Tampereen yliopisto

Joonas Jäppinen

KETTERÄT MENETELMÄT OHJELMISTOPROJEKTILIIKETOIMINNASSA

Kandidaatintutkielma
Informaatioteknologian ja viestinnän tiedekunta
Syyskuu 2023

TIIVISTELMÄ

Joonas Jäppinen: Ketterät menetelmät ohjelmistoprojektiliiketoiminnassa
Kandidaatintutkielma
Tampereen yliopisto
Tietotekniikan kandidaatintutkinto
Syyskuu 2023

Tutkielmassa oli tarkoituksena tarkastella eri ketteriä menetelmiä projektiliiketoiminnassa. Tutkielmassa tarkasteltiin aluksi projektin määritelmää, projektiliiketoimintaan liitoksissa olevia asioita ja lopuksi perinteisiä projektinhallintamenetelmiä. Perinteisistä menetelmistä siirrytään ketteriin menetelmiin, joita käsitellään erikseen. Menetelmiä tarkastellessa pyrittiin luomaan ryhmittelyä eri menetelmien kesken – mitkä menetelmät olivat pienten ja keskisuurten yritysten käyttöön ja mitkä olivat enemmän kansainvälisten, suurten yritysten käytettäväksi luotuja. Vertailu ei rajoitu pelkkään skaalautuvuuteen, vaan tarkastelua tehdään myös esim. kansainväliseen ympäristöön soveltuvuuteen.

Menetelmistä tulivat tutuiksi mm. Feature Driven Development, Large-Scale of Scrums, Nexus, PRojects IN Controlled Environments 2, Rational Unified Process, Scrum, Scrum of Scrums ja Scaled Agile Framework. Osa menetelmistä ovat skaalautumista helpommin soveltavia kuin toiset. Tutkielmassa käytetään hyväksi kirjallisuudessa tehtyä tutkimusta skaalautuvuudesta ja pyritään sen avulla luomaan valintamahdollisuus eri tilanteisiin. Tutkielmassa käydään läpi systemaattisesti eri menetelmiä ja pyritään mahdollistamaan alustavalla tasolla empiirisesti tuettu päätöksenteko menetelmien välillä.

Perinteisissä menetelmissä korostuu suunnitelmallisuus ja ketterissä menetelmissä taas työhön liittyvä epävarmuuden sietäminen. Tutkielmassa havaittiin, kuinka samankaltaisia menetelmät voivat pohjimmiltaan olla ja kuinka osittain pienillä muutoksilla saadaan aikaan eri tilanteisiin sopivia ketteriä menetelmiä. Ketteristä menetelmistä käsitellään vain osaa kaikista menetelmistä ja tutkielmassa jätettiin tietoisesti käsittelemättä mm. vanhempia ketteriä menetelmiä. Uusissa ketterissä menetelmissä havaitaan Scrumin ja Leanin roolien korostuminen alla olevina teknologioina. Ketterien menetelmien tarkastelua tehdään mm. diffuusion, menetelmän kypsyysasteen, monimutkaisuuden ja organisaation tyypin perusteella.

Tutkielmassa tarkastellaan myös tulevaisuutta, kuinka tulevaisuus ketterien menetelmien osalta tulee mahdollisesti näyttämään. Painopiste on nykyisissä mahdollisuuksissa ja niiden hyödyntämisessä. Menetelmistä jätettiin tietoisesti syvyyttä pois tutkielman laajuuden karsimiseksi ja tulevaisuuden työksi jääkin syventää havaintoja. Tutkielmassa havaittiin, että yhtä hopealuotia ei ole eri projekteille, vaan optimaalinen ratkaisu riippuu mm. projektista ja yrityksen koosta.

Avainsanat: Agile, AUP, DSDM, FDD, IT-projekti, LeSS, Nexus, PRINCE2, Projektinhallinta, Projektinhallintamenetelmä, Projektiliiketoiminta, RUP, SAFe Scrum, Scrum of Scrums

ABSTRACT

Joonas Jäppinen: Agile methods in software project business
Bachelor thesis
Tampere University
Bachelor thesis in the software engineering
September 2023

The purpose of the work was to research different Agile methods in the project business. The literature was used to first define the project, then to research other subjects which are related to the project business and lastly to research the traditional project management methods. From the traditional methods the focus moves to the Agile methods, which are covered in the separate chapter. The methods are covered systematically and tried to categorize to some categories according the scalability of the methods – which methods are better suited for small and mid-sized companies and which methods are more suitable for big, multinational companies. The comparison is not limited to the scalability of the methods while the comparison is done also from a global, multi-site perspective.

The methods which are covered among others are Feature Driven Development, Large-Scale of Scrums, Nexus, PRojects IN Controlled Environments 2, Rational Unified Process, Scrum, Scrum of Scrums and Scaled Agile Framework. Other methods are more easily scalable than others. The research uses the existing research of methods' scalability and that is the way to create a suitable solution for different situations. The research covers systematically different methods and the goal to form a basis for the empirical decision process is reached.

In traditional methods the emphasis is in the planning and in the new Agile methods the emphasis is in the tolerance of uncertainty. In thesis is observed how similar different methologies can be in the actual implementation and how different Agile methods can be created for different settings with little changes. The focus in the Agile methods is in the newer methologies and some of the older methods are left intentionally uncovered. The thesis noted that the roles of Scrum and Lean are important as underlying technologies in many different methologies. The research of Agile methologies is done by following categories: diffusion of the methology, maturity of the method, complexity and the organizational type necessary for the methology.

The thesis also inspected the future of the methologies - how does the future of different methods will look in the future. The thesis tries to keep the focus in the current usage of methologies and possibilities. Some of the depth from the methologies is intentionally left out to reduce the scope and length of the thesis – the future work is to deepen the understanding of the observations which are noticed. The results show that there is no silver bullet for different projects while the optimal solution for example depends on the size of the firm and the scope of the project.

Keywords: Agile, AUP, DSDM, FDD, IT project, LeSS, Nexus, PRINCE2, Project management, Project management method, Project business, RUP, SAFe, Scrum, Scrum of Scrums

ALKUSANAT

Aihe valikoitui käytännön kokemuksen ja oman mielenkiinnon perusteella. Olin toiminut useammassa erissä projektitiimissä urani varrella ja mietin, mikä on yhdistävä elementti eri projektien kesken – mikä takaa merkittävän onnistumisen projektinhallinnassa. Akateemista tutkimusta olen saanut tehdä merkittävissä määrin, sillä vaikuttaa siltä ettei aiheesta ole viime aikoina tehty suomeksi julkaisuja. Tampereen yliopistolle kuuluu iso kiitos mahdollisuudesta päästä lukemaan relevantteja ja viimeisimpiä artikkeleita ja muita julkaisuja, mikä on mahdollistanut työn tekemisen myös suomeksi.

Kiitos kandidaatintyön ohjauksesta ja hyvistä ideoista kuuluu seminaarin ohjaajalle, kandidaatintyönseminaarin eri opiskelijoille mainioista huomioista ja eri työnantajille, jotka ovat mahdollistaneet työhön liittyvän oppimisen ja kehityksen. Myös muiden ihmisten neuvoista ja ideoista on ollut hyötyä työn tekemisessä.

Espoo, 13.9.2023

Joonas Jäppinen

SISÄLLYSLUETTELO

LYHENTEET JA MERKINNÄT

Agile method	Ketterä kehitysmenetelmä
AUP	Agile Unified Process
DAD	Disciplined Agile Delivery
DSM	Design Structure Matrix
FDD	Feature Driven Development
LeSS	Large-Scale of Scrums
PM4A	Project Management for All
PMBOK	Project Management Body of Knowledge
PRINCE2	PRojects IN Controlled Environments 2
RUP	Rational Unified Process
Scrum	Ketterä ohjelmistokehitysmenetelmä
SAFe	Scaled Agile Framework
SoS	Scrum of Scrums

1. JOHDANTO

Tutkielman tarkoituksena on kirjallisuuden pohjalta tehdä kattava katsaus ohjelmistoprojektiliiketoiminnan harjoittamiseen ketterien menetelmien avulla. Ketteriä menetelmiä on käytössä useita ja tarkoituksena ei ole käsitellä kaikkia menetelmiä tässä tutkielmassa kattavasti, vaan tarkoituksena on huomioida menetelmien käyttö projektiliiketoiminnassa.

