



Vaasan yliopisto
UNIVERSITY OF VAASA

Sanna Talonen

Laajan tuotekehitysprojektin organisointi ja hallinta ketterässä sovelluskehityksessä

Tekniikan ja innovaatiojohtamisen akateeminen yksikkö
Tietojärjestelmätieteen Pro gradu -tutkielma
Tietojärjestelmätieteen maisteriohjelma

Vaasa 2024

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen akateeminen yksikkö**

Tekijä:	Sanna Talonen		
Tutkielman nimi:	Laajan tuotekehitysprojektin organisointi ja hallinta ketterässä sovelluskehityksessä		
Tutkinto:	Kauppatieteiden maisteri		
Oppiaine:	Tietojärjestelmätieteen maisteriohjelma		
Työn ohjaaja:	Jouni Lampinen		
Valmistumisvuosi:	2024	Sivumäärä:	99

TIIVISTELMÄ:

Tämän tutkielman tarkoituksena on luoda ymmärrys laajojen tuotekehitysprojektien projektinhallinnan haasteista ja pyrkiä löytämään niihin ratkaisuja ketteristä sovelluskehityksistä. Tutkielman kohteena ovat laajat tuotekehitysprojektit, joissa on kokeiltu perinteisiä projektinhallintamalleja ja havaittu niissä haasteita. Tutkielmassa pyritään löytämään parempi asiakastyytyväisyys käyttäen ketteriä menetelmiä.

Tutkielman teoreettisen viitekehyksen rakentaminen aloitetaan esittelemällä ketterän menetelmän peruskäsitteitä. Teoriaosuudessa pyritään keskittymään nimenomaan laajojen organisaatioiden ketteriin menetelmiin saatavilla olevien tietojen valossa. Työ jatkuu lähtötilanteen kuvauksella. Tutkielman aineisto hankittiin keskisuuren kansainvälisen IT-yrityksen Suomen osastolta.

Tutkitun aineiston perusteella voitiin todeta, että asiakkaat olivat niin pettyneitä ainasiin tuotekehitysaikataulun venymisiin, että heille sopi mikä tahansa muutos kuten esimerkiksi ketterään menetelmään siirtyminen. Riskeistä huolimatta ketterän menetelmän soveltaminen onnistui erinomaisesti ja asiakkaalle pystyttiin toimittamaan uusi tuote sovitussa ajassa. Asiakkaan innostus ja tyytyväisyys pehmittivät yrityksen ylintä kansainvälistä johtoa suhtautumaan neutraalimmin ketterään menetelmään, mutta työ jatkuu vielä tällä saralla ja kaikki mahdolliset kommunikointitilanteet pitää käyttää positiivisesti hyväksi. Tulevaisuudessa asiakkaan sovelluksiin pitää pystyä pureutumaan syvemmin, että ratkaisut ehditään aloittaa ajoissa. Tiimien tulee pilkkoa ominaisuuksia pienempiin osa-alueisiin, että työ voidaan jakaa mahdollisimman usealle tiimille.

AVAINSANAT: projektinhallinta, ohjelmistotuotanto, ohjelmistotuotantoprosessi, ohjelmistokehitys, ketterät menetelmät, ison organisaation muutos ketterään menetelmään

UNIVERSITY OF VAASA**School of Technology and Innovation****Author:** Sanna Talonen**Title of the thesis:** Laajan tuotekehitysprojektin organisointi ja hallinta ketterässä sovelluskehityksessä**Degree:** Master of Science in Economics and Business Administration**Programme:** Master's Programme in Information Systems**Supervisor:** Jouni Lampinen**Year of graduation:** 2024**Pages:** 99

ABSTRACT:

The purpose of this study is to get an understanding of the challenges of project management in the large-scale research and development projects and try to find solutions for agile methods. The target of the research is extensive research and development projects, which have been run with traditional project management models and challenges have been met. The aim of the study is to find better customer satisfaction by using agile methods.

The construction of a theoretical framework for the study starts by introducing the basic concepts of agile methodology. The aim of the theoretical part is to focus specifically in large organizations agile methods with the available information. Work continues by defining the baseline. The material of the study was got from Finland section of the medium-size international IT company.

It was found of from the researched material that customers were so disappointed from the schedule delays of development project that they were ready to accept any changes like this agile method. Regardless of the risks the agile development project success and the new product were delivered to customer on time. The satisfaction of the customers influences positive to the attitude of the directors in America, but positive communication needs continue in all possible situations. In the future, the application of the customer's needs to be studied deeply, so that solution can be started early enough. Team needs to divide feature in the smaller pieces, so that it can be done by many teams.

KEYWORDS: project management, software engineering, software production process, software development, agile methods, large-scale agile transformation

Sisällys

1	Johdanto	10
1.1	Tutkielman tarkoitus ja tavoitteet	11
1.2	Tutkielman kulku	12
2	Ketterät menetelmät teoria	13
2.1	Scrum perusperiaatteet	18
2.2	Agile-mallinnus käytännössä	19
2.2.1	Agile-mallinnuksen arvot	19
2.2.2	Keskeisimmät periaatteet	21
2.2.3	Organisaatio	22
2.2.4	Scrum toimintatavat ja aktiviteetit	24
2.3	Sovellukset laajoissa organisaatioissa	25
3	Lähtötilanne	29
3.1	Vesiputousmalli	29
3.1.1	Käytössä ollut tuotekehitysprosessi	30
3.1.2	Haasteet ja ongelmat todettiin testausvaiheessa	32
3.2	Iteratiivinen malli	34
3.3	Haasteet	37
3.4	Tavoitteet	41
4	Työn rakenne	44
4.1	Tutkimusaineisto	44
4.2	Tutkimusstrategia eli case-tutkimus	45
4.3	Tutkimuksen toteutus ja analysointi	46
5	Ketterän menetelmän soveltaminen laajaan projektiin	48

5.1	Vaatimusten priorisointi	50
5.2	Organisaation luominen	52
5.3	Hallintamenetelmien luominen	55
5.3.1	Palaverikäytäntö	56
5.3.2	Raportointi	57
5.4	Testaus	59
5.5	Työkalut	62
5.6	Metriikka	63
5.7	Projektin dokumentit	71
6	Projektinhallinnan arviointi käyttäjän näkökulmasta	72
6.1	Asiakkaat	73
6.2	Johtoryhmä	75
6.2.1	Projektin ohjausryhmä	75
6.2.2	Yrityksen Suomen osa-alueen projektijohtoryhmä	76
6.2.3	Yrityksen projektijohtoryhmä	77
6.3	Projektitiimi	79
6.4	Projektipäällikkö	80
7	Tulokset	83
7.1	Laajan tuotekehitysprojektin haasteet	83
7.2	Ketterän menetelmän soveltamisen tutkiminen laajaan tuotekehitysprojektiin	84
7.3	Ketterän menetelmän soveltaminen ja parannusehdotukset	85
8	Johtopäätökset	92
8.1	Laajan tuotekehitysprojektin haasteet	92
8.2	Ketterän menetelmän soveltamisen tutkiminen laajaan tuotekehitysprojektiin	93

8.3 Ketterän menetelmän soveltaminen ja parannusehdotukset	94
Lähteet	97

Kuvat

Kuva 1.	Ketterän menetelmän peruseriaate.	13
Kuva 2.	Vesiputousmalli ja iteratiivinen malli.	14
Kuva 3.	Vesiputous- ja iteratiivinen malli yhdistettynä ketteräksi ohjelmistokehitysmalliksi.	15
Kuva 4.	Lean, Agile, Scrum ja XP menetelmien suhde.	16
Kuva 5.	Ketterän ohjelmistokehityksen julistus.	16
Kuva 6.	Scrum kehys.	19
Kuva 7.	Tuotekehitysorganisaation eri ketterän menetelmän sovellustasot.	28
Kuva 8.	Tuotekehitysprojektin vaiheet.	29
Kuva 9.	Tyypillinen projektiorganisaatio.	30
Kuva 10.	Suomen yksikön prosessi sulautettuna globaaliin prosessiin.	32
Kuva 11.	Rational Rose integraatiomalli yrityksessä.	35
Kuva 12.	EMS-tuotekehityksen yleisiteraatiosuunnitelma.	36
Kuva 13.	Integraatiotestaussuunnitelma yhdelle uudelle toiminnolle.	37
Kuva 14.	Tuotekehitysprojektin aikataulusuunnitelma ja sen toteutuminen.	38
Kuva 15.	Projektin tuotteiden määrän (y-akseli) suunnittelu ja toteutus kuukausittain.	39
Kuva 16.	Projektin kumulatiivinen henkilövuosisuunnitelma ja –toteutumamies-työkuukausina.	41
Kuva 17.	Kehitysmallit.	43
Kuva 18.	Ensimmäinen projektiorganisaatio.	54
Kuva 19.	Uusi organisaatio kahden kuukauden jälkeen.	55
Kuva 20.	Kolmen viikon sprintin ylätasoon tapahtumat aikajanalla.	56
Kuva 21.	Uuden ominaisuuden testaustapahtumien kappalemääräinen suorittaminen scrum-tiimissä kumulatiivisesti.	59
Kuva 22.	Regressiotestitulokset neljän viikon kumulatiivinä.	61
Kuva 23.	Uusien ominaisuuksien kappalemääräinen toimitusvalmiuden suunnitelma ja toteutuma (y-akseli) kolmen viikon sprinteittäin (x-akseli).	64

Kuva 24.	Yhden tiimin tehtävien suunnittelu- ja toteumaedistymäkaavio kolmen viikon iteraatioissa.	66
Kuva 25.	Automatisoitujen testien viikottainen tilanne.	67
Kuva 26.	Viikoittain muuttuneet C++ lähdekieliset ohjelmarivit.	69
Kuva 27.	Projektin uusien ominaisuuksin kappalemääräinen toimitusvalmiuden suunnitelman ja toteutuksen muutokset kuukausittain.	70

Lyhenteet

APO	Area product owner, aluetuoteomistaja
CIP	Controlled introduction plan,
Daily Scrum	Päiväpalaveri
FO	Feature owner, ominaisuusomistaja
IOT testing	Interoperability testing, yhteentoimivuustesti
MDS testing	Multi-dimensional system testing, eri sovelluskompinaatioiden testaaminen
MPR	Management phase review, tuotekehitysprosessin virstanpylvään katselmointi ja päätös projektin johtoryhmän kanssa
MSR	Management status review, tuotekehitysprosessin määrittelemä tilannekatselmointi projektin johtoryhmän halutessa
NF testing	New feature testing, uuden ominaisuuden testaus
NPI	New product introduction, uuden tuotteen esittely
PCRG	Product council review group, tuotteen päätöksiä tekevä johtoryhmä
PLS	Product launch support, tuotteen asiakkaalle viennin tuki
PB	Product Backlog, tuotteen kehitysjojo
PO	Product Owner, tuoteomistaja
SM	Scrum Master, scrum mestari
Sprint	Sprintti
ST	Scrum Team, scrumtiimi (ts. tuoteomistaja, kehitystiimi ja scrum-mestari)
SDS testing	Single-dimensional system testing, yhden sovelluskompinaation testaaminen
SB	Sprint Backlog, sprintin kehitysjojo

1 Johdanto

IT ala on jatkuvan muutoksen kourissa. Vuosikymmenien varrella on nähty IT-alan räjähtävä nousu, uusien innovaatioiden tulo ja myöskin kuplan puhkeaminen. Innovaatiot ja asiakkaiden tarpeet ovat määritelleet markkinoita. Innovaatiot ovat joko puhjenneet kukkaansa tai kuihtuneet pois. Kaiken tämän keskellä projektinhallinta on pysynyt pääosin muuttumattomana. Vesiputousmalli on vielä paljon käytetty projektinhallintametodi. Pienet yritykset ovat kokeilleet ja siirtyneet uusiin projektinhallintamekanismeihin, mutta isot yritykset ovat hitaita kokeilemaan uutta. Toisaalta taas uusien, parempien menetelmien hyöty näkyisi nimenomaan laajoissa projekteissa. Niissä haasteet ovat mittavia, joten haasteen muuttaminen positiiviseksi voimavaraksi näkyisi myös suurempana hyötynä kuin pienen yrityksen pienessä projektissa.

Tässä tutkielmassa tarkastellaan laajan IT-tuotekehitysprojektin haasteita ja etsitään parempia ratkaisuvaihtoehtoja sen organisoimiselle ja hallinnalle ketterän menetelmän avulla, josta on hyviä kokemuksia pienillä projekteilla. IT-yrityksellä tarkoitan tietoliikennelaitteistoja ja -ohjelmistoja tuottavaa keskiuurta kansainvälistä yritystä, jolla on laaja asiakaskunta (yli 200 asiakasta) ympäri maailmaa. Asiakkaat ovat tietoliikennepalveluja tarjoavia verkko-operaattoreita. Laajalla tuotekehitysprojektilla tarkoitan henkilömäärältään isoa (250 suunnittelijaa) ja maantieteellisesti laajasti levinnyttä (kolme maanosaa).

Työn lopputulos on selvitys, jossa ketterää menetelmää on kokeiltu laajaan tuotekehitysprojektiin. Ketteristä menetelmistä käytämme Scrum menetelmään, sillä sen lisäksi, että se on yrityksessämme tunnetuin ja siitä meillä on eniten kokemusta Scrum on myös yleisesti ottaen suosituin ketteristä menetelmistä (Theocharis 2015). Kaksi kokenutta projektipäällikköä ovat käyneet Scrum mestari kurssin ja he suosittelivat nimenomaan Scrum-menetelmää. Toisaalta yrityksen verkkohallintaryhmä

on käyttänyt Scrum-menetelmää usean vuoden hyvällä menestyksellä. Vaikka tämä ryhmä ei ole osana tulevaa tuotekehitysprojektia, niin heiltä kuitenkin voimme saada käyttökokemuksia ja apua tarvittaessa.

Käytän tapaustutkimusta eli case-tutkimusta, joka soveltuu hyvin teorian ja käytännön kehittymisaskeliin ja antaa hyvät mahdollisuudet ymmärtää tapahtumien kulkua ja muutoksia. Selvitys keskittyy nimenomaan projektin organisointiin ja hallintaan, sillä näin laajan tuotekehitysprojektin siirtäminen ketterään menetelmään on erittäin haastavaa ja siitä on vain vähän kirjoitettua tietoa. Haluan jakaa yhden esimerkin ketterän menetelmän soveltamisesta laajaan tuotekehitysprojektiin. Toivon, että tämä työ auttaa ja tukee jotain toista samassa tilanteessa olevaa yritystä, joka pohtii samoja kysymyksiä.

1.1 Tutkielman tarkoitus ja tavoitteet

Tutkielman tarkoituksena on tutkia ketterän menetelmän soveltuvuutta laajaan tuotekehitysprojektiin. Tutkimusongelmaa lähestytään seuraavan kolmen tavoitteen kautta.

Ensimmäisen tavoitteena on **luoda ymmärrys laajan tuotekehitysprojektin haasteista**. Mitkä ovat suurimmat ongelmat tuotekehitysprojektin suhteen? Mitkä asiakas kokee pettymykseksi? Mitä haasteille on tehty tähän mennessä? Ensimmäiseen tavoitteeseen vastataan teorialuvuissa tarkastelemalla aiheeseen liittyvää kirjallista ja sähköistä aineistoa sekä tutkimalla tuotekehitysprojektin lähtötilanteen haasteita.

Toisena tavoitteena on **tutkia ketterän menetelmän soveltamista laajaan tuotekehitysprojektiin**. Miten laaja projekti eroaa pienestä projektista? Mitä asioita pitää erityisesti ottaa huomioon isossa organisaatiossa? Toiseen tavoitteeseen

vastataan tutkimalla ketterän projektinhallinnan kirjallisuutta ja soveltamalla sitä kansainvälisen IT-yrityksen projektinhallintaa.

Kolmantena tavoitteena on **testata ketterää menetelmää laajassa projektissa ja analysoida sen soveltuvuus mahdollisine parannusehdotuksineen**. Tähän tavoitteeseen päästään analysoimalla laajan projektin onnistumiskokeilua.

1.2 Tutkielman kulku

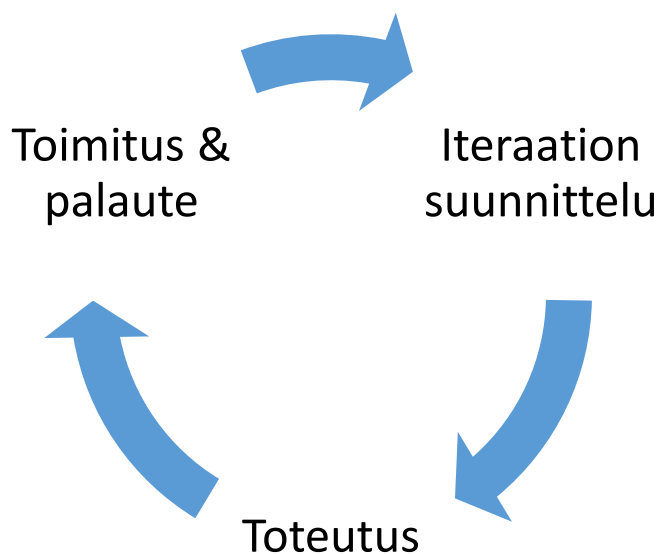
Tutkielma alkaa tärkeiden käsitteiden kuvaamisella. Syvennän niihin liittyvää teoriaa kappaleessa kaksi. Kolmannessa kappaleessa kuvaan lähtötilanteen haasteineen ja tavoitteineen. Työn rakenteen kuvaan kappaleessa neljä.

Viidennessä kappaleessa sovellan ketterän menetelmän teoria tietoa laajaan tuotekehitysprojektin muuttamisessa ketterän menetelmän mukaiseksi. Kuudennessa kappaleessa arvioin projektinhallinnan onnistumista käyttäjän näkökulmasta. Seitsemännessä kappaleessa kerään yhteen selvityksen tulokset.

Kahdeksas kappale sisältää johtopäätökset ja keskeiset tulokset. Kerron siellä myös toimenpidesuositukseni ja jatkotoimenpide-ehdotukseni.

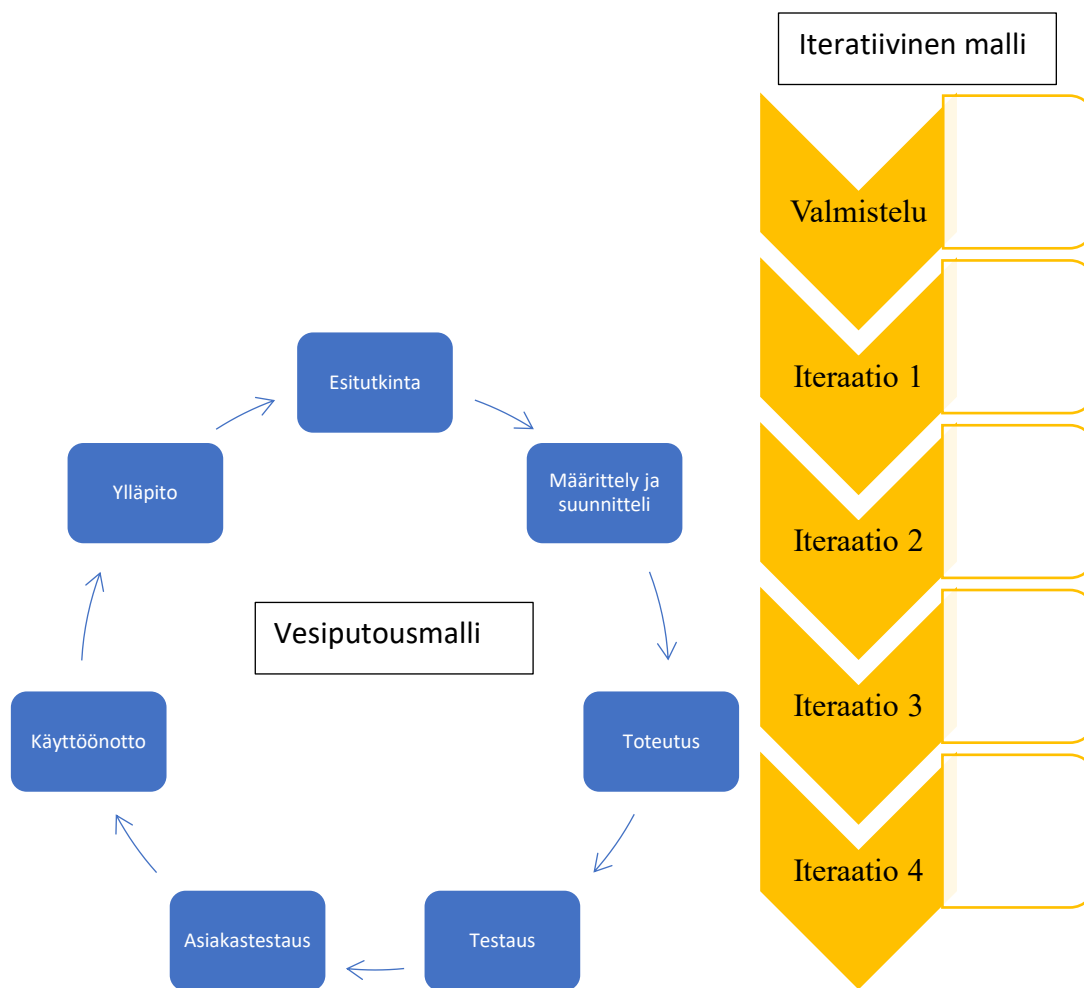
2 Ketterät menetelmät teoria

Ketterä ohjelmistokehitysmenetelmä (agile software development) on tällä hetkellä tehokkain projektinhallinnan menetelmä. Peruseriaate on, että monitaitoinen tiimi tekee iteraatiokierroksen suunnitelman, toteuttaa sen ja toimittaa toimivan ohjelmistokokonaisuuden, kommunikoi suoraa, reagoi muutoksiin nopeasti ja kehittyä jatkuvasti kuten kuvassa 1 on esitelty.



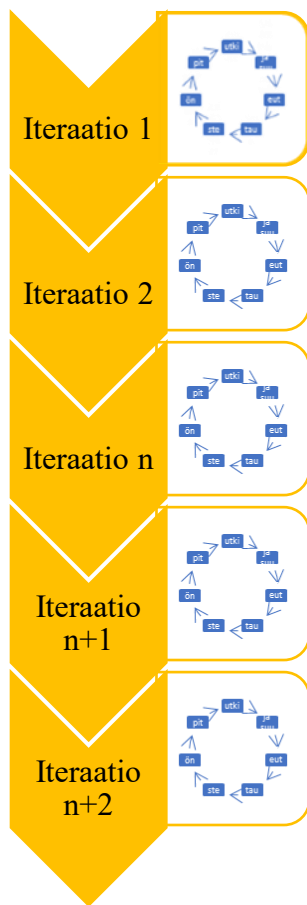
Kuva 1. Ketterän menetelmän peruseriaate.

Ketterät menetelmät ovat ihan uusi ohjelmistokehitysmenetelmä, mutta toisaalta se sisältää vanhat menetelmät erinomaisesti soveltaen. Yksinkertaistaen voisi sanoa, että ketterät menetelmät ovat yhdistäneet vesiputousprosessin iteratiiviseen prosessiin. Kuvassa 2 on yksinkertaistettu vesiputousmalli, jossa vaiheet tehdään peräkkäin sekä iteratiivinen malli, jossa tuote valmistuu pala palalta etukäteen suunnitelluissa iteraatioissa.



Kuva 2. Vesiputousmalli ja iteratiivinen malli.

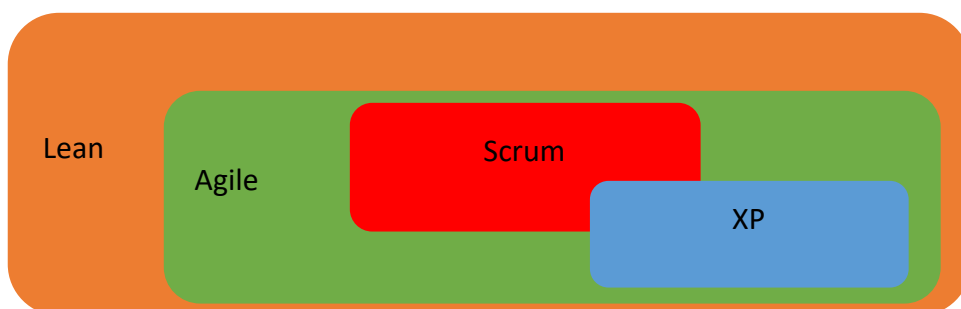
Kuvassa 3 on yhdistetty nämä kaksi mallia ketteräksi menetelmäksi. Siinä yhdessä iteraatiossa tehdään kaikki vesiputousmallin vaiheet pienelle tuotteen palaselle. Iteraation lopussa on siis olemassa testattu toimitettavaksi kelpaava pieni uusi toiminnallisuus. Toisin kuin iteratiivisessa mallissa ketterän menetelmän iteraatiokierroksien määrää ei voida tarkkaan sanoa. Tai jos kierrosten määrä sanotaan tarkkaan, niin sisältöä ei silloin voida kertoa etukäteen. Ketterissä ohjelmistotuotantomenetelmissä yhdistyy aiempien prosessien hyvät puolet.



Kuva 3. Vesiputous- ja iteratiivinen malli yhdistettynä ketteräksi ohjelmistokehitysmalliksi.

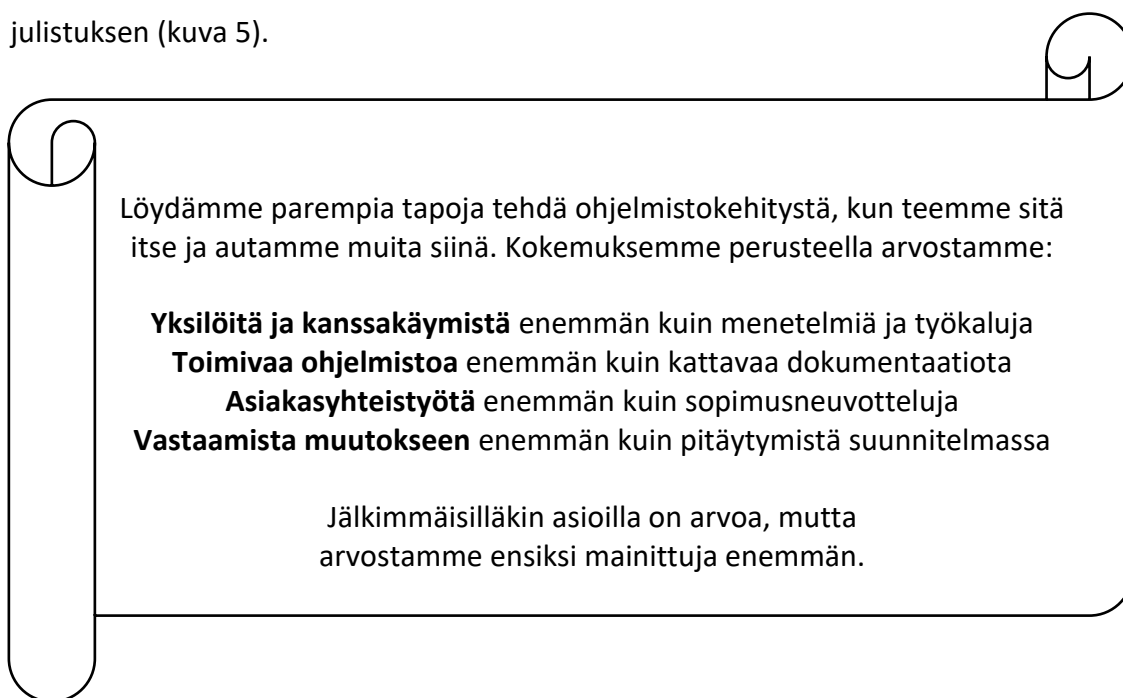
Jos kerran vesiputous- ja iteratiivinen malli ovat sulautuneet, niin herää kysymys, että onko ketterät menetelmät sittenkään puhtaasti ketteriä menetelmiä vai onko siihen sulautunut muitakin menetelmiä. Theocharis (2015) on tutkinut tiimensä kanssa vesiputousmallin ja Scrum-menetelmän yhteyttä. Hän totesi, että heidän saamansa tiedon valossa erityisesti laajat kehitysprojektit käyttävät vesiputousmallin ja erityisesti Scrumin yhdistelmää. Erityisesti laajat organisaatiot käyttävät perinteistä prosessia perushallintoon sekä rajapintana muihin yrityksiin ja liiketoimintaprosesseihin kun taas kehitystiimit käyttävät puhtaammin ketteriä menetelmiä.

