vaatii paljon kommunikaatiota muilta tahoilta tuotehallintaan päin. Vaikka ongelmia on esiintynyt, mielipiteet ovat vaihdelleet myös tuotehallinnon suhteen. Tuotekehityksen puolelta on tullut positiivista palautetta tuotehallinnon kiinni saamisesta ja tuotehallinnon korkeimpiin tahoihin kommunikaatio on toiminut hyvin.

Organisaation kommunikaatiossa nähtiin aiemmissa haastatteluissa vielä paljon parannettavaa. Virhetilanteita dokumentoivaa bugikantaa pidettiin tehokkaana ja hyvänä kommunikointikanavana tuotekehityksen ja testauksen välillä virhetilanteissa. Ongelmakohtana tuli esiin tietämättömyys muiden henkilöiden tehtävistä. Ei tiedetä, mitä kukin työntekijä tekee, jolloin ei osata kommunikoida tehokkaasti oikeiden tahojen kanssa. Rinnakkaisten projektien välinen kommunikaatio on ollut todella vähäistä ja sitä tulisi parantaa huomattavasti nykyisestä. Samaan tapaan kuin yksilöiden välinen tietämys, myös ryhmien välillä tulisi tietää enemmän, mitä kukin tekee.

Scrumin myötä organisaatiossa kommunikoinnin määrä on pakostakin kasvanut. Kasvanut kommunikoinnin määrä nähdään pääasiallisesti erittäin hyvänä asiana. Kommunikointi on lisääntynyt varsinkin tiimissä, jossa kehityksen ja testauksen välille on syntynyt entistä tiiviimpi kommunikointisuhde. Tuotehallinto on inhimillistynyt Scrumin myötä, koska tuotteen omistajan saa kiinni vähintään kahdesti sprintin aikana. Kommunikoinnin paraneminen nähdään myös helpotuksena omassa työssä, koska uskalletaan entistä herkemmin kysyä tiimin jäseniltä ongelmallisista asioista.

Vaikka kommunikointi on lisääntynyt kaikkien osapuolien välillä, aiemmin esitellyt ongelmat ovat yhä ratkaisematta. Eri projektien välisessä kommunikaatiossa ei ole tapahtunut tarvittavaa muutosta ja kokonaiskuvan hahmottaminen on hankaloitunut entisestään. Eri tiloissa työskentelevien henkilöiden kanssa kommunikointi nähdään ongelmallisena, koska kommunikoinnin määrä heidän kanssaan on lisääntynyt. Tuotehallinnon suhteen kommunikoinnin määrä on kasvanut, mutta laadullisesti taso on pysynyt samana. Haastatteluissa ilmaistiin huoli siitä, että tuotehallinto ei tiedä tarpeeksi tarkkaan, mitä pitäisi tehdä. Tällöin tiimin sisällä tehdyt päätökset vaatimuksis-

ta saattavat muuttaa niitä väärään suuntaan. Mikäli kommunikoinnin paraneminen lasketaan tapaamisten määrässä, se on muuttunut parempaan suuntaan, mutta parannettavaa löytyy yhä paljon.

5.2.9 Oma-aloitteisuus projektiryhmissä/tiimissä

Scrumin teoriassa kirjoitettiin tiimien oma-aloitteisuudesta seuraavasti: Scrum-tiimin tärkein arvo syntyy tiimin organisoitumisesta ja hallinnoinnista, roolien ylittämisestä sekä itsensä ylittämisestä. Organisoitumisella ja hallinnoinnilla tarkoitetaan tässä tapauksessa vapautta valita omat tehtävänsä ilman hallinnon puuttumista asiaan. Roolien ylittäminen tarkoittaa sitä, että tiimit tunnistavat itse jäsenistään ominaisuudet, joiden kautta se muodostaa järjestyksen toimintaansa. Jäsen, joka osaa parhaiten tehdä jonkin asian, ottaa sen omalle vastuulleen. (Leffingwell 2007, 42–43.) Haastatteluissa seurattiin alku- ja lopputilanteessa aiempien projektiryhmien ja uusien tiimien oma-aloitteisuuden astetta.

Projektiryhmissä oma-aloitteisuus vaihteli kehityspuolen ja testauksen välillä huomattavan paljon. Testauspuolen tehtävät ohjasivat yksilöitä oma-aloitteisuuteen ja itsestäänohjautuvuuteen enemmän kuin kehityspuolella. Kehityspuolella ongelmana nähtiin, että oma-aloitteisuuteen ei ole mahdollisuuksia ja tehtäväkohtaiset rajoitteet ovat suuret. Ylhäältäpäin määrätty tehtävät ovat projektipäälliköiden vastuulla, jolloin ryhmät tekevät vain, mitä heille määrätään. Haastatteluista kävi kuitenkin ilmi, että kehityksessä oma-aloitteisuus rajoittuu muutamiin yksilöihin henkilöityneiden tehtävien myötä ja pienet ryhmäkoonpanot parantavat oma-aloitteisuutta. Oma-aloitteisuus tulee tällöin esiin yksilöiden kautta tehtävistä riippuvaisena, mutta ei ryhmässä. Yleinen mielipide on myös, että oma-aloitteisuuteen tulisi olla paljon enemmän mahdollisuuksia. Paremmalla kommunikaatiolla parannettaisiin myös alihankkijoiden oma-aloitteisuutta.