Tutkielman aihe valikoitui käsiteltäväksi sen käytännöllisyyden vuoksi. Havaitsin työelämässä ollessani, että ketterien menetelmien käytöstä projektiliiketoiminnassa on konkreettista hyötyä, mutta en löytänyt viimeisintä kirjallisuutta suomeksi. Tavoitteena tutkielmassa onkin käyttää englanniksi kirjoitettua kirjallisuutta lähdemateriaalina ja mahdollisesti viitata myös suomeksi tehtyihin opinnäytetöihin. Tavoitteena on ollut luoda mahdollisimman kattava kuva eri menetelmistä mahdollisimman tuoreella akateemisella tiedolla.

Tutkielmassa käsitellään ketteriä kehitysmenetelmiä kirjallisuuden avulla ja luotiin käsitystä, minkälaisissa projektiympäristöissä menetelmiä voitaisiin käyttää. Skaalautuvuus nousi ensisijaiseksi tutkimuskohteeksi ja jotkin menetelmät havaittiin toisia enemmän skaalautuviksi – todennäköisesti tietoisen suunnittelun tuloksena.

Työssä on käytetty ChatGPT-tekoälyä alussa ideoiden luomiseen ja alustavien lähteiden hankkimiseen.

Seuraavaksi luvussa 2 käsitellään ketteriä menetelmiä ja projektiliiketoimintaa yleisesti. Kolmannessa luvussa painopiste siirtyy käsittelemään projektiliiketoimintaa tarkemmin ja neljännessä luvussa käsitellään perinteisiä projektinhallintamenetelmiä. Viidennessä luvussa painopiste siirtyy ketteriin menetelmiin ja vuorollaan käsitellään eri kokoluokkien ja skaalautuvuuden omaavia ketteriä menetelmiä. Seitsemännessä luvussa käsitellään menetelmiä käytännössä ja tulevaisuutta. Lopuksi on yhteenveto tutkielman aiheesta.

2. KETTERÄT MENETELMÄT JA PROJEKTILIIKE-TOIMINTA

Tässä luvussa ja sen aliluvuissa käsitellään ketteriä menetelmiä ja projekteja yleisesti tutustuttaen lukijaa ketteriin menetelmiin ja projektiliiketoimintaan.

2.1 Ketterät menetelmät

Ketterillä menetelmillä voidaan tarkoittaa useampaa eri asiaa, mutta yhtenä tapana määrittää ketterä menetelmä voidaan todeta niiden olevan tapoja tehdä asioita joustavasti ja mukautuen tilanteeseen pyrkien säilyttämään hyvän kommunikaation tason kattavan dokumentaation sijaan. (Lehtonen et al., 2022)

Ketterän ohjelmistokehityksen julistus luotiin vuonna 2001 selkeyttämään tilannetta ketterien kehitys metodien kentällä. Ketterän ohjelmistokehityksen julistus julistaa, että yksilöt ja interaktio tulisi arvostaa ylitse prosessien ja työkalujen. Lisäksi manifesti toteaa toimivan ohjelmiston tulisi olla päämääränä sen sijaan, että kattavaa dokumentaatiota tehdään. Manifesti julistaa myös asiakasyhteistyön olevan tärkeämpää kuin sopimusneuvottelut, ja muutokseen vastaamisen tulisi asettaa ensimmäiseksi prioriteetiksi suunnitelman noudattamisen sijaan. (Bentley, 2020)

Tiimien koko on ketteriä menetelmiä käytettäessä yleensä 5-9 henkilöä. Tiimeissä painottuu kommunikaatio eri tekijöiden välillä ja nopea reagointi tapahtuneisiin muutoksiin. (Lehtonen et al., 2022)

Joissakin ketterissä menetelmissä suositaan ns. parielementtiä, jossa ohjelmoijina toimii käytännössä kaksi henkilöä. Mielellään ohjelmoijat ovat toistensa osaamista täydentäviä siten, että molemmat pystyvät tarjoamaan tukeaan toiselle tarvittaessa. Haittapuolena pariohjelmoinnissa voidaan mainita mahdollisesti lisääntynyt ajankäyttö. (Williams, 2001)

2.2 Projektiliiketoiminta

Määriteltäessä projektiliiketoimintaa on tärkeää määritellä ensinnäkin ohjelman, projektin ja projektinhallinnan käsitteet.

“Project is a temporary endeavour undertaken to create a unique product, service, or result. The temporary nature of projects indicates a beginning and an end to the project work or a phase of the project work. Projects can stand alone or be part of a program or portfolio.” (PMBOK, 2021)

Käsitteen määritelmän käännös kuuluisi seuraavasti: projekti on väliaikainen pyrkimys, johon on ryhdytty yksilöllisen tuotteen, palvelun taikka ratkaisun tuottamiseksi. Projektin väliaikainen luonne osoittaa alun ja projektityön taikka sen osan lopun. Projektit voivat olla itsenäisiä tai osa ohjelmaa tai portfolio-kokonaisuutta.

“Program is related projects, subsidiary programs, and program activities that are managed in a coordinated manner to obtain benefits not available from managing them individually.” (PMBOK, 2021)

Ohjelma on liitoksissa olevien projektien, aliohjelmien ja ohjelmien toiminnan hallinnointi koordinoidusti, jotta saadaan hyötyjä, joita eivät ole saatavissa hallinnoidessa niitä yksittäin.

“Project management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements. Project management refers to guiding the project work to deliver the intended outcomes. Project teams can achieve the outcomes using a broad range of approaches (e.g., predictive, hybrid, and adaptive).” (PMBOK, 2021)

Projektinhallinta on tietotaidon, taitojen, kykyjen, työkalujen ja tekniikoiden hyödyntämistä projektiaktiviteetteihin projektien vaatimuksien saavuttamiseksi. Projektinhallinta viittaa projektityön ohjaamiseen toivottujen lopputuloksien saavuttamiseksi. Projektiryhmät voivat saavuttaa lopputulokset käyttäen laaja-alaisesti eri lähestymistapoja (esimerkiksi ennustavia, hybridi ja sopeutuvia).

3. PROJEKTILIIKETOIMINNASTA TARKEMMIN

Tässä luvussa käsitellään perinteisiä projektinhallinnanmenetelmiä ja luodaan tarkempi katsaus projektiin liittyviin termeihin ja teoriaan.

3.1 Projektin määritelmä

Projektin tarkoituksena on arvon tuottaminen aloittajalleen. Projektia voidaan tarkastella kolmesta eri näkökulmasta. Ensimmäinen niistä on sen luonne väliaikaisena organisaationa. Sitä varten luodaan erityinen organisaatio, joka hajoaa projektin saavutettua maalinsa, jolloin projektin henkilöstö siirtyy muihin tehtäviin. Projektia voidaan tarkastella myös tuotteen tai työn rakenteen kautta. Se voidaan jakaa pienempiin osiin, joita on helpompi hallita. Kolmanneksi se voidaan nähdä aktiviteetteina tai vaiheistettuna prosessina. Sen elinkaari koostuu aikataulusta ja resursseista, mitä hallitaan. (Artto et al., 2011)

3.2 Projektinhallintamenetelmät ja projektinhallinta

”Project management is the application of management practices aimed at achieving the project goal and objectives.” (Artto et al., 2011) Projektinhallinta on hallinnollisten tapojen täytäntöönpanoa projektin maalin ja tavoitteiden saavuttamiseksi. Sen käytännöt sisältävät tietotaidon, taidot, metodit ja työkalut, mitkä ovat tarpeen projektin maalin ja tavoitteiden saavuttamiseksi. (Artto et al., 2011)

Projektinhallintaa voidaan tarkastella kolmesta eri näkökulmasta: 1) projektinhallinta tietämysalueena ja prosesseina, 2) projektinhallinta (esimerkiksi projektipäällikön) tietotaitona ja ominaisuuksina, 3) projektinhallinta työkaluina ja dokumentteina. (Artto et al., 2011)

Projektien määrittämiseen kytköksissä oleva käsite on rautakolmio. Kolmio muodostuu esimerkiksi kolmesta eri ulottuvuudesta: 1) laajuus (scope), 2) aika (time) ja 3) hinta (cost). Rautakolmio on saanut nimensä siitä, että kaikki kolme ulottuvuutta eivät voi tulla toteutetuksi yhtä aikaa, vaan jonkin täytyy projektissa joustaa. Laajuuden sijaan voidaan joskus mahdollisesti käyttää myös laadun (quality) määritelmää yhtenä kolmion kulman osana laajuuden sijaan taikka osana kolmiota kolmion sisällä ollen. Rautakol-

miosta on tehty myös muita variaatioita, joissa kulmien osina ovat muut osa-alueet ja kolmion sisältö vaihtelee. Rautakolmion nimen perusteella voidaan tarkentaa kolmion sisältöä esim. Kliemin ja Ludin versio rautakolmiosta. (Artto et al., 2011; Hellström, 2023; Lock, 2020)

Gantt-kaavio on yksinkertainen esitys projektien resursseista ja niiden käytöstä suhteessa aikaan. Toinen kaavio on esimerkiksi DSM-diagrammi, joka suhteuttaa projektien eri vaiheita toisiinsa ja mahdollistaa ajankäytön optimoinnin muodostaman kriittisen polun koko projektille (Artto et al., 2011; Steward, 1981, Soo-Haeng et al., 2005).