Agile on yleisnimi useille erilaisille ohjelmistokehitysmetodeille kuten Scrum, eXtreme Programming, DSDM ja Crystal Clear, joiden keskinäinen suhde on kuvattu kuvassa 4. (Björkholm 2021: 9).



Kuva 4. Lean, Agile, Scrum ja XP menetelmien suhde.

Vuonna 2001 kokoontui 17 eri taustoilla varustettua asiantuntijaa keskustelemaan kahdeksi päiväksi ja lopputuloksena he allekirjoittivat ketterän ohjelmistokehityksen julistuksen (kuva 5).



Kuva 5. Ketterän ohjelmistokehityksen julistus (Agile Manifest b. 2001)

Agile sisältää kaksitoista peruseriaatetta, joista yhdeksän tärkeintä ovat lueteltu seuraavaksi (Björkholm 2021: 9).

1. **Priorisoida ja keskittyä** maksimoidakseen sijoitetun pääoman tuotto.
2. **Läpinäkyvyys** nähdä ongelma ja rakentaa luottamus.
3. **Iteratiivinen ja inkrementaalinen** kehitys oppiakseen, saadakseen palautetta ja pienentää riskejä.
4. **Yhteistyö** väärinymmärrysten minimoimiseksi ja erottaakseen ja turvata liiketoimintamallin tavoitteet.
5. **Kannustaa ja toivottaa muutokset tervetulleiksi**, sillä muutokset ovat kuitenkin todellisuutta.
6. **Yksinkertaiset työkalut** pystyäkseen soveltamaan prosessia toimintaan.
7. **Tavoitekeskeisyys** taatakseen tehokkaan työskentelyyn.
8. **Alituinen parantaminen** tullakseen tehokkaammaksi.
9. **Korkea laatu**, sillä virheet ovat kalliita.

Lean peruseriaate on minimoida aika idean ja maksettavan tuotteen välillä. Tyypillisesti keskitytään työskentelemään kauaskantoisesti, minimoimaan riskit ja olemaan tehokas välttyäkseen tekemästä tarpeettomia asioita ja näkyviä ongelmia sekä parantamaan jatkuvasti prosessia. Keskeistä on nopea toimitus eli lyhyet toimitusajat, jolloin tuotannon pullonkauloja etsitään ja poistetaan. Tähän on olemassa useita työkaluja, joita Lead-periaatteessa käytetään. (Björkholm 2021: 10-11).

Extreme Programming, josta käytetään lyhennettä XP, on kehittynyt vuosien varrella ohjelmistotuotannon parhaista menetelmistä. Parityöskentely ja -ohjelmointi yhdistää molempien kokemuksen ja usein johtaa parempaan tuottavuuteen ja laatuun. (Björkholm 2021: 16-17).

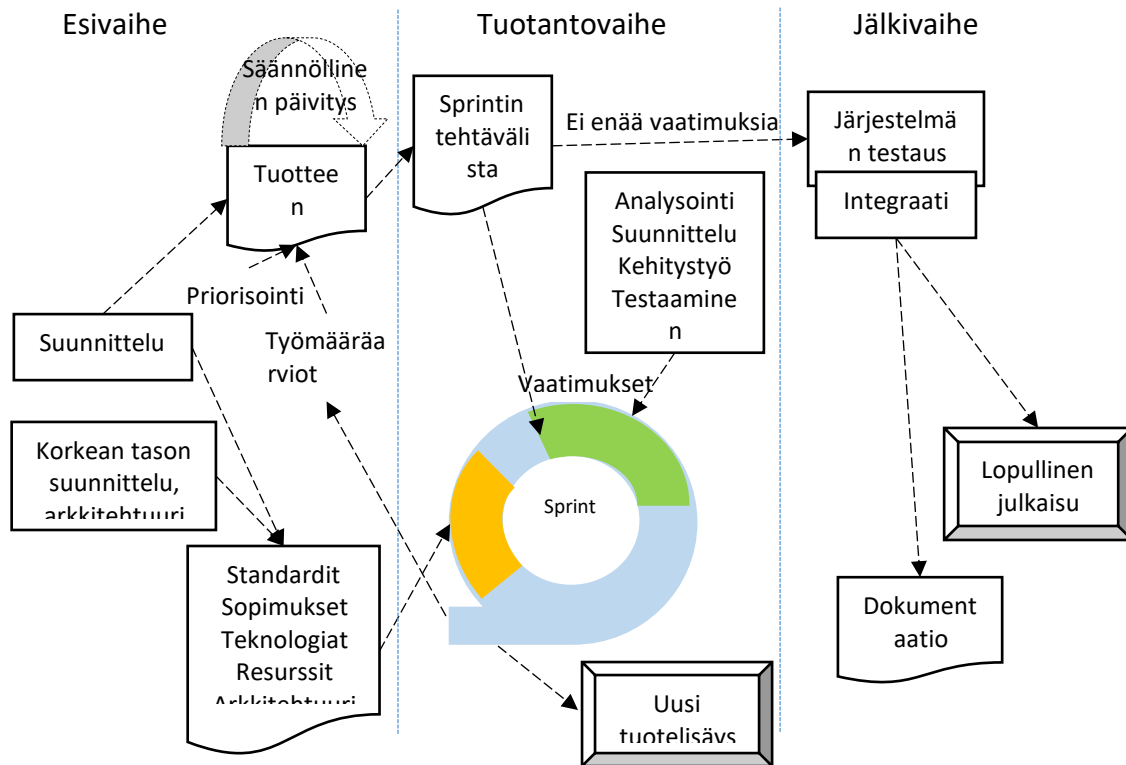
2.1 Scrum peruseriaatteet

Kuten edellä on todettu, niin ketteriä menetelmiä on useita, mutta tässä tutkielmassa keskitymme pääasiassa Scrum menetelmään. Vuonna 1986 Hirofumi Takeuchi ja Ikujiro Nonaka kuvasivat scrumin kehitysprosessin. Scrum-sana tulee rugby-joukkueesta, jossa ryhmä pyrkii etenemään yksikkönä ja tekee tiiviisti yhteistyötä. Scrum-kehittäjiä ovat Jeff Sutherlandia, Ken Schwaberiä, John Scumniotalesia ja Jeff McKenna. Ketterän ohjelmistokehitysmenetelmän keskeinen periaate on, että kehitys tiimi asettaa itselleen tavoitteet ja löytää keinot niiden saavuttamiseksi. Painopiste on joustavuudessa ja oppimisessa kierroksen aikana. (Wikipedia 2024, Scrum)

Scrum-prosessi sisältää kolme vaihetta, jotka on esitetty kuvassa 6: esivaihe, tuotantovaihe ja jälkivaihe. **Esivaiheessa** luodaan tuotteen työlista kaikkine vaatimuksineen, jotka asetetaan tärkeysjärjestykseen ja niille määritellään työmääräarviot. Tässä vaiheessa määritellään myös projektitiimi ja muut projektin puitteet sekä korkean tason suunnitelma ja arkkitehtuurisuunnitelma perustuen tuotteen työlistan vaatimuksiin. (Schwaber & Beedle 2002).

Tuotantovaiheessa kukin iteraatiokierros sisältää vaatimusten analysoinnin, suunnittelun, toteutuksen, testauksen ja ohjelmiston toimituksen. Tärkeysjärjestyksessä olevasta tuotteen työlistasta valitaan iteraatioissa toteutettavat työt, joista muodostuu sprintin työlistaan. Yksi iteraatiokierros kestää viikosta kuukauteen, iteraatiokierroksessa tehdään useiden tuotteiden palasia ja samaa tuotetta tehdään useissa iteraatioissa. Loppupalaverissa katselmoidaan tulokset ja päivitetään tekemättömien töiden lista. Kun tekemättömyyttä toiminnallisuutta ei enää ole, niin iteraatiokierros voidaan todeta viimeiseksi. (Schwaber & Beedle 2002).

Jälkivaiheessa suoritetaan ohjelmiston osien integrointi, järjestelmätestaus ja dokumentaatio. Lopullinen tuote voidaan toimittaa asiakkaalle ja projekti on valmis.



Kuva 6. Scrum kehys. (Schwaber & Beedle 2002).

2.2 Agile-mallinnus käytännössä

Seuraavaksi kuvaan Agile-mallinnuksen tärkeitä käytännön periaatteita kuten Agile-mallinnuksen arvot, keskeisimmät periaatteet, organisaatio sekä Scrum toimintatavat ja aktiviteetit.

2.2.1 Agile-mallinnuksen arvot

Ambler (2002: 19) on jakanut Agilen arvot viiteen päätekijään: kommunikointi, yksinkertaisuus, palaute, rohkeus ja nöyryys.

Sebestyén (2016: 199-200) on tutkinut eri kommunikaatiomuotoja: kertominen, näyttäminen ja toiminnan harjoitus. Taulukko 1 todistaa selkeästi, että näiden kolmen kommunikaatiomuodon yhteiskäyttö on kaikista tehokkain sekä lyhyellä, mutta varsinkin pitkällä tähtäimellä.

Taulukko 1. Oppimistehokkuus ja kyky reagoida ajan kanssa. (Sebestyén 2016: 200)

	Kommunikaatiomuoto		
	Kertoa	Kertoa ja näyttää	Kertoa, näyttää ja toiminta
Muistaa kolmen viikon päästä	70 %	72 %	85 %
Muistaa kolmen kuukauden päästä	10 %	32 %	65 %

Kommunikointi on kaksisuuntaista, sillä siinä sekä annetaan että saadaan tietoa. Tuotekehitysohjelmiston menestykseen tarvitaan tehokasta kommunikointia, jossa koko projektin henkilöstö suunnittelijasta projektin sidosryhmiin kommunikoi keskenään. Mallinnus auttaa omien ideoiden kommunikoinnissa, muiden ideoiden ymmärtämisessä sekä kokonaiskuvan hahmottamisessa. (Amber 2002: 20).

Ohjelmistokehityksessä **yksinkertaistamista** ei yksinkertaisesti voi painottaa liian paljon! Peruseriaate on pitää tänään mallinnus niin yksinkertaisena kuin mahdollista ja murehtia huomisen mahdollisista mallinnushaasteista huomenna. (Amber 2002: 22).

Palautteen saaminen varmistaa, että työ on menossa oikeaan suuntaan. Kehitystiimi itsessään on yksi tärkeä palautteen lähde. Jos mahdollista, niin kohderyhmä pitäisi osallistua ja antaa palautetta jo tuotekehityksen aikana. Varmin tapa saada palautetta

on antaa projektin sidosryhmän käyttää ohjelmistoa. Neljäntenä sidosryhmän hyväksyntätestit varmentavat vaatimusten toteutumisen oikealla tavalla. (Amber 2002: 22-23).

Agile-kehityksen aloittaminen uudessa toimintaympäristössä vaatii **rohkeutta**. Vain tarpeellisen dokumentoinnin tekeminen vaatii rohkeutta. Totuttujen toimintamallien kyseenalaistaminen vaatii rohkeutta. Monimutkaisuuden hylkääminen ja yksinkertaisuudessa pysyminen vaatii rohkeutta. Priorisointi vaatii rohkeutta. Tiimiin luottaminen vaatii rohkeutta. Vaatii rohkeutta myöntää oma vajavaisuus ja virheensä. Mutta ennen kaikkea vaatii rohkeutta luottaa, että tulevaisuuden ongelmat pystytään ratkaisemaan tulevaisuudessa. (Amber 2002: 24-25).

Parhailla suunnittelijoilla on **nöyryys** huomata, että he eivät tiedä kaikkea. Ohjelmistokehitysprojekti tarvitsee paljon erilaisia kykyjä: laaja-alaista ymmärrystä ja syvällistä osaamista. Toisen arvostaminen tasavertaisena asiantuntijana on nöyryyttä, joka auttaa menestymään. (Amber 2002: 25-26).

2.2.2 Keskeisimmät periaatteet

”**Ohjelmisto** on sinun päätavoitteesi” (Amber 2002: 28). Upea dokumentointi, mahtavat johdon statusraportit tai tuotemallit voivat oli kivoja, mutta jos ne eivät vie päätavoitetta eteenpäin, niin niihin ei kannata hukata aikaa. Täytyy löytyä rohkeutta keskittyä siihen mikä on tärkeää eli ohjelmistokehitys. (Amber 2002: 28).

”Toinen tavoite on **seuraavan ponnistuksen mahdollistaminen**” (Amber 2002: 28). Laadukkaan ohjelmiston kehittäminen ei yksin riitä, vaan pitää luoda riittävä

dokumentointi jatkon tehokkaan mahdollistamiseksi, osaamisen siirtämiseksi suunnittelijalta toiselle ja henkilöstön motivoimiseksi. Järjestelmään kehittäessä pitää osata pitää silmällä myös tulevaisuutta. (Amber 2002: 28).

Matkusta vähin matkatavaroin haluaa muistuttaa, että kaikki dokumentit ja mallinnukset, jotka päätetään pitää, täytyy myös ylläpitää matkan aikana. Vaatii rohkeutta ja hyvää kommunikointia karsia ylläpidettävä aineisto mahdollisimman pieneksi, mutta kuitenkin riittäväksi. (Amber 2002: 29).

Oletus yksinkertaisen ratkaisun hyvydestä on toimivat periaate. Yksinkertaisin ratkaisu on myös helpoin ylläpitää.

Muutoksia tapahtuu jatkuvasti. Juuri muutokset tekevät ohjelmistokehityksestä niin jännittävän! **Muutosalttius** on Agile-mallintajan peruspiirre. Kun ohjelmisto on rakennettu yksinkertaiseksi, niin vaatimusten muuttuminen viime hetkelläkin on mahdollista. (Amber 2002: 30).

2.2.3 Organisaatio

Agile toimintamallissa on hieman erilainen tapa nähdä organisaatio kuin aiemmissa malleissa, sillä metodin mukaan vaikeat asiat pitää yksinkertaistaa niin paljon kuin vain mahdollista. Projektioorganisaatio sisältää seuraavat roolit: ohjausryhmä, viiteryhmä, projektinomistaja, projektijohtaja, tiimi ja asiakas. (Gustavsson 2007: 51-52).

Näiden lisäksi on olemassa kaksi roolia, jotka harvoin määritellään projektimääritelmässä: linjaesimies ja resurssiomistaja. Linjaesimies vastaa ja omistaa

tuotteen ja sen arkkitehdin pitkällä tähtäimellä, kun projekti on loppunut. (Gustavsson 2007: 54). Resurssiomistaja vastaa projektin henkilöstöstä, ja tämä vastuu on usein jakautunut usealle päällikölle organisaatiossa. Projektit tyypillisesti sisältävät myös materiaali- ja työkaluhankintoja, joten tarvitaan myös taloudesta vastaava päällikkö. (Gustavsson 2007: 55).

Ohjausryhmä tulisi olla mahdollisimman pieni, jolloin päätöksenteko on nopeaa ja tehokasta. Palaverien tulee olla nopeita ja pitkiä keskusteluja välttävä. Jos päätöstä ei pystytä tekemään, niin sekin on tietoinen odottamispäätös ja silloin kirjataan jollekin ohjausryhmän jäsenelle lisätutkimistehtävä. (Gustavsson 2007: 56-59).

Viiteryhmässä pitää vähintäänkin olla käyttäjä ja projektitulosten edunvalvoja. Tämä foorumi muodostuu kiinnostuneista, jotka voivat vaikuttaa projektituloksiin omilla kommentteillaan ja mielipiteillään. (Gustavsson 2007: 60-61).

Tuotteen omistaja määrittelee tuotteen vaatimukset ja niiden prioriteetit toisiinsa nähden. Vaatimukset ovat yleensä hajaantuneet eri organisaation osasiin, jolloin haasteena on löytää yksi yksittäinen henkilö tähän rooliin. Tuotteen omistajan tulee pystyä puhumaan kehitystiimin kanssa samoilla termeillä ja hänen tulee pystyä vastaamaan tiimin kysymyksiin liittyen vaatimuksiin ja niiden prioriteetteihin. (Gustavsson 2007: 62-64).

Asiakas-rooli voi olla asiakas tai sisäinen projektin tilaaja, joka pystyy määrittelemään käytössä olevan budjettitason. (Gustavsson 2007: 65-66).

Projektinjohtajaa kutsutaan Scrum mestariksi erotuksena perinteiseen projektipäällikköön, sillä rooleissa on merkittäviä eroavaisuuksia. Scrum mestari on

ensikädessä valmentaja eikä johtaja. Hän tukee ja auttaa tiimiä tulemaan niin autonomiseksi ja tehokkaaksi kuin mahdollista. Scrum mestari ei tee päätöksiä tiimin puolesta, vaan hän on esteiden poistaja ja tiimin menestyksen mahdollistaja. Konkreettisia tehtäviä on epäselvien vaatimusten selvittäminen, resurssien hankkija, palaverin järjestäjä (päivittäiset-, sprintti- sekä aloitus- ja lopetuspalaverit), kommunikointi projektin sisä- ja ulkopuolelle sekä pysäyttää mahdolliset vaikeudet mahdollisimman nopeasti. (Gustavsson 2007: 67-71).

Scrum-tiimillä on rugby-joukkueen ominaisuudet: itseohjautuva, selkeä tavoite ja kollektiivinen vastuu. Ryhmäkoko ei saa olla isompi kuin, että jokainen voi nähdä jokaisen silmät ja että voi keskustella jokaisen yksittäisen tiimiläisen kanssa. Tiimin pitää muodostua jäsenistä, joilla on tarvittavat taidot ongelmien ratkaisemiseen. Kaikilla ryhmän jäsenillä pitää olla samat perustaidot ja sen lisäksi joillakin voi olla erityistaitoja. Tiimi on itse vastuussa töiden jakamisesta keskenään ja Scrum mestari on ryhmän auttaja. Laadun takaamiseksi ohjelmistotiimissä pitää olla myös testaaja. (Gustavsson 2007: 75-82).

Schwaber (1996) määrittelee, että yhtenä kehitystiimin jäsenenä on laatuvaastaava. Näin hän pystyy huolehtimaan laadusta koko projektin ajan, ja testien suunnittelu ja tavoitteet pysyvät selkeinä kehitystiimille. Nopea palaute on edellytys hyvälle laadulle, joten testit pitää automatisoida.

2.2.4 Scrum toimintatavat ja aktiviteetit

Schwaber ja Beedle (2002) listaa Scrum-toimintatavat: tuotteen työlista, työmääräarviot, sprintti, sprintin tehtäväluettelo, päiväpalaveri, sprinttikatselmus ja sprintin jälkikäteistarkastelu.

- **Tuotteen työlista** muodostuu tuotteen vaatimuksista tietyllä ajanhetkellä ja vaatimukset ovat työlistalla prioriteettijärjestyksessä.
- **Työmääräarviot**, joita tarkennetaan koko ajan, tekevät tuotteen omistaja yhdessä kehittäjien kanssa
- **Sprint** on iteraatiokierros, jonka aikana tiimi kehittää tuotteesta uuden, toimivan version, joka on halutessa valmis asiakkaalle toimitettavaksi.
- **Sprintin tehtäväluettelo** sisältää iteraatiokierroksen tehtävälistan, jonka kehitystiimi on itse valinnut suunnittelupalaverissa.
- **Päiväpalaverissa** seurataan iteraatiokierroksen tehtävien edistymistä. Jokainen kehittäjä kertoo kolme asiaa: mitä on tehnyt edellisen palaverin jälkeen, mitä aikoo tehdä ennen seuraavaa palaveria ja mahdolliset ongelmat.
- **Sprinttikatselmuksessa** sprintin viimeisenä päivänä tutkitaan iteraatiokierroksella kehitettyä tuoteversiota ja tuotteen työlista päivitetään.
- **Sprintin jälkikäteistarkastelussa** tutkitaan iteraatiokierrosta ihmisten, yhteistyön, prosessin ja työkalujen osalta sekä määritellään parannuskohteet.

2.3 Sovellukset laajoissa organisaatioissa

On olemassa vain vähän kirjallisuutta, siitä miten ketterät menetelmät tulisi skaalata laajoissa organisaatioissa. Tyypillisesti viitekehykset painottavat joko organisaation näkökulmaa tai tuote- ja ohjelmistokehitystä. Toimiva ratkaisu skaalaamiseen löytyy useiden viitekehysten, organisaation ketterän kulttuurin ja maalaisjärjen yhdistelmällä. (Lekman 2015).

Vaikka kirjallisuutta laajoihin organisaatioihin soveltamisesta on vähän, niin silti pystyy muodostamaan selkeän kuvan peruseriaatteisiin nojaten. Jos Agile ei toimi organisaation pienessä yksikössä, niin aivan varmasti se ei tule toimimaan myöskään laajassa organisaatiossa. Toisaalta jos Agile soveltuu organisaation ajatusmaailmaan ja

yksittäisiin ryhmiin, niin silloin sitä pystyy soveltamaan myös useaan ryhmään. Helppoa se ei varmasti tule olemaan, mutta ei mikään muutos ole itsessään helppoa vaan kovaa työtä.

Voimme tarkastella ketterän menetelmän soveltuvuutta pienii yksikköihin seuraavan tavoitelistan avulla (Amber 2002: 312-314).

- Agile toimintamallia ollaan soveltamassa nimenomaan ohjelmistokehitykseen.
- Agilen peruseriaatteet voidaan ja halutaan ottaa käyttöön.
- Ohjelmistokehityksessä käytetään iteratiivista ja inkrementtaalista lähestymistapaa.
- Vaatimukset ovat epävarmoja ja haihtuvia.
- Pää tavoite on ohjelmistokehitys.
- Sidosryhmä on aktiivinen ja osallistuva.
- Kehitystiimillä on valtuudet päätöksentekoon, jolloin heillä on mahdollisuus itse vaikuttaa onnistumiseensa ja epäonnistumiseensa.
- Suunnittelijat ovat vastuullisia ja motivoituneita.
- Organisaatiosta löytyy vähintään yksi, joka on erityisen innokas Agileen siirtymisestä.
- Käytössä on riittävät ja asianmukaiset resurssit.

Vastaavasti seuraava lista valottaa tilanteet, joissa Agile ei sovellu (Amber 2002: 314-315).

- Yksi tai useampi edellisestä listasta puuttuu.
- Organisaation kulttuuri on suunnattu ohjailevaan prosessiin.
- Tiimi on laaja ja se on jakaantunut

Mitä sitten tehdä, jos organisaatio ei ole ideaalinen? Ensinnäkin täytyy muistaa pitää asiat yksinkertaisena. Hyvä alku on kouluttaa tiimejä ja aloittaa keskustelu. Toiseksi täytyy muistaa, että Agileen siirtyminen ei ole helppo päätös. Siihen tarvitaan rohkeutta muuttaa ympäristöä ja totuttuja tapoja kuin myös sidosryhmien (erityisesti ylemmän johdon) tuki. (Amber 2002: 315-316).

Kun Agile toimintamallia esittelee organisaatiolle, niin vastarintaa syntyy aivan varmasti. Yleisimmät organisaation ja kulttuurin haasteet ovat skeptiset suunnittelijat, yli-innokkaat prosessipoliisit, paperigeneraattorit, keittokirjafilosofia, kyvyttömyys hyväksyä oikeaa epäonnistumisen syytä ja laaja dokumentointitarve avainhenkilöiden lähdön pelossa. (Amber 2002: 316-320).

Vastaavasti projektitiimin haasteet ovat fyysisesti jakaantunut kehitystiimi, toimitus toiselle tiimille ja kiinteä hintainen sopimus. (Amber 2002: 320-321).

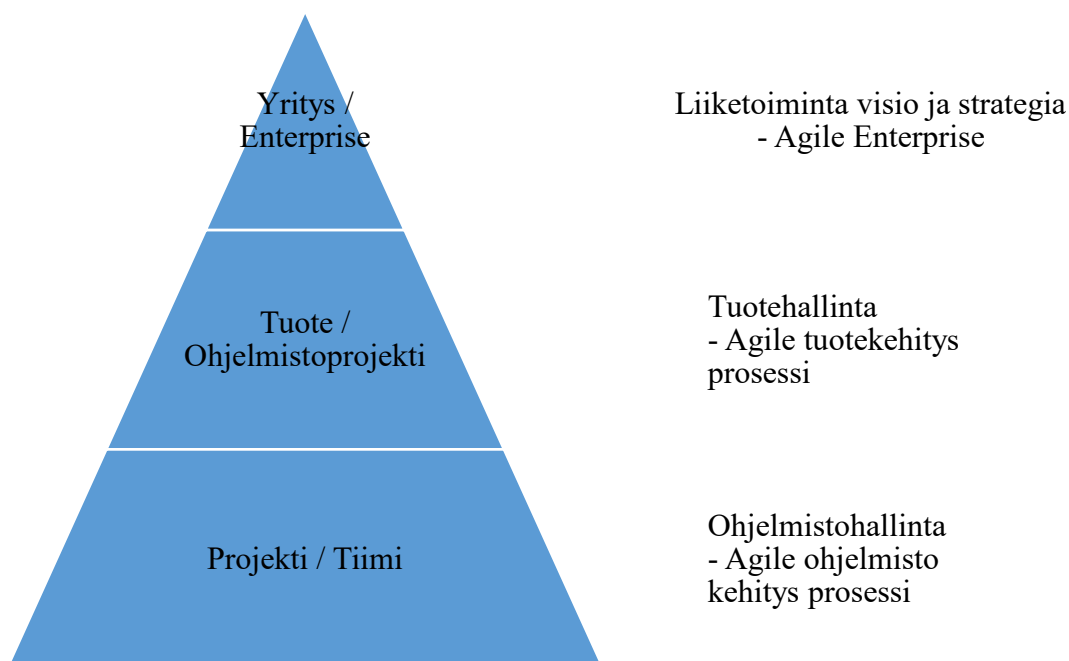
Jos Agile toimintamalliin siirtymisen haasteita on paljon, niin silloin pitää valita jokin seuraavista vaihtoehdoista: siirtyä osittain Agileen, luovuttaa Agilen suhteen tai vaihtaa työpaikkaa. (Amber 2002: 324).

Faktalista, jos organisaatio on sitoutunut Agileen (Amber 2002: 327-328):

- Asiakkaat ovat aktiivisesti osallistumassa vaatimuksiin.
- Vaatimusten muuttaminen on sallittua.
- Vaatimukset ovat priorisoitu ja korkeimman prioriteetin vaatimukset tehdään ensin.
- Ohjelmistokehitys on iteratiivista ja inkrementaalista.
- Pää tavoite on ohjelmistokehitys eikä dokumentointi.
- Tiimin jokaisen jäsenen panos on tärkeä.
- Asiat pyritään aktiivisesti pitämään niin yksinkertaisina kuin mahdollista.
- Suurin osa kehitysprosessista hylätään.