Scrumin myötä tiimien oma-aloitteisuus on pakon sanelemana kokenut muutoksen. Mielipiteet siitä, onko oma-aloitteisuus ja itsestäänohjautuvuus kasvanut Scrumin myötä, jakoutuivat ääripäihin. Oma-aloitteisuuden kasvua on ollut selkeästi havaittavissa ja siihen on myös haastateltavien mukaan tarjottu mahdollisuuksia tarpeeksi ja paremmin kuin aiemmin. Kovan kiireen myötä on ollut pakko työskennellä oma-

aloitteisesti. Tämä kulminoituu positiivisena puolena erityisesti ohjelmointivirheiden korjaamisen suhteen sprintin aikana.

Myös päinvastaisia mielipiteitä esiintyi. Oma-aloitteisuus ei ole kasvanut, koska siihen ei vielä anneta mahdollisuuksia. Vaatimukset on järjestelty sprinttiin yksilöitä silmällä pitäen, jolloin jokainen on tiennyt tarkalleen, mitä heiltä odotetaan sprintin aikana. Tällöin ei ole mahdollista valita tehtäviä oma-aloitteisesti, koska omat osaamisalueet eivät saata kohdata muiden vaatimusten kanssa. Omien osaamisalueiden tehtävien puuttuminen sprinteistä on aiheuttanut turhautumista, varsinkin kun ei ole ollut aikaa kiireen takia perehtyä tarpeeksi muihin tehtäviin. Sprintin aikana suoritettavat priorisoidut tehtävät ovat olleet selkeästi yksilöityjä, minkä seurauksena tuotekehityksen katsotaan muuttuneen suorittavaksi koneeksi, joka on riippuvainen tuotteen omistajasta.

Tilanne on loppujen lopuksi lähestulkoon sama kuin aiemmin. Vielä ei ole ilmeisen selkeästi huomattavissa suurempaa vaikutusta positiiviseen tai negatiiviseen suuntaan. Oma-aloitteisuutta tulisi siis vielä parantaa, kannustaa enemmän ja kommunikoinnin kautta kasvattaa oma-aloitteisuutta. Poikkeustilanteissa tulisi käyttää myös omaa järkeä ja ottaa enemmän vastuuta oma-aloitteisesti tilanteen mukaan. Parantamisen varaa on siis huomattavissa oma-aloitteisuuden suhteen.

5.2.10 Muutos vaatimusten hallinnassa

Muutos vaatimusten hallinnassa -luku on jaettu kahteen osaan: työmäärien arviointiin ja priorisointiin. Tiimin tehtäviin kuuluu vaatimusten perusteella luoda arvio työmäärästä, joka tarvitaan vaatimusten toteuttamiseen. Tuotteen omistajalla on puolestaan vaatimuksia kohtaan tehtävänä niiden tärkeyden priorisoiminen, korkealle priorisoidut vaatimuksen suoritetaan alemman prioriteetin omaavia vaatimuksia ennen. Kysymys esitettiin haastateltavan roolin mukaisesti.

Vaatimusten työmäärien arviointi

Ennen muutosta vaatimusten työmäärien arviointi nähtiin todella hankalana tehtävänä. Arvioiden tekeminen oli puhtaasti arvailua varsinkin, jos henkilö ei tiennyt tarkalleen, mitä hänen tulee tehdä. Ajan ja kokemuksen myötä vaatimusten arviointiin pystyy saamaan tietyn näppituntuman, mutta silloinkin on täysin arpapeliä, pätevätkö arviot.

Suurien arvioiden tekemistä ja ylimalkaista päiväarviointia pidettiin suhteellisen help-
pona.

Työmäärien arviointiin vaikuttavia tekijöitä on useita. Vaatimusmäärittelyn laatu saat-
taa aiheuttaa muutoksia vaatimukseen, jolloin vaatimukset ja arviot muuttuvat. Lisä-
tehtävien ennustettavuus on mahdotonta työmääriä arvioitaessa. Vaatimuksia saattavat
olla tekemässä väärät tahot, jolloin kokemattomuus vaatimusten mukaisesta työstä
hankaloittaa arviointia. Vaikka arviot olisivat kuinka tarkkoja tahansa ja kokemuksen
kautta hyväksyttäviä, ongelmien ennakoiminen on täysin mahdotonta ja arviot menet-
tävät luotettavuutensa.