Varsinaisesti organisaation kontrollijärjestelmiä eivät kaaviot ja diagrammit korvaa, vaan niiden toteuttamiseksi tarvitaan tarkkaan harkittuja prosesseja: PMBOK:ia (PMBOK, 2021), PM4A:ta (Bentley, 2020) taikka PRINCE2:ta (PRojects IN Controlled Environments) (Hunt, 2006).

3.3 Sidosryhmät ja ympäristö

Projektin eri sidosryhmät voidaan määritellä usealla eri tavalla samoin kuin sen ympäristö. Sidosryhmien määrittelyä on tehty useamman eri akateemisen tutkimuksen verran, ja määrittelyssä voidaan tehdä erittäin hienovaraisia eroja.

”Stakeholders are individuals, groups, or organizations that the project may affect or that can affect the project” on Arton et al. (2011) määritelmä sidosryhmälle. Suomeksi määritelmä kuuluu seuraavasti: ”Sidosryhmiä ovat yksilöt, ryhmät ja organisaatiot, joihin projekti saattaa vaikuttaa tai jotka voivat vaikuttaa projektiin.” (Artto et al., 2011).

Sidosryhmillä voi olla joko suora tai epäsuora yhteys projektiin tai lopputuotoksena olevaan tuotteeseen. Sidosryhmänä voi olla myös ryhmä, johon projekti vaikuttaa, mutta jolla ei ole suoraa keinoa vaikuttaa lopputulokseen (Artto et al., 2011).

Tyypillisesti projektilla on seuraavat sidosryhmät, joilla on suora yhteys projektiin (Artto et al., 2011):

- projektimanageri
- projektiorganisaatio
- projektitiimi
- organisaation yksikkö yhtiössä, joka tekee projektin
- asiakas
- käyttäjä

- ostaja
- sponsori tai projektin omistaja.

Lisäksi voi olla muita sidosryhmiä, joilla on yhteys projektiin kuten

- toimittaja taikka palvelun tarjoaja
- virkamiehet taikka viranomaiset
- rahoittajat
- media
- muut kohderyhmät
- kilpailijat
- projektiin osallistuvat henkilöt
- yhteiskunta laajemmassa merkityksessä.

3.4 Elinkaari ja toteutus

Määritelmä kirjallisuudessa projektin elinkaaresta on tarkkaan määritelty. Sen elinkaari viittaa vaiheiden ketjuun, jossa ideat, odotukset, ja mahdollisuudet projektille on identifioitu, projekti on toteutettu ja hyödyt on saatu projektin tuotteesta, ja tuotteen käyttöä tuetaan (Artto et al., 2011).

Projektin voidaan katsoa koostuvan kolmesta eri vaiheesta: 1) työ ennen projektin alkua (ideointi, vaihtoehtojen kartoitus, valmistelu), 2) työ projektin aikana (projektin suorittaminen), ja 3) työ projektin jälkeen (projektin tuloksen käyttö ja tuotteen käytön tuki) (Artto et al., 2011).

Projektin tuotoksen toteuttamisessa voi olla useita eri tapoja. Ensinnäkin tuotos saatetaan tuottaa itse tai osa siitä. Toinen vaihtoehto on kehittää osittain tai kokonaan projektin tuotos ulkoisen kumppanin kanssa, jolloin yrityksen sisäisten resurssien käyttö vähenee (Artto et al., 2011).

3.5 Projektissa onnistuminen

Projektissa onnistumisen voi määrittää monella eri tavalla. Yksi tapa hahmottaa asiaa on tarkastella sitä ajan, resurssien ja laajuuden näkökulmasta. Onko siinä saatu ajoissa tehtyä riittävän laaja suoritus ilman ylimääräisiä resursseja? Mikäli on, voidaan se ainakin tästä näkökulmasta katsoa onnistuneeksi. Toinen näkökulma tarkastelee projektin

onnistumista sidosryhmien kannalta. Onko se onnistunut vastaamaan avainsidosryhmien tarpeisiin riittävän hyvin vai ovatko sidosryhmät vielä tyytymättömiä tilanteeseen? Kolmas kriteeri projektin onnistumiselle voi olla tuotettu laatu. Onko siinä tuotettu tarpeeksi laadukasta tuotosta, joka kelpaa laatuvaatimukset täyttäen? Joissakin projekteissa voi olla käytössä standardi, joka määrittää minimivaatimukset laadulle.

4. PERINTEISET PROJEKTINHALLINTAMENETELMÄT

Tässä luvussa käsitellään projektinhallinnanmenetelmiä ja luodaan pikainen katsaus kyseisten menetelmien sisältöön.

4.1 Project Management for All

Project Management for All (PM4A) on projektinhallintamenetelmä, joka noudattaa perinteistä vesiputousmallista ajattelutapaa, jossa projektiehdotuksen (muodostuu alustavasta projektin ideasta) ja suunnittelun jälkeen tehdään itse toteutus ja testaus. Tämän jälkeen projektin katsotaan olevan valmis ja toteutuksen vastaavan toivottua. PM4A on kehitetty Colin Bentleyn ja Andy Brittonin toimesta pieniä projekteja varten, joissa on tarpeen eri osien kontrollit. Erityistä huomiota ovat saaneet laatu-, riski-, versio- ja muutoskontrollit. Projektin eri osat erotettu toisistaan porteilla ja seuraavaan osaan siirtyminen edellyttää edellisen osan hyväksyttyä läpäisyä (Bentley, 2020).

4.2 Project Management Body of Knowledge

Project Management Body of Knowledge (PMBOK) on projektinhallinnan tiedonhallintakehys. Sen on kehittänyt Project Management Institute (PMI). PMBOK tarjoaa projektinhallinnan käytäntöjä, periaatteita ja prosesseja. Niitä voidaan soveltaa erilaisissa projekteissa eri toimialoilla. Project Management Body of Knowledge määrittelee myös projektinhallinnan prosessit. Ne kuvaavat projektin elinkaaren aikana eri vaiheita. Aloituksen, suunnittelun, toteutuksen ja seurannan sekä projektin päättämisen kattavat eri prosessit (PMBOK Guide, 2021).

4.3 Projects IN Controlled Environments 2

Projects IN Controlled Environments 2 (PRINCE2) on perinteinen prosessipohjainen projektinhallintamenetelmä, joka kehitettiin valtionhallinnon projekteja varten. Se tarjoaa selkeän viitekehyksen projektien suunnitteluun, toteutukseen, hallintaan ja seurantaan.

PRINCE2 omaa seitsemän eri periaatetta, jotka ohjaavat projektin onnistumiseen. Periaatteet ovat: 1) jatkuva liiketoiminnallinen oikeutus (continued business justification), 2) kokemuksesta oppiminen (learn from experience), 3) määritellyt roolit ja vastuut (defined roles and responsibilities), 4) hallitse vaiheittain (manage by stages), 5) hallitse poikkeuksin (manage by exception), 6) keskity tuotteisiin (focus on products), ja 7) räätälöi sopimaan projektiin (tailor to suit the project) (Hunt, 2006).

