

Joni Luomala

**KETTERIEN MENETELMIEN KÄSITYS  
PROJEKTIHALLINNASTA: ESIMERKKINÄ SCRUM**

Tietojärjestelmätieteen  
kandidaatintutkielma  
30.3.2009



**JYVÄSKYLÄN YLIOPISTO  
TIETOJENKÄSITTELYTIETEIDEN LAITOS**

## TIIVISTELMÄ

Luomala, Joni Petteri

Ketterien kehitysmenetelmien käsitys projektinhallinnasta: esimerkkinä Scrum  
/ Luomala Joni Petteri

Jyväskylä: Jyväskylän yliopisto, 2009.

31 s.

Kandidaatintutkielma

Yhä useammat ohjelmistojenkehitysprojektit toteutetaan käyttäen perinteisten suunnittelulähtöisten menetelmien sijasta ketteriä menetelmiä. Ketterät menetelmät ovat asiakaslähtöisiä menetelmiä, joissa asiakas on tiiviisti mukana järjestelmän kehittämisessä. Ketterille menetelmille on ominaista toimivan ohjelman tuottaminen lyhyissä tuotantokierroksissa, joiden jälkeen esitetään asiakkaalle aikaansaannokset ja asiakas pääsee esittämään kommentteja syntyvästä järjestelmästä ja sen vastaavuudesta vaatimuksiin.

Tässä tutkielmassa perehdytään ketterien menetelmien ja tarkemmin Scrum-menetelmän käsityksiin projektinhallinnasta. Tutkielmassa esitellään kolme yleistä jäsenystä projektinhallintaan, muutamia näkemyksiä ketterien menetelmien projektinhallinnasta sekä tutkitaan kirjallisuuskatsauksen avulla, miten ketterät menetelmät yleensä sekä tarkemmin Scrum täyttävät yleisimpien projektinhallinnan jäsenyyksien määrittämiä projektinhallinnan käytänteitä.

Tutkimuksesta käy ilmi, että vaikka ketterissä menetelmissä lähtökohdat projektinhallintaan ovat hyvin erilaisia, kuin yleisissä projektinhallinnan jäsenyyksissä, niin useimmat yleisistä käytänteistä voidaan täyttää käyttämällä ketteriä menetelmiä.

AVAINSANAT: Projektinhallinta, ketterät kehitysmenetelmät, Scrum, PMBOK, CMMI

Ohjaaja: Mauri Leppänen

Tietojenkäsittelytieteiden laitos

Jyväskylän Yliopisto

Tarkistaja: Mauri Leppänen

Tietojenkäsittelytieteiden laitos

Jyväskylän Yliopisto

# SISÄLLYSLUETTELO

1 JOHDANTO .....	5
2 PROJEKTIN HALLINTA .....	8
2.1 Projekti ja projektinhallinta .....	8
2.2 Project Management Body of Knowledge (PMBOK) .....	9
2.3 CMMI projektinhallinnan prosessialueet.....	11
2.4 Valittava näkökulma .....	14
3 PROJEKTINHALLINTA KETTERISSÄ MENETELMISSÄ .....	15
3.1 Agile Project Management .....	15
3.2 Ketterät menetelmät vs. Project Management Body of Knowledge .....	19
4 PROJEKTINHALLINTA SCRUM - MENETELMÄSSÄ .....	21
4.1 Scrum.....	21
4.2 Scrum vs. Project Management Body of Knowledge .....	24
5 YHTEENVETO .....	28
6 LÄHTEET .....	30

# 1 JOHDANTO

Viimevuosien aikana ketterien kehitysmenetelmien käyttö on yleistynyt dramaattisesti. (Lindvall ym. 2004) Myös menetelmien määrä on lisääntynyt eikä merkkejä lisääntymisen loppumisesta ole näkyvissä. Tästä johtuen tutkijoiden ja menetelmien käyttäjien on vaikea pysyä perässä, mikä menetelmä sopii mihinkin tarkoitukseen parhaiten. (Abrahamson ym. 2003) Ketterillä kehitysmenetelmillä tarkoitetaan ohjelmistojen kehitysmenetelmiä, jotka pystyvät vastaamaan ohjelmistoprojektien muutoksiin paremmin ja nopeammin kuin tavalliset, niin sanotut suunnitelmaohjautuvat prosessimallit. Ketterien menetelmien muita ominaispiirteitä ovat asiakaslähtöisyys, toimivien ohjelmiston osien julkaisu lyhyiden kehitysjaksojen välillä sekä menetelmän oppimisen helppous. (Abrahamson ym. 2003) Ketterien kehitysmenetelmien perusarvot on kirjattu ”Ketterän ohjelmistokehityksen manifestiin” (Agile Manifesto 2001). Ketteristä menetelmistä suosituimpia ovat Scrum, eXtreme programming sekä näiden kahden erilaiset risteytykset.

Monet pienet yritykset suosivat ketteriä menetelmiä, koska kokevat perinteiset menetelmät monimutkaisiksi, byrokraattisiksi ja joustamattomiksi. Samoihin näkökantoihin on törmätty tutkittaessa myös isojen yritysten tarpeita. Myös isoilla yrityksillä on paineita parantaa tuottavuutta ja siksi etsiä uusia tapoja ohjelmiston kehittämiseen. Perinteisten menetelmien suurimpina ongelmina on se, että käyttäjien vaatimuksia ei saada selville tarkasti, ja aikataulut pakottavat aloittamaan ohjelmiston toteuttamisen ennen kuin kaikkia vaatimuksia on saatu kartoitettua. Tällaiset ongelmat ajavat organisaation koosta riippumatta etsimään uusia tapoja. Myös suurissa organisaatioissa on otettu onnistuneesti käyttöön ketteriä menetelmiä. (Lindvall ym. 2004)

Projektinhallinnan näkökulmat eri menetelmien välillä eroavat toisistaan laajalti. Myös yleisesti ketterien menetelmien projektinhallintaa käsittelevät tutkimukset eroavat toisistaan. Toisissa jäsenyksissä keskitytään pelkästään

projektin läpiviennin hallintaan, kun toisissa otetaan lisäksi kantaa myös teknisen toteutuksen hallintaan. Yleisesti ketterien menetelmien projektinhallinnalle on yhteistä asiakaslähtöisyys, pienet tiimikoot, itsestään ohjautuvat tiimit, kevyt johtamistyyli ja teknisempinä asioina testilähtöinen kehitys, jatkuva integraatio sekä tiimin kesken yhdessä tehtävä kehitystyö. Menetelmien välillä on kuitenkin eroja. Jotkut menetelmät eivät ota kantaa projektinhallintatapoihin, kun taas esimerkiksi Scrum on kehitetty helpottamaan projektien hallintaa. (Lindvall ym. 2004) Scrumin kehittäjät pohtivat, että ohjelmistojen kehittäminen on monimutkainen ja arvaamaton prosessi, jonka on sopeuduttava muuttuvaan ympäristöön. (Marçal ym. 2008)

Barnes (1990) on määritellyt projektinhallinnan keskeisimmät osuudet seuraavasti: "Projektinhallinnan avulla organisoidaan joukko henkilöitä toimimaan siten, että projektille asetetut tavoitteet saavutetaan, ja työ saadaan päätökseen." Ruuskan (2001) mukaan projektinhallinta voidaan jakaa kahtia ohjausprosessiin ja toteutusprosessiin. Toteutusprosessi tähtää suoranaisesti projektin lopputuloksen aikaansaamiseen. Ohjausprosessin avulla taas pyritään saavuttamaan laadullisesti tavoitteiden mukainen lopputulos mahdollisimman tehokkaasti.