- Asiakasrajapinta tekee liiketoimintapäätökset ja suunnittelijat tekevät tekniset päätökset.
- Mallin sisältö on merkittävästi tärkeämpi kuin sisällön esitysmuoto.
- Testaaminen on kriittinen mallinnuksen jatkon kannalta.

Laani (2012: 98-99) todennut, että ketterän menetelmän vaikutus ulettuu koko yrityksen organisaatioon. Kuvassa 7 on henkilömääräisesti isoin osa-alue eli paikallinen projektitiimi kuvataan alimmaisena ja pienimääräisin yrityksen johto kolmion kärjessä. Jos ketterää menetelmää käytetään vain kehitystiimissä, niin ongelmaksi muodostui tuoteomistajan tuottama priorisoitu vaatimuslista sekä perinteiset toimitusvaatimukset aikatauluineen. Ketterän menetelmän laajentaminen koko ohjelmistotuotantoon ratkaisi useita ongelmia, mutta toi mukanaan uusia haasteita kuten perinteiseen veriputousmalliseen tavoiteasetantaan ja aikatauluun sitoutumiseen. Kun ketterä menetelmä laajennettiin koskemaan myös yrityksen johtoa, niin päästiin vesiputousmallin kahleista ja liiketoiminta alkoi ajatella uudella tavalla. (Laani 2012:98-106).



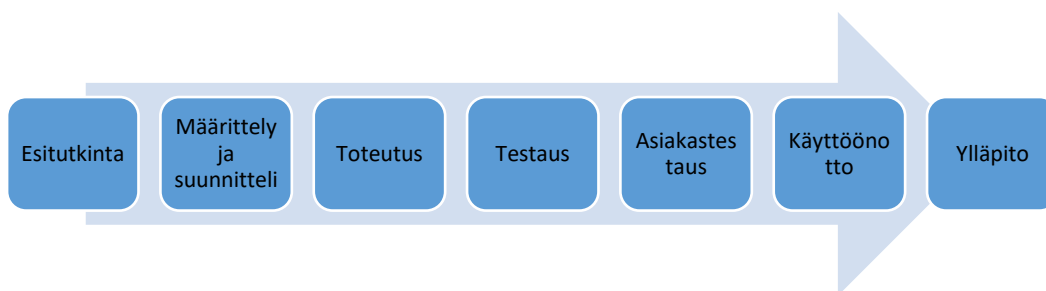
Kuva 7. Tuotekehitysorganisaation eri ketterän menetelmän sovellustasot.

3 Lähtötilanne

Amerikkalaisen keskisuuren IT-yrityksen Suomen yksikön projektit ovat suoritettu perinteisellä vesiputousmallilla. Sen lisäksi yksikön 30 henkilön osastolla on kokemusta iteratiivisesta mallista, kun he olivat samassa projektissa Amerikan yksikön kanssa. Tyypillisesti yrityksen projektien projektihenkilöstö on jakaantunut usealle eri paikkakunnalle ja maahan. Projektikieli on englanti, joka on vain harvan äidinkieli. Suomen, Amerikan ja Intian väliset kulttuuri- ja aikaerot ovat iso haaste, mutta oikein käytettynä myös valtava mahdollisuus käyttää eri kulttuurien vahvuuksia ja vuorokauden jokaista tuntia projektin eduksi.

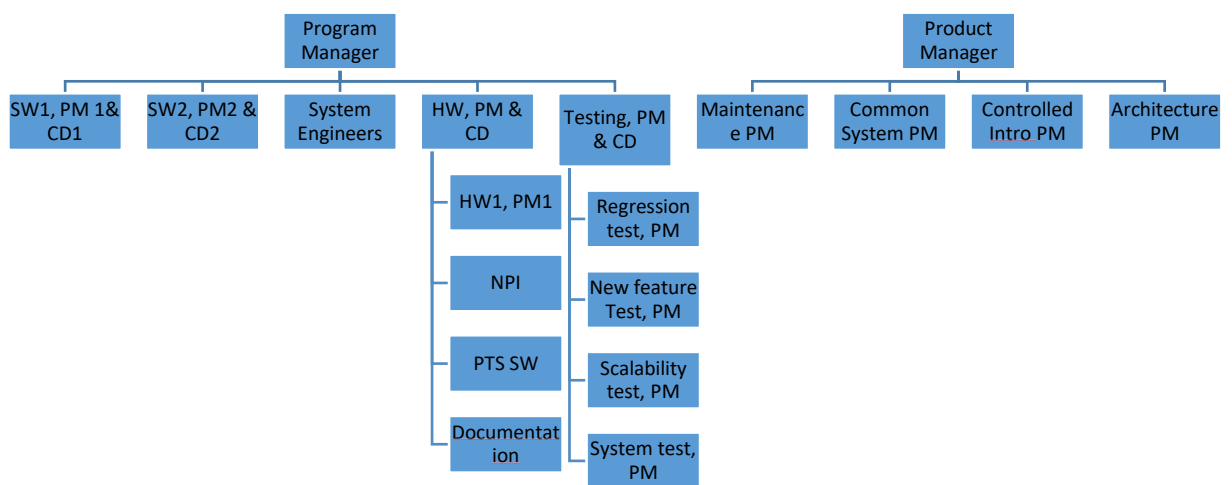
3.1 Vesiputousmalli

500 henkilön tuotekehitysorganisaatiossa oli käytössä perinteinen vesiputousmalli, joka etenee suoraviivaisesti vaihe vaiheelta kuten kuvassa 8 on esitetty. Alussa tehtiin projektisuunnitelma aikatauluineen, henkilöstöineen, tuotteineen ja budjetineen. Kuten varmaan kaikissa tuotekehitysprojektissa suunnitelmissa oli kaksi isoa ongelmaa: aikataulu ja tuotteet. Projekti kesti liian kauan ja asiakkaat halusivat enemmän tuotteita. Projektisuunnitelmaa optimoitiin positiivisella ajatuksella, kunnes siitä jotenkin päästiin yksimielisyyteen. Sitten alkoi projekti.



Kuva 8. Tuotekehitysprojektin vaiheet.

Tuotepäällikkö ja projektipäällikkö lähtivät kasaamaan projektia yhteistyössä molemmat omasta näkökulmasta: tuotepäällikkö tuotteeseen tarvittavia ominaisuuksia ja projektipäällikkö organisaation tutkimisella. Projektioorganisaatioon (kuva 9) saatiin ensin paikalle tuotepäällikkötiimi, mutta muun organisaation saatavuus olikin ongelma, sillä he olivat vielä kiinni myöhästyneissä projekteissa.



Kuva 9. Tyypillinen projektioorganisaatio.

3.1.1 Käytössä ollut tuotekehitysprosessi

Yrityksellä on globaali prosessi ja Suomen yksiköllä on sen lisäksi tarkempi prosessi ja nämä molemmat prosessit noudattavat laatu järjestelmää. Kuten taulukossa 2 on esitetty globaali prosessi sisältää kuusi vaihetta (MPR, Management Phase Review), joista jokainen vaihe päättyy globaalin johtoryhmän katelmointiin ja päätökseen, että jatketaanko vai lopetetaanko projekti.

Taulukko 2. Yrityksen globaalin prosessin vaiheet.

MPR	Vaiheen nimi	Määritelmä
0	Esikatselmointi	Projektin liikevaihtosuunnitelma ja projektin kustannusrakenne
1	Suunnittelu	Tuotteiden määrittelyvaihe alkaa
2	Toteutus	Projektin toteutus alkaa ja tehdään lupaus pilottiasiakastoimituspäivästä
3	Testaus	Testaus alkaa sisältäen integrointi- ja järjestelmätestauksen
4	FCS	Toimitus pilottiasiakkaiden tuotantoverkkoon
5	GA	Tuote on yleisesti saatavilla kaikille asiakkaille

Suomen yksiköllä on tarkempi prosessi, joka kuvaa eri vaiheet (QR, Quality Review) yksityiskohtaisella tasolla. Sen lisäksi jokaisella osa-alueella on oma prosessinsa kuten järjestelmäsuunnittelu, ohjelmistokehitys, laitteistokehitys, dokumentointi, integrointitestaus, järjestelmätestaus, projektinhallinta, asiakaspilotointi ja tuotanto. Nämä kaikki prosessit ovat keskenään yhteensopivia ja ajan tasalla olevia, sillä projektin aikataulussa pysymistä on yritetty korjata kaikilla tavoin.

Suomen prosessin QR-vaiheet linkittyvät saumattomasti globaaliin prosessiin MPR-vaiheisiin (kuva 10). Tämä ei ole sinänsä yllätys, sillä Suomessa oli kehitetty uusi prosessi useita vuosia aiemmin ja sitä oli ylläpidetty ja päivitetty vuosittain. Toisaalta globaali prosessi kehitettiin kaksi vuotta aiemmin ja sen pohjaksi oli yllätykseksi otettu Suomen prosessi. Tämä oli tietysti myös helpotus, sillä meidän ei tarvinnut muuttaa mitään toimintaa sulautuaksemme uuteen globaaliin prosessiin. Samalla tuli itsevarmuutta, että me osaamme tehdä jotain oikein, sillä jos meidän prosessi kelpasi malliprosessiksi. Suomen QR-vaiheiden katselmointi ja jatkamispäätös tehtiin Suomen johtoryhmässä, mutta kun MPR- ja QR-vaiheet linkittyvät niin saumattomasti yhteen,

QR-vaiheet päätöspalaverit korvattiin MPR-vaiheen päätöspalaverilla. Huomioitavaa on, että Suomen kaksi alkuperäistä vaihetta, QR7 ja QR5, olivat jo jääneet tarpeettomina pois.



Kuva 10. Suomen yksikön prosessi sulautettuna globaaliin prosessiin.

Suomen prosessi perustui takaisinkytkentää eli kun aiemmissa vaiheissa ollaan tehty suunnitelma, niin myöhemmässä vaiheessa testataan tätä suunnitelmaa vasten. Arkkitehtuurisuunnitelma ja järjestelmäsuunnitelma tehdään QR8 vaiheessa. QR6 vaiheessa aloitettiin moduulitestisuunnitelman tekeminen sekä lähdekielisen ohjelman toteutus hyväksytyä arkkitehtisuunnitelmaa vasten. QR4 vaiheessa lähdekielinen ohjelmisto oli valmista ja aloitettiin moduulitestaus sekä integrointitestisuunnitelman teko. QR3 vaiheessa moduulitestaus oli hyväksytysti läpäisty ja integrointitestaus alkoi. QR2 vaiheessa integrointitestaus oli onnistuneesti ohi ja alkoi järjestelmättestaus.

3.1.2 Haasteet ja ongelmat todettiin testausvaiheessa

Prosessit ovat hyvin mietittyjä ja niitä noudatettiin hyvin, joten miten ihmeessä projektit voivat myöhästyä toistuvasti niin paljon? Esikatselmointi (MPR0) ja suunnitteluvaihe (MPR1) onnistuivat yleensä hyvin, kun systeemisuunnittelijat kirjoittivat dokumentit ylätasoa suunnitelmista. Suunnittelijat olivat tällöin tyypillisesti kiireisiä edellisen projektin loppuun saattamisessa ja kriittisten vikojen korjaamisessa, joten syvällisiä kommentteja perusteluineen ei ehditty tehdä.

Toteutusvaihe (MPR2) alkoi yleensä aavistuksen myöhässä, mutta suurin ongelma oli se, että taitavimmat suunnittelijat olivat vielä kiinni edellisessä projektissa. Tilanne paperilla näytti ehkä hyvältä, mutta tosiasia oli se, että edellisen projektin ongelmia ei pysty ennustamaan tarkasti, mutta se oli varmaa, että apuun kaivattiin juuri niitä taitavampia henkilöitä, jotka olivat avainasemassa uuden projektin pääsuunnittelijoina. Valintatilanteessa olemassa oleva projekti, jolla oli kiire asiakkaalle liikevaihdon toivossa, sai aina korkeamman prioriteetin. Näin aloitteleva projekti ajelehtii epämääräisenä luottaen sitoutuneiden ja osaavien pääsuunnittelijoiden venymiskykyyn.

Ongelmat konkretisoituivat, kun integraatiotestaus pääsi eroon edellisestä projektista ja aloitti uuden tuotekehitysprojektin testaamisen. Vanhat testit eivät enää menneetkään läpi, sillä integraatiotestaus on ollut täysin edellisen projektin käytössä ja uutta lähdekielistä ohjelmaa on tuotettu valtava määrä ilman pääsuunnittelijoiden täyttä ohjausta. Yleensä päätettiin, että ohjelmoidaan nyt tuotteet valmiiksi ja korjataan viat sitten myöhemmin juuri ennen tuotteen toimittamista asiakkaalle. Tämä kostautui. Pahimmissa tapauksissa vikoja ei enää pystynyt korjaamaan ilman arkkitehtuurista remonttia, joten edessä oli vaikea päätös: otetaanko vielä lisää aikaa, poistetaanko uusi tuote vai poistetaanko vanha tuote. Tämä tapahtui aina loppumetreillä, jolloin aikaa oli jo käytetty paljon, sillä ensimmäinen vaihtoehto oli jo valittu useaan kertaan. Lopulta jouduttiin tekemään vielä yksi vaikea päätös eli poistamaan jokin vanha, olemassa oleva tuote tästä uudesta ohjelmistosta.

Prosessi oli siis kunnossa ja sitä noudatettiin, mutta vain päästäkseen läpi katselmoinnista seuraavaan vaiheeseen. Katsantokanta oli, että onko asioita tehty oikein ja riittävästi sen sijaan, että paljonko meillä on piilotyötä jäljellä. Esimerkiksi toteutusvaiheen loppukatselmoinnissa todettiin, että uusien tuotteiden ohjelmisto oli kirjoitettu valmiiksi, joten voimme edetä seuraavaan vaiheeseen. Kun epäröimme, että kirjoitettu ohjelmisto voi olla täysi pommi eikä toimi ollenkaan halutulla tavalla, niin totesimme, että testaushan on nimenomaan seuraavassa vaiheessa eli hyväksytään

äkkiä tämä toteutusvaihe tehdyksi, että päästään testausvaiheeseen selvittämään tätä mysteeriä. Testausvaiheessa sitten todettiin, että ohjelmisto on mennyt täysin rikki, joten toteutusvaihe alkoi käytännössä alusta. Emme kuitenkaan palanneet prosessissa toteutusvaiheeseen, sillä tilanne paljastui vähitellen testauksen edetessä. Lopulta totuus oli näkyvässä ja asiakkaalle piti lähteä kertomaan, että hyvin edenneestä projektista huolimatta tulemme jälleen kerran myöhästymään paljon.

Asiakkaat olivat todella kärsivällisiä, sillä tuotteet olivat hyviä ja odottamisen arvoisia. Suurin syy oli kuitenkin rautainen ammattitaito niin asiakasrajapinnassa kuin tuotekehityksessä sekä yhteen hiileen puhaltaminen. Ilman sitä korttitalo olisi hajonnut omaan mahdottomuuteensa.

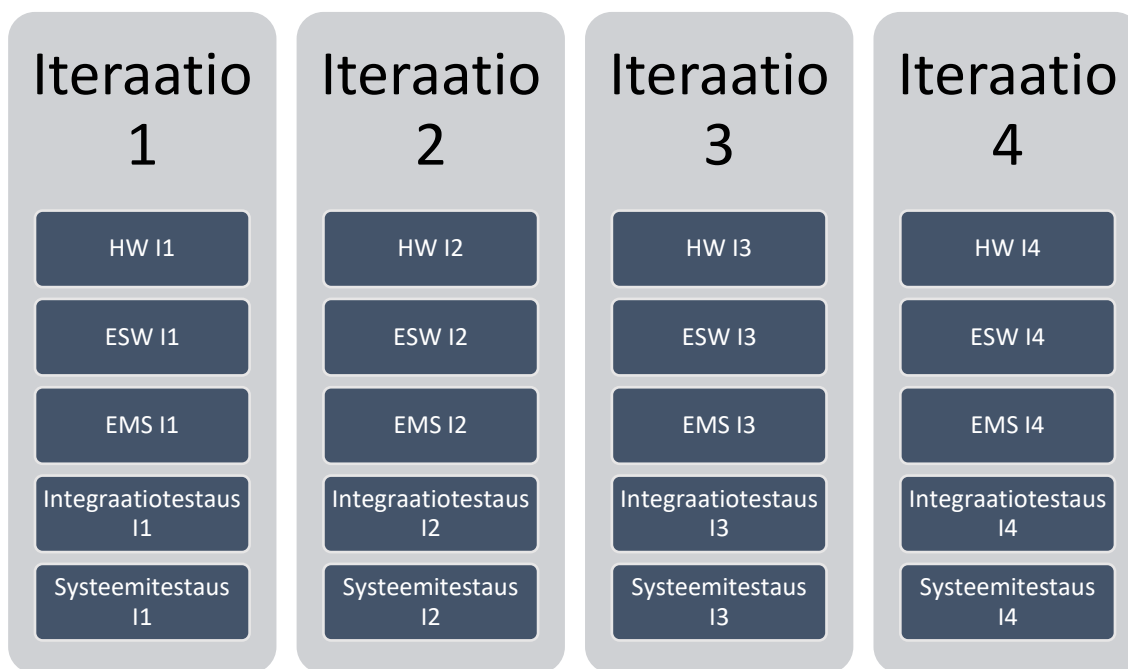
3.2 Iteratiivinen malli

Rational Rose iteratiivinen ohjelmistokehityksen prosessi (RUP, Rational Unified Process) on kehitetty IBM:llä 2003, mutta se perustuu iteratiiviseen elinkaarimalliin 1960-luvulta. Iteratiivisessa elinkaarimallissa tehdään useita pieniä iteraatiokierroksia, joissa jokainen kierros sisältää samat vaiheet kuin vesiputousmallissa eli esitutkinnasta ylläpitoon. Jokaisella iteraatiokierroksella syntyy uusi versio sovelluksesta. Kun kaikki määritellyt iteraatiokierrokset on tehty, niin tuote julkaistaan asiakkaalle. (Wikipedia 2024, Rational Unified Process)

Yrityksessämme RUP oli käytössä USA:n toimipisteessä. Kun 30 henkilön Suomen tuotekehitysosasto siirrettiin auttamaan USA:n tuotekehitystä, niin iteratiivinen kehitysmalli oli yksi uusi opetettava asia. Vesiputousmallin jälkeen se tuntui aggressiiviselta hosumiselta, mutta erittäin tehokkaalta ja selkeältä. Ydinasiaksi ja suureksi eduksi nousi kommunikoinnin tehokkuus. Tuote muodostui laitteistosta,

ohjelmistopakettista sekä EMS-ohjelmistosta. Tässä projektissa uusia laitteistoja tuli neljä ja jokaiseen niihin uusi, ladattava ohjelmistopaketti. EMS-ohjelmisto oli yksi ladattava ohjelmisto, joka tuki kaikkia näitä neljää uutta laitteistoa kuin myös kaikkia aiempia.

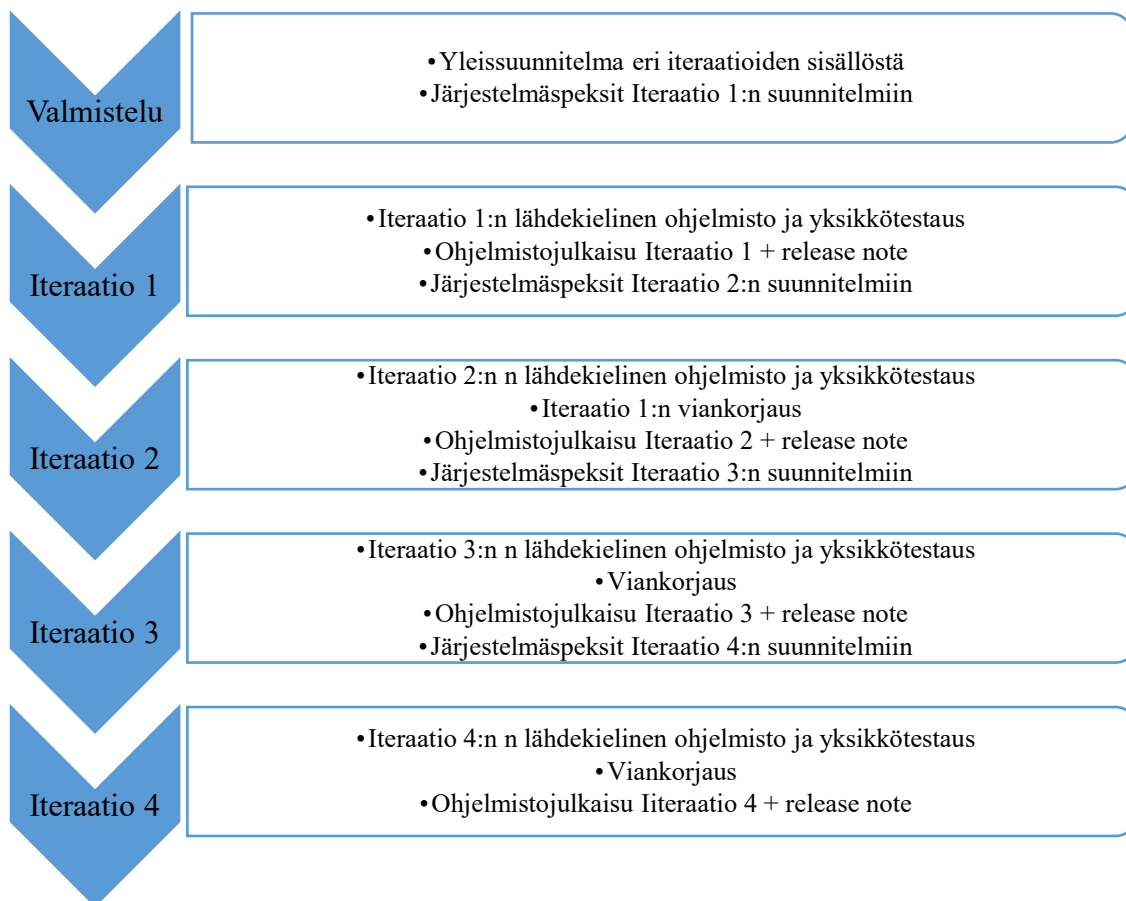
Rational Rose Iteratiivista kehitysmallia sovellettiin siten, että koko kehitysjakso jaettiin neljään noin kuusi viikkoa kestävään iteraatiojaksoon (kuva 11), jossa tapahtui tuotekehitys ja yksikkötestaus. Laitteistokehitys (HW), sulautettuohjelmistokehitys (ESW) ja ohjelmistokehitys (EMS) tekivät omat iteraatiosuunnitelmat, joista sitten koottiin yksi yhteinen. Jokaisen iteraatiovaiheen jälkeen alkoi ao. uusien osioiden integrointitesti sekä vanhojen tuotteiden systeemitestit.



Kuva 11. Rational Rose integraatiomalli yrityksessä.

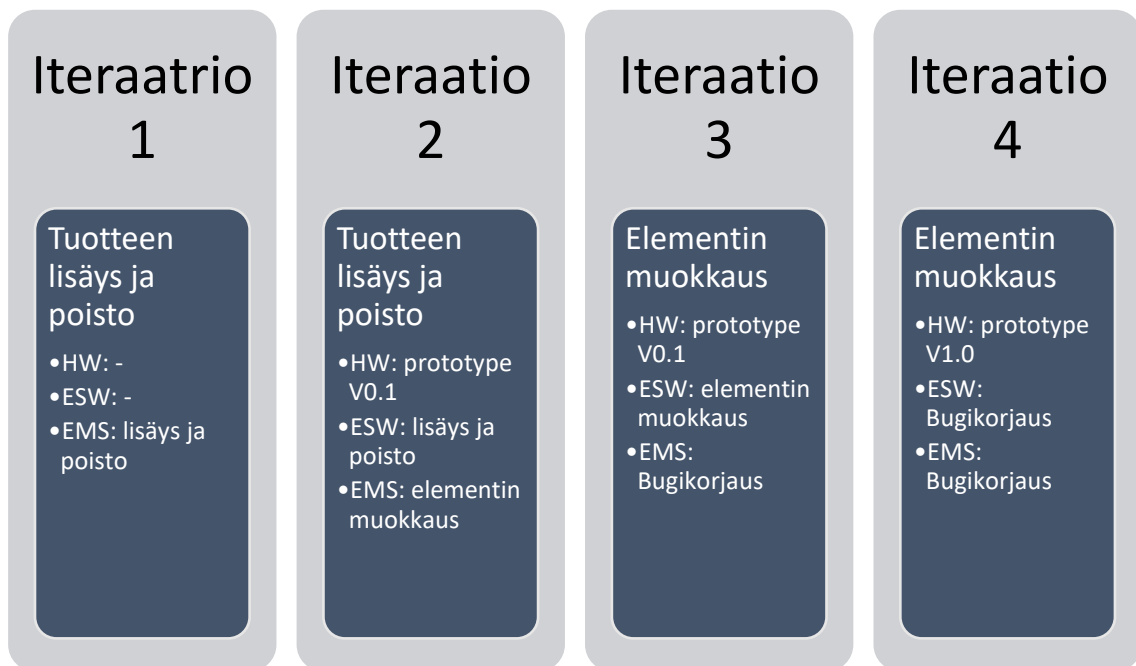
Koska tuote muodostuu eri osien kokonaisuuksista, niin iteraatiosuunnitelman tekeminen vaati paljon kommunikointia eri ryhmien välillä. Integraatioryhmä ei testaa

toimintaa ennen kuin se on toteutettu kaikissa kolmessa kehitystiimissä. Kuva 12 esittää EMS-tuotekehityksen iteraatiosuunnitelman yleiskuvaa.



Kuva 12. EMS-tuotekehityksen yleisiteraatiosuunnitelma.

Projekti sisältää useita uusia ominaisuuksia, jotka kehitetään ja testataan pieninä palasina eri iteraatioissa. Uuden ominaisuuden kehittäminen tulee aloittaa kaikkein vaikeimmasta ja epävarmimmasta osasta, jolloin yllätysten ratkaisemiseen on eniten aikaa (kuin jos haastavimmat osat tehtäisiin viimeisessä iteraatiossa). Kuva 13 esittää iteraatioiden integraatiotestaussuunnitelmaa, joka siis pohjautuu eri tuotekehityksen (HW, ESW ja EMS) iteraatiosuunnitelmiin.



Kuva 13. Integraatiotestaussuunnitelma yhdelle uudelle toiminnolle.