Scrum on vaikuttanut työmäärien arviointiin ristiriitaisesti. Suunnittelupäivän aikana
tiimin välinen kommunikointi ja yhteistyö ovat helpottaneet huomattavasti arvioinnin
tekemistä. Iteratiivisuus ja pienemmät arvioitavat kohteet ovat helpottaneet arviointia.
Huomioitavaa on kuitenkin vahva henkilöityminen tehtäviin, jolloin yhteinen mielipi-
de on loppuen lopuksi suorittavan osapuolen oma arvio. Vaikka työmääräarviointi
nähdään helpompana kuin aiemmin, se on yhä edelleen epäonnistunut. Arviot ovat
olleet liian optimistisia ja ei ole otettu tarpeeksi huomioon ongelmiin kuluva aikaa.
Vaikka työmääräarviot yhä heittelevät paljon, Scrumin myötä pystytään huomioimaan
ongelmat entistä aiemmin. Usko siihen, että kokemuksen kautta opitaan tarkempaan ja
parempaan arviointiin, on vahva haastateltavien keskuudessa, mutta oppimista on vie-
lä paljon jäljellä vaatimusten työmäärien arvioinnissa.

Vaatimusten priorisointi

Vaatimusten priorisointi nähtiin ennen muutosta Scrumiin hyvin ristiriitaisena toimin-
tona. Haastateltavien mielipiteet vaihtelivat hyvin voimakkaasti helposta hankalaa.
Priorisoinnissa nähtiin vaikeaksi se, että on monta pakollista vaatimusta, jotka täytyy
saada sisällytettyä projekteihin korkealle prioriteetille. Toisaalta muitten vaatimusten
prioriteettia ei tahdottaisi laskea kokonaisuuden eheyden puolesta.

Muutoksen jälkeen vaatimusten priorisoinnin nähtiin helpottuneen verrattuna aiem-
paan. Koska tietyt asiat on pakko suorittaa, että pystytään esittämään demotilaisuudes-
sa perusasioita, on helpompaa priorisoida vaatimuksia demon avulla. Vaikka on hel-
pompaa priorisoida ja muuttaa vaatimuksia, sitoutuminen tietyn vaatimuksen suorit-
tamiseen on vähentynyt. Iteratiivisuus on helpottanut priorisointia mutta tuonut muka-

naan haittapuolena näkemysten rajoittumisen vain kuukauden päähän. Yhden projektin kohdalla työ on helpottanut, mutta kokonaiskuvan muodostaminen useiden projektien osalta on vaikeutunut. Vaikeudet kokonaiskuvan hallinnassa useiden projektien osalta ovat tuoneet vaikeuksia ohjausryhmän toimintaan, joka hallinnoi kaikkia projekteja. Kommunikoinnin vähyys tuotehallinnassa on saattanut ohjausryhmän pimentoon projektien kulusta. Tuloksena on henkilöresurssien vaikeampi hallinnointi ja se, että tilanteisiin reagoiminen on hankaloitunut. Mikäli tietyllä prioriteetilla oleva vaatimus päätetään siirtää eteenpäin, saatetaan jättää huomioimatta, millainen merkitys sillä on asiakkaalle. Mikäli tiedonkulku muutoksista saavuttaisi ohjausryhmän, pystyttäisiin siirtämään henkilöresursseja nopeuttamaan muiden vaatimusten valmistumista ja tätä kautta tyydyttämään asiakkaan tarpeet.

5.2.11 Onko muutoksesta ollut hyötyä?

Loppuhaastattelun viimeinen kysymys koski opinnäytetyön kannalta tärkeintä kysymystä: onko muutoksesta ollut hyötyä. Haastatteluiden perusteella Scrum on tuonut paljon hyviä ominaisuuksia, mutta myös negatiivisia puolia on nähtävillä. Kysymyksen teki ongelmalliseksi haastatteluiden ajankohta. Kuten aiemmin on mainittu, optimaalinen aikaväli olisi yhden – kahden vuoden kuluttua muutoksen alkamisesta. Tästä syystä useiden henkilöiden oli vielä hankalaa tunnistaa konkreettisia hyötyjä tai haittoja.

Muutos Scrumiin on ollut organisaatiota virkistävä toiminto, joka on kasvattanut työntekijöiden motivaatiota. Selkeän rytmin saaminen omaan työhön ja oman työn konkreettisempi suunnittelu kuukaudeksi eteenpäin ovat myös lisänneet motivaatiota.

Ammattitaito kehittämissuunnitelmien läpivientiin on kohentunut uuden asian oppimisen ansiosta. Motivaatiota parantaa myös yksilöiden kaipaama työnteon rauhoittuminen ja selkeytyminen. Lisäksi kommunikaation paraneminen on motivoinut työntekijöitä.