Tässä tutkielmassa esitellään yleisen määritelmän lisäksi kaksi jäsenystä: Project Management Body of Knowledge (PMBOK) sekä CMMI - projektinhallinnan prosessialueet. PMBOK on Project Management Instituten määrittelemä joukko hyväksi havaittuja käytänteitä, joita organisaatiot voivat ottaa käyttöön oman harkintansa mukaan. (Sliger 2007) CMMI on kypsyyssmalli prosessien kehittämiseen tuotteiden ja palveluiden tuotannossa. Myös CMMI koostuu hyväksi havaituista käytänteistä. (Marçal ym. 2008) Projektinhallinnan prosessialueet ovat vain yksi osa kypsyyssmallia. Tutkielman tarkoituksena on kirjallisuuskatsauksen avulla ottaa selvää, millainen on ketterien menetelmien näkemys projektinhallinnasta, sekä millainen on Scrum-menetelmän näkemys projektinhallinnasta. Tutkielma on jaettu kolmeen osaan. Ensimmäisessä osassa

(Luku 2) esitellään projektinhallintaa yleensä. Toisessa osassa (Luku 3) keskitytään ketterien menetelmien käsitykseen projektinhallinnasta, esitellään Agile Project Managementia ja tarkastellaan sen suhdetta PMBOK:in käytänteisiin. Luvussa 4 esitellään tarkemmin Scrum-menetelmää ja sen projektinhallintaa sekä tarkastellaan sen suhdetta PMBOK:iin. Viimeisessä luvussa esitellään yhteenveto ja kerrotaan tutkielman tärkeimmät johtopäätökset ja tutkimustulokset.

## 2 PROJEKTIN HALLINTA

Tässä luvussa esitetään määritelmiä projektinhallinnalle eri kirjallisuuslähteiden mukaan. Yleisen määrittelyn lisäksi esitellään kaksi jäsenystä projektinhallintaan: (a) Project Management Body of Knowledge, joka on Software Management instituten määrittelemä joukko hyväksi havaittuja käytänteitä, ja (b) CMMI projektinhallinnan prosessialueet, joka on CMMI-DEV kypsyysmallin projektinhallintaa käsittelevien prosessialueiden joukko.

### 2.1 Projekti ja projektinhallinta

Ruuska (2001) kuvaa kirjassaan projekteja joukoksi ihmisiä ja muita resursseja, jotka on tilapäisesti koottu suorittamaan tiettyä tehtävää. Yleisestikin projektiin liitetään kertaluontoisuus, ainutlaatuisuus sekä riskit. Projektinhallinnasta on alan kirjallisuudessa useita määritelmiä. Ruuskan (2001) mukaan projektinhallinta perustuu projektisuunnitelmaan, joka on projektin toiminnan kannalta keskeinen asiapaperi. Linjaorganisaation hallintaan verrattuna projektinhallinta eroaakin juuri epävarmuuden, taloudellisen riskin sekä projektin ainutlaatuisuuden osalta. Linjaorganisaation johtaminen voidaan usein perustaa siihen, miten ennen on toimittu. Epävarmuudesta ja dynaamisuudesta huolimatta projektin alkuperäinen tavoite ei yleensä ratkaisevasti muutu projektin kuluessa. Projektinhallinnalta vaaditaankin ennakoimista ja valmiutta reagoida muutoksiin. Projektinhallinta voidaan jakaa kahtia ohjausprosessiin ja toteutusprosessiin. Toteutusprosessi tähtää suoranaisesti projektin lopputuloksen aikaansaamiseen. Ohjausprosessin avulla taas pyritään saavuttamaan laadullisesti tavoitteiden mukainen lopputulos mahdollisimman tehokkaasti. Ruuska (2001) esittää myös toisen kahtiajaon, jonka mukaan projektinhallinta voidaan jakaa koviin (management) ja pehmeisiin (leadership) tekniikoihin. Koviin tekniikoihin kuuluu projektinhallinnan käytännölliset asiat, kuten aikataulujen ja kustannusten



laadinta, valvonta ja seuranta sekä laadunvalvonta. Pehmeissä teknikoissa taas on kyse ihmisten johtamisesta, vuorovaikutuksesta ja viestinnästä.

## **2.2 Project Management Body of Knowledge (PMBOK)**

PMBOK (IEEE Computer Society 2004) määrittelee projektin lähes samalla tavalla kuin Ruuska. Projekti on väliaikainen tavoite luoda ainutkertainen tuote tai palvelu. Projekteja toteutetaan organisaation kaikilla tasoilla, ja projekti saattaa olla yhden henkilön toteuttama tai saattaa sisältää jopa tuhansia työntekijöitä ja voi olla kestoltaan muutamasta tunnista moneen vuoteen. Väliaikainen tarkoittaa sitä, että jokaisella projektilla on määritelty alku ja loppu. Projektin loppu saavutetaan, kun tavoitteena ollut tuote tai palvelu on valmis, tai kun huomataan, että projektia ei saada toteutettua olemassa olevilla resursseilla. Projektien tarkoituksena on tuottaa jotakin, mitä ei ole aikaisemmin tehty. Projekti voi olla ainutlaatuinen, vaikka se kuuluisi suurempaan kategoriaan. Esimerkiksi toimistorakennuksia on rakennettu tuhansia, mutta silti jokainen uusi rakennusprojekti on jollakin tapaa ainutkertainen. PMBOK:in mukaan projektinhallinta on sovellus tietämystä, taitoja, työkaluja ja tekniikoita projektin toimintojen sovittamiseksi projektin vaatimuksiin. Projektinhallinta kuuluu jokaiseen projektin vaiheeseen.

PMBOK jakaa projektinhallinnan kahteen alueeseen, jotka ovat projektinhallinnan viitekehys ja projektinhallinnan tietämyksen alueet. Projektinhallinnan viitekehys tarjoaa perusrakenteen projektinhallinnan ymmärtämiselle. Se jakautuu projektinhallinnan kontekstiin, joka määrittelee projektin toimintaympäristön sekä projektinhallinnan prosesseihin, jotka kertovat miten erilaiset projektinhallinnan prosessit ovat vuorovaikutuksessa keskenään. (IEEE Computer Society 2004) Tietämyksen alueet on jaettu yhdeksään alueeseen:

- Projektin integroinnin hallinta

Tämä alue kuvaa prosessit, jotka tarvitaan varmistamaan, että projektin eri elementit ovat oikein koordinoitu. Alue koostuu projektisuunnitelman kehittämisestä ja toteuttamisesta sekä integroidusta muutosten kontrolloinnista

- Projektin laajuuden hallinta

Tämä alue kuvaa prosessit, jotka tarvitaan varmistamaan, että projekti sisältää kaikki vaaditut työt, ja vain vaaditut työt projektin menestyksekkääseen valmistumiseen. Laajuuden hallinta koostuu projektin aloittamisesta, laajuuden suunnittelusta, tarkentamisesta, hyväksymisestä sekä laajuuden muutoksiin vastaamisesta.

- Projektin ajan hallinta

Tämä alue kuvaa prosessit, jotka tarvitaan varmistamaan projektin valmistuminen ajallaan. Ajan hallinta koostuu toimintojen määrittelystä, jaottelusta, toimintojen kestojen arvioinnista, aikataulun tekemisestä ja valvonnasta.