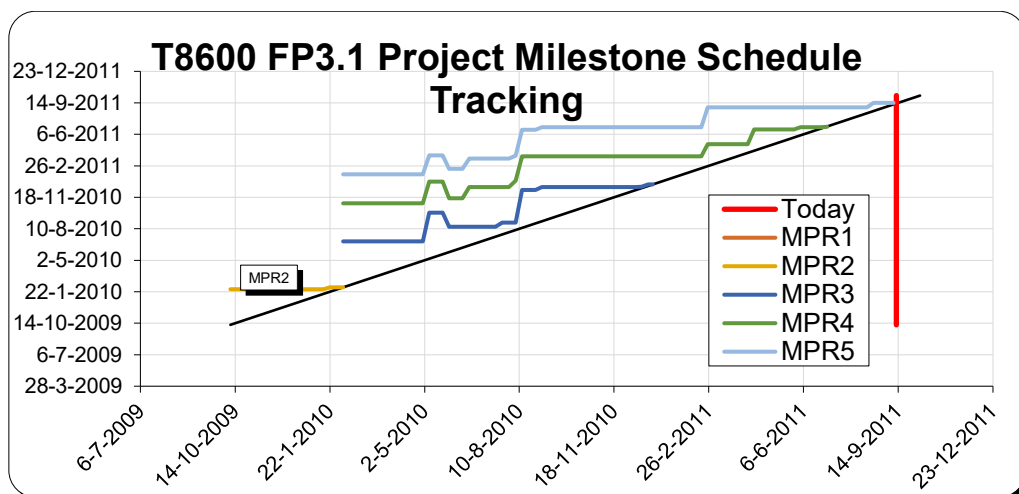
Iteratiivinen kehysmalli oli tehokas ja toimiva, mutta ennen kaikkea erittäin kommunikoiva. Kun kahden vuoden projekti loppui, niin 30 henkilön Suomen tuotekehitysosasto palasi yrityksen Suomen tuotekehityksen tehtäviin. Siellä oli edelleen käytössä vesiputousmalli eikä yrityksen Suomen tuotekehityksen johto ollut vielä kiinnostunut iteratiivisen tai scrum-kehitysmalliin siirtymisestä.

3.3 Haasteet

Pitkän aikataulun perusteella voimme todeta, että tämän yrityksen tuotekehitysprojektien suurimmat haasteet ovat aikataulussa pysyminen, sovitun sisällön tuottaminen sekä laatu. Margaret Combe on todennut (Dye 1999: 363-364), että laajan organisaation haaste on sovittaa samanaikaisesti sekä kasvu- että ylläpitostrategia. Voimme todeta, että ongelmat ovat vuosikymmenestä toiseen samat,

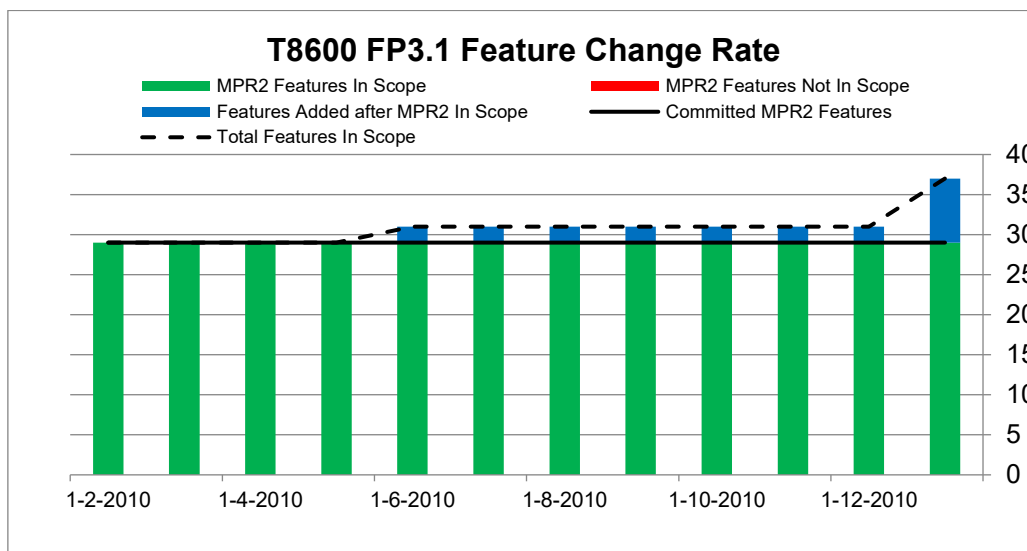
sillä kasvu on käytännössä uutta sisältöä ja ylläpito laatua. Myös ratkaisut olivat edelleen hyvin samansuuntaiset eli tee selkeä strategia, jalkauta vastuu jokaiselle saavuttamaan tuo strategia ja mittaa saavutuksia tuota strategiaa vasten (Dye 1999: 368). Tulokset eivät kuitenkaan ole niin lupaavia, sillä varsinkin laajassa projektissa myös strategia on laaja, joten sen pilkkominen ja jakaminen koko organisaatiolle siten, että strategia on edelleen muuttumaton esim. kuuden kuukauden päästä on haastavaa, ellei jopa mahdotonta.

Asiakkaan suurin pettymys on myöhästyvä aikataulu, sillä yrityksen antaman etenemissuunnitelma (= aikataulu + sisältö) perusteella asiakas suunnittelee omat päivityksensä ja omien asiakkaiden kommunikoinnin. Kyseessä saattaa olla hyvinkin laaja hanke, jossa yrityksemme on mukana vain yhtenä toimittajana. Kun muiden toimittajien laitteistoja ostetaan ja asennetaan aikataulun mukaan ja meidän laitteet ja ohjelmisto ovat myöhässä, niin asiakas joutuu hankalaan tilanteeseen. Hänen on mietittävä, että lykkääkö hän koko päivitystä tai tekeekö kaksi päivitystä (ensin muiden toimittajien ja sitten meidän myöhässä oleva). Tämä on kallista, kun aikataulu on myöhässä yli neljä kuukautta kuten kuva 14 esittää yrityksen viimeisestä vesiputousmallilla suoritetusta projektista.



Kuva 14. Tuotekehitysprojektin aikataulusuunnitelma ja sen toteutuminen.

Sovitun sisällön vaihtuminen ei käytännössä ole ollut ongelma, sillä yrityksen asiakasrajapinta tuntee hyvin asiakkaat ja neuvottelee sisällöstä ja sen pois jättämisestä. Käytännössä kaikkein tärkeimpiä asiakkaita kuunnellaan hyvin tarkasti, että he jatkossakin ostaisivat paljon ja tuottaisivat liikevaihtoa. Jos tärkeiden asiakkaan suunnitelmat muuttuvat, niin silloin sovitun uuden toiminallisuuden tärkeys laskee ja jos tuotekehityksessä on ongelmia, niin ominaisuus voidaan tiputtaa kokonaan pois. Kuva 15 esittää yrityksen viimeisen vesiputousmallilla suoritettujen projektin sisällön muutoksia. Kuvasta voi todeta, että sisältö on kasvanut projektin edetessä. Käytännössä lisätyt ominaisuudet ovat pieniä parannuksia, joiden tärkeys asiakkaalle on kasvanut sitä mukaa kuin projekti on myöhästynyt.



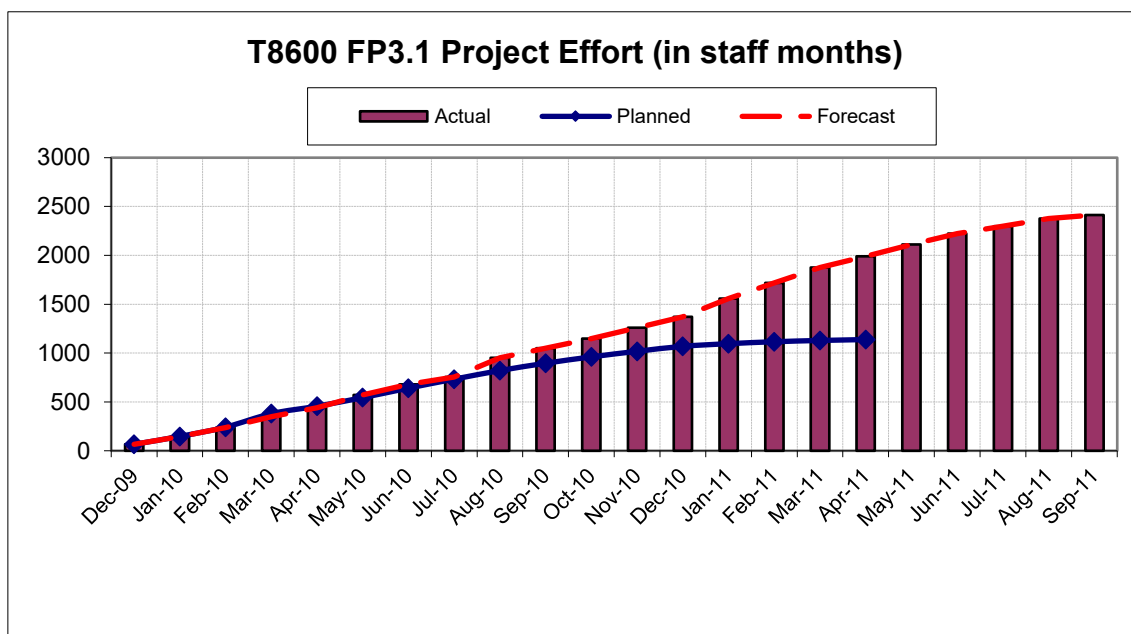
Kuva 15. Projektin tuotteiden määrän (y-akseli) suunnittelu ja toteutus kuukausittain.

Laadusta tulee ongelma, kun laitteiston myynti kasvaa ja kaikkea mahdollista asiakkaan konfiguraatiota ei pystytä ennustamaan ja testaamaan. Silloin pitää tehdä päätöksiä, että miten testataan ja toisaalta, että miten tulkitaan testitulosten vakavuutta. Päätös panostaa laatuun eikä uusiin ominaisuuksiin ei ole yksiviivainen. Uusien ominaisuuksien tekeminen laadun kustannuksella taas tulee maksamaan tulevaisuudessa paljon.

Henkilöstön usko omaan tekemiseen ja johtoryhmän päätöksen on haaste, joka syö positiivista ilmapiiriä ja innovatiivista intoa sisältäpäin. Jos tuotekehitystiimi ”tietää” jo projektia aloitettaessa, että se tulee myöhästymään eikä jaksakaan siitä enää raportoida katselmointikommenttina, niin tiimi ei selkeästi enää luota projektin johtoon. Silloin on aivan varma, että projekti tulee takuuvarmasti myöhästymään. Mutta miten saada kaiken nähneen ja kokeneen tuotekehitystiimin uskomaan, että tällä kertaa me oikeasti kuunnellaan teitä? Ja uskommeko itse, että tilanne on jotenkin merkittävästi toinen juuri tällä kertaa?

Projektin ohjausryhmän lisäksi projektia seuraa yrityksen Suomen sekä Amerikan johto. Tässä törmätään kulttuurieroihin. Siinä missä Suomessa on totuttu raportoimaan haasteet, niin Amerikassa raportoidaan saavutukset. Suomessa saavutukset vain sivutetaan nopeasti, että päästään käsittelemään oikeita ongelmia yhdessä. Amerikassa ongelmat raportoidaan nopeasti positiivisin sanakääntein korjauslistan kanssa. Tässä on paljon opittavaa puolin ja toisin, mutta tosia-asia on se, että yritys on amerikkalainen ja Amerikassa tehdään isot päätökset, joten Suomen osa-alueen on opittava heidän raportointitapansa.

Projektin henkilövuosisuunnitelma suurenee jokaisesta myöhästymisestä ja lisää projektin kustannuksia kuten kuva 16 osoittaa. Projekti on myöhässä yli neljä kuukautta, jolloin suunniteltu henkilöstötarve on yli neljä kuukautta pidempi. Sen lisäksi projekti on ylittänyt suunnitellun henkilöstöbudjetin jo kahdeksan kuukautta aiemmin, kun huomattiin, että tavoitteessa ei pysytä. Kaiken kaikkiaan henkilöstösuunnitelma yli kaksinkertaistui suunnitellun 1138 henkilöstökuukaudesta 2414 henkilöstökuukauteen. On sanomattakin selvää, että tämä on erittäin kallista yritykselle sen lisäksi, että projekti oli vielä myöhässä, jolloin takaisinmaksu tulee suunniteltua myöhemmin.



Kuva 16. Projektin kumulatiivinen henkilövuosisuunnitelma ja –toteutuma miestyökuukausina.

3.4 Tavoitteet

Tuotekehitysprojektien yksinkertainen tavoite on liikevaihdon lisääminen, joka onnistuu asiakkaiden liikevaihdon lisäämisen takaamisella. Tästä päästään tosiasiaan, että pitää tuntea asiakkaan tarpeet ja on pystyttävä vastaamaan niihin silloin kun asiakas niitä tarvitsee.

Edellisen tuotekehitysprojektin palautteesta on tehty kehitystavoitteet

- Pidä ohjelmistotuotteen laatu tasaisena ja korkeana.
- Rohkaise kehitystiimejä antamaan realistiset aika-arviot ja toimintasuunnitelma aikataulumyöhästymisien välttämiseksi.
- Tarkka tuotemääritys pitää tehdä yhdessä toteutuksen kanssa.

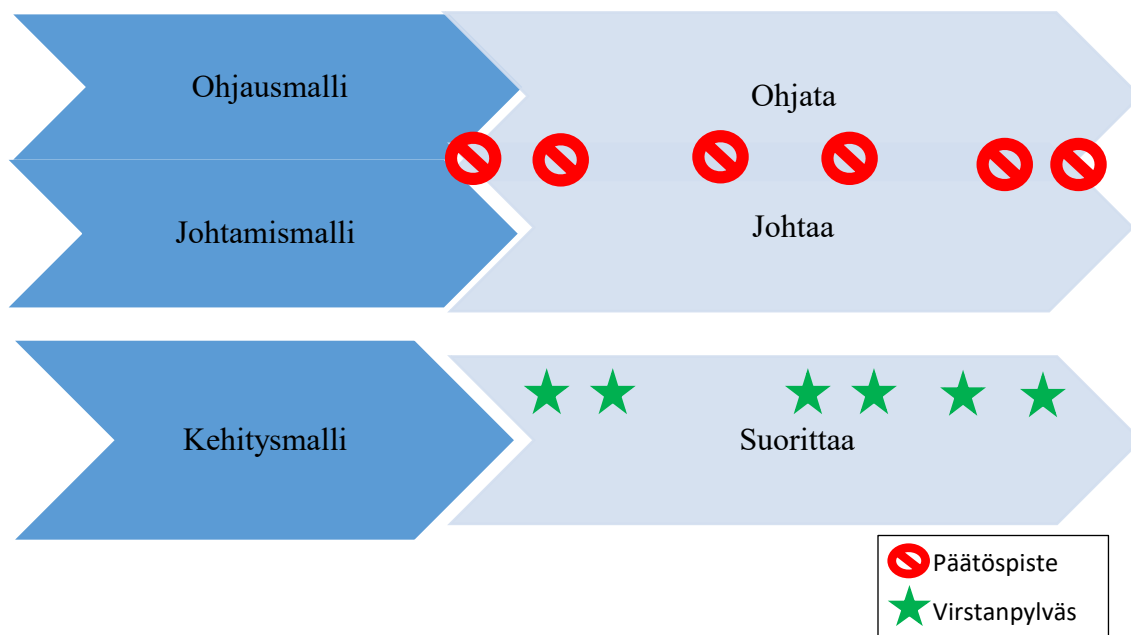
Yritys oli tilanteessa, että yhteenkään epäonnistumiseen ei ole enää varaa. Nyt on tehtävä jotain radikaalisti toisin, otettava isoja riskejä, sillä saman jatkaminen johtaisi väistämättömään tuhoon. Agile-kehityksestä on paljon positiivisia kokemuksia ja tuloksia, mutta miten se soveltuisi näin laajaan tuotekehitysorganisaatioon?

Toinen kysymys oli aika, sillä sitä ei nyt ole. Käytännössä Agileen pitäisi siirtyä välittömästi, mutta uuteen malliin siirtyminen näin laajalla joukolla tarvitsee pilotointiaikaa, kokeilua ja suunnittelua. Ja juuri aikaa ei nyt ole.

Kommunikointiin tarvittaisiin myös aikaa, sillä Amerikan ylin yritysjohto on nähnyt aikatauluista lipsumista liikaa eikä ole myötämielinen uusille, riskialttiille projektinjohtomenetelmän muuttamiselle.

Ja miten asiakkaat suhtautuvat? Kuinka he jaksavat ottaa vastaan uuden toimintatavan, kun heidän asiakkaat huutavat uusia toimintoja ja ominaisuuksia mahdollisimman nopealla aikataululla? Kun asiakkaille kerrottiin uudesta toimintamallisuunnitelmasta, niin he yllättivät täysin positiivisella vastaanotolla. He näkivät vaikutusmahdollisuutensa tuotteiden sisältöön ja julkaisuaikatauluun.

Tonnquistin (2024: 409) mukaan ohjaus ja johtamismallit määrittelevät, että miten projekteja tulee ohjata ja johtaa kuin taas kehitysmalli määrittelee, että kuinka tulokset otetaan esille (usein hyvinkin tarkalla tasolla). Yrityksen projektit on nimenomaan ohjattu ja johdettu määriteltyjen päätöspisteiden mukaan (kuva 17), jossa ollaan katselmoitu, että miten projekti on onnistuttu johtamaan. Dynaamiset kehitysmallit, kuten esimerkiksi Agile ja Scrum, ovat malleja, joissa keskitytään suorittamaan pieniä virstanpylväitä (kuva 14), jotka sisältävät mahdollisuuden toimittaa asiakkaalle tämän pienen virstanpylvään sisällön eli esimerkiksi ohjelmistotuotteen.



Kuva 17. Kehitysmallit (Tonnquist 2024: 409).

Tässä tilanteessa on paljon epävarmuutta ja riskinottoa, mutta jotain on muutettava ja nopeasti sillä muuten yritys kokee suuren alamäen. Toisaalta taas yrityksen Suomen johto- ja tuotekehityshenkilöstö ovat pitkään yrityksessä olleita huippuasiantuntijoita, jotka ovat valmiita venymään mihin vain, että tämä huippuosaaminen saadaan asiakkaiden käyttöön ja omaksi liikevaihdoksi.

Päätös oli tehty: siirrymme Scrum-projektinvetomalliin nopeasti soveltaen. Sitten voimme ainakin sanoa, että kaikkemme yritimme viimeiseen asti! Yritetään siis toimia 250 tuotekehityshenkilön kanssa siten, että asiakas saa mahdollisimman oikean tuotteen mahdollisimman oikeaan aikaan!

4 Työn rakenne

Tässä kappaleessa kuvataan tämän tutkielman rakenne. Tutkielman tarkoitus ja tavoitteet sekä tutkimuskysymykset ovat kuvattuna jo tutkielman kappaleessa 1.1. Tässä jatketaan kuvaamalla tutkimusainestoa eli kohdeorganisaatio. Toiseksi esitellään tutkimusstrategia eli case-tutkimus. Lopuksi tarkastellaan tutkielman toteutusta ja analysointia.

4.1 Tutkimusaineisto

Tässä tutkielmassa havainnoin kansainvälisen keskisuuren IT-yrityksen laajaa tuotekehitysprojektia. Yritys on amerikkalainen ja havainnointi tapahtuu Suomesta. Projektin kolmas kansallisuus on intialainen yritys, jota käytetään alihankintana. Tässä tutkielmassa kaikki tuotekehitysprojektit johdetaan ensimmäistä kertaa saman sateenvarjoprojektin alla. Henkilömäärä on 250 ja tutkielman tekijä toimi projektipäällikkönä.

Tutkielmassa käytetty aineisto on kerätty aikavälillä 2011 – 2013, jolloin alussa vielä käytettiin vesiputousmallia ja 2013 koko organisaatio siirtyi ketterään menetelmään. Työn pääpaino on tarkastella tilannetta teoreettisen viitekehyksen läpi.

Seuraavaksi esittelen kolme tyypillistä palautepuheenvuoroa 2011 projektipalautteesta.

”Teillä on loistava tuote ja innovatiivinen, hyödyllinen sovellus, mutta aikataulun siirtyminen ja sisällön muuttuminen kerta toisensa jälkeen syövät meidän

asiakasuskottavuuttamme. Haluamme, että annatte sellaisen aikataulun, joka oikeasti pitää.” (Asiakas).

”Asiakas on tyytyväinen tuotteeseen, mutta myöhästymisen takia myynti ja sitä myötä rahavirta ovat auttamattomasti myöhässä.!” (Yrityksen johto).

”Aikataulussa pysymisen on tärkeää ja alussa uskonkin siihen. Matkan varrella projektiin kuitenkin lisätään asiakkaalle tärkeitä sovelluksia, vaikka aikatauluongelmia on jo nykyisenkin sisällön kanssa.” (Projektipäällikkö).

4.2 Tutkimusstrategia eli case-tutkimus

Tässä tutkielmassa käytetään Case-tutkimusta, joka on havainnoivaa ja sopii hyvin ongelmalähtöiseen tutkielmaan kuten uuden kehitysprosessin käyttöönotto organisaatiossa. Tapaustutkimuksessa on tärkeä tapauksen määrittely, analysointi sekä ratkaisu. (Wikipedia 2024, Tapaustutkimus).

Case-tutkimuksessa tutkitaan ja analysoidaan tapauksen taustalla olevia tekijöitä, syy-seuraussuhteita sekä ympäristöä.

”Laadullisiksi nimitetyt tutkimukset rakentuvat Töttöä (2004, 9-20) mukaillen 1) aiemmista, tutkittavasta aiheesta tehdyistä tutkimuksista ja muotoilluista teorioista, 2) empiirisistä aineistoista (suurimmaksi osaksi tekstimuotoisia tai sellaiseksi muutettuja aineistoja) sekä 3) tutkijan omasta ajattelusta ja päättelystä.” (Saaranen-Kauppinen & Puusniekka 2009: 9).

Yiniä (2009) mukailten Case-tutkimuksen pääpiirteet ovat

1. Tarkastellaan tutkittavaa aihetta syvällisesti.
2. Huomioidaan ympäristö, historia ja muut vaikuttavat alueet.
3. Hyödynnetään monipuolisesti kirjallisuutta, haastatteluja ja havaintoja.
4. Käytetään yleisesti tiedossa olevaa teoreettista mallia, jota testataan tutkielman aikana.
5. Pyritään ymmärtämään laajasti ja yleistämään, jos mahdollista.

4.3 Tutkimuksen toteutus ja analysointi

Tutkielma toteutettiin analysoimalla lähtötilanne. Keskeinen lähde oli palaute, jota tuli niin asiakkailta, ylimmältä johdolta kuin projektitiimeiltä. Amerikkalainen yli johto antoi enää yhden mahdollisuuden, joten Suomalainen yksikkö oli pakko-onnistumisen edessä. Ketteristä menetelmistä oli puhutta yrityksen sisällä ja muutama ryhmä yrittänytkin käyttää niitä, joten tiedettiin niiden hyödyt ja ymmärrettiin myös, että koko organisaatio pitää siirtyä käyttämään ketteriä mentelmiä. Toisaalta myös tiedettiin, että näin ison organisaation muuttaminen yhdessä yössä vesiputousmallista ketteräksi menetelmäksi on suuri riski -ellei jopa tuhoon tuomittu yritys. Näin kuitenkin päätettiin tehdä.

Ostettiin muutama Agile konsulttipäivä auttamaan ylätasen organisaatiokaavio tekemisessä, ja sitten vain alettiin systemaattisesti tiedottamaan sekä kuuntelemaan tuotekehitystiimien vastarintaa. Osa keskeisistä ja kokeneista pääsuunnittelijoista pitivät muutosta järjettömänä ja riskialttiina, mutta kun he ymmärsivät, että muuta vaihtoehtoa ei ole ja nyt pitää keskittyä tekemiseen eikä jo tehdyn päätöksen analysointiin, niin tiimit saatiin puhaltamaan samaan hiileen.

Keskeisessä asemassa oli kommunikaatio ja läpinäkyvyys, että kehitystiimit voivat luottaa keskijohdon osaamiseen ja voivat itse keskittyä omaan osaamisalueeseensa. Kehitystiimit kokivat projektipäällikkö luotettavaksi, rehelliseksi ja tasapuoliseksi, joten kolmen viikon välein tapahtuvat koko projektia käsittävät informaatiopalaverit olivat keskeisessä osassa niin tiedon jakamisen kuin myös opetusmielessä. Siellä mitattiin kehitystiimin itse itselleen asettamia tavoitteita heidän sprint saavuttamiin tavoitteisiin.

Ensimmäisen sprintin jälkeen vain intian tiimit olivat saavuttaneet tavoitteensa. Tämä oli äärimmäinen kolaus taivataville Suomen tiimeille ja he purnasivat, että intian tiimien tavoitteet olivat lapsellisen helppoja ja toisaalta ne olivat kysyneet heiltä apua niidenkin tekemiseen. Oppiminen tapahtui hitaasti, mutta varmasti, sillä intian tiimit olivat toimineet sinänsä Agilen mukaisesti, mutta Suomen tiimien ei pidä auttaa heitä, jos siitä ei ole etukäteen sovittu ja jos se ei ole heidän työtlistalla.

Toisaalta Suomen tiimien pitää miettiä, että pitääkö jokaisen toiminnan olla 100%:sti tehty vai voisiko 70% riittää asiakkaan tarpeisiin? Tavoitteena on miettiä, että mikä on minimi, mitä asiakas tästä ominaisuudesta tarvitsee? Toteutetaan se ja laitetaan loput tulevaisuuden työlistalle ja otetaan seuraava tärkeä ominaisuus työllä minimitavoitteen. Tämä oli valtava muutos, mutta kun sitä toistettiin, niin siitä alettiin näkemään hyötyjä. Ensimmäistä kertaa pääsuunnittelijoille tuli tunne, että me tosiaankin saadaan toimivia, pieniä palasia valmiiksi.

5 Ketterän menetelmän soveltaminen laajaan projektiin

Meillä on siis 250 tuotekehitysinsinööriä kolmessa eri maassa neljällä eri paikkakunnalla. Laaja asiakaskunta, joka on innostunut tuotteesta, mutta kyllästynyt jatkuviin myöhästelyihin ja laatuongelmiin. Kilpailijat, jotka saavuttavat etulyöntiasemaamme joka kuukausi moninkertaisella tuotekehityshenkilöstöllään. Yrityksen amerikkalainen johtoryhmä, joka näkee laatuongelmat ja myöhästelyt, mutta ei välttämättä potentiaalia. Osakkeen omistajat, jotka vaativat tuloksia neljännesvuosittain.

Vesiputousmallia on kokeiltu vuosikymmenet ja se on tiensä päässä. Sovellettua Scrum-mallia kokeiltiin laajassa tuotekehitysprojektissa muutamalla ohjelmistotiimillä. Siinä todettiin, että osittain soveltaminen ei tuota tarvittavaa tulosta, joten on vain uskallettava siirtää koko tuotekehitys scrum-malliin. Toisaalta taas olemme yksi osa laajaa kansainvälistä yritystä, jolla on omat prosessinsa, joista ei ole lupa joustaa. Amerikan laatutiimi oli erittäin huolestunut tulevista muutoksista. Lukemattomista yrityksistä huolimatta he eivät suostuneet ymmärtämään, että muutoksessa raportoimme oleellista tietoa paljon aiempaa enemmän, vaikka jätämmekin epäoleellisen tiedon mittaamatta. Prosessiin ei ole tulossa mitään muutosta ketterän menetelmän soveltamisen helpottamiseksi. Lopulta yksinkertaistimme kommunikointia kertomalla, että seuraamme täysin olemassa olevaa prosessia. Yrityksen ylin Amerikan johto rauhoittui, ja laatutiimi halusi sukeltaa meidän scrum-metriikkaamme.

Jos vielä vertaamme tuotekehityksen tilannetta aiempiin tavoitelistaan (kappaleessa 2.3 Sovellukset laajoissa organisaatioissa), niin huomaamme, että ketterään menetelmään siirtyminen sisältää paljon riskejä.