Kommunikaation paraneminen on tuonut muitakin hyötyjä kuin motivaation kasvamisen. Se on antanut tiimeille selkeämmän kuvan siitä, missä mennään, jolloin seuranta sprintin on parantunut. Keskustelut muiden tiimin jäsenten kanssa ovat helpottaneet omaa työskentelyä ja suunnittelupäivä on opettanut ymmärtämään, mitä muut henkilöt tekevät. Informaation kulusta on ollut jokaiselle osapuolelle todellista hyötyä. Kommunikointia on tapahtunut tiiviimmin myös asiakkaaseen päin tuotehallinnosta. Keskustelukanava asiakkaaseen päin on voitu avata aikaisemmin konkreettisten asioiden

kerä. Tuotteen omistaja on ollut tiiviimmässä yhteydessä asiakkaaseen päin vaatimusten selventämisessä. Tällöin ei tule turhia ominaisuuksia ja turhaa työtä tiimille, vaan voidaan keskittyä olennaisiin toiminnallisuuksiin. Asiakkaille pystytään Scrumin myötä lupaamaan asioita ja lupaukset pystytään pitämään entistä paremmin vaatimusten priorisoinnin avulla. Muutos lyhyisiin iteraatiosykleihin on parantanut tuotehallinnon reagoitokykyä, mistä on asiakkaalle hyötyä. Scrumista on ollut hyötyä ennen kaikkea sprinttien seurantaan. Vaikka ennustettavuus on heikentynyt pidemmällä aikavälillä, se on parantunut yhden kuukauden osalta huomattavasti. Enää ei voida piiloutua kolmeksi kuukaudeksi työskentelemään, vaan läpinäkyvyys on tullut mukaan entistä voimakkaammin. Läpinäkyvyys on paljastanut ohjelmistokehitystä haitanneita esteitä ja poikkeamista saadaan tieto entistä herkemmin. Myös jokapäiväinen tieto sprintin seurannasta on ollut tarkemmin ja laajemmin nähtävillä kuin aiemmin. Parannusta on tapahtunut aiemmin esiteltyjen asioiden lisäksi myös henkilöressurssien organisoinnissa ja työmäärän arvioinnissa. Avoimia ongelmia sovelluksessa on ollut vähemmän ja tuotteen omistaja näkee kasvokkain tiimin ainakin kahdesti sprintin aikana. Myös entiset vesiputousmallin raskaat hallinnointimenetelmät ovat kokeneet tarpeelliseksi nähdyn kevennyksen.

Haastateltavien mukaan hyötyä on ollut yhtä paljon kuin haittaakin. Varsinaisia haittoja on tullut tiimin tehon vähenemisenä, jonka uskotaan kehittyvän kuntoon ajallaan. Ennustettavuuden supistuminen kuukauden pituiseksi on ollut negatiivinen ominaisuus, mutta kokemuksen kautta tämänkin uskotaan parantuvan. Yksilöiden kannalta negatiivista on ollut useissa samanaikaisissa projekteissa työskentely ja se, että työt ovat vaihdelleet liian paljon kuukauden aikana. Tehtävien jakamisessa ei ole nähty tapahtuneen vielä kukaan mitään uudistusta, joten henkilöityminen on yhä pinnalla. Hallinnoinnin osalta negatiivista on ollut projektin kustannusten ennustettavuuden vaikeutuminen ja aikataulujen venyminen.

Luottamus Scrumin hyödyllisyyttä kohtaan on korkealla. Toimintapisteinä hyödyn lisäämiseksi haastateltavat näkivät ketterien menetelmien käytänteiden kehittämisen toimiviksi, testausympäristöjen valmistelun ennen sprinttiä ja osallistumisen useampiin demoihin kuin vain omaan. Toiveissa on, että tuotehallinto kehittyisi tehokkaammaksi ja alkanut muutos jatkuisi eikä jäisi puolitiehen, mikä pienentäisi tehokkuutta.

6 JOHTOPÄÄTÖKSET JA POHDINTA

Johtopäätökset ja pohdinta -luku on opinnäytetyön viimeinen luku. Se sisältää kaksi pääotsikkotasoa: päätelmiä tuloksista ja tutkimusprosessin analysointi. Ensimmäinen luku sisältää pohdinnanomaisia päätelmiä muutoksesta ja Scrumin osa-alueista. Toinen luku sisältää analyysin tutkimuksesta kohdistuen tutkimuksen ongelmiin, hyödynnettävyyteen ja jatkotutkimukseen.

6.1 Päätelmiä tuloksista

Muutos

Muutos ei ole koskaan täysin kivuton prosessi. Toisaalta kun huomataan kipua prosesseissa, voidaan olla varmoja, että muutosta on tapahtunut. Kaikki muutoksen vaikutukset eivät tule hetkessä esiin, vaan kestää aikaa tottua ja mukautua muutokseen.

Vaikka uusia asioita ei heti ymmärtäisikään, lyhyellä aikavälillä on tärkeää saada positiivinen kuva asiasta, jolloin muutosvastarintaa ei pääse syntymään. Yksilön muutos vaatii vahvasti tukea organisaatiolta. Kun tuki on kohdallaan, ei muutoksesta tule ongelmia itselle, ja se nähdään helppona. Muutoksen tukena organisaation on hyvä käyttää sitä ajavien henkilöiden lisäksi ulkopuolista tukea.