4.4 Rational Unified Process

Rational Unified Process (RUP) on prosessi, joka kehitettiin kattamaan koko ohjelmiston elinkaaren. Nimensä prosessi on saanut siitä, että se pyrkii määrittelemään: 1) kuka tekee mitä, 2) koska se tehdään, 3) kuinka saavuttaa tietty päämäärä, ja 4) syötteet ja ulostulot jokaisesta aktiviteetista. RUP koostuu useasta hierarkkisesta elementistä. Prosessi koostuu alhaisen tason aktiviteeteista, jotka yhdistetään ns. työvirroiksi (kuvastaa kuinka yksi aktiviteetti syöttää toiseen). Nämä vaiheet organisoidaan iteraatioiksi. Jokainen iteraatio identifioi jonkin systeemin osan. Iteraatioista muodostuu vaiheet, jotka keskittyvät eri osiin suunnitteluprosessia esimerkiksi vaatimukset, analyysi, design ja toteutus. Vaiheet (phases) voidaan ryhmitellä sykleiksi (cycles), jotka keskittyvät peräkkäisiin julkaisuihin järjestelmästä (Hunt, 2006).

5. PROJEKTINHALLINTAMENETELMÄT KETTERISSÄ PROJEKTEISSA

Tässä luvussa on tarkoitus käsitellä yleisesti ketteriä menetelmiä, jotka ovat käytössä taikka joiden käyttöä voisi suositella.

5.1 Pienen koon ja vähäisen byrokratian ketterät menetelmät

Tässä kohdassa on tarkoitus käsitellä kooltaan pieniin organisaatioihin soveltuvia ketteriä menetelmiä, joissa byrokratian ja hallinnollisen työn määrä on vähäinen suhteutettuna kehittämisprosessiin.

5.1.1 Lean development

Lean development on kattava termi kaikelle Lean-ajattelun sisältävälle ketterälle kehittämiselle. Lean-ajattelun alkuperä on valmistuksessa, jossa sitä käytetään tuotantolinjojen hukan vähentämisessä. Päämääränä Lean-kehityksessä oli ja on hukan minimointi ja arvon maksimointi asiakkaalle. Tehokkuus, jatkuva parantaminen ja asiakasarvon saaminen enimmäistasolle ovat periaatteen keskeisiä tekijöitä. Lean-ajattelussa keskitytään resurssitehokkuuden sijaan virtaustehokkuuteen, pyritään pienentämään eri vaiheiden kokonaiskestoa ja sitä kautta parantamaan tehokkuutta.

Lean-ohjelmistokehityksessä on seitsemän periaatetta, joita on syytä noudattaa: 1) eliminoi hukka (eliminate waste), 2) rakenna laatu sisään (build quality in), 3) luo tietotaitoa (create knowledge), 4) lykkää sitoutumista (defer commitment), 5) toimita nopeasti (deliver fast), 6) kunnioita ihmisiä (respect people), ja 7) optimoi kokonaisuus (optimize the whole) (Poppendieck et al., 2007).

Menetelmät ja työkalut voivat vaihdella prosessista toiseen, mutta yleisesti käytetään Leanin mukaisia työkaluja: arvovirtakartoitusta, pull-periaatetta (kysyntä ohjaa tuotantoa), Kanban-järjestelmiä (esim. ilmoitustaululla lappuja) ja jatkuvaan virtauksen ajatukseen sopivia menetelmiä. Menetelmä sopii erityisen hyvin nopeasti vaihteleviin ja epävarmoihin prosesseihin, joissa asiakkaiden tarpeet ja markkinoiden olosuhteet voi-

vat vaihdella merkittävästi – menetelmä korostaa joustavuutta, sopeutuvaisuutta ja nopeaa reagointia muutoksiin.

5.1.2 Scrum

Scrum on kehitysprosessi projekti- ja tuotehallintaan. Erityisesti Scrumia käytetään ohjelmistokehitystyössä. Scrum rakentuu kolmen pilarin varaan: läpinäkyvyys, tarkastettavuus ja sopeutuvaisuus. Kolmen pilarin malli tukee työskentelyä iteratiivisesti ja vaatii lisätuekseen myös luottamusta. Ilman luottamusta voi syntyä jännitteitä ja pullonkauloja (Scrum, 2023).

Scrumiin kuuluvia arvoja ovat 1) rohkeus, 2) keskittyminen, 3) sitoutuneisuus, 4) arvostus ja 5) avoimuus. Arvoja harjoitetaan sprinteissä, joissa tehdään kehitystöitä. Niissä tehdään jatkuvasti tarkastelua ja sopeutumista prosessiin ja toimitettavaan tuotteeseen. Scrumissa rooleja ovat scrum master, tuoteomistaja ja kehittäjät, jotka tekevät kehittämistyön.

Kehitysjono esittää merkittävää roolia Agile Scrum -metodologiassa. Kaikki, mitä on tarkoituksena tehdä projektissa taikka ohjelmassa, tulisi olla kehitysjonossa. Jokainen tehtävä tulisi olla enintään 24 tuntia pitkä. Pidemmät kuin 24 tuntia kestävät tehtävät tulisi pilkkoa erillisiksi alitehtäviksi. Suunnittelu tulisi tehdä sprintin suunnittelupalaverissa, jossa scrum master ja tuoteomistaja tapaavat ja päättävät, mitä tehdään seuraavassa sprintissä. Sprintit ovat syklejä, joita käytetään ohjelman kehitysprosessissa. Ne kestävät tyypillisesti kahdesta neljään viikkoa. Sprinteissä ratkaistavat ”ongelmat” eli kehityskohteet ovat ”Työn alla” -tilassa ennen kuin ne voidaan merkitä ratkaistuiksi eli kehitykseltään valmiiksi (Antunes et al., 2015).

5.2 Skaalautumattomat ketterät menetelmät

Tässä kohdassa on tarkoitus käsitellä ketteriä menetelmiä, joiden skaalautuvuus on erittäin heikko taikka heikko ja joiden käyttämisessä on jonkin verran byrokratiaa ja hallinnollista työtä.

5.2.1 PRINCE2 Agile

PRINCE2 Agile on menetelmänä jalostettu versio PRINCE2-menetelmästä. PRINCE2 suunniteltiin aluksi perinteisiin projekteihin soveltuvaksi ja siinä ei ollut ketteriä osia. Uudessa PRINCE2 Agilessa on ketteriä komponentteja mukana ja se soveltuu ketterää

menetelmää tarvitsevaan projektiin. Menetelmä hyödyntää PRINCE2:n hallinnollista rakennetta ja ketterien menetelmien iteratiivista työotetta (Hunt, 2006).

PRINCE2 Agile hyödyntää PRINCE2-menetelmän periaatteita, teemoja ja prosesseja, mutta tarjoaa lisäksi ketterien menetelmien käytäntöjä ja työkaluja. Menetelmässä tehdään ketterästi töitä huomioiden kuusi eri ulottuvuutta. Nämä ulottuvuudet jaotellaan joustamattomiin, joustaviin ja mahdollisesti joustaviin. Ulottuvuudet ovat: 1) hinta, 2) kesto, 3) laajuus, 4) laatu, 5) hyödyt, 6) riskit.

Menetelmä sisältää seuraavat ominaisuudet, jotka ohjaavat työntekoa:

1. Ketterät periaatteet: menetelmä hyödyntää jatkuvaa parantamista, iteratiivista kehitystä, itseorganisoituvuutta ja jatkuvaa oppimista.
2. Priorisointi ja iteraatiot: menetelmä edistää priorisointia ja työn jakamista pienempiin osiin, joita sanotaan iteraatioiksi tai sprinteiksi. Iteraatioita toistetaan projektin aikana, mikä mahdollistaa reagoinnin muutoksiin ja paremman tuotosten toimittamisen.
3. Työkalut ja tekniikat: PRINCE2 Agile sisältää useita ketteriä työkaluja ja tekniikoita. Esimerkkinä voidaan mainita Kanban-taulut, retrospektiivit, jatkuva integrointi ja automatisoitu testaus.
4. Projektin hallinta ja ketterä tuotoksen toimittaminen: PRINCE2 Agile yhdistää PRINCE2:n projektinhallinnan rakenteen ketteriin menetelmiin, kuten Scrumiin tai Kanbaniin (Axelos, 2015).