- Projektin kustannusten hallinta

Tämä alue kuvaa prosessit, jotka tarvitaan varmistamaan projektin toteutuminen annetun budjetin raameissa. Kustannusten hallinta koostuu resurssien suunnittelusta, kustannusarviosta, budjetoinnista sekä kustannusten kontrolloinnista.

- Projektin laadun hallinta

Tämä alue kuvaa prosessit, jotka vaaditaan varmistamaan, että projekti tyydyttää sille määritetyt tarpeet. Laadun hallinta koostuu laadun suunnittelusta, laadun takaamisesta ja laadun kontrolloinnista.

- Projektin henkilöressurssien hallinta

Tämä alue kuvaa prosessit, jotka vaaditaan projektin henkilöressurssien tehokkaaseen allokointiin. Henkilöressurssien hallinta koostuu organisaation suunnittelusta, henkilöstön hankkimisesta sekä tiimien kehittämisestä.

- Projektin viestinnän hallinta

Tämä alue kuvaa prosessit, jotka vaaditaan varmistamaan oikea-aikainen ja -sisältöinen kokoelma, levitys ja varastointi projektin sisältämästä informaatiosta. Viestinnän hallinta koostuu viestinnän suunnittelusta, informaation jakelusta, esitysten raportoinnista sekä hallinnoidusta projektin päätöksestä.

- Projektin riskien hallinta

Tämä alue kuvaa prosessit, jotka liittyvät projektin riskien tunnistamiseen, analysointiin, laadullisten ja määrällisten riskien analysointiin, riskeihin vastaamisen suunnitteluun sekä riskien valvontaan ja kontrollointiin.

- Projektin hankintojen hallinta

Tämä alue kuvaa prosessit, jotka vaaditaan varmistamaan organisaation ulkopuolelta tulevien tavaroiden ja palveluiden hankinta. Hankintojen hallinta koostuu hankintojen suunnittelusta, raha-anomuksista, lähteiden valinnasta, sopimusten hallinnasta ja päättämisestä.

### **2.3 CMMI projektinhallinnan prosessialueet**

Software Engineering Instituutin (SEI) mukaan CMMI (Capability Maturity Model Integration) on prosessien parantamiseen tarkoitettu kypsyyssmalli tuotteiden ja palveluiden kehittämiseen. Se koostuu hyväksi havaituista käytännöistä, jotka keskittyvät kehitys- ja ylläpitotoimintoihin tuotteen elinkaaren kaikissa vaiheissa aina idean kehittelystä tuotantoon ja ylläpitoon.

CMMI:stä on olemassa kaksi esitystä: tasoittainen ja jatkuva. Tasoittainen esitys keskittyy organisaation prosesseihin kokonaisuudessaan ja kartoittaa prosessien kehittämistä ennalta määrättyjen prosessialueiden ryhmittelyiden perusteella. Jatkuva esitys keskittyy yksittäisten prosessialueiden kehittämiseen. (Marçal ym. 2008)

CMMI-mallin komponentit on ryhmitelty kolmeen kategoriaan, jotka kuvastavat kuinka niitä tulisi tulkita (Chrissis ym. 2003):

- Vaadittavat (tarkat ja yleiset tavoitteet): Mitä organisaation tulee saada aikaan tyydyttääkseen tietyn prosessialueen.
- Odotetut (tarkat ja yleiset käytännöt): Mitä organisaatio yleensä toteuttaa saavuttaakseen vaaditun komponentin. Odotetut komponentit ohjaavat heitä, jotka toteuttavat parannuksia tai tekevät arviointeja.
- Informatiiviset: Kertovat yksityiskohtia, jotka auttavat organisaatiota saavuttamaan vaadittavat ja odotetut komponentit.

CMMI for Development (CMMI-DEV) kuvaa 22 prosessialuetta. Prosessialue on joukko samankaltaisia toimintoja, joilla pyritään saavuttamaan tiettyjä tavoitteita. Näiden alueiden tavoitteena on määritellä mitä tulisi tehdä eikä niinkään, että miten tulisi tehdä. Prosessialueet voidaan jakaa neljään kategoriaan: prosessinhallinta, projektinhallinta, valmistus ja tuki. Projektinhallintakategoria keskittyy johtamisen toimintoihin, kuten suunnitteluun seurantaan ja kontrollointiin. (Marçal ym. 2008)

Projektinhallinta-kategoriaan kuuluu kuusi prosessialuetta (CMMI-DEV V1.2: 55-58):

- Projektin suunnittelu

Suunnittelu sisältää projektisuunnitelman tekemisen, sidosryhmien tarkoituksenmukaisen sisällyttämisen, suunnitelmaan sitouttamisen sekä suunnitelman ylläpitämisen.

- Projektin valvonta

Valvonta sisältää valvonta- sekä korjaustoimet. Projektisuunnitelmasta selviää projektin valvonnan oikea taso, raportoinnin tiheys sekä mittarit edistymisen valvontaan.

- Alihankkijoiden hallinta

Selvittää mitä osia projektista teetetään alihankkijoilla. Alueeseen kuuluu myös alihankkijoiden valinta sekä sopimusten tekeminen alihankkijoiden kanssa.

- Integroitu projektinhallinta

Perustaa ja ylläpitää projektin määritellyt prosessit, jotka on muokattu organisaation yleisistä prosesseista.

- Riskien hallinta

Riskien hallinta hallitsee jatkuvasti eteen tulevia mahdollisia riskejä toiminnoilla, jotka arvioivat riskejä ja mahdollisuuksia niiden lieventämiseen.

- Kvantitatiivinen projektinhallinta

Tarjoaa kvantitatiiviset ja tilastolliset tekniikat projektin edistymisen ja tuotteen laadun arviointiin.

Näistä kolme ensimmäistä prosessialuetta kuuluu CMMI:n kypsyystasolle kaksi, integroitu projektinhallinta ja riskien hallinta tasolle kolme ja viimeinen tasolle neljä.

## 2.4 Valittava näkökulma

Molemmat tutkielmassa esitellyistä projektinhallinnan jäsenyksistä sisältävät hyvin samantyyppisiä alueita. Jaottelu on vain hieman erilainen. PMBOK on jaettu useampaan tietämyksen alueeseen, mutta suurin osa tietämyksen alueista löytyy myös CMMI:n projektinhallinnan prosessialueista. CMMI on kypsyysmalli, joka sisältää projektinhallinnan prosessialueiden lisäksi paljon muitakin prosessialueita, joihin peilaamalla voidaan selvittää organisaation kypsyys taso. Tässä tutkielmassa keskitytään ketterien menetelmien ja tarkemmin Scrumin käsitykseen projektinhallinnasta, käyttäen viitekehystenä PMBOK - viitekehystä koska se keskittyy pelkästään projektin hallintaan.

### 3 PROJEKTINHALLINTA KETTERISSÄ MENETELMISSÄ

Tässä luvussa esitellään ketterien menetelmien projektinhallintaa, ja tarkastellaan miten PMBOK:in mukaiset käytänteet täyttyvät ketterissä menetelmissä.