Organisaation tulisi kehittyä jatkuvasti, mutta pikemminkin pienin askelin kuin kaikilla kerralla. Tällöin minimoidaan riski rikkoa liikaa paikkoja yhdellä kertaa prosessista. Kun muutos on suoritettu onnistuneesti yhteen osaan prosessia, voidaan siirtyä seuraavaan muutokseen. On huomioitava, että tarvitaan jatkuvaa arviointia tapahtuneista ja tulevista muutoksista. Mikäli muutoksen takana on vahva organisaatio, voivat suuretkin turbulenssia aiheuttavat muutokset onnistua pitkällä aikavälillä. Tällöin täytyy uskoa omaan tekemiseensä ja yrityksessä on oltava johtohahmo viemässä muutosta eteenpäin ja kehittämässä sitä. Empiirinen malli sanookin, että kokeilun kautta oppii. Tämä pätee varmasti myös muutoksessa. Organisaation on ensin tunnistettava, millainen kenttä sillä on, miten se reagoi tiettyihin asioihin ja selvittää kompastuskivet, minkä jälkeen voidaan sopeutua.

Testaus

Testauksen ongelmana on se, että heille tulee jatkuvasti enemmän tekemistä ja kaikkia virheitä ei ehditä korjata kuukauden aikana. Mikäli ei ole alusta lähtien täysin selkeä kuva vaatimuksista ja siitä, kuinka ne tulisi testata, ei päästä miettimään, mitä voi mennä vikaan. Tästä syystä myös kommunikointi kehittäjiä kanssa on tärkeää, koska vain siten saadaan selville, kuinka vaatimus toteutetaan, ja ymmärretään, mitä pitää testata. Tällöin opitaan myös soveltamaan itse ja löytämään vikoja sekä pystytään integroimaan testattavat asiat paremmin suurempaan kokonaisuuteen. Ympäristöjen asentamiset ja kaikki muut häirittelevät tulee karsia pois, jolloin pystytään keskittymään olennaiseen asiaan koko sprintin ajan. Järkevin tilanne syntyy, kun pystytään automatisoimaan testausta, jolloin myös testauksen jatkuva määrä pysyy suhteellisen vakiona. Mikäli testaajat ovat useissa projekteissa, heille tulee yliannostus tietoa, jolloin ei ymmärretä täysin, mitä tapahtuu ja kokonaiskuva testattavista asioista hämärtyy. Tällöin sitoutuminen ja käytettävä aika tiettyyn asiaan vähenevät ja huolimattomuus kasvaa, jolloin virheet menevät asiakkaalle testattaviksi.

Tuotteen omistaja

Scrumin myötä tuotteen omistajan tulisi pystyä fokuusoimaan enemmän asiakkaiden suuntaan. Suunnittelupäivän aikana on mahdollista suunnitella tehtävät sille tasolle, että tiimi pystyy oma-aloitteisesti työskentelemään sprintin ajan. Tällöin tuotteen omistajan ei tarvitse olla seuraamassa koko ajan tiimin etenemistä, vaan hän voi uhrata enemmän aikaa asiakkaalle. Tuotteen omistajan kontaktissa asiakkaaseen on tapahtunut parannusta, koska saadaan helpommin konkreettista näytettävää yhä aikaisemmassa vaiheessa. Hankaluutena on lupauksien pitäminen asiakkaalle toteutettavista toiminnallisuuksista. Empiirinen malli sovelluskehityksessä saa aikaan sen, että aikataulut muuttuvat kuten ennenkin. Tämän lisäksi tiimi saattaa huomata, että ei pysty toteuttamaan luvattua asiaa tietyssä ajassa, ja pahimmillaan tuotteen omistajan lupaus venähtää yhden julkaisusyklin eteenpäin. Tiimi ei ymmärrä, kuinka ominaisuuksien pudottaminen voi vaikuttaa asiakkaaseen asti, koska he eivät näe kokonaiskuvaa asiakkaan silmin vaan omasta näkökulmastaan suorittavana osapuolena.

Kommunikointi

Kommunikointia tulisi tapahtua tehokkaammin tiimin ja testauksen välillä. Tuotteen omistajan suuntaan täytyy saada tehostettu yhteys, mutta kommunikaation ei tule olla

jatkuvaa. Demon ja suunnittelupäivän tulisi riittää. Demotilaisuudessa tulisi saada asiakas enemmän mukaan keskustelemaan, kuinka kehittää tuotetta, eikä vain mainostaa, mitä on saatu aikaiseksi. Tuotteen omistajan on helpompi keskittyä omiin töihin, kun tiimi on mukana asiakkaan kanssa keskustelemassa tuotteesta. Tällöin myös tiimille muodostuu parempi kuva kehitettävän tuotteen tulevaisuudesta ja sitoutuminen paranee.