- *Ketterään menetelmään siirtymisessä tarvitaan rohkeutta muuttaa ympäristöä ja totuttuja tapoja kuin myös sidosryhmien (erityisesti ylemmän johdon) tuki.* Valitettavasti meillä ei ole Amerikan ylimmän johdon tukea ollenkaan.
- *Suurin osa kehitysprosessista hylätään.* Valitettavasti mitään ylätasen prosessimuutoksia ei sallittu, vaan olemassa olevaa prosessia pitää edelleenkin noudattaa kuten vesiputousmallissakin.
- *Tiimi sijaitsee samassa toimitilassa.* Valitettavasti tiimi on laaja ja jakaantunut kolmeen eri maanosaan.
- *Ohjelmistokehityksessä käytetään iteratiivista ja inkrementtaalista lähestymistapaa.* Valitettavasti vain pienimuotoinen pilotti on tehty ja suurin osa tuotekehityksestä seuraa edelleenkin vesiputousmallia.
- *Vaatimukset ovat priorisoitu ja korkeimman prioriteetin vaatimukset tehdään ensin.* Valitettavasti suurin osa vaatimuksista ovat kriittisiä ja asiakasrajapinta ei ole pystynyt laittamaan niitä tärkeysjärjestykseen.
- Näiden lisäksi ohjelmistoprojektiin kuuluu kiinteästi laitteistotiimit, jotka kehittävät fyysisesti uudet tuotteet ohjelmistotuotekehityksen rinnalla.

Kaikesta vastarinnasta ja riskeistä huolimatta päätimme siirtyä ketterään tuotekehitykseen, sillä saman toimintatavan jatkaminen olisi varman epäonnistuminen. Seuraavaksi esitellään ketterän tuotekehitysprojektin keskeisimmät tiedot.

250 henkilön projektitiimi rooleihin jaettuna (maa)

- 1 projektipäällikkö (Suomi)
- 10 aliprojektipäälliköitä (Suomi ja USA)
- 3 tuotepäällikköä (Suomi)
- 11 systeemisuunnittelijoita (Suomi ja USA)
- 125 ohjelmistosuunnittelijoita (Suomi, USA ja Intia)
- 15 laitteistosuunnittelijoita (Suomi)

- 61 uuden tuotteen testisuunnittelijoita (Suomi, USA ja Intia)
- 15 Regressiotestaus yms. testaus
- 9 sekalaista: tuotemuutokset, pilottitestausta, systeemitestausta

Ohjausryhmä

- Projektipäällikkö
- Projektihallinnan vetäjä
- Tuotekehitysjohtaja
- Tuotehallinnanjohtaja
- Tekninen johtaja

Johtoryhmä

- Ohjausryhmä
- Suomen toimitusjohtaja
- Amerikan tuotekehitysjohtaja
- Amerikan laatujohtaja

5.1 Vaatimusten priorisointi

Ensimmäinen vaihe on suunnitteluvaihe. Siinä projektipäällikkö yhdessä systeemisuunnittelijoiden ja tuotepäälliköiden kanssa hahmottaa asiakkaan tarpeita, aikatauluarviota, tuotekehitystiimin osaamista ja vapautumista edellisestä projektista. Projektihenkilöstö oli alle 20 henkilö. Koska kyseessä on hyvin luova vaihe, niin ketterän menetelmän soveltaminen ei tähän sovellu. Mutta koulutusmielessä puhuimme

kuitenkin ”scrum-kieltä”, sillä on erittäin oleellista, että tulevat tuotteenomistajat alkavat sisäistää scrum-metodologiaa ja -termejä.

Keskeiseksi aiheeksi nousi vaatimus-lista. Tajusimme matkan varrella, että meillä on monenlaista priorisoitavaa ennen kuin projekti alkaa: tuotepäälliköt priorisoivat asiakkaiden toivomia ominaisuuksia ja systeemisuunnittelijat priorisoivat tuotteen teknologiaominaisuuksia. Nämä pitää saada samalle listalle ja saman priorisoinnin piiriin, että ketterää menetelmää pystytään hyödyntämään.

Tästä syntyi kauan toivottu Excel-taulukko, johon kaikki toivotut uudet ominaisuudet olivat listattuna. Taulukkoon laitettiin paljon yksityiskohtaistakin tietoa, mutta pääasia oli, että nyt uudet ominaisuudet olivat listassa prioriteettijärjestyksessä. Listaan laitettiin myös työmääräarvioita. Mitä korkeampana ominaisuus oli listalla, niin sitä tarkempia speksejä ja arvioita siitä aloitettiin tekemään. Tämä ei aina ollut helppoa, sillä kärkituotteet olivat yleensä hyvin hankalia ja vaikeita. Mutta scrumin metodologian mukaisesti noudatimme tätä.

Vähitellen projektin ylätason suunnitelma alkaa muotoutua. Esiin nousivat ominaisuudet, jotka ovat aivan pakko saada työnalle. Oli pakko priorisoida, että tärkein ominaisuus on organisoitava ensimmäisenä ja sille on etsittävä parhaat tekijät –vaikka se tarkoittaisi, että ”pikkuhomma” sijalla viisi jää tekemättä. Tämä oli hyvin vapauttavaa, sillä ennen näitä lukuisia ”pikkuhommia” on otettu työnalle, jolloin asiakkaalle on saatu ominaisuuksia, mutta se tärkein ominaisuus jää aina aloittamatta. Nyt se otetaan työnalle ensimmäisenä! Näin yksi aiemmin esitetty riski on saatu poistettua: *Vaatimukset ovat priorisoitu ja korkeimman prioriteetin vaatimukset tehdään ensin.*

5.2 Organisaation luominen

Kolmessa maassa olevien 250 tuotekehityssuunnittelijan organisointi scrum-tiimeihin ei ollut ihan suoraviivainen asia. Periaatteeksi haluttiin, että tiimin on oltava samalla paikkakunnalla, mieluiten samassa rakennuksessa kuten aiemmin esitetty tavoite määritteli: *Tiimi sijaitsee samassa toimitilassa*. Tämä poissulki osaavien suomalaisten huippusuunnittelijoiden ja uusien kansainvälisten suunnittelijoiden samaan tiimiin laittamisen, joka olisi tiedon oppimisen lisäksi sulattanut myös kulttuurieroja. Ajatus oli siis muodostaa itsenäisiä scrum tiimejä kolmeen eri maahan.

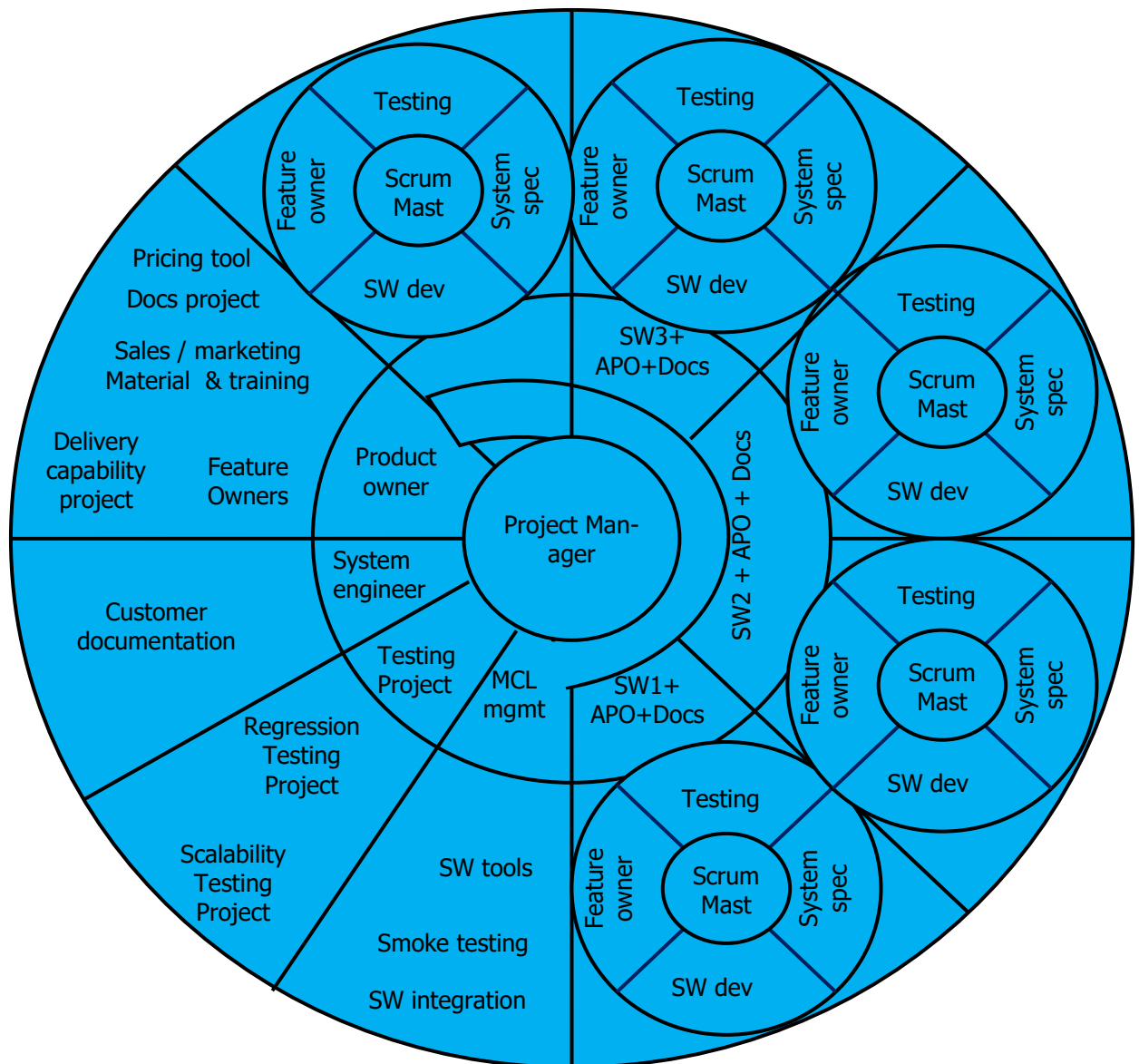
Seuraava päätös oli, että laitetaanko scrum-menetelmien mukaisesti testaajat scrum-tiimeihin. Testisuunnittelijat ovat muotoutuneet vuosien varrella huippuasiantuntijoiksi, jotka tekevät uskomattoman määrän työtä ja joustavat aina. He ovat omassa organisaatiossa, jossa he tukevat toisiaan hyvässä, positiivisessa hengessä. Toisaalta taas nämä testisuunnittelijat ovat erittäin palveluhenkisiä ja toteuttavat joustavasti systeemi – ja ohjelmistosuunnittelijoiden toiveita. Samassa tiimissä testisuunnittelijoiden taidot laajenevat entuudestaan ja myös heidän laaja systeeminäkemyksensä saataisiin ohjelmistosuunnittelijoiden tietouteen, jolloin todennäköisesti välttyttäisiin isoilta, kohtalokkailta virheiltä, jotka maksavat paljon projektin loppuvaiheessa. Kolmanneksi testisuunnittelijoita on erittäin vähän, joten miten vähästä jaetaan moneen tiimiin. Päätettiin, että laitetaan testisuunnittelijat scrum-tiimiin, mutta pidetään heidät matriisiorganisaation mukaisesti testiorganisaatiossa, jossa testipäällikkö on heidän esimiehensä ja vastaa testauksen kehittämisestä.

Kolmas keskeinen päätös oli Scrum mestareiden valinta, tiimien koko ja henkilöstö. Olemassa olevista projektipäälliköistä leivottiin Scrum mestarit. Scrum-periaatteiden mukaisesti scrum-tiimin koko oli 7 +/- 2 henkilöä. Testaajien mukaan tulo lisäsi tiimin

koon ylälukemiin, mutta jokainen tiimi käsiteltiin erikseen ja valittiin paras mahdollinen lähestymistapa. Jotkin tiimit sisälsivät 14 suunnittelijaa ja jokin tiimi vain neljä. Nämä olivat kuitenkin perusteltuja ja parempaa ratkaisua ei löydetty, joten päätimme lähteä liikkeelle 18 scrum ohjelmistotiimillä.

Lisäksi oli kaksi laitteistotiimiä, jotka kehittävät laitteiston perinteisellä vesiputousmallilla, mutta soveltuvat muiden projektitiimien uuteen malliin. Koska laitteistotiimit koostuivat osaavista, toisensa ja muun projektihenkilöstön hyvin tuntevista huippuammattilaisista ja scrum ei tosiaankaan ole tarkoitettu laitteistokehitykseen, niin uskalsimme lähteä liikkeelle.

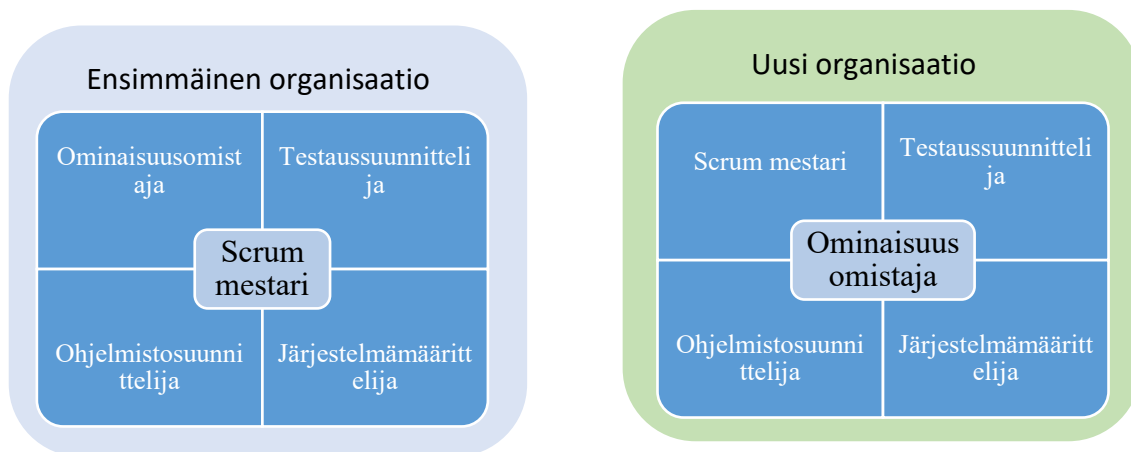
Tuotepäälliköt nimettiin aluetuoteomistajiksi (APO, area product owner) ja systeemisuunnittelijat ominaisuusomistajiksi (FO, feature owner). Vastuualueet olivat selkeät. Jokaiselle tiimille nimettiin ominaisuusomistaja, joka on viikoittain yhteydessä tiimiin sekä aluetuoteomistaja, joka on ominaisuusomistajan työpari. Integrointi- ja systeemitestaus jatkavat omassa tiimissään. Scrum mestari on raportointivastuussa projektipäällikölle. Kuva 18 esittää pelkistetysti projektin ketterää organisaatiota, sillä tiimejä on todellisuudessa 18 sekä lisäksi vielä on kaksi laitteistotiimiä ja järjestelmätestaustiimi.



Kuva 18. Ensimmäinen projektiorganisaatio.

Kun projektia oli ajettu kaksi kuukautta, niin huomasi, että emme saa scrum-menetelmästä parasta tehoa. Olemme liikaa vesiputousmallin kahleissa. Uutena ideana oli keikauttaa vastuunjako ihan toisinpäin (kuva 19). Scrum mestareiden sijaan laitetaan ominaisuusomistajat raportointivastuuseen. Scrum mestarit ovat vastuussa hallinnosta, tiimin henkilöstöstä, kokousten järjestämisestä, metriikan laatimisesta ja kaikista muustakin käytännön asioita, mutta ominaisuusomistajat ovat vastuussa edistymäkaaviosta eli siitä, että miten valmis ominaisuus oikeasti on.

Vaihto tuntuu isolta riskiltä, sillä ominaisuusomistajat eivät koe osaavansa projektipäällikön hommia, mutta he olivat valmiita yrittämään. Joillekin Scrum mestareille järjestely tuntuu syövä heidän arvostusta ja vaarana oli vetäytyminen, joten muutoksen kommunikoinnilla oli suuri merkitys. Ominaisuusomistajat ovat huippuasiantuntijoita ja nöyriä uuden tehtävän edessä, joten se edesauttoi muutoksen tekemisessä.



Kuva 19. Uusi organisaatio kahden kuukauden jälkeen.

Uusi organisaatio loksautti paikoilleen ja vastuu siirtyi aidosti ominaisuusomistajille ja scrum-tiimille, joten tämä organisaatio jatkui projektin loppuun asti.

5.3 Hallintamenetelmien luominen

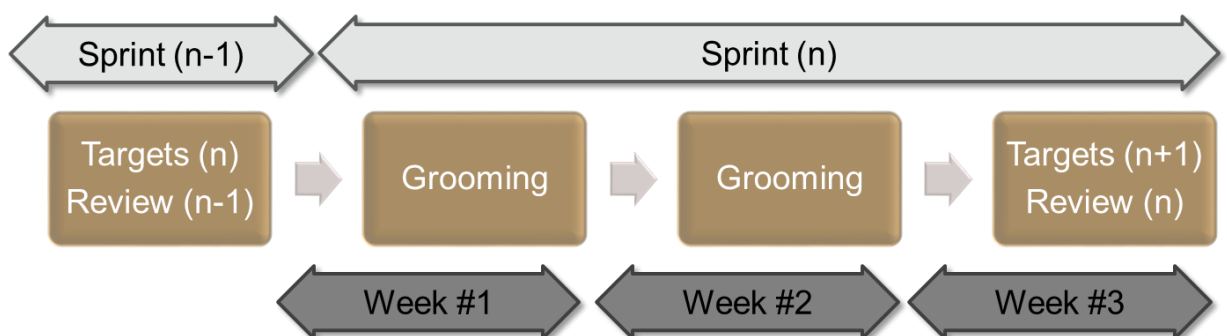
Amerikan johdon kanssa on sovittu, että kansainvälistä prosessia noudatetaan. Toisaalta taas scrum-tiimejä ei saa häiritä vesiputousmallin tyylisillä ennustuksilla ja lupauksilla. Päätin, että minä projektipäällikkönä olen tässä välissä ja tuotan kaikki tarvittavat prosessin vaatimat katselmoinnit ja dokumentit systeemisuunnittelijoiden avustuksella.

Epäilijöistä huolimatta tämä sujui oikein hyvin ja pidemmällä tähtäimellä meillä oli enemmän dataa tarjolla kuin aiemmissa projekteissa ja kansainvälisissä projekteissa.

Koska tiimit ovat uusia ja tiimien nopeudesta ei ole tietoa, niin päätimme, että jokaisen tiimin on ajettava 2-3 sprinttiä ennen kuin se antaa aikataululupauksen. Tämä ei tietystikään ole yrityksen kansainvälisen prosessin mukaista, sillä siellä aikataululupaus annetaan suunnittelun jälkeen ennen toteutuksen aloittamista. Ja näistä aikataululupauksista meillä on huonot kokemukset! Siitä huolimatta otti oman aikansa, että yrityksen USA:n johtoryhmä hyväksyi tämän uuden käytännön.

5.3.1 Palaverikäytäntö

Projektihenkilöstö yleensä kritisoi projektin palaverien määrää, ja on yleisesti tiedossa, että scrum-menetelmässä ovat päivittäiset tiimipalaverit. Haasteena on saada tiimit innostumaan määrällisesti enemmistä palavereista ja tekemään työtä sen eteen, että ne ovat tehokkaita pikapalavereita. Kuten kuva 20 esittää yksi sprintti on kolmen viikon mittainen ja sen alussa on tavoiteasetanta- ja lopussa katselmointipalaveri jokaiselle tiimille sekä tietysti tavoitteen etenemisen seurantalpalaveri.



Kuva 20. Kolmen viikon sprintin ylätasen tapahtumat aikajanalla.

Näin laajan projektin palaverimäärä on päätähuimaava. Palaverissa istuminen ei edistä asiakkaalle tehtävää tuotetta, mutta kommunikaatio pitää pelata, että jotain järkevää saadaan ulos tiukassa aikataulussa. Listasin kaikki palaverit (Taulukko 3) yhteen dokumenttiin seuraavine tietoineen: palaverin nimi, kohderyhmä, agenda, tiheys, kesto. Palaverien määrä on valtava, mutta kyseessä on 250 henkilön tuotekehitysprojekti, joten yhden henkilön palaverimäärä ja -aika ei ole niin iso.

Taulukko 3. Yhden sprintin palaverit.

Palaveri	Kesto / h	Määrä	PM	APO	FO	SM	Testaus, HW, NPI, CIP, doc, INM, Maintenance, Laatu	Kaikki
APO	1,5	3	1	x	x	x		
Sprint	1	2	x	x				
Info	0,5	1	x	x	x	x	x	x
Feature	2	1	x	x	x			
Core team	1,5	1	x		x	*	x	
Ohjausryhmä	1	3	x					
CAPEX	1	1	x			x	*	
SOS	0,25	3-15	x			x	*	
ECE#1	1	3	x	*	*	*	*	
Yht.			16	7	6	8	2	1

* osallistujia soveltuvien osin

5.3.2 Raportointi

Projektipäällikkö raportoi kerran kuukaudessa yrityksen Amerikan johtoryhmälle ja joka toinen viikko projektin ohjausryhmälle. Sen lisäksi oli Management Phase Review (MPR)

ja Management Status Review (MSR) Suomen johtoryhmälle. Asiakasrajapinnalle tein status-kalvoetin joka viikko, jota he voivat käyttää omiin esityksiinsä. Projekti-info koko projektihenkilöstölle oli kolmen viikon välein.

Projektin riskien raportointi oli erittäin keskeisessä asemassa. Opin nopeasti, että jos muuta ei ehditä katsoa, niin yrityksen Amerikan johto haluaa nähdä riskitaulukon (Taulukko 4). Se olikin oiva työkalu avun saamiseen eli jos riski oli kasvanut, niin johtajalla oli tyypillisesti kolme kysymystä: mitä olet tehnyt asian eteen, mitä aiot tehdä ja kuinka minä voin auttaa. Viimeinen kysymys oli ratkaiseva monissa tilanteissa, sillä joskus hankintaprosessi oli tyssännyt, joten protoja ei tullut, joskus muiden tuotelinjojen avunpyynnöt häiritsivät projektiani ja joskus asiakkaan lisävaatimukset tulivat kohtalokkaalla hetkellä, joten oli mainio tilaisuus antaa Amerikan johtajan tehdä päätökset, sillä hän kuitenkin näki koko tilanteen.

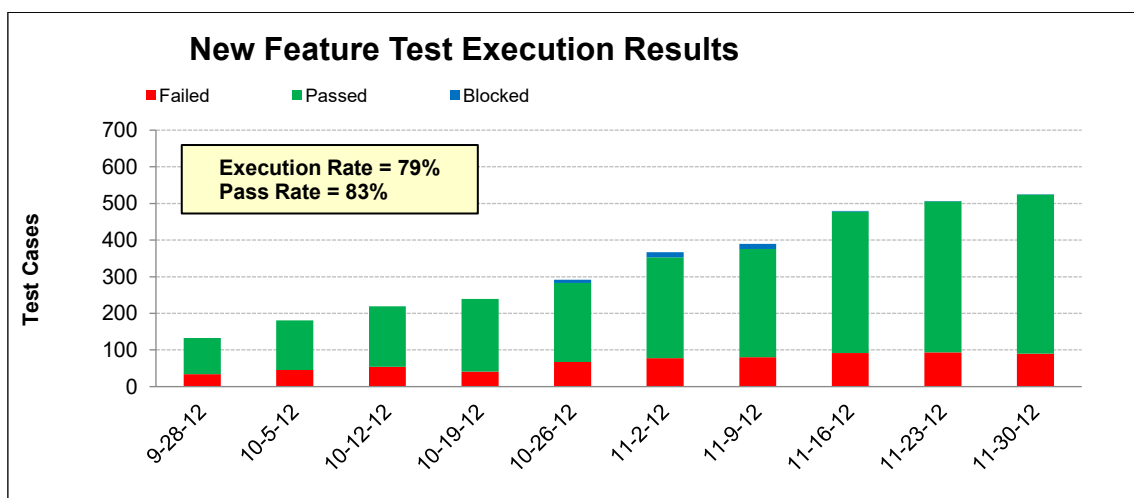
Taulukko 4. Projektin riskit.

#	Määritelmä	Mahdollinen	Vaikutus	Ratkaisu	Tilanne nyt
1	CDC2 ei tule valmiiksi ajoissa (2.5)	Korkea	Iso	Priorisoidaan ominaisuudet	Työnalla
2	ECE#1 ei tule valmiiksi 23.11 mennessä	Korkea	Iso	Korkeampi prioriteetti	Hallinnassa
3	ECE#1 laatu on huonompaa kuin yleensä (< 97 %)	Keskitaso	Iso	Enemmän testausta aiemmin	Hallinnassa

5.4 Testaus

Uuden tuotteen ominaisuustestaus tapahtuu scrum-tiimeissä, joiden lisäksi oli vesiputousmallista tutut regressio-, systeemi-, applikaatio, MDS-, SDS-, IOT- ja PLS-testausta. Nämä ovat vesiputousmallissa olleet sen jälkeen, kun kaikki ohjelmiston tuotteet ovat valmiita, jolloin testausorganisaatio on tosi lujilla ja toisaalta virheet löytyvät aivan liian myöhään. Nyt yritetään venyttää niitä koko projektin ajalle, vaikka näin ensimmäisessä scrum-projektissa testihenkilöstä pääsee tulemaan projektiin vasta puoli vuotta myöhässä koska edellinen projekti vedettiin perinteisellä menetelmällä, joten testaaaja tarvittiin erityisesti projektin lopussa.