#### 3.1 Agile Project Management

Augustine ym. (2005) esittelevät artikkelissaan CAS:iin (Complex adaptive systems) pohjautuvan Agile Project Management - kehyksen. CAS tarkoittaa monimutkaista ja mukautuvaa järjestelmää, joka koostuu puoliautonomisista agenteista, jotka pyrkivät maksimoimaan joidenkin asioiden sopivuutta toisiinsa reagoimalla havaitsemiinsa muutoksiin. Projektit, jotka käyttävät ketteriä menetelmiä, ovat artikkelin mukaan CAS:n - tyyppisiä. Tutkimuksen mukaan projektipäälliköt siirtyvät yleensä johtamaan projekteja perinteisin menetelmin, vaikka ketterät menetelmät on otettu käyttöön. (Augustine ym. 2005) Artikkelissa määritellään kuusi käytäntöä ketterien projektien hallintaan:

- **Orgaaniset 7-9 henkilön tiimit.** Yhtenäisten tiimien luominen vähentää kommunikoinnin hankaluuksia. Yhtenäinen tiimi myös kehittää optimaaliset keinot kommunikointiin. Vaikka projektissa tarvittaisiin isompia, esimerkiksi 15 henkilön tiimejä, voidaan ne jakaa pienemmiksi osatiimeiksi.
- **Ohjaava näkemys.** Kun projektin kokonaisvisiosta kerrotaan ryhmäläisille vain yksinkertaistettu versio, alkavat jäsenet toimia tehokkaasti. Ketterissä menetelmissä projektipäällikön tarkoitus on ohjata ryhmäläisiä käyttämään omia kykyjään ja päätöksen tekoa.

- **Yksinkertaiset säännöt.** APM:n käytännöissä on määritelty yksinkertaisia sääntöjä, jotka kaikkien tiimin jäsenten tulee hyväksyä. Projektin johtaja etsii sääntöjä, joita ei noudateta ja ottaa selvää miksi niitä ei noudateta. Tämän pohjalta hän poistaa hankalasti hyväksyttävät osat säännöistä.
- **Vapaa ja avoin pääsy kaikkeen informaatioon.** CAS:ssä informaatio suunnittelusta, prosesseista sekä organisaatiosta on tärkeä osa jokaisen tiimin jäsenen sopeuttamista projektiin. Tiimin jäsenten välisen kommunikaation rikkaus riippuu suuresti informaation vaihdon avoimuudesta. APM korostaa informaation avoimuutta ja vapaata jakamista.
- **Kevyt johtamistyyli.** Perinteisissä lähestymistavoissa johtaminen nähdään yleensä muutoksen, riskin ja ihmisten hallintana. Aikataulun muuttuessa yksityiskohtaiset menetelmät, työkalut ja periaatteet usein epäonnistuvat. Kun johtajille annetaan enemmän valtaa hallita, he saattavat unohtaa alkuperäisen tarkoituksensa järjestyksen luoja.
- **Mukautuva johtaminen.** Liian rakenteiset järjestelmät ovat liian kankeita, ja liian löyhästi johdetut organisaatiot ajautuvat kaaokseen. Kaikkien näiden periaatteiden noudattaminen tiimiä johtaessa on haastavaa. Mukautuva johtaminen vaatii "systemiajattelua", jotta voi ymmärtää projektin sisäiset voimat. Ketterän projektin johtajan on ymmärrettävä projektin eri osien väliset vuorovaikutukset sekä johdatettava niitä jatkuvaan oppimiseen ja muuntumiskykyyn. Mukautuva APM:iin pohjautuva johtamiskehys sisältää muutamia käytäntöjä. Näitä ovat kyky johtaa muutosta ja mukautua siihen, näkemys projektista muuttuvana ja mukautuvana systeeminä, ulkoisen valvonnan rajojen tunnistaminen vakiintuneessa järjestyksessä sekä kokonaisvaltainen lähestymistapa ihmislähtöisten ongelmien ratkaisemiseen.



Hass (2007) kertoo artikkelissaan ketterien menetelmien projektinhallinnan (Agile Project Management) olevan toistuva ja vähittäin etenevä prosessi, jossa kehittäjät ja projektin sidosryhmät työskentelevät aktiivisesti yhdessä ymmärtääkseen alueen, tunnistaakseen mitä pitää tehdä ja priorisoidakseen toiminnallisuudet. Artikkelissa APM jaetaan yhdeksään eri komponenttiin, jotka ovat:

- **Visuaalinen valvonta.** Tarkoittaa ”kortteja seinällä” menetelmää, eli käytetään erilaisia visuaalisia menetelmiä kuvaamaan projektin etenemistä. Esimerkiksi projektin vaiheet merkitään seinälle ja markatuista valmiit osat ovat eri värillä kuin keskeneräiset tai kokonaan aloittamattomat. Näin tiimi pystyy näkemään helposti meneillään olevan vaiheen.
- **Keskitetty huipputehokkaat tiimit.** Ketterässä kehitystyössä kaikki tiimin avainjäsenet työskentelevät samassa paikassa. Tämä parantaa kommunikaation ja koordinaation laatua tiimin sisällä.
- **Testilähtöinen kehittäminen.** Testitapaukset kehitettävälle ominaisuudelle tehdään ennen kuin itse ominaisuutta toteutetaan. Testitapaukset tehdään usein samalla kun vaatimuksia määritetään.
- **Mukautuva valvonta.** Kaikkien tiimin jäsenten tulee mukautua tilanteisiin koko ajan. Tästä johtuen projektipäällikön tulee olla enemmän johtaja kuin orjapiiskuri.
- **Yhteistyössä tehtävä kehitystyö.** APM perustuu yhteistyöhön tiimin sisällä. Kaikki tiimin jäsenet tekevät tulosta ja saavat palautetta siitä, ja siten voivat kehittyä seuraavaan iteraatioon. Tämä on APM:n yksi vahvuuksista.
- **Ominaisuuslähtöinen kehittäminen.** Vähentää huomattavasti projektin monimutkaisuutta, koska kehittäjät keskittyvät vain yhteen ominaisuuteen kerrallaan. Projektin johtajat päättävät, mitkä ominaisuudet ovat prioriteettilistalla korkeimmalla ja toteutetaan ensin. Yleensä korkean riskin

komponentit toteutetaan etusijaisesti, ja tavoitteena on tehdä komponentit vain yhteen suuntaan riippuviksi ydinjärjestelmästä. Näin muut komponentit ovat toisistaan riippumattomia ja niitä voidaan kehittää missä järjestyksessä tahansa.

- **Ihmisten johtaminen (leadership) ja yhteistyö enemmän kuin käskeminen ja valvonta.** Projektinjohtaja toimii muiden johtoportaiden kanssa yhteistyössä, lisäksi projektinjohtaja toimii tiiminvetäjänä poistaen rajoja ydintiimien välillä.
- **Siirtyminen kustannusnäkökulmasta liikevaihtonäkökulmaan.** Liiketoiminta-analyytikon tehtävänä on varmistaa, ettei tiimi investoi enempää rahaa kehitystyöhön kuin mitä projektin tuotto tulee olemaan. Projektin johtaja keskittyy projektin kustannuksiin, kun liiketoiminta-analyytikko keskittyy projektin omistamisen kokonaiskustannuksiin, joihin kuuluu myös järjestelmän julkaisun jälkeiset kulut.
- **Opiksi ottaminen.** Jokaisen kehityssyklin jälkeen tiimi pitää palaverin, jossa mietitään mitä voidaan tehdä paremmin seuraavassa iteraatiossa.