6.2 Tutkimusprosessin analysointi

Tutkimuksen ongelmat

Tutkimuksen kannalta suurimmaksi ongelmaksi muodostui aikaväli haastatteluiden välillä. Muutos on hidasta, joten lopetushaastattelun ajankohta olisi pitänyt olla myöhemmin kuin toteutettu ajankohta. Vastaavanlaisessa akateemisessa tutkimuksessa (Mann, C. & Maurer, F. 2005. A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction.) haastatteluiden aikaväli on ollut kaksi vuotta. Opinnäytetyön aikatauluun ei ollut mahdollista järjestää haastatteluja yhtä pitkällä aikavälillä. Neljän kuukauden aikana muutos on saanut aikaiseksi monia asioita, mutta ei tarpeeksi, jotta mielipiteet olisivat ehtineet tekeytyä muuttumattomiksi. Muutoksen aikaisessa vaiheessa haastateltavien mielipiteet muuttuvat kokemusten myötä jatkuvasti ja tulokset perustuvat yhden päivän mielipiteisiin, jotka seuraavana päivänä saattavat olla täysin toisenlaiset. Pidemmällä aikavälillä mielipiteet olisivat varmasti ehtineet juurtua jolloin olisi saatu luotettavia tuloksia.

Tutkimuksen toisena ongelmana on haastateltavien määrä. Saatavilla olevien henkilöiden määrä rajoitettiin pilottiprojekteissa oleviin henkilöihin, joka on looginen rajoitus. Toisaalta, mikäli haastateltavien määrää olisi saanut kasvatettua suuremmaksi, virhemarginaali pienenesi ja mielipiteitä/kehitysideoita olisi kerääntynyt enemmän.

Tutkimustulosten hyödynnettävyys ja yleistettävyys

Opinnäytetyön empiirinen osuus toteutettiin toimeksiantajalle, joten hyödyn tutkimustuloksista saa myös ennen kaikkea toimeksiantaja. Tutkimustuloksille saadaan kuitenkin tukea muista tapauskohtaisista tutkimuksista, kuten aiemmin viitatussa Mann, C. & Maurer, F. 2005. A Case Study on the Impact of Scrum on Overtime and Customer

Satisfaction. Opinnäytetyöhön kuuluva teoriaosuus puolestaan on täysin yleistettävissä ja hyödynnettävissä, koska tieto perustuu lähteistä saataviin korkeatasoisiin materiaaleihin.

Jatkotutkimusaiheet

Jatkokehityksen kannalta oleellisin asia on seuranta pidemmällä aikavälillä. Seuranta tulisi jakaa kahteen osaan: muutokseen ja Scrum-prosessiin. Organisaation muutosta tulisi tutkia pidemmällä aikavälillä ja muutoksen skaalaa tulisi laajentaa. Tutkimukseen tulisi lisätä asiakasrajapinta ja määritellä laajemmin muutoksesta odotettavat hyödyt. Hyötyjen seuranta ja niiden mittaaminen toisi tutkimukseen lisäksi kvantitatiivisen otteen. Muutoksen kokeneita henkilöitä tulisi myös tutkia pidemmällä aikavälillä. Ovatko mietteet muutoksesta muuttuneet vuoden tai kahden jälkeen? Seurannassa tulisi tarkastella Scrum-prosessia pidemmällä aikavälillä. Jatkon kannalta oleellista olisi selvittää, millä tavoin Scrumin viitekehystä on muutettu ajan myötä ja mitkä osat siitä ovat muuttuneet. Tällöin saataisiin pidemmältä aikaväliltä tietoa Scrum-prosessin elämisestä organisaatiossa ja viitekehysten muutoksesta. Hyödynnettävyyden kannalta useamman vuoden seuranta prosessin kehittymisestä antaisi entistä tarkemman kuvan Scrumista osana ohjelmistokehitystä.

LÄHTEET

Beck, K. & Beedle, M. & Bennekum, A, Van etc 2001. Agile Manifesto.

[Http://agilemanifesto.org/](http://agilemanifesto.org/)

Bird, C. 2006. Process guidance for team system. Viitattu 22.1.2008, 19.2.2008.

[Http://www.scrumforteamssystem.com/ProcessGuidance/](http://www.scrumforteamssystem.com/ProcessGuidance/)

Clinton, K. 2006. Agile game development. Viitattu 17.1.2008..

[Http://www.agilegamedevelopment.com/images/complexity_production.jpg](http://www.agilegamedevelopment.com/images/complexity_production.jpg)

Haikala, I. & Märijärvi, J. 2002. Ohjelmistotuotanto. Helsinki: Talentum Media.

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. Helsinki: Talentum.

Hannus, J. 1994. Prosessijohtaminen: Ydinprosessien uudistaminen ja yrityksen suorituskyky. Jyväskylä: Gummerus Kirjapaino.

Hirjärvi, S. & Remes, P. & Sajavaara, P. 2004. Tutki ja kirjoita. Helsinki: Kustannusosakeyhtiö Tammi.