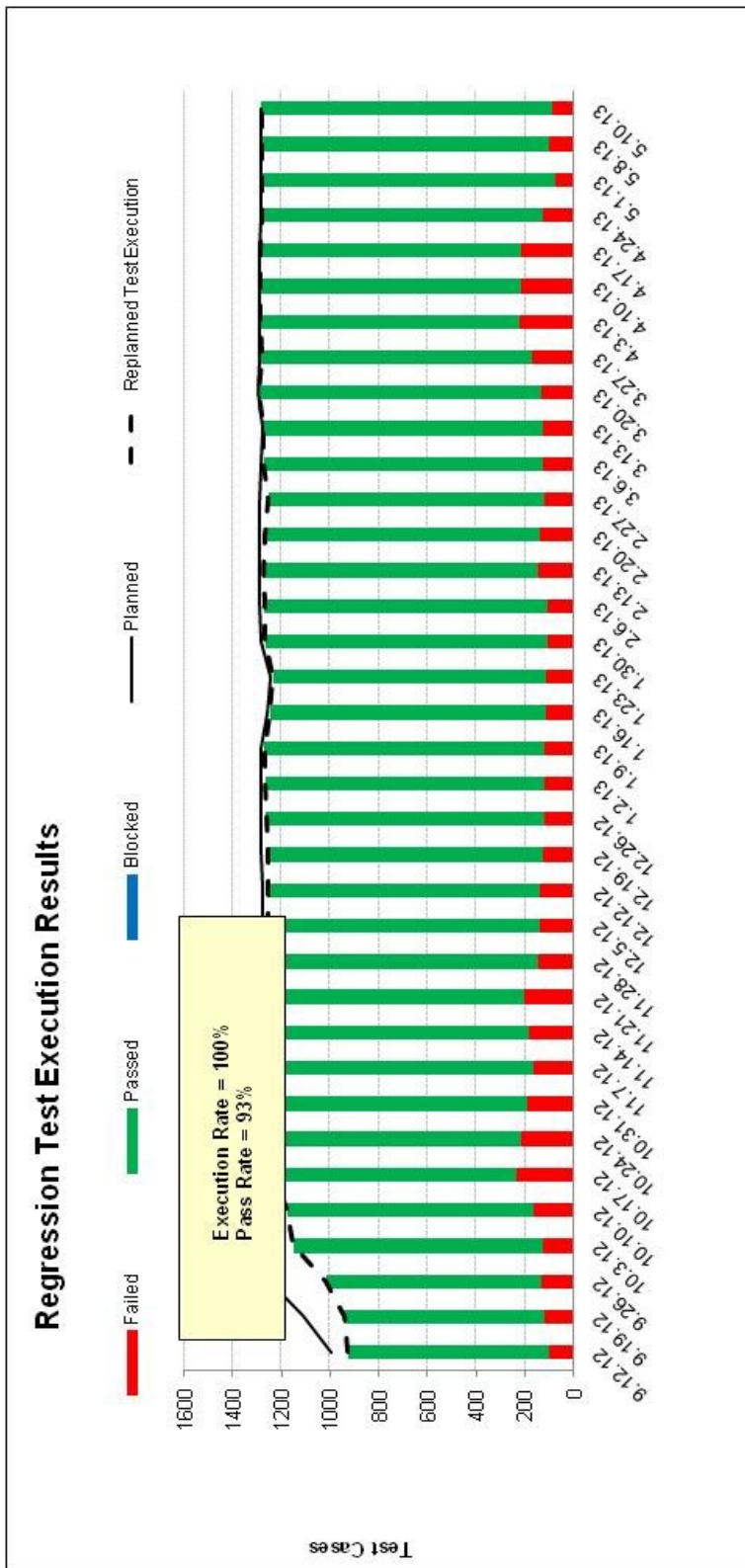
Kuva 21 esittää miten uusien piirteiden testien määrä kasvaa sekä näiden testien testitulokset. Tämä on ensimmäinen projekti, jossa uusia testejä on tehty ennen kuin ohjelmistokehitys on loppunut. Tämä on todella iso askel eteenpäin ja se näkyy myös uusien piirteiden laatutasossa. Kun testaus aloitettiin, kun ominaisuutta vielä kehitettiin, niin ohjelmistosuunnittelija sai palautetta työstään jo hyvin alkuvaiheessa ja näin ollen isoimmat sudenkuopat pystyttiin välttämään.



Kuva 21. Uuden ominaisuuden testaustapahtumien kappalemääräinen suorittaminen scrum-tiimissä kumulatiivisesti.

Uuden ominaisuuden testauksen lisäksi pääpaino on testien automatisointi, jotta ne voidaan suoraan siirtää testipatteriin. Kun testi pystyttiin automatisoimaan jo heti alkuvaiheessa, niin sen uudelleen ajaminen oli nopeaa ja pystyttiin pitämään korkea laatu viikosta toiseen ilman testisuunnittelijan manuaalista työtä. Tämä on huikea kehitys aiempiin projekteihin, joissa testejä on aloitettu automatisoimaan vasta projektin lopussa.

Regressiotestauksen tarkoitus on varmistaa viikoittain (kuva 22), että ohjelmiston aiemmat ominaisuudet vielä toimivat. Regressiotestaus on automaattitestausta hyvin laajalla testipatterilla, joka kasvaa koko ajan uusien ominaisuuksien lisääntyttyä. Haasteena on aika ja laatu. Miten tiheästi pitää testata, että tuote ei pääse huomaamatta rikkoontumaan ja toisaalta, miten käyttää rajalliset resurssit järkevästi, sillä ”turhaan” testaamiseen ei ole varaa?



Kuva 22. Regressiotestitulokset neljän viikon kumulatiivinä.

Näiden testien lisäksi on järjestelmätestaus, jossa varmistetaan, että uusi tuote toimii edelleenkin olemassa olevien muiden tuotteiden kanssa kuten järjestelmänhallinnan sekä kolmannen osapuolten laitteiden kanssa. Soveltuvuustestissä varmistetaan, että ääriarvot ja usean arvon konfigurointi yhtäaikaaisesti toimivat edelleenkin. Pilottitestaus on asiakkaan toimitiloissa tapahtuva testaus, joka suoritetaan asiakkaan testilaitteistolle.

5.5 Työkalut

Olimme aiemmin käyttäneet Microsoft Project työkalua projektin suunnitteluun ja hallintaan. Hyvin pian huomasin, että tämä ei sovellu scrum-projektinhallintaan. Siirryimme Excelin käyttöön ja samalla aloitimme etsimään parhaiten sopivaa scrum-työkalua. Työkaluryhmä oli tutustunut muutamaaan scrum-työkaluun ja valitsimme mielestämme parhaiten meille sopivan pilottikäyttöön: ScrumWorksPro.

Pilotointi tehtiin 15 henkilön voimin viikon aikana, siten että kaikki roolit, osastot, maat ja tuoteryhmät olivat edustettuna. Pilotoinnin kommentit ovat Taulukossa 5, johon on myös lisätty Rational Team Concert -työkalun näkemys ao. kommentista. Koska meillä ei ollut tästä työkalusta pilottiversiota, niin kysymysmerkki tarkoittaa, että tilanne pitää tarkistaa asiakastuesta.

Taulukko 5. Pilotointikommentit.

Kommentti	ScrumWorksPro	Rational Team Concert
Tehtävien riippuvuus	-	+
Viimeisen tehtävän päivitys	-	+
Käyttäjien määrän rajoitus	+	+
Teknologia	Ratkaistavissa	?
Hierarkia	-	?
Vaikeakäyttöinen	+	-
Liityntä vikatietokantaan	-	?
Metriikka	-	+
Ominaisuustason priorisointimahdollisuus	-	-
Ominaisuuden jakaminen aliominaisuuksiin ja linkitys	+	?
Työkalu pakottaa kiinteisiin rakenteisiin ja terminologiaan	-	-

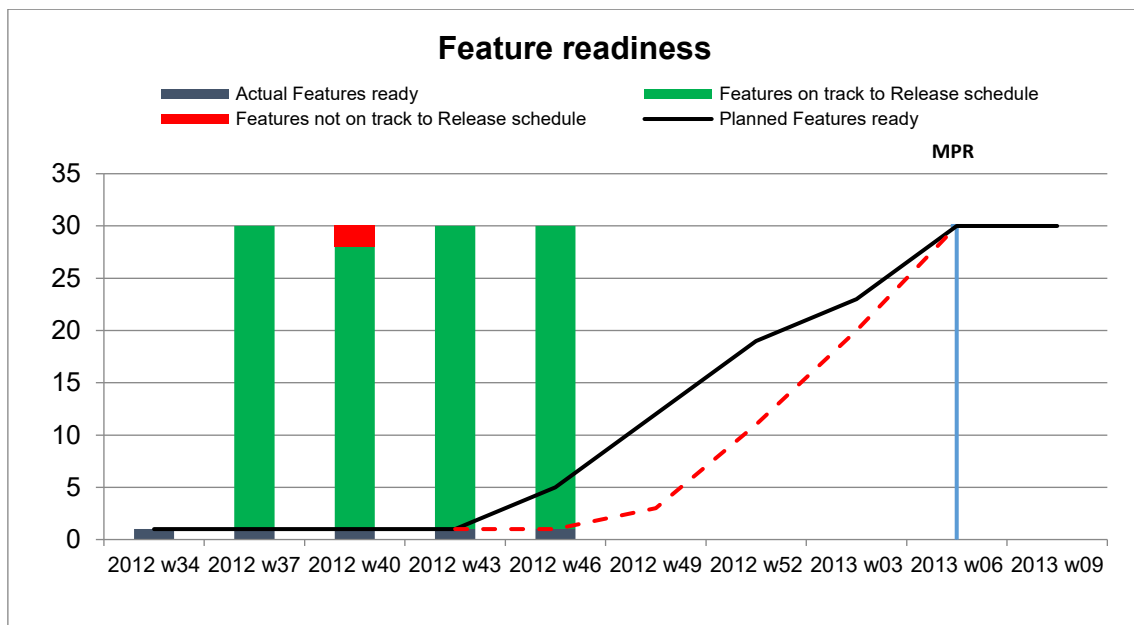
Teimme päätöksen, että tällä hetkellä juuri tässä aloitustilanteessa Excel tukee meitä parhaiten, sillä projektimme on niin suuri, että työkalua olisi niin paljon pitänyt soveltaa. Päätimme kuitenkin olla avoimia scrum-työkalulle tulevaisuudessa.

5.6 Metriikka

Yllä olevat testaus-metriikat ovat kuuluneet perusprojekteihin ennen scrumiakin, mutta yllätykseksemme Scrum-projektinhallintaan siirryttäessä teimme lisää metriikkaa. Yleensä paperityöt vähenevät scrumiin siirryttäessä. Näin kävikin turhien

suunnitelmien suhteen, mutta scrum-tyylistä metriikkaa, jossa ennustetaan tulevaisuutta historian valossa, tuli lisää.

Uutena metriikkana aloimme raportoida uusien ominaisuuksien valmistumista (kuva 23). Aluksi ennustettiin uuden ominaisuuden valmistumisajankohta ja sitten raportoitiin sprinteittäin tätä valmistumisajankohtaa vasten. Tämä on hyvin yksinkertainen ja havainnollinen metriikka, jota yrityksen Amerikan ylin johto piti jopa parempana kuin heidän määrittelemät metriikat. Tässä mittarissa valmis ominaisuus tarkoittaa täysin valmista, testattua ja dokumentoitua tuotetta, joka voidaan toimittaa sprintin lopussa asiakkaalle.



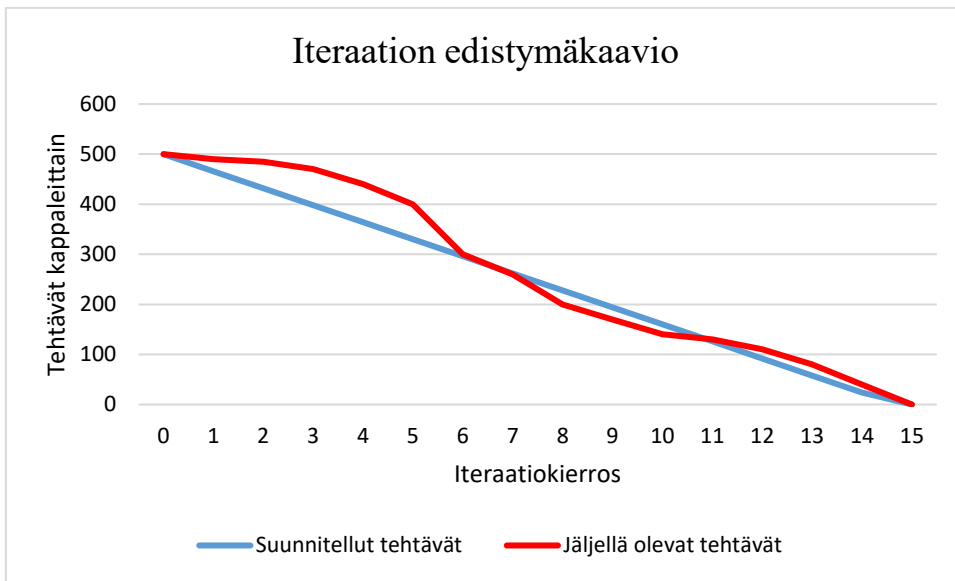
Kuva 23. Uusien ominaisuuksien kappalemääräinen toimitusvalmiuden suunnitelma ja toteutuma (y-akseli) kolmen viikon sprinteittäin (x-akseli).

Tiedot tähän kuvaajaan saatiin taulukosta (Taulukko 6), jossa ylläpidettiin perustietoja uusista ominaisuuksista kuten tiimi, riskitaso, valmistumistavoite sekä tiimin omistautuminen.

Taulukko 6. Uuden ominaisuuden tilannetieto.

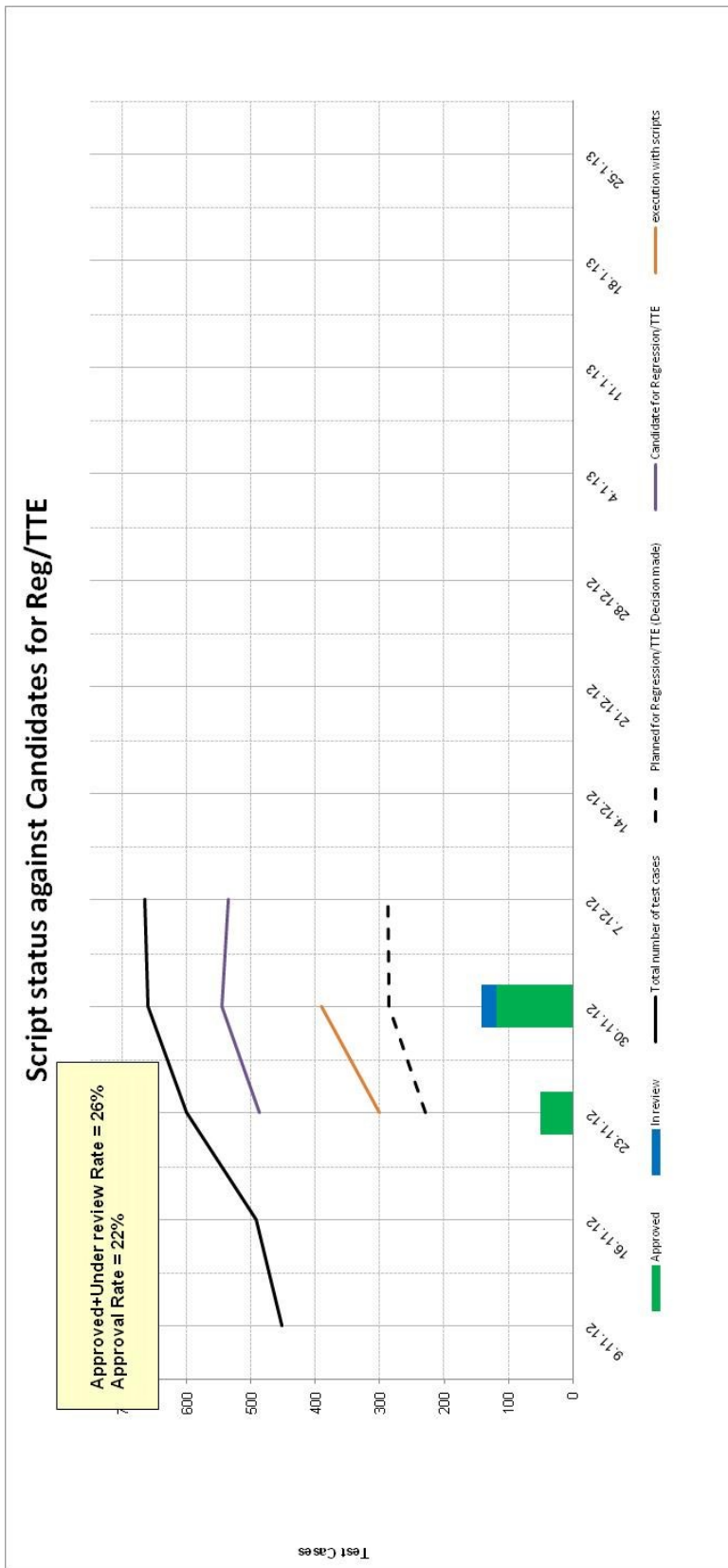
Nr o	Ominaisuus	Tiimi t	Riskitas o	Tavoitepäivä	Tiimin sitoutuminen	Kommentti
1	CDC2	E6, E7, C1, HW1	Keski	8.2	Keski	Riskisuunnitelma työn alla
2	Ethernet	C2	Pieni	7.12	Korkea	Sisältyy ECE:ään
3	Portit	E5	Pieni	7.12	Korkea	Sisältyy ECE:ään
4	VLAN	E5	Pieni	7.12	Korkea	Sisältyy ECE:ään
5	PWE silta	C2, C6	Keski	28.12	Korkea	Uusi ominaisuus

Ketterän menetelmän yksi parhaista mittareista on iteraatiokierroksen edistymäkaavio (kuva 24), johon on piirretty suunniteltujen tehtävien tavoitejana sekä todellinen jäljellä olevien tehtävien määrä. Tavoitejanassa koko iteraatiokierroksen tehtävät ovat jaettu iteraatiokierroksen päivillä, joka tässä projektissa on kolme viikkoa eli 15 työpäivää. Todellinen edistymä, että kuinka paljon tehtäviä on vielä jäljellä, kirjataan päivittäin pidetyssä sprinttipalaverissa. Tästä edistymäkaaviosta näkee yhdellä yleissilmäyksellä, että kuinka tiimin iteraatiokierros on edistynyt tavoitteeseen nähden, jolloin jo iteraation ensimmäisistä päivistä lähtien voidaan korjata toimintaa tämän yksinkertaisen kaavion palautteen perusteella.



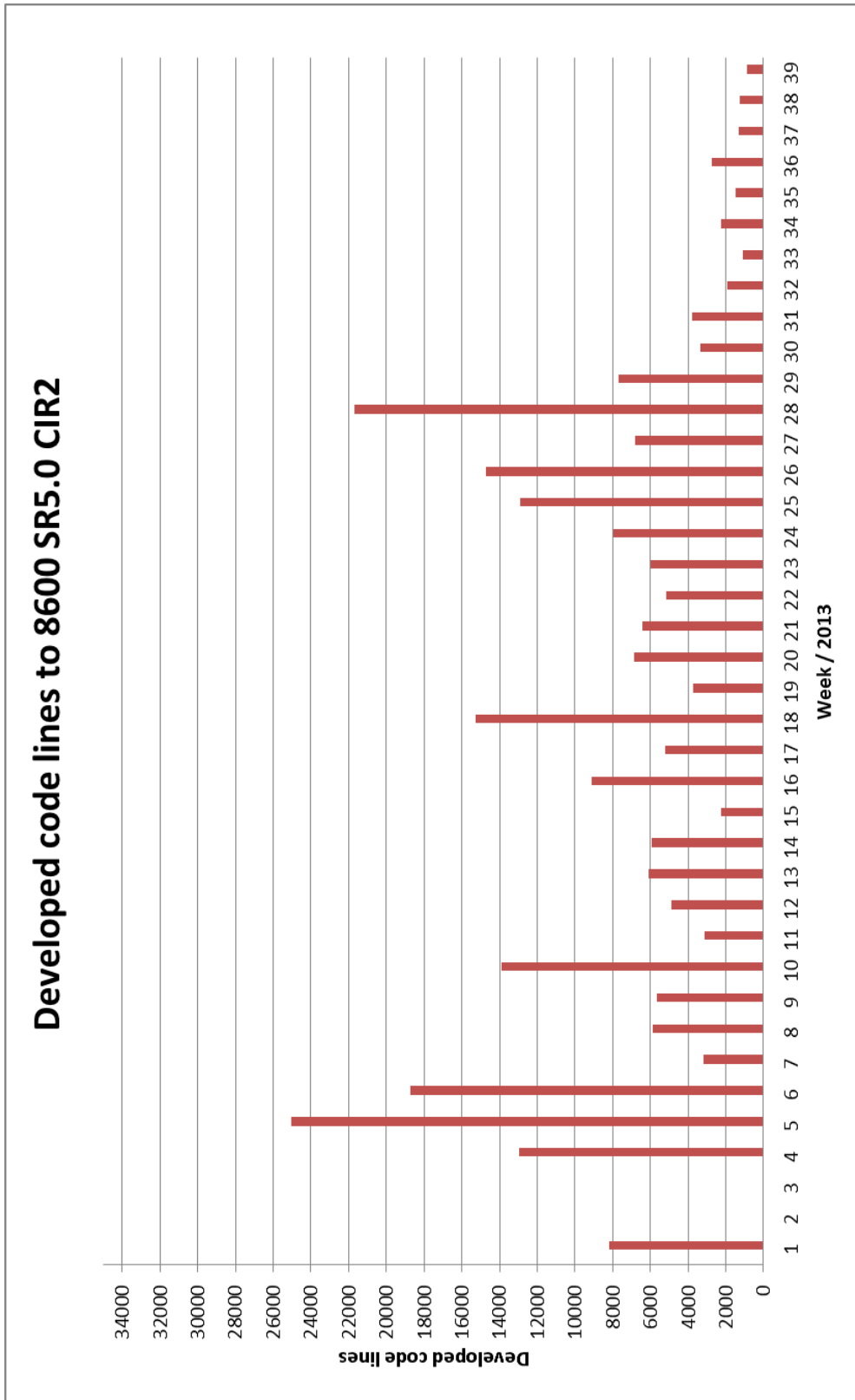
Kuva 24. Yhden tiimin tehtävien suunnittelu- ja toteumaedistymäkaavio kolmen viikon iteraatioissa.

Uusien ominaisuuksien testaus on aina aiemmissa projekteissa aloitettu vasta ominaisuuden koodaamisen jälkeen. Koska aikaa on ollut vähän, niin testaus on aloitettu manuaalisesti ja automaattiset skriptit on tehty vasta projektin loppuvaiheessa tai jopa asiakastoimituksen jälkeen. Nyt tavoitteena on tehdä ensin automaattiset skriptit, jolloin vältetään manuaaliselta työltä sekä saadaan samanlaista testausta viikosta toiseen. Ongelmana on testisuunnittelijoiden kiireellisyys edellisen projektin kanssa, johon he siis kehittävät tämän edellisen projektin automaattiskriptit. Kuva 25 raportoi automaattiskriptien tilannetta, jolloin tämä tärkeä työ tulee näkyviin.



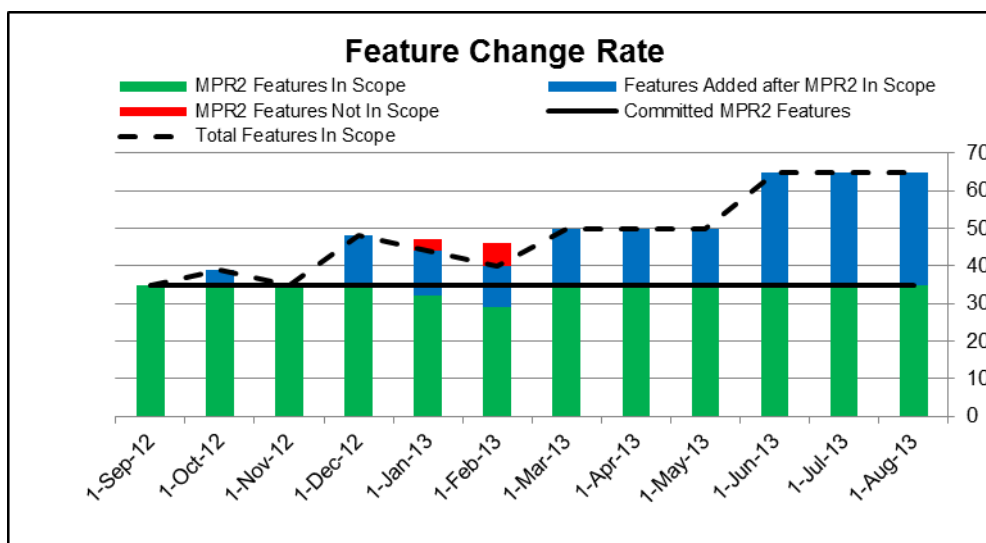
Kuva 25. Automatisoitujen testien viikottainen tilanne.

Aiemmissa projekteissa ominaisuudet ovat raportoitu valmiiksi ja ollaan siirrytty testausvaiheeseen. Sen jälkeen projekti on venynyt, sillä testauksessa on havaittu paljon virheitä. Virheiden korjaus ja suppeneminen on vienyt todella paljon aikaa ja jälkikäteen on huomattu, että raportoitu ominaisuus ei sisältänyt kaikkia haluttuja piirteitä, joten toteutusta on jouduttu tekemään testausvaiheessa. Ketterään menetelmään siirryttäessä testaus tehdään heti toteutuksen jälkeen palasissa, mutta varmistaakseen projektin kypsyyden tehtiin yksinkertainen mittari raportoimaan muutettuja lähdekielisiä ohjelmarivejä (Kuva 26). Lähdekielisen ohjelmarivien muutokset pitäisi supeta radikaalisti lähdekielisen ohjelma- ja testausvaiheen jälkeen, jolloin tuote on lopputestauksessa ennen asiakkaalle toimittamista.



Kuva 26. Viikoittain muuttuneet C++ lähdekieliset ohjelmarivit.

Ketterän menetelmän positiiviset puolet alkoivat näkyä projektin ominaisuuksien muutosmetriikassa (Kuva 25). Aluksi luvattiin valmistaa vain ne ominaisuudet, jotka pystyttiin heti aloittamaan olemassa olevilla tiimeillä, mutta luvattiin myös ottaa työnalle asiakkaalle tärkeitä ominaisuuksia prioriteettijärjestyksessä, kun tiimit vapautuvat. Kuva 27 osoittaa, että projekti pystyi tuottamaan melkein kaksinkertaisen määrän uusia ominaisuuksia.



Kuva 27. Projektin uusien ominaisuuksien kappalemääräinen toimitusvalmiuden suunnitelman ja toteutuksen muutokset kuukausittain.

Uusi tarkoituksenmukainen metriikka puolusti erinomaisesti ketterän projektimenetelmän hyviä puolia ja toimi näin ollen hyvänä viestinnän apuna Amerikan ylemmälle johdolle sekä laatuosastolle.

5.7 Projektin dokumentit

Tyypillisesti scrumiin siirryttäessä dokumenttien määrä vähenee. Tämä oli meidänkin tarkoitus, mutta koska olimme sopineet noudattavamme yrityksemme kansainvälistä ylätasoa prosessia, niin jokainen dokumentti oli tapauskohtainen.

Esimerkiksi projektisuunnitelmadokumentti oli yksi keskeisimmistä dokumenteista, joka tosin vanhenee heti seuraavana päivänä eikä sitä ylläpidetä, joten scrum-menetelmässä tällainen dokumentti on tarpeeton ja vie vain turhaan aikaa. Kun aloin tarkemmin katsomaan, niin ylätasoa prosessi ei ole tarkkoja määritteitä, että millainen projektisuunnitelma pitää olla, joten tein yksityiskohtaisesti ylätasoa dokumentin, jossa oli asiat, jotka eivät muutu ja muilta osin dokumentissa oli linkki sinne, missä ajantasainen tieto on. Tämä tuntui mielekkäältä ja toimivalta, joten otimme tämän toimintatavan käyttöön.

Ennen projektin aloitusta teimme ja katselmoimme seuraavat dokumentit: projektisuunnitelma, HW-projektisuunnitelma, testisuunnitelma, CIP-suunnitelma, PLS-projektisuunnitelma, NPI-projektisuunnitelma ja asiakasdokumentointiprojektisuunnitelma.

6 Projektinhallinnan arviointi käyttäjän näkökulmasta

Scrum-menetelmän mukaisesti palautetta kerättiin sprinteittäin koko matkan ajan eikä jätetty viimeiseksi työksi vesiputousmallin mukaisesti. Tämä on ehdottomasti oikea ja tehokas tapa, sillä projektin lopussa katseet ovat jo uudessa projektissa, muistetaan vain isot asiat ja on ainainen kiire.