Augustine ym. (2005) keskittyvät jaottelussaan lähinnä vain projektinpäällikön toimintaan sekä projektin läpiviemisen onnistumiseen. Hassin (2007) jaottelun lähtökohta on huomattavasti teknisempi ja jaottelu on tehty koko tiimin näkökulmasta. Molemmissa jaotteluissa oletetaan, että pienet tiimit ovat tehokkaita, ja niihin löydetään tarvittavat osaajat jokaiselta osa-alueelta. Molemmat jaottelut myös asettavat projektipäällikölle melkoisia haasteita toimia tilanteiden vaatimalla tavalla.

### 3.2 Ketterät menetelmät vs. Project Management Body of Knowledge

Vaikka PMBOK:n ja ketterien menetelmien filosofiat projektihallinnasta ovat perustaltaan hyvin erilaisia, monet PMBOK:issa määritellyt käytänteet ovat lähes yhteensopivia ketterien käytänteiden kanssa. Suurimmat erot tulevat siinä, milloin ja miten nämä käytännöt toteutetaan, sekä sanaston eroavaisuudessa. (Sliger 2007)

Integraation hallinta jakautuu PMBOK:in mukaan kolmeen osaan, jotka ovat projektisuunnitelman kehittäminen, julkaiseminen ja muutoksen hallinta. Ketterissä menetelmissä projektisuunnitelma ei välttämättä sisällä kaikkia projektin elementtejä tarkasti kuvattuna, vaan se voi koostua esimerkiksi projektin työtilan seinillä olevista kirjoitustauluista, joissa eri vaiheet ovat eri väreillä. (Sliger 2007) Muutoksenhallinta on yksi ketterien menetelmien tärkeimmistä ominaisuuksista. Agile Manifeston (Agile Alliance 2001) yksi arvo on muutokseen vastaaminen ennen suunnitelman seuraamista.

Projektin laajuuden ja ajan hallinta ovat tärkeässä asemassa PMBOK:ssa. Myös ketterissä menetelmissä keskitytään näihin, mutta hieman eri näkökulmasta kuin perinteisissä menetelmissä, jotka pyrkivät välttämään muutoksia. Ketterät menetelmät ovat valmistautuneet muutoksiin, koska ne pyrkivät aikataulua ja kustannuksia muokkaamalla tuottamaan asiakkaalle ominaisuuksia, joita se todella tarvitsee. (Sliger 2007)

Hassin (2007) mukaan yksi Agile Project Managementin komponenteista on siirtyminen kustannusnäkökulmasta liikevaihtonäkökulmaan. Sen mukaan liiketoiminta-analyytikon tehtävänä on varmistaa, ettei kehitystiimi investoi kehitystyöhön liikaa. Kun projektinpäällikkö keskittyy projektin kustannuksiin, liiketoiminta-analyytikon tehtävänä on tarkkailla projektin todellisia kokonaiskustannuksia, joihin kuuluu myös projektin valmistumisen jälkeiset kustannukset. Näin ollen myös projektin kustannusten hallinta otetaan

huomioon ketterissä menetelmissä. Kustannusten hallintana voidaan pitää myös asiakkaan mukana oloa päättämässä, mitä ominaisuuksia toteutetaan ja missä vaiheessa.

PMBOK:in mukaan laadunhallinnan täytyy kohdistua sekä projektin johtamiseen että projektin tuotokseen. Johtuen ketterien menetelmien asiakaslähtöisyydestä, sekä toimivan ohjelman jatkuvasta tuottamisesta ja asiakkaalla hyväksymisestä, projektin tuotoksen laatu on aina korkea. Testilähtöinen kehittäminen taas varmistaa, että kehitettävät ominaisuudet ovat toimivia. Kun itse ominaisuuden ohjelmointi aloitetaan vasta kaikkien mahdollisten testitapausten kehittämisen jälkeen, ja ominaisuus on valmis testien mentyä läpi, voidaan tätä pitää perustavana laadun suunnitteluna, lupauksena ja valvontana. (Sliger 2007)

Henkilöstön johtaminen jaetaan PMBOK:issa kolmeen osaan: organisatorinen suunnittelu, työntekijöiden hankinta sekä tiimin luominen. Projektin tarvittavat roolit ja vastuut jaetaan tarkasti, ja kuhunkin rooliin pyritään löytämään oikea ihminen. Ketterissä menetelmissä tiimit ovat itsestään organisoituvia, ja tiimissä on kaikki tarvittavat osaajat toimivan ohjelman tuottamiseen. Tiimin kehittyessä kaikki sen jäsenet oppivat enemmän muiden tehtävistä ja ovat jatkossa halukkaampia ja osaavampia auttamaan alueilla, jotka eivät varsinaisesti aluksi kuuluneet omiin osaamisalueisiin. (Sliger 2007)

Riskien hallintaan ketterät menetelmät eivät tarjoa suoria ratkaisuja. Riskien hallinta otetaan huomioon jatkuvalla kommunikoinnilla sekä läheisellä asiakasyhteistyöllä, mutta suoranaisia käytänteitä riskien kartoittamiseen tai arviointiin ei yleensä ole tarjolla. (Marçal ym. 2008)

## 4 PROJEKTIHALLINTA SCRUM - MENETELMÄSSÄ

Tässä luvussa esitellään Scrum-menetelmää ja sen projektinhallinta-käytänteitä. Lopussa tarkastellaan, kuinka Scrum menetelmän projektinhallintakäytännöt ovat yhteneviä PMBOK-jäsennyksen kanssa. Luvun lopussa on taulukkona kuvattu, mitkä ketterien menetelmien ja Scrum-menetelmän käytänteet toteuttavat PMBOK:in tietämyksen alueita.

### 4.1 Scrum

Ken Schwaber esitteli Scrum-menetelmän ensimmäisen kerran vuonna 1996 prosessina, joka hyväksyy, että kehitysprosessi on ennalta arvaamaton. Scrumin lähtökohtana on, että ohjelmiston kehittäminen on liian monimutkaista ja ennalta arvaamatonta etukäteen tarkasti suunniteltavaksi. Erilaiset ympäristömuuttujat ja tekniset muuttujat (kuten aikarajat, laatu, vaatimukset, resurssit, toteutusteknologiat ja työkalut ja kehitysmenetelmät) tulee ottaa valvontaan, jotta voidaan reagoida muutoksiin joustavasti. (Marçal ym. 2008) Scrum sisältää kolme roolia (Schwaber 2004: 142 - 143):

- **Tuotteen omistaja** edustaa kaikkia projektin sidosryhmiä, ylläpitää tuotteen työlistaa (product backlog) sekä priorisoi projektin vaatimuslistoja.
- **Tiimi** on vastuussa toiminnallisuuksien toteuttamisesta. Ne ovat itse itseään johtavia ja organisoivia sekä toimivat kaikilla teknisillä osa-alueilla. Tiimit ovat vastuussa siitä, kuinka tuotteen työlistan vaatimukset toteutetaan iteraatioiden aikana, sekä suunnittelevat työnsä sen mukaan. Tiimin jäsenet ovat yhdessä vastuussa iteraatioiden onnistumisesta.
- **Scrum-mestari** (ScrumMaster) on vastuussa Scrum-prosessin johtamisesta, Scrumin käytänteiden oikeanlaisesta toteutuksesta sekä projektin tuoton maksimoinnista.