5.2.2 Dynamic Systems Development Method

Dynamic Systems Development Method (DSDM) rakentuu usean arvon varaan – prosessissa arvostetaan tervettä järkeä ja käytännöllisyyttä. Myös selkeä ketterä prosessi, ihmiset (joiden rooli on selkeästi määritelty), selkeästi määritellyt tuotteet ja suositellut käytännöt määrittävät prosessia. DSDM voidaan myös kuvata projektinhallinnan kolmion avulla. Aika, laatu ja hinta ovat pysyviä arvoja, kun taas ominaisuudet (features) on vaihteleva, toisin kuin normaalissa prosessissa. DSDM toimittaa oikean tuotteen oikeaan aikaan, mikäli prosessia on oikein noudatettu. Alhaisen prioriteetin vaatimukset jätetään tekemättä, kun suojataan lisäyksen (increment) tai aikaikkunan (timebox) lopetuspäivämäärää (DSDM, 2023).

DSDM:ään kuuluu kahdeksan periaatetta, jotka ovat 1) keskity liiketoiminnan tarpeisiin (focus on the business need), 2) toimita ajallaan (deliver on time), 3) tee yhteistyötä (collaborate), 4) älä tee kompromisseja laadusta (never compromise quality), 5) raken-

na inkrementaalisesti pysyvistä perusteista (build incrementally from firm foundations), 6) kehitä iteratiivisesti (develop iteratively), 7) kommunikoi jatkuvasti ja selkeästi (communicate continuously and clearly), 8) demonstroi kontrollia (demonstrate control) (DSDM, 2023).

DSDM:ssä on erilaisia rooleja – jotkin rooleista ovat tuttuja muista ketteristä viitekehyksistä eri nimillä. Roolit ovat: 1) liiketoiminnan sponsori (business sponsor), 2) liiketoiminnan visionääri (business visionary), 3) tekninen koordinaattori (technical coordinator), 4) projektipäällikkö (project manager), 5) liiketoiminta-analyytikko (business analyst), 6) tekninen neuvonantaja (technical advisor), 7) työpajan fasilitaattori (workshop facilitator), 8) DSDM valmentaja (DSDM coach), 9) liiketoiminnan neuvonantaja (business advisor), 10) liiketoiminnan suurlähettiläs (business ambassador), 11) ratkaisutestaaja (solution tester), 12) tiimin vetäjä (team leader), and 13) ratkaisukehittäjä (solution developer). Roolit eivät ole yhteen henkilöön sidottuja – yksi henkilö voi omata yhden taikka useamman roolin ja rooli voi olla jaettuna eri henkilöiden kesken (DSDM, 2023).

5.2.3 Feature-Driven Development

Feature-Driven Development (FDD) on iteratiivinen ja inkrementaalinen malliperustainen ohjelmistokehitysprosessi. Malli pohjautuu lyhyisiin iteraatioihin ja viiteen aktiviteettiin. Aktiviteetit ovat: 1) yleisen mallin rakentaminen (overall model development), 2) ominaisuuslistan kerääminen (feature list building), 3) suunnittelu ominaisuuksittain (plan by feature), 4) suunnittelu ominaisuuksittain (design by feature) ja 5) rakentaminen ominaisuuksittain (build by feature). Ensimmäiset kaksi aktiviteettia on yleisiä aktiviteetteja, jotka määrittävät yleisen mallin ja sen rakenteen. Loput kolme aktiviteettia ovat iteratiivisia, joissa erityinen ominaisuus rakennetaan (Danenas et al., 2012).

FDD:ssä projektissa voi olla käytössä modulaarinen rakenne, joka määrittää kuinka eri kerrokset kommunikoivat keskenään. Palmerin ja Felsingin (2002) mukaan rakenne voidaan jakaa neljään osaan: käyttöliittymä (UI), ongelma-alue (problem domain), systeemi-interaktio (system interaction) ja tiedonhallinta (data management).

FDD:ssä ominaisuuden rakentamisen priorisoinnissa voidaan käyttää kolmea eri kriteeriä: 1) tärkeys liiketoiminnalle, 2) riskitaso, 3) toteutusvaatimukset. Niiden perusteella rakennetaan lopullinen ominaisuus, jonka kehittäjät toteuttavat. Iteraatiossa valitaan toteutettavaksi osa ominaisuuksista, jotka ovat listattuna listassa. Toteutuneille iteraation ominaisuuksille tehdään testaus ja niiden toiminta varmistetaan ennen kuin iteraatio päättyy (Hunt, 2006).

5.2.4 Agile Unified Process

Agile Unified Process eli AUP koostuu Rational Unified Process:ista ja ketteristä menetelmistä. Prosessi käyttää ketterää mallintamista (agile modeling), testivetoista suunnittelua (Test-Driven Development), ketterää muutoksen hallintaa ja tietokannan uudelleenkirjoittamista tehokkuuden parantamiseksi. Ketterä mallintaminen on yksi avaintekijöistä ketterän prosessin mallintamiseksi ja dokumentoimiseksi. Ketterä mallintaminen on kokoelma arvoja, periaatteita ja käytänteitä ohjelman suunnittelemiseksi kevyellä tavalla. Se arvostaa kommunikointia, yksinkertaisuutta, rohkeutta, palautetta ja nöyryyttä. Mallintamisen arvoihin kuuluu 1) yksinkertaisuuden olettaminen, 2) rohkeuden syleily, 3) inkrementaalinen muutos, ja 4) nopea palaute (Ioannis, 2010).

AUP:ssa on RUP:in tapaan 4 merkittävää vaihetta. Ensinnä on aloittaminen (inception), toisena on tarkentaminen (elaboration), kolmantena on rakentaminen (construction) ja viimeisenä, neljäntenä on siirtyminen (transition). Ensimmäisessä vaiheessa tiimi idenfioi alkuperäisen laajuuden, potentiaalisen arkkitehtuurin, hankkii rahoituksen ja sidosryhmien hyväksynnän (Ioannis, 2010).

Toisessa vaiheessa tiimi varmistaa järjestelmän järkevyyden ja ehdotetun arkkitehtuurin. Kolmannessa vaiheessa tiimi alkaa rakentamaan ohjelmistoa säännöllisesti, vaiheittaisella tavalla, joka huomioi sidosryhmien korkeimman prioriteetin tarpeet. Neljänneksi siirtymisessä tiimi validoi ja ottaa käyttöön järjestelmän tuotantoympäristössä (Ioannis, 2010).

5.3 Skaalatut ketterät menetelmät (suur)yrityksille

Tässä kohdassa on tarkoitus käsitellä skaalautuvia menetelmiä, joiden byrokratian määrä vaihtelee ja kohteena olevan organisaation koko vaihtelee pienestä paikallisesta firmasta suureen kansainväliseen organisaatioon.

5.3.1 Nexus

Nexus tunnetaan myös nimellä Meta Scrum. Se lisää Scrumiin ainoastaan yhden uuden roolin. Se sopii yrityksille, jotka käyttävät 3-9 tiimiä yhden tuotteen valmistamiseen. Tiimit jakavat yhteisen tuotteen kehitysjononon, tuoteomistajan ja scrum masterin. Nexukseen kuuluu olennaisena osana Nexus Integration Team (Nexus integraatiotiimi), joka on Scrum-tiimi ja johon kuuluvat tuoteomistaja ja scrum master. Integraatiotiimiin kuuluu olennaisena osana tuoteomistaja, joka on ainoa pysyvä jäsen tiimissä – muut

tiimiläiset voivat tarpeen ja ajan mukaan vaihtua. Nexuksessa on tyypillistä järjestää korkeamman asteen tapaamisia, jotka pohjautuvat Scrumin tapaamisiin. Esimerkkinä voidaan mainita Nexus Sprint Planning (Nexus sprintin suunnittelu), jossa suunnitellaan korkeammalla tasolla seuraavan sprintin suorituksia (Scrum.org, 2023).

5.3.2 Scrum of Scrums

Scrum of Scrums (SoS) on pienen mittakaavan skaalautuva viitekehys Scrum-muotoiseen kehitystyöhön. Jokaiseen Scrum-tiimiin voi kuulua Scrumin mukainen määrä henkilöitä ja jokaisella Scrum-tiimillä on oma lähettiläs ”ambassador”, joka päivän päätteeksi osallistuu tiimien väliseen palaveriin, johon osallistuu muiden tiimien lähettiläät. Lähettiläälksi voidaan valita kuka tahansa – kehittäjä, scrum master taikka tiimin päällikkö. Jokaisessa kokouksessa käsitellään tehdyt muutokset, seuraavalla listalla olevat asiat ja esteet. Kirjaa käsitellyistä asioista pidetään erillisellä kehitysjonolla, jonka tarkoituksena on helpottaa tiimien välistä kommunikointia (AgileAlliance.org, 2023).