Tässä projektissa ensimmäinen iso muutos tapahtui kaksi kuukautta sen jälkeen, kun scrum-tiimit olivat aloittaneet. Tuntui, että tiimien aikatauluvastuu ei ollut oikeasti oikein kellään (vaikka paperilla tietysti jokainen oli vastuussa). Scrum mestarit olivat organisaattoreita, tiimin työn mahdollistajia ja tiimin ääni ulkomaailmaan. He siis kysyivät tiimin sitoutumisastetta ja saivat vastauksen, että tiimi oli sitoutunut tämän sprintin tavoitteisiin ja yltäasolla oman tuotteen tekemiseen. Sitoutuminen oli tullut ja projekti eteni.

Ominaisuusomistajat olivat oikeasti vastuussa tuotteen synnystä ja he miettivät, että miten asiakkaan haluama tuote voitaisiin saada vielä tehokkaammin ja nopeammin aikaiseksi. Ominaisuusomistajat ovat kuitenkin käytännössä teknisiä asiantuntijoita, joten epäilytti siirtää tuotteen raportointivastuuta heille. Kun päätös oli tehty, niin se lokshti heti paikoilleen. Toki ominaisuusomistajat kokivat epävarmuutta olemaan tiimin ääni, mutta selkeästi he olivat koko ajan kantaneet vastuuta, että tuote saadaan tehtyä. Hehän vastasivat vaatimusten prioriteettilistasta.

Tämän ison muutoksen jälkeen tuli taas tunne, että rohkeita päätöksiä ja riskejä pitää uskaltaa tehdä. Odottaminen, pelkääminen ja tekemättä jättäminen ovat pahin vaihtoehto, sillä silloin ei ainakaan voida voittaa. Vääräkin päätös on parempi, sillä silloin ainakin opimme jotain – ja uuden, paremman päätöksen voi sitten tehdä sen pohjalta!

6.1 Asiakkaat

Suurin osa asiakkaista oli ollut pitkään yrityksen asiakkaana aiemmalle tuoteperheelle, joten toimitsijan vaihtaminen olisi ollut iso kynnyks. Asiakkaat joustivat aikataulun suhteen myös sen takia, että tarjottu tuote oli innovatiivinen ja kilpailukykyinen markkinoiden muihin tuotteisiin verrattuna. Asiakkaiden toive oli saada käyttöönsä mahdollisimman paljon uusia tuotteita mahdollisimman nopeasti, että he voivat palvella omia asiakkaitaan kilpailukykyisesti.

Vesiputousmallin aikatauluongelmat konkretisoituivat asiakkaille ensimmäisenä olemassa olevan tuotteen laatuongelmina, kun ylläpitoa vähennettiin, että saatiin maksimaalinen henkilöstö uusien ominaisuuksien tekemiseen. Pian aikatauluongelmat näkyivät myös siinä, että uusia ominaisuuksia jouduttiin karsimaan ja silti uuden tuotteen aikataulu oli auttamattomasti myöhässä.

Asiakkaat olivat jo todenneet, että yritys teki kaikkensa heidän palvelemiseksi ja oli rehellisesti tunnustanut ongelmansa. Kun avainasiakkaille puhuttiin ketteriin menetelmiin siirtymisestä, niin he olivat avoimia mille tahansa muutokselle ja toisaalta he uskoivat, että yritys teki kaikkensa. Olikin yllättävää, että asiakkaat ”ostivat” ketterät menetelmät ennen kuin yrityksen amerikkalainen johto. Asiakkaat uskoivat yrityksen näkemyksiin, että ketterällä menetelmällä myös asiakkaat voisivat vaikuttaa enemmän tuoteportfolioon. Sitäkin tärkeämpi oli tieto, että asiakkaat voisivat priorisoida uusia piirteitä koko ajan eikä vain vuosittain.

Asiakkaille kerrottiin, että he voivat vapaasti vaihtaa uusien ominaisuuksien prioriteettijärjestystä niin kauan, kunnes ominaisuutta aletaan tehdä, ja kun tekeminen aloitetaan, niin se myös saadaan valmiiksi. Kun uusi tuote on valmis, niin se periaatteessa voidaan antaa asiakkaalle, jos niin sovitaan. Tämä oli huikea muutos

kahden vuoden välein tuleviin tuotepaketteihin, jotka olivat myöhässä ja uusien tuotteiden tärkeyskin oli muuttunut matkan varrella.

Loistavat asiakassuhteet siis takasivat, että yritys pystyi siirtymään viimeiseen oljenkorteen, ketteriin projektinhallintamenetelmiin, amerikkalaisen johdon jarrutellessa. Asiakkaat olivat innokkaana mukana priorisoimassa omia tuotteitaan. Tietysti he halusivat kaikkiin uusiin tuotteisiin lupaukset mahdollisimman nopeasta toimituksesta, mutta ymmärsivät kokemuksesta, että lupauksella ei vielä omaa asiakasta rahasta. Nyt siis asiakas sai itse pitää tuoteprioriteettilistaa, hänelle kerrottiin suunnitelma kärkituotteen työnalle ottamisesta ja sitten kun tuotetta elettiin tehdä, niin tilanteesta raportoitiin. Kun tuote oli saatu tehtyä, niin asiakas sai halutessaan ottaa sen käyttöön tai odottaa seuraavan tuotteen valmistumista.

Asiakas oli sitä mieltä, että resurssit käytettiin maksimaalisen tehokkaasti ja että hän saa oman tuotteen mahdollisimman nopeasti tällä menetelmällä. Asiakkaan näkökulmasta tilanne oli läpinäkyvämpi kuin aiemmin ja jaettu tieto oli todellista ja uskottavaa. Tieto oli hyvin selkeää ja ymmärrettävää, jolloin asiakkaat kokivat, että kommunikointi on parantanut. Tietysti asiakkaat toivoivat, että heidän tuotetoiveet nousisivat yrityksen listalla korkeammalle ja että tuote tulisi valmiiksi nopeammin, mutta se on varmasti normaali toive kaikilta asiakkailta. Pääasia oli, että asiakas sai nopeammin uusia tuotteita ja että ne olivat juuri oikeita tärkeitä tuotteita ja että kommunikointi oli selkeää ja tarkoituksenmukaista.

6.2 Johtoryhmä

Projektin johtoryhmä muodostuu kolmesta osa-alueesta

- Projektin ohjausryhmä
- Yrityksen Suomen osa-alueen projektijohtoryhmä
- Yrityksen projektijohtoryhmä

Projektin ohjausryhmä on muodostettu tätä nimenomaista projektia varten ja se lakkautetaan, kun projekti loppuu. Yrityksen Suomen osa-alueen projektijohtoryhmä on Suomen alueen ylin päätäntäelin, joka tekee projektin suhteen joko päätöksen tai päätöksen viedä päätösvalta eteenpäin yrityksen projektijohtoryhmälle. Yrityksen projektijohtoryhmä on Amerikasta hallinnoitava projektin ylin päätöselin, joka tekee päätökset projektin aloittamisesta, jatkamisesta ja lopettamisesta.

6.2.1 Projektin ohjausryhmä

Projektin johtoryhmän jäsenet ovat yrityksen Suomen osa-alueen tuotekehitysjohtaja, Suomen osa-alueen projektihallinnanjohtaja, yrityksen tekninen huippuasiantuntija sekä Suomen osa-alueen tuotehallinnanjohtaja. Koska heillä kaikilla on koko tuotelinjaa koskeva vastuu omasta osa-alueestaan, heidän päätöksensä sisältävät myös pitkän aikavälin näkökulman.

Johtoryhmän jäsenet ovat olleet pitkään yrityksen palveluksessa ja tuntevat asiakkaat ja yrityksen toimintamallin erinomaisesti. He ovat olleet vastuussa, kun aiemmilla kerroilla projektit ovat venyneet ja tuotteen laatu on heikentynyt uuden kehittämisen varjossa.

Selvitysten antaminen sekä asiakkaille että yrityksen amerikkalaiselle johdolle on tuttua heille ja he tekevät kaikkensa, että projekti onnistuu ja turhilta selityksiltä vältyttäisiin. Yrityksen johtoryhmä valitsi projektipäällikön ja menetelmän, jota halusivat hänen käyttävän. Koska projektipäällikkö oli kokenut ja tunsu yrityksen toimintatavat ja henkilöt erinomaisesti, niin yhteistyö sujui alusta asti hyvin. Johtoryhmä halusi suoraa ja selkeää toimintaa, sekä rohkeita korjausliikkeitä, jos asiat eivät suju. Raportointi oli selkeää, läpinäkyvää ja luottamuksellista korostaen, että miten johtoryhmä voisi auttaa projektia eteenpäin.

6.2.2 Yrityksen Suomen osa-alueen projektijohtoryhmä

Joka toinen viikko kokoontuva PCRG on vastuussa Suomen tuoteryhmän projektien päätöksenteosta. Jokaisen osa-alueen korkein vastuhenkilö on edustettuna henkilökohtaisesti. Esitettävä materiaali oli jäsenten tutustuttavana vuorokausi etukäteen. Palaveri toteutuu videoneuvotteluna, konferenssipuheluna sekä liveMeetinginä.

Ryhmän jäsenet ovat olleet tekemässä uudistus- ja järjestäytymispäätöstä jo ennen kuin tämä SR5.0 projekti käynnistettiin, joten he olivat positiivisella asenteella odottamassa tehokasta ja nopeaa kehitystä nimenomaan aikataulun suhteen. Projektipäällikkö yhdessä ohjausryhmän kanssa tekivät kehityspäätökset itsenäisesti, joten yrityksen Suomen osa-alueen projektijohtoryhmä ei ollut siinä päätöksentekijänä.

Vaikka johtoryhmä oli varautunut uudistuksiin ja kovaankin taisteluun saadakseen nämä muutokset läpi, niin silti pettymyksiltä ei voitu välttyä. Ei suunnittelu- ja testaustyö sinällään ole muuttunut tai vähentynyt, kun menetelmää muutettiin, sillä suunnittelijat

olivat jo huippuunsa viritettyjä asiantuntijoita. Nyt työ vain tehtiin pienemmissä, paremmin arvioiduissa palasissa ja ongelmat ratkaistiin nopeammin. Kaikkea haluttua ei edelleenkaan saada tehtyä, vaikka kuinka unelmoitaisiin. Mutta se mitä tehdään, niin se saadaan valmiiksi luvatussa aikataulussa.

Projektin johtoryhmä on aiemmin tottunut lisäämään 'pieniä' asiakasta suuresti ilahduttavia ominaisuuksia pitkin aiempaa vesiputousprojektia, jolloin projektit venyivät ja asiakas ei saanut mitään luvatussa aikataulussa. Nyt tämä toimintamalli, jossa pieniä, asiakasta suuresti hyödyttäviä ominaisuuksia lisättiin projektiin, sopii oikein hyvin Scrum-projektiin. Tämä ilahdutti niin johtoryhmää kuin asiakastakin, kun päätöksiä voitiin tehdä useammin -ja mikä tärkeintä sovelluksia voitiin myös toimittaa asiakkaalle.

6.2.3 Yrityksen projektijohtoryhmä

Ryhmä muodostuu kaikkien tuotelinjojen johtajista ja päätöksentekijöistä. Amerikan laatuosasto vastasi kokouksen järjestelyistä ja kaikkien tuotelinjojen johtaja oli ylin päättävä taho. Ryhmä kokoontui kuukauden välein katselmoimaan kaikki projektit. Tiedot piti kirjata Dashboardiin vuorokautta aiemmin tietyssä muodossa, että projektin tilanne oli helppo nähdä yhdellä silmäyksellä.

Hyvin nopeasti huomasin raportoinnin kulttuurierot: suomalainen sivuuttavat saavutukset ja positiiviset asiat nopeasti, että aikaa jää ongelmien pohtimiseen ja amerikkalainen sivuttaa ongelmien pohtimisen yhteen sivulauseeseen ja pitää esityksen positiivisena. Koska kuulijakunta oli amerikkalaista, niin suomalainen tyyli antoi ihan liian negatiivisen kuvan ja näytti siltä, että suomalaiset eivät osaa hoitaa hommiaan.

Kun aikaa yhtä esitystä kohti oli rajallisesti, niin huomasin, että pääsivun lisäksi yksi asia käytiin aina läpi: riskit. Siinäkin sävy oli pidettävä positiivisena ja vaihtoehtoja etsivänä amerikkalaiseen tapaan. Harjoittelun jälkeen opin itsekin lukemaan amerikkalaisten rivien välistä ne suomalaiset ”todelliset asiat” eli ongelmat.

Amerikkalaisella johtajalla oli aina yksi kysymys, kun projekti oli vaikeuksissa: miten minä voin auttaa? Miltei joka kerta minulla projektipäällikkönä oli jokin avunpyyntö amerikkalaiselle johtajalle kuten esim. investoinnit eivät menneet läpi kankeasta prosessista läpi, laitteiston lisääminen nopeuttaisi tai huippuasiantuntijan käyttö jne. Amerikkalainen johtaja seisoj lupautensa takana ja todellakin vauhditti projektia omalta osaltaan.

Toisaalta yrityksen amerikkalainen johto ei hyväksynyt scrum-projektinhallintaan siirtymistä ja se todella haittasi etenemistä. Minulta projektipäällikkönä sekä suomalaisen yrityksen johdolta meni paljon aikaa ja energiaa uuden projektinhallintamenetelmän perustelemisessa. Vastaanotto oli hyvin vastahankainen ja he pelkäsivät, että homma karkaa käsistä. Lopulta (ensin kun asiakkaat itse oli saatu vakuutettua) asiakkaat vaativat scrum-projektinhallintaan siirtymistä amerikkalaiselta johdolta, jolloin heidän oli pakko antaa periksi. Toisaalta taas amerikkalainen laatuosasto ryhtyi seuraamaan erittäin tarkasti toimintaamme ja että varmasti toimimme vesiputousmallin edellyttämässä raamissa. Soveltaminen onnistui hyvin vahvan yrityksen prosessi- ja projektihallintatietämyksen pohjalta sekä positiivisella asenteella. Laatuosasto ei edelleenkään uskonut, että meidän menetelmällä voidaan menestyä, mutta he amerikkalaisina auttoivat oikean viestin ja painotuksen löytämisessä raportoinnissa amerikkalaiselle johtajalle. Meidän projektin metriikat olivat aina ajantasaiset, josta saimme kiitosta.

Kun asiakas tarvitsi yllättäen uuden sovelluksen voittaakseen itselleen asiakkaan ja kun me pystyimme sen toimittamaan sovitusti kuuden viikon aikataululla, niin yrityksen

amerikkalainen johto todella ihmetteli ketteryyttä. Tosin sävy oli vielä epäileväinen ja he pelkäsivät, että laatu on huonoa, mutta kun kaikki menivät hyvin, niin yrityksen amerikkalainen johtokin hieman kiinnostui tästä uudesta, ketterästä toimintatavastamme. Asiakkaalta tuleva positiivinen palaute oli suurin syy amerikkalaisen johdon mielipiteiden kääntyminen negatiivisesta myötämielisempään.

6.3 Projektitiimi

Projektitiimi oli vastahankainen uusien toimintojen aloittamiselle, mutta kun asia vain päätettiin nopealla aikataululla, niin vastarintaan ei juuri jäänyt aikaa. Projektitiimille ajatus myytiin siten, että te itse päätätte, että miten paljon te lupaudutte toimittamaan jokaisen sprintin aikana. Ajatus positiivisesta onnistumisesta houkutteli, vaikka projektitiimi ei uskonut, että koko hanke voisi onnistua.

Monet projektitiimit toimivat hyvin samallailla kuin aiemminkin, sillä vaikeimmat työt tarvitsivat erikoisosaamista, joten tiimin jäsenet eivät voineet scrum-tyylisesti valita tehtäviä tärkeysjärjestyksessä. Toisaalta taas osatehtävät voitiin priorisoida ja tiimi otti ne työnalle tärkeysjärjestyksessä.

Ensimmäisen sprintin jälkeen oli paljon melkein-valmista... Projektihenkilöstö oli oppinut vesiputousmallissa, että melkein valmis on melkein valmis. Nyt melkein valmis oli toimittamiseksi kelpaamaton eli hommaa ei saatu valmiiksi. Tämän ajattelutavan sisäänajo tiimeille otti todella paljon aikaa. Ylimmän johdon pelko, että tiimit lupaavat liian vähän, ei ikinä toteutunut muiden kuin intialaisten konsulttitiimien kanssa, sillä heidän palkanmaksu perustui sprintissä onnistumiseen.

Tästä tulikin yksi suurimmista kritiikeistä, kun sprinttimittareiden mukaan intialaiset konsulttitiimit onnistuivat parhaiten ja suomalaiset huippuasiantuntijat olivat hännillä. Suomalaiset asiantuntijat kritisoivat, että intialaisten konsulttitiimien työ ei ole yhtä vaativaa ja he eivät ole yhtä tehokkaita jne. Tämä on aivan totta, mutta sprintin arvioinnissa ei ole kyse työn vaativuudesta, vaan siitä miten hyvin tiimi tuntee kykynsä toimittaa valmista. Projektipäällikkönä korotin jatkuvasti suomalaisille tiimeillä, että älkää luvatko liikoja! Ja silti he yliarvioivat tiimin ajan ja eivät saavuttaneet kaikkea lupaamaansa. Tässä kesti todella kauan, että suomalaiset tiimit hyväksyivät oman kapasiteettinsa. Vuosien varrella oli opittu puristamaan ja uskomaan mahdottomiin. Nyt emme projektinhallinnassa halunneet tiimiltä hienoja, mahdottomia lupauksia, vaan luvattun toimitettavaksi kelpaavan tuotteen palasen, joka on toteutettu, testattu ja dokumentoitu.

Kun matkan varrella muutimme hieman pisteytystä ottamaan huomioon myös lopputavoitteen, niin suomalaiset tiimit pärjäsivät paremmin ja intialaisille tiimeille tuli mukaan myös pitkän tähtäimen suunnitelma.

Alkumuutosvastarinnan jälkeen projektitiimit mukautuivat uuteen, ketterään scrum-projektinhallintaan hyvin ja alkoivat nähdä positiiviset puolet ja oman vaikutusmahdollisuutensa. Kolmen viikon välein tapahtuva info-palaveriin, jossa kerrottiin jokaisesta tiimistä jotain positiivisessa hengessä, oli mukava ja odotettu tilaisuus.

6.4 Projektipäällikkö

Projektipäällikkönä olin kokenut ja tunsin yrityksen suomalaiset prosessit, henkilöstön, tuotteet ja toimintatavat hyvin. Olin toiminut vajaa kolme vuotta amerikkalaisen

tuotelinjan asiantuntijana ja projektipäällikkönä, joten tunsin amerikkalaisen kulttuurin melko hyvin, vaikka yllätyksiä edelleenkin tuli. Kaksi intialaista konsulttitiimiä oli toiminut aiemmassa projektissani, joten minulla oli käsitys heidän taidoistaan ja toimintatavoistaan ja toisaalta he konsultteina sopeutuivat hyvin yrityksen toimintatapoihin.

Minulle tämä projekti oli erittäin mieluinen haaste, jonka koin suurena luottamuksen osoituksena. Sain käyttää hyvin laajasti taitojani ja opin uutta ja koin kehittyväni niin työntekijänä kuin ihmisenäkin. Vastuu oli suuri, mutta ohjausryhmä luotti minuun ja minä heihin. Avoin, rehellinen ja positiivinen ilmapiiri mahdollisti kehittämisen ja kehittymisen. Tavoite oli koko ajan kirkaana mielessä eli miten pystymme parhaiten tarjoamaan asiakkaille maksimaalisen hyödyn organisaatiostamme. Uusien päätösten tekeminen ei ollut ollenkaan raskasta, sillä katseet olivat edessäpäin eikä vanhoja murehdittu. Esimerkiksi tiimien organisaatio- ja vastuurakenteiden muuttaminen Scrum mestarista ominaisuusomistajalle ei ollut vaikea päätös, sillä ensimmäinen organisaatorakenne ei toiminut halutulla tavalla, joten korjaava toimenpide piti tehdä.

Vanhan vesiputousmallin mukainen raportointi Amerikan johdolle ei sinänsä rasittanut minua sen kummemmin. Joka tapauksessa jotain pitää raportoida ja koska tunsin olemassa olevat käytännöt niin hyvin, niin itse prosessi ei vienyt aikaani vaan pääsin keskittymään sisältöön. Totta kai olisi ollut helpompi raportoida pelkästään Scrum-tyylisesti, mutta kun se ei ollut mahdollista, niin otin raportoinnin positiivisena haasteena. Tavoitteeni oli olla näiden kahden maailman välissä: projektitiimit toimivat ja raportoivat scrum-tyylisesti ja ylempi johto näkee totuttuja vesiputousmalliraportteja. Alun miettimisen ja suunnittelun jälkeen tämä sujuikin oikein jouhevasti.

Projektipäällikkönä haasteena oli tavoitteiden ja motivaation eteenpäin puskeminen siten, että jokainen 250 projektihenkilöstä tuntisi, että hänellä on tärkeä tehtävä ja se

motivoisi häntä ylittämään itsensä ja antamaan kaikkensa tiimin hyväksi. Kolmen viikon välein olevassa info-palaverissa valitsin yksi tai kaksi tavoitetta, jotka toistin kolmessa eri paikassa taatakseni tavoitteiden läpimenon. Motivaatiota yritin lisätä kertomalla info-palaverissa jokaisesta tiimistä jotain tärkeää ja merkittävää, että he tunnistaisivat itsensä ja että myös muut tiimit näkisivät heidän osuutensa.

Jos Scrum mestari oli negatiivinen tai vastahankainen, niin sanoman eteenpäin vieminen oli erittäin vaikeaa. Scrum mestari ja feature owner oli kuitenkin valittu niin teknisiltä kuin henkilökohtaisilta ominaisuuksilta tarkoin niin, että he sopivasti haastoivat ja täydensivät toisiaan. Muutamana tiimin kanssa tein enemmän töitä, että sain scrum-tyyliä ja positiivisuutta eteenpäin. Lopulta tiimit alkoivat löytää omia toimivia toimintatapoja ja tottua scrum-tyyliin. Tiimeillä, joiden scrum mestari tai feature owner eivät olleet suunnitelmallisia, oli jatkuvasti haasteita ymmärtää, että tehdyt lupaukset pitää pysyä muuttumattomana ja ne pitää toteuttaa kolmen viikon eli sprintin aikana.

Projektipäällikölle yksi keskeinen onnistumisen mittari on projektihenkilöstön vastuunottaminen. Scrummissa perinteisesti vastuu siirtyy nopeasti tiimille ja se tapahtui myös tässä projektissa.

7 Tulokset

Tässä tutkielmassa oli kolme tutkimustavoitetta, jotka esitetään keskeisten tulosten kanssa alla.

7.1 Laajan tuotekehitysprojektin haasteet

Ensimmäisen tavoitteena on luoda ymmärrys laajan tuotekehitysprojektin haasteista. Kirjallisuutta ja erityisesti lähtötilannetta tutkimalla selvisi, että tuotekehitysprojektin kolme suurinta haastetta ovat aikataulussa pysyminen, sovitun sisällön tuottaminen ja laatu.

Näistä asiakas kokee isoimmaksi ongelmaksi **aikataulun**, sillä asiakas on antanut lupauksia omille asiakkaille tämän aikataulun perusteella. Tämän perustuu myös heidän asiakkaiden laskutus, joten aikataulun viivästyminen lykkää myös asiakkaan maksuaikataulua.

Heikentynyt laatu näkyy asiakkaan verkossa käyttökatkoksina ja pidemmällä aikavälillä asiakastyytymättömyytenä, mutta se ei vaikuta välittömästi rahaliikenteeseen. Tukioorganisaatio tekee ammattitaitoista työtä, että ongelmalle löytyy mahdollisimman pian ainakin väliaikainen ratkaisu ja ongelma saadaan poistettua. Näin ollen yksittäiset laatuongelmat kantautuvat harvoin yrityksen johdon korviin, mutta pidempiaikaiset ongelmat ovat todellinen ongelma.

Sovitun sisällön tuottaminen ei ole niin suuri ongelma asiakkaalle, sillä he huomaavat sen aikataulun venymisenä, jolloin pääpaino on aikataululla. Isompi ongelma on tuotekehitysprojektin sisällä, jossa 'melkein valmis' ominaisuus, jonka prioriteetti on laskenut, roikotetaan mukana ja oletetaan ylioptimistisesti, että se tulee kohta

valmiiksi. Tämä positiivinen ajattelu tulee hyvin kalliiksi, sillä asiakas ei enää ole kiinnostunut ominaisuudesta, joten sille ei ole maksajaa ja toisaalta ominaisuus ei ole valmis, joten se on laadullinen riski.

Projektiryhmä, ohjausryhmä ja yrityksen johto ovat tehneet kaiken voitavansa vuosien varrella saadakseen nämä haasteet hallintaan. Silti aikataulut ovat venyneet, laatu heikentynyt kiihtyvällä vauhdilla ja sisältö on kutistunut entisestään. Epärealistiseen aikatauluun uskominen ei ole ratkaissut ongelmaa. Myöskään henkilömäärän lisääminen ei ole ollut ratkaisu (varsinkin kun henkilöstö on kokemattomia konsultteja vieraasta maasta), sillä edelleenkin kaikkein haastavimmat uudet piirteet ovat myöhässä, jolloin koko projekti on myöhässä. Ratkaisu pitää tulla uudesta näkökulmasta, joka muuttaa rakenteita ja toimintaa ratkaisevalla tavalla. Tällainen ratkaisu voisi olla Agile, ketterän kehityksen projektimallinnus.

7.2 Ketterän menetelmän soveltamisen tutkiminen laajaan tuotekehitysprojektiin

Toisena tavoitteena on tutkia ketterän menetelmän soveltamista laajaan tuotekehitysprojektiin. Yrityksellä oli kokemusta pienistä ja isoista tuotekehitysprojekteista, jotka käyttävät samaa vesiputoustuotekehitysprosessia. Kaikissa projekteissa oli sama ongelma eli avainhenkilöiden saatavuus. He olivat kriittisiä resursseja projektin käynnistämisvaiheessa, jolloin piti tehdä arkkitehtuurisia päätöksiä ja aikatauluarvioita. Toisaalta uuden projektin käynnistysvaiheessa nämä avainhenkilöt olivat täystyöllistettyjä käynnissä olevan projektin loppuvaiheessa, jolloin piti tehdä päätöksiä vikojen vakavuudesta ja korjaustärkeysjärjestyksestä. Näiden avainhenkilöiden puuttuminen oli yleisin syy projektin aloituksen myöhästymiselle sekä pienessä että isossa tuotekehitysprojektissa.

Merkittävä ero pienen ja ison tuotekehitysprojektin tuloksissa oli se, että pieni tuotekehitysprojekti pysyi paremmin aikataulussa ja myöhästyminen oli muutamia kuukausia ison tuotekehityksen vuoden myöhästymisen sijaan. Tarkemmin tutkittaessa huomattiin, että pienessä tuotekehitysprojektissa uudet ominaisuudet olivat paljon pienempiä ja sitä mukaan paremmin tunnettuja ja helpommin arvioitavia. Yleisin syy myöhästymiselle oli uusi, tuntematon ominaisuus, jota vasta lähdettiin määrittelemään ja arvioimaan, mutta saman aikaisesti sille piti osata arvioida kokonaistyöaika. Kun ominaisuutta lähdettiin tekemään, niin optimistinen työmääräarvio kohtasi realiteetit ja aikataulut venyivät.