Scrum-projektit alkavat tulevan tuotteen visioinnista korkealla tasolla. Tämän jälkeen luodaan tuotteen työlista (product backlog), joka on lista tuotteen vaatimuksista. Jokainen listan vaatimus priorisoidaan ja jaetaan osiin joita kutsutaan sprinteiksi. (Marçal ym. 2008)

Scrumin yksi avainsana on ajan rajaaminen (time-boxing), joka tarkoittaa kaikille vaiheille ennalta määriteltyä aikaa, jota ei saa ylittää. Sprintin suunnittelukokoukselle on ennalta määrätty kahdeksan tunnin kesto ja se on jaettu kahteen neljän tunnin vaiheeseen. Ensimmäisessä vaiheessa valitaan ne tuotteen työlistan kohdat, jotka toteutetaan seuraavan sprintin aikana. Tässä vaiheessa tiimi voi tehdä ehdotuksia, mutta lopullisen päätöksen tekee tuotteen omistaja. Tiimillä on vastuu päättää, kuinka paljon tuotteen omistajan haluamista ominaisuuksista voidaan toteuttaa sprintin aikana. Ajan rajaaminen neljään tuntiin tarkoittaa sitä, että enempää aikaa tuotteen työlistan analysointiin ei ole. Loput analysoinnista suoritetaan sprintin aikana. Toinen vaihe suunnittelukokouksesta pidetään heti ensimmäisen vaiheen loputtua. Siihenkin on varattu aikaa tasan neljä tuntia. Tässä vaiheessa tiimi suunnittelee, miten äskeisessä vaiheessa valitut toiminnallisuudet toteutetaan valmiiksi tuotteeksi. Toisen vaiheen tuotoksena syntyy toteutusvaiheen työlista (sprint backlog), joka on lista tehtävistä, tehtävien arvioinneista sekä työtehtävistä, joiden parissa tiimi alkaa työskennellä uuden ominaisuuden toteuttamiseksi. (Schwaber 2004: 133 - 139)

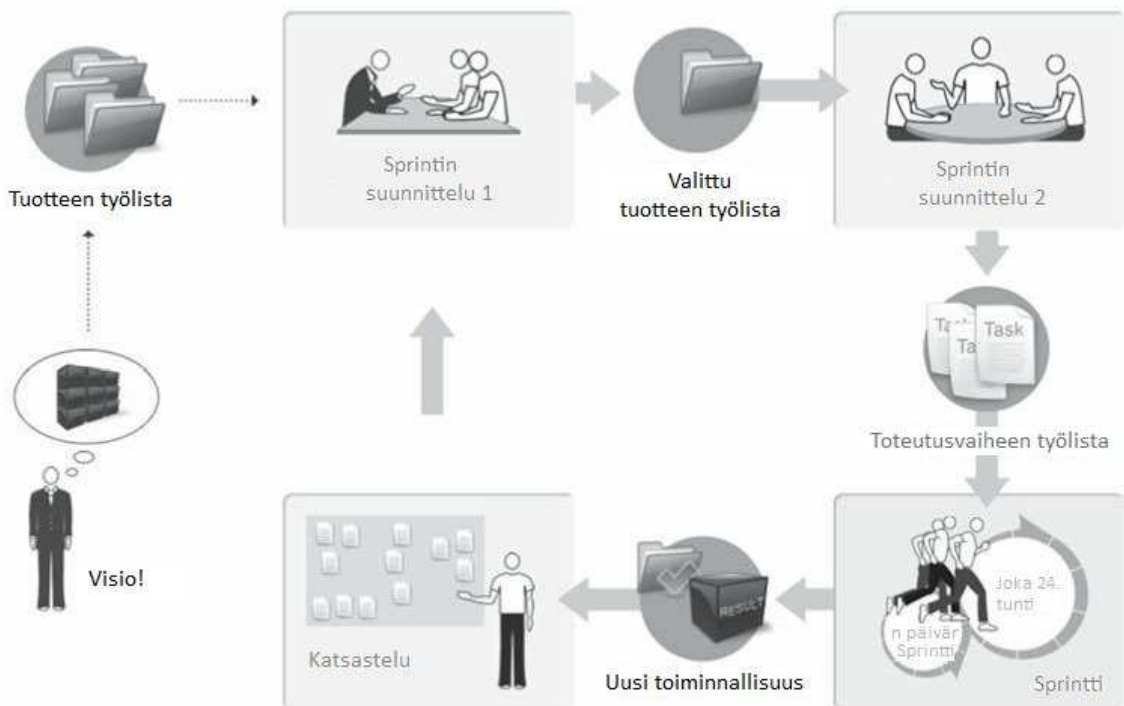
Päivittäisen Scrum-tapaamisen kesto on määrätty viideksitoista minuutiksi riippumatta tiimin jäsenten määrästä. Tässä tapaamisessa jokainen tiimin jäsen vastaa kolmeen kysymykseen: mitä on tehnyt viimeisen palaverin jälkeen, mitä aikoo tehdä seuraavaan palaveriin mennessä ja onko ilmennyt ongelmia. (Schwaber 2004: 135 - 136)

Kaikki toiminnallisuus toteutetaan sprinteissä. Sprintti on rajattu, kolmenkymmenen peräkkäisen kalenteripäivän ajanjakso. Tässä ajassa tiimin

vaaditaan toteuttavan jotakin tuotteen omistajan vaatimuksia. Tämä on myös pisin aika, mitä projektin sidosryhmien täytyy odottaa. Sprintin aikana tiimi voi etsiä apua ulkopuolelta, mutta kukaan ei saa tarjota apua, ohjeita tai kommentteja tiimille. Tiimi on itseohjautuva. (Schwaber 2004: 136 - 137)

Sprintin jälkeen järjestetään neljän tunnin mittainen sprintin tarkastelukokous. Tässä kokouksessa tiimi esittelee projektin sidosryhmille ne toiminnallisuudet, jotka ovat valmiita. Mitään puolivalmiita tuotoksia kokouksessa ei esitellä. Sidosryhmät voivat tässä kokouksessa vapaasti esittää kysymyksiä ja kommentteja tiimille. (Schwaber 2004: 137 - 138)

Aivan sprintin lopuksi järjestetään kolme tuntia kestävä sprintin katsastelutapaaminen. Tapaamiseen osallistuu tiimi, Scrum-mestari sekä mahdollisesti tuotteen omistaja. Tapaamisessa jokainen tiimin jäsen vastaa kysymyksiin mikä meni hyvin sprintin aikana ja mitä voidaan tehdä paremmin seuraavassa sprintissä. (Schwaber 2004: 138 - 139) Scrumin vaiheet on kuvattu kuvassa 1.



Kuva 1. Scrum-menetelmän vaiheet (Suomennettu lähteestä Marçal ym. 2008)

## 4.2 Scrum vs. Project Management Body of Knowledge

Lähtökohdat projektinhallintaan Scrumissa ovat hyvin erilaiset kuin PMBOK:issa. Kuitenkin suurin osa PMBOK:in tietämyksen alueista voidaan toteuttaa Scrumin käytänteillä.