5.3.3 Scaled Agile Framework

Scaled Agile Framework (SAFe) voidaan katsoa olevan laajennettu ja skaalattu Scrum käytännössä. Sen tarkoituksena on kattaa seitsemän eri osa-aluetta ja mahdollistaa suurten organisaatioiden ketteryys. Seitsemän eri kategoriaa ovat: 1) Lean-portfolionhallinta (Lean Portfolio Management) 2) organisaation ketteryys (Organizational Agility), 3) jatkuvan oppimisen kulttuuri (Continuous Learning Culture), 4) Lean-Agile johtajuus (Lean-Agile Leadership), 5) tiimi ja tekninen ketteryys (Team and Technical Agility), 6) ketterä tuotteen toimitus (Agile Product Delivery), ja 7) yrityksen tuotteen toimitus (Enterprise Solution Delivery).

SAFe koostuu useasta eri tiimistä, jotka voivat tehdä joko samaa tuotetta tai erillisiä tuotteita. SAFe ei ole yksittäinen kokonaisuus, vaan se koostuu useammasta viitekehyksestä. SAFe:n viitekehyksiä ovat: 1) ydin (essential), 2) suuri (large), 3) portfolio ja 4) koko (full) (SAFe, 2023).

Ydin-tasoon kuuluvat kaikki Scrumin elementit, mutta lisäksi viitekehykseen kuuluu myös Agile Product Delivery ryhmä, johon kuuluvat tuotteen hallinta, systeemiarkkitehti ja julkaisujunan päällikkö (release train engineer). Suuri-tasoon kuuluu lisäksi Enterprise solution delivery -osa, jota ohjaa ratkaisupäällikkö (solution train manager), -arkkitehti (solution train architect) ja ratkaisujunasuunnittelija (solution train engineer). Portofolio-konfiguraatiossa uutena osiona tulee Lean-portfoliohallinta, joka tuo muka-

naan kaksi uutta roolia: 1) Aihion omistajan (epic owner) ja kokonaisarkkitehdin (enterprise architect) (SAFe, 2023).

5.3.4 Large-Scale Scrum Framework

Large-Scale Scrum Framework eli LeSS on Scrum, jossa useampi tiimi työskentelee yhden tuotteen parissa. LeSS voidaan jakaa kahteen osaan: 1) pienempi LeSS ja 2) valtava LeSS. Pienempi LeSS sopii alle 9 tiimin organisaatioille, kun taas suuri LeSS sopii kahdeksasta usean tuhannen ihmisen organisaatioihin. Tarve suuremman joukon viitekehykselle tulee ilmi, kun tuoteomistaja ei enää yksin kykene hahmottamaan tuotetta kokonaisuutena (Larman et al., 2016).

LeSS kietoutuu kymmenen eri periaatteen ympärille. Periaatteet ovat: 1) suuren mittakaavan Scrum on Scrum (large-Scale Scrum is Scrum), 2) läpinäkyvyys (transparency), 3) enemmän vähemmällä (more with less), 4) kokonaisvaltainen tuotteeseen keskittyminen (whole-product focus), 5) asiakaskeskeisyys (customer-centric), 6) jatkuva parantaminen kohti täydellisyyttä (continuous improvement towards perfection), 7) Lean-ajattelu (Lean thinking), 8) järjestelmäajattelu (systems thinking), 9) empiirinen prosessihallinta (empirical process control), 10) jonoteoria (queuing theory) (Larman et al., 2016).

5.4 Muut mainitsemisen arvoiset menetelmät

Tässä kohdassa käsitellään ketteriä menetelmiä, jotka eivät ole sopineet kategorisointiin muulla tavoin. Kohta sisältää muutaman ketterän toimintamallin/menetelmän ja on suhteessa tarkoituksellisesti pienempi kuin muut kappaleet.

5.4.1 Kokonaisketterä

”Kokonaisketterä on Suomessa kehitetty toimintamalli-innovaatio, joka on suunnattu palveluorganisaatiolle ja kehityshankkeille. Siinä painottuu käytännöllisyys ja liiketoimintalähtöisyys.” (Kokonaiskettera.fi, 2023) Kokonaisketterä on suomalainen ketterän menetelmän viitekehys.

Kokonaisketterässä toimintamallissa ongelma on aluksi 1) idea-tasolla. Idea-tasolta ongelma siirtyy 2) konseptointi-tasolle. Konseptointi-tasolta ongelma siirtyy 3) kehitysjono-tasolle, josta se siirtyy 4) kehitys-tasolle ja lopulta 5) hyödyt-tasolle.

Idea-tasolla jono sisältää ainoastaan asioita, joiden uskotaan täyttävän liiketoimintatavoitteet. Jono elää jatkuvasti ja siitä edistetään asioita, joiden arvioidaan omaavan korkeimman hyöty-panossuhteen. Konseptointi-tasolla paketit muutetaan täydellisiksi kokonaisuuksiksi, joilla on tavoitteet, sisältö, rajaukset ja mittarit. Kestoltaan konseptointitason paketit ovat 1-3 kk mittaisia. Ensin määritetään ongelmat, jonka jälkeen etsitään ongelmalle ratkaisu (Kokonaiskettera.fi, 2023).

Kehitysjono-tasolla pyritään ylläpitämään läpäisyastetta hyvänä ja arvopaketit ”priorisoidaan pisteyttämällä ne eri liiketoimintatavoitteiden suhteen ja jakamalla arvopaketin koolla.” Kehitysjonossa olevat asiat voidaan kehittää esim. Scrum-metodia käyttämällä. Lopuksi käsitellään muutos ja arvioidaan hyödyt – muutos tulee johtaa jotenkin ja arvopakettien tulee tulla hyödynnetyksi kokonaisvaltaisesti (Kokonaiskettera.fi, 2023).

5.4.2 Disciplined Agile Delivery

Disciplined Agile Delivery eli DAD ei varsinaisesti ole mitään sitovia ohjeita antava viitekehys, vaan sen tarkoituksena on auttaa projektin suorituksessa tarjoamalla ohjeita ja menetelmiä eri ongelmien ja haasteiden ratkaisemiseksi. DAD:ia voidaan kuvata erittäin monimutkaiseksi malliksi, jonka sopeuttaminen erilaisiin projekteihin voi onnistua helposti (PMI.org, 2023).

Menetelmää on alettu käyttämään, mutta sen käyttöä jarruttaa fakta ettei sitä voida skaalata käytettäväksi eri yhteyksissä. Sen monimutkaisuus on keskivertomääräistä tasoa ja kustannukset jäävät kalleimmista viitekehyksistä alhaisemmiksi. Globaaleihin, hajautetuille tiimeille menetelmä ei oikeastaan sovi tai sen sopiminen on katsottu hankalaksi (Ebert & Paasivaara, 2017).

6. PROJEKTINHALLINTAMENETELMÄT KÄYTÄNNÖSSÄ JA TULEVAISUUS

Tässä luvussa ja sen aliluvuissa käsitellään projektinhallintaa ketterässä projektissa. Pääosa huomiosta keskittyy kehitykseen, jossa ei ole parielementtiä ohjelmistokehitystyössä. Tarkemmin pariohjelmoinnista on alaluvussa 2.1.

6.1 Vertailua eri menetelmien kesken

Ketterissä projekteissa yhteistä on neljän eri periaatteen noudattaminen: 1) yksilöiden ja kommunikoinnin merkitys ylitse prosessien ja työkalujen (individuals and interactions over processes and tools), 2) toimiva ohjelmisto kattavan dokumentoinnin sijaan (working software over comprehensive documentation), 3) asiakasyhteistyö sopimusneuvottelun sijaan (customer collaboration over contract negotiation), 4) vastaus muutokseen suunnitelman noudattamisen sijaan (responding to change over following a plan (Hunt, 2006).