Kniberg, H. 2006. Scrum and XP from the Trenches.

[Http://www.infoq.com/minibooks/scrum-xp-from-the-trenches](http://www.infoq.com/minibooks/scrum-xp-from-the-trenches).

Leffingwell, D. 2007. Scaling software agility: best practices for large enterprises. United States: Pearson Education, Inc.

Petäjä, M. & Koponen, E. & Lautamatti, L. & Bauer, M. 2000. Muutosprosessien ohjaaminen: Aikuiskouluttajan opas. Työvoimaministeriö.

[Http://verkkolehti.peda.net/jyu/img/oplaa/muutosprosessien_ohjaaminen.pdf](http://verkkolehti.peda.net/jyu/img/oplaa/muutosprosessien_ohjaaminen.pdf)

Royce, W. 1970. Managing the development of larger software systems. Viitattu

12.3.2008. [Http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf](http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf)

Schuh, P. 2005. Integrating agile development in the real world. United States: Hingham (MA) : Charles River Media.

Schwaber, K. & Beedle, M. 2002. Agile software development with Scrum. United States: Upper Saddle River (NJ) , Prentice Hall.

Schwaber, K. 2007. The Enterprise and Scrum. United States: Microsoft Press.

Schwaber, K. 2006. Introduction to Scrum Viitattu 19.2.2008.
[Http://www.controlchaos.com/resources/index.php](http://www.controlchaos.com/resources/index.php).

Schwaber, K, & Leffingwell, D. & Smits, H. 2005. A CIO's Playbook for Adopting the Scrum Method of Achieving Software Agility. Rally Software Development Corp.

Viitattu 1.3.2008.

[Http://www.leffingwell.org/Document_Store/CIO_Playbook_For_Adopting_Scrum_080805.pdf](http://www.leffingwell.org/Document_Store/CIO_Playbook_For_Adopting_Scrum_080805.pdf).

Tietoa Landis+Gyristä. 2006. Landis+Gyr -sivusto. Viitattu 5.3.2008.

[Http://landisgyr.fi/](http://landisgyr.fi/)

LIITTEET

Liite 1. Ketterän manifestin käytänteet.

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. (Beck, K. & Beedle, M. & Bennekum, A, Van etc 2001. Agile Manifesto.)

Liite 2. Ketterän ohjelmistokehityksen käytänteet

1. Jatkuva integrointi.

Jatkuvan integroinnin tarkoituksena on, että kehittäjä merkitsee versionhallintaan sisään jokaisen valmistuneen toiminnallisuuden sen valmistuessa. Tällöin ongelmien löytäminen nopeutuu huomattavasti, estää kehittäjää lähtemästä sivuraiteille toiminnallisuuden kehittämisessä ja järjestelmän versio pysyy puhtaana. (Schuh 2005, 76–80.)

2. Automaattinen build

Automaattinen buildin tekeminen vapauttaa aikaa kehittäjältä, kenen on aiemmin täytynyt tehdä päivittäinen tai viikoittainen rituaalinsa buildin kokoamisessa. Tämä vapauttaa kyseisen kehittäjän aikaa hyödyllisempiin tehtäviin ja poistaa buildin rakentamisen henkilökeskeisyyden. Ajan säästämiseksi kehittäjä voi verrata uuden toiminnallisuuden kääntymistä uusimpaan buildiin. Hänen ei tarvitse käyttää aikaa muiden kehittäjien kelpaamattomien toiminnallisuuksien kääntämiseen. (Mts. 71–74.)

3. Automaattinen käyttöönotto

Automaattinen käyttöönotto on automaattisen buildin kanssa samantyyppinen, nopeutava ja työvoimaa säästävä toimenpide. Sillä tarkoitetaan esimerkiksi testaukseen käytettävän palvelimen asentamista uusimmalla sovellusversiolla automaattisesti. Tällöin uuden toiminnallisuuden integroiminen järjestelmään voidaan testata koko järjestelmän osalta tehokkaammin.

4. Refaktorointi

Refaktoroinnin tarkoituksena on saada muut ketterän ohjelmistokehityksen käytänteet toimimaan paremmin. Sen perusolemus on muuttaa järjestelmän suunnittelua ilman toiminnallisuuksien muuttamista. Tällöin suunnittelu paranee, kaksinkertaisuudet poistuvat koodista, koodi yksinkertaistuu sekä operaatioiden uudelleenkäytettävyys paranee. (mts. 92–94.)

5. Koodin kollektiivinen omistaminen

Kollektiivisella omistamisella vältetään toiminnallisuuksien sekä osa-alueiden yksilöitymistä. Kun kehittäjät ymmärtävät ja osaavat kehittää eri osa-alueita, säästytään kaksinkertaisuuksilta koodissa, säästetään aikaa, järjestelmän laatu paranee kun useampi henkilö seuraa tapahtuvia muutoksia ja edistetään tiimin välistä kommunikaatiota. Hankalissa tilanteissa voidaan hyödyntää useiden henkilöiden osaamisesta. (mts. 83–87.)