- Projektin integroinnin hallinta

Scrumissa projektisuunnitelma on tuotteen työlista. Sen tekee yhteistyössä tuotteen omistaja ja Scrum-mestari. Tuotteen työlistan toteuttamisesta pidetään huoli jokaista sprintiä edeltävässä sprintin suunnittelupalaverissa, jossa valitaan sprintin aikana toteutettavat ominaisuudet. Myös integroitu muutoksen kontrollointi tapahtuu samassa palaverissa, sekä sprintin tarkastelupalaverissa, jossa tiimi esittelee sprintin aikana toteutetut ominaisuudet projektin sidosryhmille ja projektin osakkaat voivat esittää haluamiaan muutoksia. (Schwaber 2004: 133 - 139)

- Projektin laajuuden hallinta

Scrum-projekteissa laajuudenhallinta tapahtuu sprintin suunnittelupalavereissa sekä sprintin tarkastelupalavereissa. Näissä palavereissa päätetään, mitä tullaan tekemään ja tarkastellaan mitä on tehty. Palavereissa myös pidetään huoli siitä, että kaikki projektin sidosryhmät tietävät, mitä tuotteen työlistassa kirjatut ominaisuudet sisältävät, sekä varmistetaan, että toteutetut ominaisuudet vastaavat niille asetettuja vaatimuksia. Myös muutoksiin vastaaminen tapahtuu näissä palavereissa. (Schwaber 2004: 133 - 139)

- Projektin ajan ja kustannusten hallinta

Sprintien tiukasti määrätystä 30 päivän pituudesta johtuen voidaan sanoa, että siltä osin ajan hallinta on melko tiukka. Marçalin ym. (2008) mukaan projektin elinkaarenhallinta on Scrumissa täysin huomioon otettu. Tutkimuksen mukaan projektin elinkaari jaetaan neljään osaan, jotka ovat suunnittelu, järjestäminen,



sprintit sekä julkaisu. Näin ollen projektin sprintien määrä tulee kiinnittää suunnitteluvaiheessa. Tässä tapauksessa ajan hallinta on melko tiukkaa. Kustannusten hallintaan Scrum ei tarjoa suoranaisia työkaluja. (Marçal ym. 2008)

- Projektin laadun hallinta

Laadunhallintaa voidaan pitää hyvin huomioon otettuna, koska projektin sidosryhmät pääsevät jokaisen sprintin jälkeen näkemään mitä on saatu aikaan ja voivat vaikuttaa siihen mitä tulisi saada aikaan.

- Projektin henkilöresurssien hallinta

Henkilöresurssien hallinta Scrumissa tapahtuu tiimin kokoamisessa. Scrum-tiimit ovat itseohjautuva. Tiimi suunnittelee itse, kuinka se voi maksimoida tuottavuutensa. Suunnittelu kuuluu tasaisesti koko tiimille. Scrum-mestari ja muut osalliset voivat opastaa, neuvoa ja informoida tiimiä, mutta tiimi on silti itse vastuussa johtaa itseään. (Schwaber 2004: 101 - 102)

- Projektin viestinnän hallinta

PMBOK:in viestinnän hallinnalla tarkoitetaan oikea-aikaista ja -sisältöistä kokoelmaa, levitystä ja varastointia projektin informaatiosta. Scrumissa viestintä tapahtuu tiimin sisällä jatkuvasti, koska tiimi työskentelee fyysisesti samassa tilassa. Mahdollinen tiimien välinen viestintä tapahtuu esimerkiksi Scrum-mestareiden välillä, jotka voivat muodostaa yhden Scrum-tiimin ja tavata omassa Scrum-tapaamisessa päivittäin. Viestintä projektin sidosryhmille tapahtuu taasen sprintin suunnittelu- ja tarkastelukokouksissa. Sprintin jälkeen tehdään muodollisia raportteja, joilla pyritään saamaan kaikille projektin osapuolille näkymä projektin tilasta ja pitää projekti läpinäkyvänä kaikille. (Schwaber 2004: 133 - 139)

- Projektin riskien ja hankintojen hallinta

Näihin kahteen PMBOK:in tietämysalueeseen Scrum ei suoraan tarjoa ratkaisua samalla tavoin kuin muihin alueisiin. Scrum määrittelee riskin mahdolliseksi esteeksi projektille. Riskien tunnistaminen tapahtuu päivittäisissä tapaamisissa, mutta mitään käytänteitä riskien lähteiden löytämiseen ei ole, eikä myöskään parametreja tai kategorioita riskien kontrollointiin. (Marçal ym. 2008)

Alla olevassa taulukossa (Taulukko 1) eritellään ketterien menetelmien ja Scrum-menetelmän käytänteitä, ja sitä mitkä käytännöt vastaavat mihinkin PMBOK:in tietämysalueisiin. Taulukosta näkyy, että Scrum edustaa pitkälti yleistä ketterien menetelmien linjaa projektinhallintakäytänteissä. Moniin alueisiin ei ketterät menetelmät eikä Scrum tarjoa suoranaisia ohjeita, mutta läheisellä asiakasyhteistyöllä voidaan kattaa monia prosessialueita. Esimerkiksi laajuuden, ajan ja laadunhallinta ovat tällaisia alueita. Johtuen suorien ohjeiden puuttumisesta on näiden alueiden täyttyminen paljolti kiinni asiakasyhteistyön toimivuudesta ja asiakkaan aktiivisuudesta. Tällaisiin asioihin ei ole ketterissä menetelmissä sääntöjä koska menetelmät pyritään pitämään kevyinä ja helppoina oppia. Tämä tuo omat hankaluutensa projektin johtamiseen.

Taulukko 1. Ketterien menetelmien ja Scrumin vastaavuudet PMBOK:in tietämyksen alueisiin.

	Ketterät menetelmät	Scrum
Integraation hallinta	<ul style="list-style-type: none"> <li>➤ Projektisuunnitelma</li> <li>➤ Muutoksiin vastaaminen</li> </ul>	<ul style="list-style-type: none"> <li>➤ Tuotteen työlista</li> <li>➤ Muutoksiin vastaaminen</li> </ul>
Laajuuden hallinta	<ul style="list-style-type: none"> <li>➤ Asiakas mukana päättämässä mitä toteutetaan</li> </ul>	<ul style="list-style-type: none"> <li>➤ Asiakas mukana päättämässä mitä toteutetaan</li> </ul>
Ajan hallinta	<ul style="list-style-type: none"> <li>➤ Lyhyet tuotantocyklit</li> </ul>	<ul style="list-style-type: none"> <li>➤ Ajan rajaaminen</li> <li>➤ Kaikki toiminnallisuudet toteutetaan sprinteissä</li> </ul>
Kustannusten hallinta	<ul style="list-style-type: none"> <li>➤ Kustannusnäkökulmasta liikevaihtonäkökulmaan</li> <li>➤ Asiakas valitsee toteutettavat ominaisuudet</li> </ul>	<ul style="list-style-type: none"> <li>➤ Asiakas valitsee toteutettavat ominaisuudet</li> </ul>
Laadun hallinta	<ul style="list-style-type: none"> <li>➤ Asiakkaalla hyväksyttäminen</li> <li>➤ Testilähtöisyys</li> </ul>	<ul style="list-style-type: none"> <li>➤ Sprintin suunnittelu ja tarkastelukokoukset</li> <li>➤ Asiakas mukana kokoajan</li> </ul>
Henkilöressurssien hallinta	<ul style="list-style-type: none"> <li>➤ Tiimien kasaaminen</li> <li>➤ Tiimien moniosaaminen</li> <li>➤ Tiimien itseohjautuvuus</li> </ul>	<ul style="list-style-type: none"> <li>➤ Tiimien kokoaminen eri osa-alueiden osaajista</li> <li>➤ Tiimien itseohjautuvuus</li> </ul>
Kommunikaation hallinta	<ul style="list-style-type: none"> <li>➤ Tiimi työskentelee samassa paikassa</li> <li>➤ Raportointi sidosryhmille</li> </ul>	<ul style="list-style-type: none"> <li>➤ Tiimi samassa tilassa</li> <li>➤ Sprintin suunnittelu ja tarkastelukokoukset</li> <li>➤ Raportointi sprintien jälkeen</li> </ul>
Riskien hallinta	<ul style="list-style-type: none"> <li>➤ Tiimin sisäinen jatkuva kommunikaatio</li> <li>➤ Jatkuva raportointi asiakkaalle</li> </ul>	<ul style="list-style-type: none"> <li>➤ Päivittäisissä tapaamisissa</li> <li>➤ Merkattuna tauluihin</li> </ul>