Kriteeri	**Viitekehys**				
	Scrum of Scrums (SoS)	Scaled Agile Framework (SAFe)	Large Scale Scrum (LeSS)	Disciplined Agile Delivery (DAD)	Lean Scalable Agility for Engineering (LeanSAFE)
Laajuus	Ohjelmisto, laitteisto, ja järjestelmät; joustava	Ohjelmisto	Ohjelmisto	Ohjelmisto	Ohjelmisto, laitteisto, ja järjestelmät
Erottava tekijä	Mahdollistaa scrumin kaikkiin käyttötapauksiin ja skaalautuu	On monimutkainen, omaa monia artefakteja, rooleja ja ohjeita	Mahdollistaa joustavuutta tarjotessaan vain suosituksia	On monimutkainen, kattaa monet mallit	Toimii kriittisten järjestelmien kanssa
Alla oleva teknologia	Scrum	Scrum ja muut agile-menetelmät, lean	Scrum	Scrum, lean	Scrum, lean
Käyttöönotto	Käytössä monissa yrityksissä	Käytössä useissa yrityksissä	Käytössä useissa yrityksissä	Käyttö on aloitettu	Käyttö on aloitettu
Skaalautuvuus	On joustava ja soveltuu hyvin erilaisiin tapahtumapaikkoihin	Kohdistuu isoihin yrityksiin, mutta on nähty raskaana	Voidaan soveltaa erilaisiin tapahtumapaikkoihin	Voidaan soveltaa erilaisiin tapahtumapaikkoihin	Voidaan soveltaa erilaisiin tapahtumapaikkoihin
Monimutkaisuus	Alhainen	Korkea	Keskitasoinen	Keskitasoinen	Keskitasoinen
Hinta	Alhainen	Korkea	Keskitasoinen	Keskitasoinen	Alhainen
Globaalisti hajautetut tiimit	Mahdollinen	Mahdollinen	Mahdollinen	Hankala	Mahdollinen

Taulukko 1. Viiden ketterän menetelmän vertailu suomennettu lähteestä (Paasivaara et al., 2017).

Taulukossa 1 on selostettu eri menetelmiä: Scrum of Scrums (SoS), Scaled Agile Framework (SAFe), Large Scale Scrum (LeSS), Disciplined Agile Delivery (DAD), Lean Scalable Agility for Engineering (LeanSAFE). Menetelmiä on käsitelty erinäkökulmista, kuten laajuus, erottava tekijä, alla oleva teknologia, käyttöönotto, skaalautuvuus, monimutkaisuus, hinta, globaalisti hajautetut tiimit. Laajuuden sisältö vaihtelee menetelmittäin – soveltamisalue vaihtelee ohjelmistoista järjestelmiin. Erottava tekijä vaihtelee merkittävästi myös menetelmittäin. Alla oleva teknologia taas ei juurikaan vaihtele viitekehyksittäin, sillä teknologia on pääasiassa Scrum tai jokin ketterä johdannainen. Käyttöönotto vaihtelee myöskin menetelmittäin – kaikki eivät ole vielä laajasti käytössä kuten DAD tai LeanSAFE. Jokainen taulukon 1 viitekehyksistä on hyvin skaalautuva. Monimutkaisuus ja hinta vaihtelevat menetelmittäin. Globaalisti hajautetuissa tiimeissä toimivat kaikki taulukon 1 menetelmät paitsi DAD.

Aspekti	SAFe	SoS	LeSS	DAD	Nexus	RAGE
Tiimin koko	50-120 henkeä julkaisujunissa	5-10 tiimiä	10 Scrum tiimiä, 7 henkeä x tiimi	200 henkeä tai enemmän	3-9 Scrum-tiimiä	Ei erityistä kokoa
Diffuusio	Korkea	Korkea	Keskiarvoinen	Matala	Matala	Matala
Kypsyysaste	Korkea	Korkea	Korkea	Keskiarvoinen	Matala	Matala
Monimutkaisuus	Korkea/Keskiarvoinen	Keskiarvoinen/Matala	Keskiarvoinen/Matala Scrum-tietoisille	Korkea (useita käytäntöjä)	Keskiarvoinen/Matala Scrum-tietoisille	Keskiarvoinen/Matala Scrum-tietoisille
Organisaation tyyppi	Perinteiset organisaatiot	Perinteiset ja ketterät organisaatiot	Suuret yritykset	Useita organisaatioita ja yrityksiä	Perinteiset ja ketterät organisaatiot	Perinteiset ja ketterät organisaatiot

Taulukko 2. Tiivistetysti ketteriä menetelmiä ja niiden ominaisuuksia suomennettu lähteestä (Kalenda et al., 2018).

Taulukossa 2 käsiteltävinä menetelminä ovat SAFe, SoS, LeSS, DAD, Nexus, RAGE (Recipes for Agile Governance in the Enterprise). Käsiteltävinä aspekteina ovat tiimin koko, diffuusio, kypsyysaste, monimutkaisuus, organisaation tyyppi. Menetelmät vaihtelevat toisistaan jokaiselta aspektiltaan merkittävästi – joissakin menetelmissä on samaa keskenään, mutta menetelmistä erottuu selvästi mm. erilaiset käyttötarkoitukset ja kypsyysasteet.

Suomennetusta Kalendan ja muiden (2018) kehittämästä taulukosta näkyy selvästi, kuinka laajaksi menetelmien voidaan ajatella skaalautuvan. Esimerkiksi Nexuksen voidaan ajatella skaalautuvan yhdeksään tiimiin asti, mutta mietittäessä organisaation rakennetta voidaan todeta, että yhdeksän on aika paljon Nexuksen tyyppiselle menetelmälle, jossa pyritään minimaalisilla lisäyksillä saamaan aikaan skaalautuva rakenne. Parhaiten skaalautuvat menetelmät ovat SAFe ja DAD taulukon 1 mukaan, sillä ne tarjoavat mahdollisuuden kehittää yli sadan hengen rakenteissa ohjelmistoa. Kuitenkin on

huomattava, että moni ketteristä menetelmistä pohjautuu Scrumiin (Paasivaara et al., 2018) ja niissä on käytössä Scrumin mukaiset sprintit ja jossakin muodossa myös roolit. Kokonaisketterä voi olla myös yksi ison mittakaavan projektimenetelmistä – toimintamalli on kehitetty Suomessa ja on luonteeltaan enemmän sopiva kokoluokaltaan suuriin projekteihin. Kehittäminen myös Kokonaisketterässä-toimintamallissa perustuu Scrumiin.

6.2 Ketterät menetelmät käytännössä

Tarkasteltaessa ketteriä menetelmiä projektiliiketoiminnassa tulee ottaa huomioon projektiliiketoiminnan eritysluonteisuus mahdollisesti lyhyine kestoineen ja ajallisesti epäjatkuvuuksineen. Projektit eivät välttämättä seuraa toinen toisiaan peräjälkeen ja projektien luonteen vuoksi täysin valmiita tuotteita ei välttämättä saada aikaiseksi. Tämän syyn vuoksi pitää jaotella projektit kolmeen erilaiseen kategoriaan, jossa käytettävissä oleva aika määrittää projektin ja tavoitteen luonteen. Ensimmäinen kategoria on 1) ns. proof-of-concept-projektit (PoC-projektit), joissa luodaan jonkinlainen ensimmäinen toteutus projektin jostakin osasta ja varmistetaan, että idea on toimiva.

Toisena projektityyppinä voidaan pitää ns. normaaleja projekteja, jotka kestävä esim. muutamasta viikosta muutamaan kuukauteen. Näiden projektien aikana on mahdollista toteuttaa useita sprinttejä ja luoda Minimum Viable Product eli MVP eli vähimmäinen kelvollinen tuote, joka toteuttaa raakileen tasoisena käyttäjän vaatimukset.

Kolmantena projektityyppinä voidaan pitää useita kuukausia taikka useita projekteja sisältävät ohjelmat, joissa toteutetaan ensin MVP ja myöhemmin parannetaan MVP:n pohjalta tuotetta vastaamaan tarvetta.

Tarkasteltaessa Scrumin soveltuvuutta on huomattava, että Scrumissa on aina vähintää kolme henkilöä – tuoteomistaja, scrum master ja kehitystiimin jäsen. Rakennelma on sinänsä jäykkä, kun ajatellaan pienimuotoisia projekteja, jotka kestävät kuukaudesta kahteen. Täysipäiväisesti osallistuminen Scrumin mukaisesti ei ole järkevää scrum masterin ja tuoteomistajan kannalta katsottuna, sillä resurssit ovat kuukaudesta kahteen kuukauteen kestävässä projektissa huomattavasti pienemmät kuin normaaleissa projekteissa.