6. Pariohjelmointi

Pariohjelmoinnin tarkoituksena on, että kehitys tapahtuu pareittain. Yhdistämällä kaksi henkilöä saadaan laadukkaampi ja vähemmän virheitä sisältävä lopputulos. Lähtökohta on ihanteellinen uuden työntekijän opettamiseen ja tietoa jakautuu laajemmin tiimissä. Pariohjelmointi vähentää henkilökeskeisten osa-alueiden syntymistä ja se toimii motivoivana lähteenä vastuullisten tehtävien sitoutumiseen. (mts. 95–99.)

7. Yksinkertainen suunnittelu

Vaatimusten muuttuminen nopealla aikavälillä johtaa siihen, että suunnitteluun käytetty aika, voi hyvinkin nopeasti näkyä pääoman tuottoasteessa negatiivisena. Tästä syystä yksinkertainen suunnittelu parantaa koodin muokattavuutta, omaksumista, käyttönotettavuutta ja parantaa koko tiimin itsevarmuutta työtään kohtaan. (mts. 80–83.)

8. Automaattinen yksikkötestaus

Automaattinen yksikkötestaus tarkoittaa sitä, että jokaiselle toiminnallisuudelle tehdään testejä, jotka suoritetaan muutosten jälkeen. Kun yksikkötestejä voidaan suorittaa jatkuvasti, nopeutuu virheiden löytäminen, tiimin usko sovelluksen toimivuuteen kasvaa, koodin virheettömyys paranee ja testit toimivat jatkossa regressiotestinä. (mts. 108–113.)

9. Hyväksyttävyytestaus (järjestelmättestaus)

Hyväksyttävyytestaus toimii toiminnallisuuksien testaamisena. Sen lähtökohtana toimii parhaimmassa tapauksessa asiakkaan laatimat käyttötapaukset, joiden toimintaa testataan. Tarkoituksena ei ole satunnaisesti etsiä virheitä, vaan toimia mekanismina, todeta toiminnallisuuden toimimisesta. Hyväksyttävyytestauksen automatisoinnin

etuna on, että kasvavan järjestelmän työpanos testaukseen pysyy vakiona, koska aiemmat testaukset todetaan automaattisesti. Myös kehityksen laatu paranee koska, uusi ominaisuus ei katsota olevan valmis mikäli se ei läpäise testiä. (mts. 113–116.)

10. TDD, testilähtöinen kehitys (Test Driven Development)

Testilähtöisen kehityksen ideana on tuottaa kaikesta järjestelmään toteutettavista toiminnallisuudesta testaus, mieluusti automatisoitu. Testilähtöisyys estää testaamattomien toiminnallisuuksien syntymistä järjestelmään ja yksinkertaistaa suunnittelua. Testauksen ollessa kattavaa, kehittäjä voi muokata toiminnallisuuksia ilman pelkoa, että se jää testaamatta. (mts. 116–118.)

Liite 3 Aloituskyselyn kysymykset

1. Kuinka hyvin yleinen kommunikaatio on toiminut tuotehallinnon – kehityksen – testauksen – muiden tahojen välillä? Saatko tarvittavat henkilöt kiinni?
2. Koetko että projektiryhmät ovat oma-aloitteisesti ja itsestään ohjautuvia?
3. Koetko vaatimusten työmääräarvioinnin/priorisoinnin hankalaksi, miksi?
4. Millaisena koet nykyisen toimintatavan, vesiputousmallin yrityksessä?
 - 4.1 Mitkä ovat vesiputousmallin hyvät puolet?
 - 4.2 Mitkä ovat vesiputousmallin huonot puolet?
5. Mitä tulisi muuttaa nykyisessä mallissa?
6. Millaisia odotuksia sinulla on Scrumia kohtaan?

Liite 4 Lopetuskyselyn kysymykset

1. Millaisia ajatuksia/tuntemuksia sinulla on henkilökohtaisesti muutoksesta Scrumiin? Positiivisia/negatiivisia ajatuksia?
2. Kuinka olet itse sopeutunut muutokseen? Onko muutokseen ollut helppoa vai vaikeaa sopeutua?
3. Kuinka näet organisaation sopeutuneen muutokseen? Onko muutokseen ollut helppoa vai vaikeaa sopeutua?
4. Onko yleinen kommunikaatio parantunut tuotehallinnon – tiimin ja muiden tahojen välillä? Saatko tarvittavat henkilöt kiinni?
5. Onko tiimien oma-aloitteisuus kasvanut?
6. Onko vaatimusten työmäärien arviointi/priorisointi helpottunut vai vaikeutunut?
7. Mikä Scrumin toiminnoista on ollut hyödyllisin/vähiten hyödyllinen (planning day, daily Scrum, demo, retrospective), Miksi?
8. Onko muutoksesta Scrumiin ollut hyötyä vai ei?