Yrityksen omat kokemukset ja kirjallisuustutkimukset antavat samansuuntaisen viestin, että ketterä projektimalli hallitsee erinomaisesti epämääräisiä projekteja, joiden työmääräarviota ei pystytä kertomaan tarkasti projektin alussa. Toisaalta ketterän menetelmän soveltaminen laajaan tuotekehitysprojektiin toteutetaan ensin soveltamalla ketterää menetelmää pieneen projektiin ja siitä sitten vähitellen laajentaen isompiin projekteihin. Yrityksen ongelmana on nimenomaan aika, sillä enää ei ole aikaa pieneen kokeiluun, vaan on tehtävä isoja päätöksiä.

7.3 Ketterän menetelmän soveltaminen ja parannusehdotukset

Kolmantena tavoitteena on testata ketterää menetelmää laajassa projektissa ja analysoida sen soveltuvuus mahdollisine parannusehdotuksineen. 250 henkilöstön projekti käynnistettiin ketterällä menetelmällä kolmessa eri maassa. Muutosvastarinta oli voimakkainta projektitiimin lisäksi yrityksen amerikkalaisella johdolla. Asiakkaat ja lopulta myös yrityksen Suomen johtoryhmä olivat vakuuttuneita, että kokeilu on riskin arvoinen, sillä vanha tapa pienine korjausliikkeineen johtaa takuuvarmasti epäonnistumiseen.

Ketterän menetelmän tuotekehitys onnistui erinomaisesti kaikista riskeistä ja vastarinnasta huolimatta. Vihdoinkin pystyttiin keskittymään tekemiseen ja uusien ominaisuuksien priorisoimiseen aiemman myöhästyneen aikataulun selittelyn ja korjaamisen sijaan. Asiakkaalle pystyttiin toimittamaan luvatut ominaisuudet sovitussa aikataulussa ja projektiin pystyttiin lisäämään uusia ominaisuuksia, kun kehitystiimi vapautui. Kaikkein parasta oli, että tekemämme ennustukset historiasta opitun nopeuden pohjalta olivat paljon luotettavammalla kuin mitkään aiemmat laskelmamme vesiputousmallissa. Tämä oli todella suuri edistysaskel!

Asiakkaat luottivat, että yritys tekee kaikkensa saadakseen heille ja heidän asiakkaille maksimaalisen hyödyn. He lähtivät innolla mukaan ketterään menetelmään ja priorisoivat omia tuotteitaan. Heidän luottamuksensa oli avainasemassa, että yrityksen ylin johtokin lopulta antoi luvan ketteriin menetelmiin. Asiakkaat uskoivat, että tällä menetelmällä vähäisiä resursseja käytetään tehokkaimmin ja he saavat maksimaalisen hyödyn. Kommunikointi oli erittäin avointa asiakkaiden kanssa ja he saivat konkreettista, rehellistä tietoa projektin etenemisestä. He huomasivat, että luvatut, pilkotut ominaisuudet tulivat valmiiksi luvatussa ajassa ja heillä oli mahdollisuus saada uudesta ominaisuudesta demo ja halutessaan jopa uusi tuoteversion. Nyt keskustelu oli konkreettista ja ajantasaista eikä projektin myöhästymisen selitysten kuuntelua ja korjausliikkeiden suunnittelua pelastaakseen omat asiakassuhteensa.

Asiakassuhdetta voisi parantaa entisestään sukeltamalla syvemmälle asiakkaan tulevaisuuden suunnitelmiin. Jos asiakas avaisi tarkemmin heidän pitkän tähtäimen tavoitteensa, niin asiakasrajapinta voisi miettiä ratkaisua ja tekninen tiimi voisi pilkkoa ominaisuuden ja tehdä ratkaisun valmiiksi vähitellen palapalalta. Tämä olisi erittäin hyödyllistä yritykselle, sillä jos yksi asiakas tarvitsee uuden piirteen, niin voi olla, että sama ratkaisu soveltuisi myös jollekin toiselle asiakkaalle eli uusi ominaisuus saataisiin toiselle asiakkaalle valmiiksi jo ennen kuin he ovat ehtineet kaivata sitä.

Projektin johtoryhmän jäsenet tuntevat asiakkaat ja yrityksen toimintamallin erinomaisesti ja he tekivät rohkean päätöksen ketteriin menetelmiin siirtymisestä. Johtoryhmän kommunikointi asiakkaille ja projektipäällikölle oli erittäin luottamuksellista ja rehellistä ja yhteistyötä korostavaa. He tekivät kaikkensa auttaakseen projektipäällikköä ja projektia ja he kannattivat suoraa, rohkeaa toiminnan korjausta heti odottamisen sijaan.

Yrityksen Suomen osa-alueen johtoryhmä olivat mukana uudistus- ja järjestäytymispäätöksessä ennen ketterän kehitysprojektin aloitusta, joten he odottivat parannuksia ja aikataulussa pysymistä. Lievänä pettymyksenä tuli se, että koska suunnittelutiimi oli jo äärimmilleen viritetty, niin itse suunnittelu ja testaustyö eivät nopeutuneet. Työ toki tehtiin pienissä, paremmin ennustettavissa palasissa ja ongelmat ratkaistiin tehokkaasti ennen kuin ne ehtivät paisua suureksi, mutta perustyö piti silti tehdä normaalisti. Yrityksen Suomen johtoryhmää ilahdutti, että pieniä, mutta liiketoiminnan kannalta suuria uusia ominaisuuksia pystyttiin lisäämään tähän ketterään tuotekehitysprojektiin ja uusi tuoteversio voitiin toimittaa asiakkaalle, jolloin voitiin saada liikevaihtoa.

Yrityksen projektijohtoryhmä muodostui amerikkalaisista johtajista, jotka suhtautuivat ensin kielteisesti ja lopulta varautuneesti ketterään kehitykseen siirtymisessä. Perustelu ja kommunikointi veivät paljon projektipäällikön ja yrityksen Suomen johtoryhmän aikaa. Yrityksen laatuosasto seurasi projektia erittäin tarkkaan ja raportoi välittömästi Yrityksen Amerikan johtoryhmälle pienimmästäkin poikkeamasta. Koska laaturyhmä ei tuntenut ketteriä menetelmiä ja muutenkin ymmärsi asioita väärin negatiivisen asenteensa takia, niin eteneminen oli todella raskasta. Kun tulokset alkoivat näkyä parin kuukauden jälkeen, niin tilanne vähän helpotti.

Laatuosasto, jolla oli suora kommunikointiyhteys yrityksen Amerikan johtoryhmään, seurasi edelleenkin projektin jokaista vaihetta suurennuslasilla ja tästä alettiin saada

hyötyä, sillä he raportoivat myös onnistumisia sekä auttoivat projektipäällikköä raportoimaan tehokkaasti amerikkalaistyyliisesti, jolloin yrityksen Amerikan johtoryhmä pystyi konkreettisesti auttamaan esimerkiksi investointihyväksynnillä. Raportointimme oli aina ajantasaista ja meitä alettiinkin käyttää hyvänä esimerkkinä muille projekteille. Kun pystyimme toimittamaan uuden, liiketointakriittisen sovelluksen sovitusti asiakkaalle, niin yrityksen Amerikan johtoryhmä alkoi ymmärtää, että ketterässä menetelmässä on erittäin hyviä puolia. Asiakkaalta tuleva kiitos oli suurin syy yrityksen Amerikan johtoryhmän myötämielisempään suhtautumiseen.

Yrityksen kannalta selkeä parannusehdotus on ketterän menetelmän tarkempi tutkiminen ja siihen siirtyminen muidenkin maiden tuoteprojektien kanssa. Tämä on kuitenkin sellainen päätös, johon Suomen osasto ei voi vaikuttaa muuten kuin positiivisella esimerkillä ja ketterien menetelmien korostamisella onnistumisen hetkellä. Tällöin viesti alkaa menemään vähitellen eteenpäin ja yrityksen johtoryhmä toivottavasti päättää kokeilla ketteriä menetelmiä muissakin projekteissa ja muuttaa yrityksen prosessi tukemaan ketteriä menetelmiä.

Projektitiimit eivät uskoneet ketterän menetelmän toimivan koko projektille, mutta alkuvastarinnan päättivät onnistua omassa ketterässä projektissaan. Erikoisosaamisen takia monet projektitiimit toimivat työnjaon osalta kuten aiemminkin, mutta ketterät elementit kuitenkin integroituivat kuin itsestään tiimeihin. Tiimin tehtävien priorisointi ja yhdessä päättäminen korosti kaikkien jäsenien tärkeyttä ja säännölliset lyhyet palaverit lisäsivät kommunikaatiota ja tietoa koko tiimin välillä. Sprintin tavoitteiden asettaminen ja saavutusten tutkiminen siirsi vastuun tiimille.

Tiimit olivat aluksi ylioptimistisia tavoitteiden asettamisessa ja lisäksi sprintin alussa kaikki luvatut ominaisuudet aloitettiin yhtä aikaa, joten lopputuloksena oli monta melkein valmista ominaisuutta, mutta ei yhtään täysin valmista. Tiimille suurin pettymys oli se, että melkein valmis ominaisuus oli julkaisukelvoton, joten työ tulkittiin

tekemättömäksi. Toisaalta taas intialaiset konsulttitiimit osasivat laittaa riittävän matalat tavoitteet ja he onnistuivat parhaiten tavoitteiden saavuttamisessa. Tämä oli karvas pala suomalaisille huippuasiantuntijoille, jotka osasit itse työn erinomaisesti, mutta laittoivat liian optimistiset tavoitteet, joten he epäonnistuivat sprintin tavoitemittareiden suhteen. Tämä vuosien varrella opittu optimismi kostautui ja kesti kauan, että suomalaiset tiimit oppivat asettamaan realistiset tavoitteet ja tunnustamaan, että julkaisukelpoinen tuote on täysin valmis testauksineen ja dokumentointineen.

Huomasimme palautepalavereissa, että pitkän tähtäimen tavoite ei ole tarpeeksi korostettuna, joten muutimme pisteytystä ottaman myös tämän huomioon. Vähitellen tiimit alkoivat nähdä ketterän menetelmän positiiviset puolet myös koko projektin näkökulmasta ja tämä loi yhteishenkeä koko projektin suhteen.

Parannusehdotuksista konkreettisin tiimin osalta on, että työtä pitäisi pysytää pilkkomaan laajemmin osa-alueiksi siten, että yksi kokonaisuus pystyttäisiin tekemään eri tiimien avulla, aluksi eri sprinteissä ja jopa eri projekteissa, mutta tavoitteena, että jos on tarvetta, niin osattaisiin tehdä isokin kokonaisuus samassa sprintissä monien tiimien avulla. Tästä tulisi todellista kilpailuetua, jos muutamassa sprintissä pystyttäisiin toimittamaan asiakkaalle tärkeä ominaisuus, jolla hän pystyisi voittamaan kauppoja ja rahastamaan omia asiakkaitaan. Jos 250 henkilöä puhaltaa yhteen hiileen yhden sprintin ajan (3 viikkoa), niin siinä saadaan 15 vuotta tehtyä kolmessa viikossa ($250 \cdot 3 = 750$ viikkoa = $50 \cdot 15$ vuotta).

Toinen parannusehdotus on, että automaattinen testiohjelma pitää olla valmiina, kun toteutusta aloitetaan tekemään. Tällöin koodaaja hyötyy automaattisesta testiohjelmasta kahdestakin näkökulmasta: aikaa säätyy, kun ei tarvitse tehdä omaa testiohjelmaa ja toisaalta tulee testattua, että tavoite on ymmärretty oikein.

Dokumentointi oli myös hyvä tehdä ennen automaattista testiohjelmaa eli yhtä aikaa määrittelyn kanssa. Näin ollen dokumentaatio olisi suunnittelijoiden käytössä testiohjelmaa ja lähdekielistä ohjelmaa tehtäessä. Tämän lisäksi dokumentaation pitäisi kehittyä enemmän automaattiseen suuntaan niin, että pdf-manuaalit ja paperiversiot voitaisiin korvata web-manuaaleilla.

Projektipäällikkö oli kokenut ja tunsii yritykset prosessit, henkilöstön, tuotteet ja toimintatavat erinomaisesti. Hänellä oli ohjausryhmän täysi luottamus ja tavoite kirkkana mielessä maksimaalisen hyödyn saamisesta organisaatiosta. Ajantasaisen päätöksen tekeminen oli helppoa ja huojentavaa, kun ei tarvitse miettiä, että jos aiempi päätös oli 'väärä'. Kun päätös ketterästä menetelmästä ja toisaalta ylätason vesiputousmallin prosessin noudattamisesta oli tehty, niin metriikan ja raportoinnin suunnittelu pääsi käyntiin. Koska raportointi oli tuttua ennestään ja ketterän menetelmän raportointi oli avointa ja tarkoitushakuista, niin palaset loksahivat nopeasti paikoilleen ja näiden kahden maailman välinen raportointi täytti prosessin tarpeet sekä antoi kuin huomaamattaan lisätieto ketterän menetelmän etenemisestä.

Motivaation ja jokaisen tarpeellisuuden ylätason tavoitteeseen peilaten oli näin laajassa projektissa haaste, mutta ketterät menetelmät nimenomaan auttoivat tässä. Säännölliset info-palaverit sprintin vaihtuessa olivat mainio tapa korosta saavutuksia ja tavoitteita jopa tiimikohtaisesti. Tiimin Scrum mestari ja ominaisuuden omistaja olivat keskeisessä asemassa motivaation suhteen, joten projektipäällikön tärkeä tehtävä oli varmistaa näiden avainhenkilöiden oma motivaatio ja tavoitteet, että sama viesti menee tiimeille myös heidän kautta. Ketterän menetelmän yksi isoimpia muutoksia ja onnistumisen askeleita on vastuun siirtyminen tiimille, ja näin tapahtui myös tässä projektissa.

Suomen asiantuntijoiden suurimpia kritiikkejä oli intialaisten konsulttitiimien hyvä menestyminen heidän omaan menestykseen nähden. Konsulttitiimit olivat tavoitteiden ja metriikan asiantuntijoita, ja he tosiaankin toimittivat juuri sitä mitä tilattiin. Mittareissa ei ollut selkeää pitkän tähtäimen tavoitetta eli koko ominaisuuden valmistumista, jolloin erinomaisista sprinttituloksista huolimatta koko ominaisuus oli erittäin paljon myöhässä. Suomen tiimeillä taas pitkä tähtäin oli koko ajan mukana, joten vaikka sprinttitulokset olivat huonoja, niin koko ominaisuus tuli valmiiksi sovitussa ajassa. Parannusehdotus on, että pitkän tähtäimen tavoite sidotaan metriikkaan, jolloin jokaisessa sprinttikatselmoinnissa katselmoidaan myös koko ominaisuuden tilannetta.

HW-tiimit ja core-tuotekehitys ovat keskeisesti mukana tuotekehityksessä ja ne seurasivat ketterän kehityksen kulkua joustavasti mukautuen. Parannusehdotuksena pitkän tähtäimen ominaisuudet pitää pystyä myös priorisoimaan, jolloin HW- ja core-tiimit pääsevät ajoissa tekemään näitä pitkäaikaista kehitystä tarvitseviä ominaisuuksia.

8 Johtopäätökset

Tämän tutkielman tarkoituksena oli miettiä ja analysoida laajan tuotekehitysprojektin haasteita ja kokeilla niihin ketterää projektinhallintamenetelmää parempien tuloksien saamiseksi. Teknologian kehityksen myötä alati muuttuvassa ympäristössä asiakas ei pysty kertomaan vaatimuksiaan ja toisaalta yrityksen on vaikea nähdä parin vuoden päähän, jolloin tuote tulee ulos asiakkaan ulottuville. Perinteisellä vesiputousmallilla tuote on auttamattomasti myöhässä. Pienet yritykset ovat siirtyneet ketteriin menetelmiin ja havainneet ne hyödyllisiksi. Laajassa tuotekehitysprojektissa menetelmän vaihtaminen on iso riski, mutta onnistuessaan saavutettavat hyödyt olisivat myös mittavia.

Tutkielman kohteena oli kansainvälisen keskisuuren IT-yrityksen tuotekehitysprojekti. Alun perin tuoteprojekteja on ollut useita, mutta nyt oli tarkoitus tehdä kaikki yhden sateenvarjoprojektin alla. Miltei koko tuotekehitysorganisaatio kaikessa kolmessa maassa (Suomi, USA, Intia) siirtyi tekemään sovelluksia tämän sateenvarjoprojektin alle.

8.1 Laajan tuotekehitysprojektin haasteet

Tutkielma rakentui kolmen tavoitteen kautta. Ensimmäisen tavoitteena on luoda ymmärrys laajan tuotekehitysprojektin haasteista. Työn alussa kerroin vesiputousmallista, joka oli alkuperäinen, käytössä oleva projektinhallintametodi. Listasin asiakkaan, yrityksen johdon, projektipäällikön ja projektihenkilöstön näkökulmasta olevat haasteet, joiden voittamiseksi on tehty kaikki voitavat vuosien varrella. Epärealistiseen aikatauluun uskomisen tai henkilömäärän lisääminen eivät ratkaisseet aikataulujen venymistä, laadun heikentymistä ja sisällön kutistumista.

Tarvitaan ratkaisu, joka tuo selkeästi uutta näkökulmaa. Työn tulokset osoittavat, että ketterät menetelmä ovat tämä ratkaisu, sillä ne soveltuvat hyvin epämääräisiin, vaikeasti määriteltäviin ja ennustettaviin projekteihin.

8.2 Ketterän menetelmän soveltamisen tutkiminen laajaan tuotekehitysprojektiin

Tutkielman toisena tavoitteena on tutkia ketterän menetelmän soveltamista laajaan tuotekehitysprojektiin. Tähän tavoitteeseen vastattiin tarkastelemalla kirjallisuutta. Tutkimuksessa selvisi, että aiheesta ei ole paljon kirjallisuutta. Aihetta on pääosin käsitelty esimerkein, joten selkeää, yksiselitteistä vastauksia ei ole. Näistä sai kuitenkin hyvän käsityksen aihealueista, että mitä pitää selvittää laajan tuotekehitysprojektin muuttamisessa ketteräksi.

Ketterät menetelmät soveltuvat erittäin hyvin epämääräisiin, kompleksisiin ja vaikeasti arvioitaviin projekteihin, sillä menetelmän kulmakivenä on ominaisuuksien jatkuva pilkkominen pienempiin osasiin, ja uudellen arviointi uuden tiedon ja kokemuksen karttuessa. Yrityksen omat kokemukset ja kirjallisuustutkimukset viestittävät, että pilkkottua, pienempää ominaisuutta on helpompi arvioida kuin epämääräistä, laajaa ominaisuuskompleksia, joten työmääräarvioista tulee realistiset ensin lyhyellä tähtäimellä ja sitten myös pidemmällä tähtäimellä, kun on opittu matkan varrella.

8.3 Ketterän menetelmän soveltaminen ja parannusehdotukset

Kolmantena tavoitteena on testata ketterää menetelmää laajassa projektissa ja analysoida sen soveltuvuus mahdollisine parannusehdotuksineen. Tähän tavoitteeseen päästiin tutkimalla ja analysoimalla laajan projektin onnistumiskokeilua.

- Ominaisuuksien priorisointi yhdessä asiakkaiden kanssa auttoi kaikkia ymmärtämään tärkeysjärjestyksen.
- Luottamuksellinen ja rakentavat kommunikointi niin asiakkaiden, johtoryhmän ja projektiorganisaation välillä suuntasi asioista keskusteluun, joka poisti stressiä.
- Pieniä tärkeitä ominaisuuksia pystyttiin lisäämään, sillä sprintissä tehtiin pieniä toimivia kokonaisuuksia eikä isoja, toimimattomia jättiläisominaisuuksia.
- Amerikan johtoryhmän sekä laaturyhmän negatiivinen asenne muuttui ensin neurtaaliksi ja projektin lopussa jopa hyväksyväksi positiivisten tulosten myötä.
- Asiakkaiden luottamus ja innostus lisääntyi, kun aikataulut pitivät.
- Koska aikataulut pitivät, niin kehitystiimi vapautui sovitusti uusien ominaisuuksien toteuttamiseen.
- Ominaisuuksien pilkkominen ja työmäärän arviointi helpottuivat ajan saatossa.
- Projektitiimi oppivat, että sprintin tuotos pitää olla täysin toimiva palanen -eikä monta 'melkein valmista' palasta.
- Projektitiimit oppivat käytännön kautta ketterän menetelmän hyödyt ominaisuuksien pilkkomisessa ja aikatauluarvioimisessa, mikä vähensi stressiä.
- Pilkkottujen ominaisuuksien suunnittelussa, toteutuksessa ja testauksessa ei tapahtunut nopeutumista, sillä ne olivat jo tehokkaita ja äärimmilleen viritettyjä.

Ketterän menetelmän testaamisessa laajaan tuotekehitysprojektiin nousivat esille myös parannusehdotukset tulevia projekteja silmällä pitäen.

Parannusehdotukset tärkeysjärjestyksessä

1. Asiakkaan pitkän tähtäimen tulevaisuudensuunnitelmat pitää saada selville, että tulevaisuuden ongelmille ehditään kehittää ratkaisut ajoissa. Näin asiakas on valmis maksamaan ratkaisusta enemmän ja toisaalta yritys voi myydä saman ratkaisun toisellekin asiakkaalle.
2. Työtä pitäisi pysytää pilkkomaan enemmän osa-alueiksi siten, että yksi kokonaisuus pystyttäisiin tekemään eri tiimien avulla, aluksi eri sprinteissä ja jopa eri projekteissa, mutta tavoitteena, että jos on tarvetta, niin osattaisiin tehdä isokin kokonaisuus samassa sprintissä monien tiimien avulla. Tästä tulisi todellista kilpailuetua, jos muutamassa sprintissä pystyttäisiin toimittamaan asiakkaalle tärkeä ominaisuus, jolla hän pystyisi voittamaan kauppvoja ja rahastamaan omia asiakkaitaan. Jos 250 henkilöä puhaltaa yhteen hiileen yhden sprintin ajan (3 viikkoa), niin siinä saadaan 15 vuotta tehtyä kolmessa viikossa ($250 \cdot 3 = 750$ viikkoa = $50 \cdot 15$ vuotta).
3. Metriikkaan pitää saada sidottua pitkän tähtäimen tavoite. Tiimin nopeus selviää ajan kanssa, mutta kehitettävänä on vielä tiimin kyky arvioida omaa toimintakykyään realistisesti.
4. Ketterän menetelmän korostaminen projektin onnistumishetkellä, jolloin yrityksen johtoryhmä alkaisi vähitellen ymmärtää ketterän menetelmän hyödyt. Yrityksen tuotekehityksen prosessin muuttaminen ketteriä menetelmiä tukevaksi olisi ensimmäinen parannustavoite. Ketterän menetelmän soveltaminen muissakin maissa oleviin projekteihin olisi toinen selkeä parannustavoite.
5. Automaattinen testiohjelma pitäisi olla valmiina, kun toteutusta aloitetaan tekemään. Tällöin ohjelmistosuunnittelija hyötyy automaattisesta testiohjelmasta kahdestakin näkökulmasta: aikaa säätyy, kun ei tarvitse tehdä omaa testiohjelmaa ja toisaalta tulee testattua, että tavoite on ymmärretty oikein.
6. HW-projektit ja core-tuotekehitys pitää kuljettaa mukana ja ne pitää pystyä priorisoimaan pitkän tähtäimen näkökulmasta.

7. Dokumentointi oli myös hyvä tehdä ennen automaattista testiohjelmaa eli yhtä aikaa määrittelyn kanssa. Enemmän saisi myös mennä automaattiseen suuntaan web-maailmaan niin, että pdf-manuaalit ja paperiversiot voisi jättää tekemättä.

Kaiken kaikkiaan ketterä menetelmä soveltui erinomaisesti tähän laajaan tuotekehitysprojektiin ja sillä saatiin parempien, konkreettisten tulosten lisäksi positiivinen yhteisen projektin tavoitteisiin ja tekemiseen uskomisen. Ketterää menetelmää täytyy ehdottomasti jatkaa ja yllä olevia parannusehdotuksia tulee ottaa mukaan tuleviin projekteihin.

Lähteet

Agile Manifesto b, (2001): *Ketterän ohjelmistokehityksen julistus*. Agilemanifesto.org. (24.10.2023). Saatavissa: <https://agilemanifesto.org/iso/fi/manifesto.html>

Ambler, Scott (2002). *Agile modeling. Effective practices for extreme programming and the unified process*. United States of America: Robert Ipsen. 384. ISBN: 0-471-20282-7.

Anita Saaranen-Kauppinen & Anna Puusniekka (2009) *Menetelmäopetuksen tietovaranto KvaliMOTV kvalitatiivisten menetelmien verkko-oppikirja*. Tampereen yliopisto: Yhteiskuntatieteellinen tietoaarkisto. Saatavissa: [kvalimotv.pdf \(tuni.fi\)](#)

Björkholm, Tomas & Hans Brattberg (2021). *Prioritera, fokusera, leverera- din snabbguide till Lean, Agile, Scrum och XP*. version 1.7 Beta. Crisp. 113. ISBN: 978-91-978630-5-6. Saatavissa: https://www.crisp.se/wp-content/uploads/2021/05/Prioritera-fokuser-leverera17_beta-web.pdf

Gustavsson, Tom (2007). *Agile – konsten att slutföra projekt*. Karlstad. Tryckeri Scandbook AB, Falun 2010. 221. ISBN 978-91-976217-1-7.

Laani, Maarit (2012). *Agile methods in large-scale software development organizations. Applicability and model for adoption*. Juvenes print. Tampere 2012. Acta Univ. Oul. 192. ISBN 978-952-62-0034-7. Saatavilla: <http://jultika.oulu.fi/Record/isbn978-952-62-0034-7>

Lekman, Lare. (2015). Suomenkielinen Nexus Guide 2015 julkaistu. Lekman consulting [verkkojulkaisu]. Saatavilla: <https://www.lekman.fi/files/ugd/7e24321bbf3c9eab724cb29780ae10cd57d2d0.pdf>

Schwaber, Ken & Mike Beedle (2002). *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice Hall. ISBN 0-13-067634-9.

Sebestyén, Ulla (2016). *Multiprojekt ledning av portföljstyrda project*. Parmatur Handelsbolag 2005-2016. Tryck: Skogs Rulloffset, Vellinge 2005. 238. ISBN 978-91-637-8363-0.

Theocharis, Georgios, Marco Kuhrmann, Jürgen Münch & Philipp Diebold (2015). *Is Water-Scrum-Fall Reality? On the Use of Agile and Traditional Development Practices*. Lecture Notes in Computer Science. Springer. 18. Saatavilla: http://dx.doi.org/10.1007/978-3-319-26844-6_11

Tonnquist, Bo (2024). *Projektledning* (upplaga 9). Sanoma utbildning. Printed in Polen by Interak. 538. ISBN 978-91-523-6557-1.

Töttö, Pertti (2004): *Syvällistä ja pinnallista. Teoria, empiria ja kausaalisuus sosiaalitutkimuksessa*. Tampere: Vastapaino 2004. 317. ISBN 951-768-127-5.

Yin, Robert. K. (2009). *Case study research: Design and methods* (4th Ed.). Thousand Oaks, CA: Sage. 282. ISBN: 978-1452242569.

Wikipedia. (2024). Rational Unified Process. (19.11.2024). Saatavilla:

https://en.wikipedia.org/wiki/Rational_Unified_Process

Wikipedia. (2024). Scrum. (19.11.2024). Saatavilla:

<https://fi.wikipedia.org/wiki/Scrum>

Wikipedia. (2024). Tapaustutkimus. (19.11.2024). Saatavilla:

<https://fi.wikipedia.org/wiki/Tapaustutkimus>