Suppeassa projektissa (enimmillään pari kuukautta) tulee kenties merkitykselliseksi Lean Development -kehityssuunta, jossa ei noudateta välttämättä kaikkia Scrumin piirteitä, vaan keskitytään kehittämiseen suurimmaksi osaksi.

Isompien projektien puitteissa tulee tähdelliseksi käyttää jotakin muuta menetelmää kuin Lean Development. Kenties SAFe tai DAD on hyvä ratkaisu tarkasteltaessa menetelmien skaalautuvuutta – toisaalta jos skaalautuvuutta ei tarvita, voi jokin muukin malli olla hyvä esim. perusmuotoinen Scrum tai skaalautuva Nexus.

6.3 Tulevaisuutta: tekoälyn käyttäminen ketterässä ohjelmistokehityksessä

Tekoälyn käyttämistä on mietitty käytettävän erilaisissa ketterissä prosesseissa. Tutkittaessa on havaittu, että tekoälyä voisi käyttää useissa tilanteissa apuna. Tekoälyä voidaan käyttää näyttämisessä, mitä pitää tulla tehdyksi. Sitä voidaan myös käyttää analysoitaessa analytiikkajärjestelmässä kuvailemaan, mitä tulee tapahtumaan, ja mitä on tapahtunut. Analytiikkamoottoria voidaan käyttää kehitysjonon tapahtumien tunnistamiseen, riskien välttämiseen, kehitysjonon tapahtumien kehittämiseen, riskien ennakoimiseen ja kehitysjonon tapahtumien arviointiin. Suunnitelmamoottoria voidaan myös käyttää sprinttien suunnitteluun (Dam et al., 2019).

7. YHTEENVETO

Projektiliiketoimintaa voidaan harjoittaa eri muodoissa ja yhtä määritelmää projektille ja projektiliiketoiminnalle ei ole olemassa. Projektit ja siihen kytkeytyneet käsitteet ovat moniulotteisia ja vaihtelevat kirjoittajasta kirjoittajaan.

Projektinhallintamenetelmiä on useita ja niillä on tarkoitus tehdä projektien hallinnasta enemmän empiiristä ja suunnitelmallista. Perinteiset menetelmät hyödyntävät vesiputous- tai V-mallista toimintatapaa, jossa kehittäminen tehdään yhtenä osana projektia. Nykypäivänä entistä suositummaksi on tullut ketterät projektinhallintamenetelmät. Ketterissä menetelmissä työskentely on tehty inkrementaaliseksi ja toisteiseksi. Myös suunnittelu on joissain malleissa tehty ketteräksi ja itsensä päälle rakentavaksi.

Ketteriä menetelmiä käsiteltäessä on syytä huomata, ettei ole olemassa yhtä ainoaa ketterää menetelmää, vaan ketteriä menetelmiä on useita erilaisia. Erilaiset ketterät menetelmät sisältävät erilaisia rooleja ja hierarkioita. Myös rakenteet vaihtelevat ketterien menetelmien välillä. Toisaalta on huomattava, kuinka Scrum on vaikuttanut eri ketteriin menetelmiin – useissa menetelmissä ketteryys rakentuu Scrumin kaltaiseen rakenteeseen, joka perustuu itseään toteuttavaan kehitystiimiin.

LÄHTEET

AgileBusiness.org (2023) DSDM-project-framework. [Viitattu 5.9.2023]. Saatavissa: https://www.agilebusiness.org/dsdm-project-framework.html

AgileAlliance.org (2023) Agile Alliance. [Viitattu 5.9.2023]. Saatavissa: https://www.agilealliance.org/glossary/scrum-of-scrums

Antunes B., Santos D., Lopes E., Fidalgo F., Alves P. (2015) Blisstrail: An Agile Project Business Case Study. Procedia computer science. [Online] 64529–536.

Artto K., Martinsuo M., Kujala J. (2011) Project business. Helsinki, Finland, http://pbgroup.tkk.fi/en/

Project Management Institute (2021) The standard for project management and a guide to the project management body of knowledge (PMBOK guide). Seventh edition. Newtown Square, Pennsylvania: Project Management Institute, Inc.

AXELOS (2015) PRINCE2 Agile. TSO.

Bentley, C. (2020) Adaptable Project Management : a combination of Agile and Project Management for All (PM4A). 1st edition. Ely, Cambridgeshire: IT Governance Publishing.

Christou I., Ponis S., Palaiologou E. (2010) Using the Agile Unified Process in Banking. IEEE software. [Online] 27 (3), 72–79.

Cooke, J. L. (2021) The PRINCE2 Agile® Practical Implementation Guide - Step-By-step Advice for Every Project Type, Second Edition. Second Edition. Ely: IT Governance Ltd.

Danenas, P. & Garsva, G. (2012) Domain Driven Development and Feature Driven Development for Development of Decision Support Systems, in Information and Software Technologies. [Online]. Berlin, Heidelberg: Springer Berlin Heidelberg. pp. 187–198.

Ebert, C. & Paasivaara, M. (2017) Scaling Agile. IEEE software. [Online] 34 (6), 98–103.

Dam H. K., Tran T., Grundy J., Ghose A., Kamei Y. (2019) Towards Effective AI-Powered Agile Project Management, in 2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). [Online]. 2019 IEEE. pp. 41–44.

Hellström, M. (2023) Scheduling and time challenges in a complex project environment. Professorin luento Strategic Project Management- kurssilla 17.8.2023 Aalto yliopistossa.

Hunt, J. (2006) Agile Software Construction. 1st ed. 2006. [Online]. London: Springer London.

Kalenda M., Hyna P., Rossi B. (2018) Scaling agile in large organizations: Practices, challenges, and success factors. Journal of software: evolution and process. [Online] 30 (10), e1954–n/a.

Kisielnicki, J. & Misiak, A. M. (2017) Effectiveness of Agile Compared to Waterfall Implementation Methods in it Projects: Analysis Based on Business Intelligence Projects. Foundations of management. [Online] 9 (1), 273–286.

Larman, C. & Vodde, B. (2016) Large-Scale Scrum: More with LeSS. 1st edition. Addison-Wesley Professional.

Lehtonen T., Tuomivaara S., Rantala V., Känsälä M., Mäkilä T., Jokela T., Könnölä K., Kaisti M., Suomi S., Ylitolva M. et al. (2022) Sulautettujen järjestelmien ketterä käsikirja. Saatavissa: https://tt.utu.fi/embedded_kasikirja/index.html

Lock, D. (2020) Project management. Routledge.

Palmer, S. R. & Felsing, J. M. (2002) A Practical Guide to Feature-Driven Development. Prentice Hall

PMI.org (2023) Disciplined Agile. Saatavissa: https://www.pmi.org/disciplined-agile/introduction-to-disciplined-agile

Poppendieck, M. & Poppendieck, T. D. (2007) Implementing lean software development : from concept to cash. 1st edition. Place of publication not identified: Addison Wesley.

Scrum.org (2023) Scrum. [Viitattu 5.9.2023]. Saatavissa: https://www.scrum.org/learning-series/what-is-scrum

ScaledAgile (2023) Scaled Agile. [Viitattu 5.9.2023]. Saatavissa: https://scaledagile.com/what-is-safe/

ScaledAgileFramework (2023) Scaled Agile Framework. Saatavissa: https://scaledagileframework.com/

Soo-Haeng C. & Eppinger, S.D. (2005) A simulation-based process model for managing complex design projects. IEEE transactions on engineering management. [Online] 52 (3), 316–328.

Steward, D. V. (1981) The design structure system: A method for managing the design of complex systems. IEEE transactions on engineering management. [Online] EM-28 (3), 71–74.

Williams, L. 2001. Integrating pair programming into a software development process. Teoksessa: Ramsey, D., Bourque, P. & Dupuis, R. (eds.) Proceedings 14th Conference on Software Engineering Education and Training. 'In search of a software engineering profession' (Cat. No.PR01059). Conference on Software Engineering Education & Training (CSEE&T). February 19–21 2001 Charlotte, North Carolina. 27–36. [Viitattu 5.9.2023]. Saatavissa: https://doi.org/10.1109/CSEE.2001.913816