## 5 YHTEENVETO

Ketterien menetelmien käyttö ohjelmistokehityksessä on yleistynyt huomasti viimeisen vuosikymmenen aikana. Ketterät menetelmät mielletään usein pienten ohjelmistotalojen menetelmiksi ja pieniin projekteihin sopiviksi. Ketteriä menetelmiä on kuitenkin otettu käyttöön myös isoissa ohjelmistotaloissa, ja ne skaalautuvat tarvittaessa hyvinkin isoihin projekteihin. (Schwaber 2004: 119 - 120) Ketterien menetelmien suosion kasvaessa myös niiden määrä on lisääntynyt. Menetelmien näkemykset projektinhallinnasta eroavat toisistaan laajalti. Toiset menetelmät eivät tarjoa projektinhallintaan mitään suoranaisia työkaluja vaan jättävät vapaat kädet. Scrum on ketterä kehitysmenetelmä, joka on kehitetty projektinhallinnan käytänteisiin. (Marchenko ym. 2008) Myös Sulaiman ym. (2006) määrittelevät Scrumin projektinhallintaviitekehikseksi.

Tässä tutkielmassa tarkasteltiin ketterien menetelmien ja tarkemmin Scrum-menetelmän projektinhallintakäytänteitä. Tutkielman tarkoituksena oli selvittää yleisiä projektinhallinnan jäsennyksiä, sekä ottaa selvää miten ketterien menetelmien projektinhallintakäytänteet sopivat yleisimpien projektinhallinnan jäsenysten vaatimuksiin.

Tutkimuksessa selvisi, että lähtökohdat projektinhallintaan ketterissä menetelmissä ovat hyvin erilaiset kuin perinteisissä projektinhallinnan jäsenyksissä. Tutkielmasta käy ilmi, että lähtökohtaeroista huolimatta ketterien menetelmien käytöllä pystytään vastaamaan perinteisten projektinhallinnan jäsenyksien tietämysalueisiin ja prosessialueisiin melko laajasti. Joihinkin alueisiin, kuten riskienhallintaan, ei löydetty suoranaisia käytänteitä Scrumista eikä muista ketteristä menetelmistä. Ei voida kuitenkaan sanoa, että ketterissä menetelmissä ei tapahtuisi minkäänlaista riskienhallintaa. Jatkuvalle kommunikaatiolle sekä asiakasyhteistyöllä pystytään välttämään yleisimpiä riskejä. Ketterien menetelmien luonteesta johtuen kaikkiin alueisiin

ei ole ylöskirjattuja sääntöjä. Useat projektinhallinnan jäsenysten vaatimat alueet täyttyvät menetelmien yleisiä periaatteita noudattaen, ja tiukkoja sääntöjä esimerkiksi riskien tai laajuudenhallintaan ei ole syytä kehittää. Näin varmistetaan myös ketterien menetelmien helppo oppiminen, vähäinen byrokratia sekä turhan dokumentaation poisjääminen.

Ketteristä menetelmistä tehtyä tutkimusta on paljon, ja siitä johtuen eri tutkimuksissa samasta asiasta voi olla hieman eriäviä tuloksia. Tämä johtuu usein siitä, että monet vertailtavat asiat ovat hyvin tulkinnanvaraisia ja tulokset riippuvat pitkälti näkökulman valinnasta. Myös sillä on vaikutusta asiaan, että ketterien menetelmien käyttöönotto on hyvin eritasoista eri organisaatioissa, ja yleensä ketterän menetelmän käyttöönotossa muokataan tietyn menetelmän käytänteet omalle organisaatiolle sopiviksi.

Jatkotutkimuksena voisi mahdollisesti tutkia empiirisesti jonkun organisaation projektinhallintakäytänteitä ennen ketteriin menetelmiin siirtymistä, siirtymävaiheen aikana sekä ketterän menetelmän käyttöönoton jälkeen. Toinen mahdollinen jatkotutkimusaihe voisi olla se, kuinka tarkasti ketterät menetelmät yleensä otetaan käyttöön organisaatioissa.

## 6 LÄHTEET

Abrahamson, P., Warsta, J., Siponen, M. T., & Ronkainen, J., 2003. New directions on agile methods: A comparative analysis. 25<sup>th</sup> International Conference on Software engineering 2003. 244 - 254

Agile Alliance 2001. Agile Manifesto. Agile Alliance. Saatavilla www-osoitteessa <http://agilemanifesto.org/>

Augustine S., 2006. Managing Agile Projects. Prentice Hall PTR

Augustine S., Woodcock S., 2003. Agile Project Management: Emergent Order through Visionary Leadership.

<http://www.ccpace.com/Resources/documents/AgileProjectManagement.pdf>

Chrissis, M. B., Konrad, M., Shrum, S., 2003. CMMI Guidelines for Process integration and Product Improvement. SEI Series in software engineering.

CMMI Product team, 2006. CMMI for Development, Version 1.2. Software engineering institute.

Hass, K., B., 2007. The Blending of Traditional and Agile Project Management. PM World Today, vol IX, issue V, May 2007

IEEE Guide Adoption of PMI Standard, 2004. A Guide to the Project Management Body of Knowledge. The institute of Electrical and Electronics Engineers, Inc. May 2004

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., 2004. Agile software development in large organizations. IEEE Computer 37, 12 2004 26 - 34

Marçal, A., de Feritas B., Soares, F., Furtado, M., Maciel, t., Belchior, a., 2008.

Blending Scrum practices and CMMI project management process areas.

Springer-Verlag London limited. Saatavilla [www.osoitteessa.com](http://www.osoitteessa.com):

<http://www.springerlink.com/content/u34hng336827w284/fulltext.pdf>

Marchenko, A., & Abrahamsson, P., 2008. Scrum in a multiproject environment:

An ethnographically-inspired case study on the adoption challenges. Agile

'08 conference. 15 - 26

Payne, B., Sencindiver, F., Woodcock, S., Augustine, S., 2005. Agile project

management: steering from the edges. Communications of the ACM, Dec

2005, pp. 318-322

Ruuska, K., 2001. Projekti hallintaan. Talentum Media Oy 2001

Sliger, M., 2006. A Project Manager's Survival Guide to Going Agile. Rally

Software Development Corporation. Saatavilla [WWW-osoitteessa.com](http://www.osoitteessa.com):

[http://www.rallydev.com/documents/rally\\_survival\\_guide.pdf](http://www.rallydev.com/documents/rally_survival_guide.pdf)

Schwaber, K., 2004. Agile Project Management With Scrum. Microsoft Press.

Sulaiman, T., Barton, B., Blackburn, T., 2006. AgileEVM - Earned Value

Management in Scrum Projects. Agile '06 Conference, IEEE Computer

Society.