

Jenni Peltomäki

Kvartaalitestauksen kehittäminen julkisella sektorilla – ketterät menetelmät

Opinnäytetyö

Tradenomi (ylempi AMK)

Tietojenkäsittelyn ja liiketoimintaosaamisen koulutus

Kevät 2026



**KAMK • University
of Applied Sciences**

Tiivistelmä

Tekijä: Peltomäki Jenni

Työn nimi: Kvartaalitestauksen kehittäminen julkisella sektorilla – ketterät menetelmät

Tutkintonimike: Tradenomi (ylempi AMK), tietojenkäsittely ja liiketoimintaosaaminen

Asiasanat: ohjelmistotestaus, ketterät menetelmät, Scrumban, testiautomaatio, kvartaalitestaus, HR-järjestelmät

Opinnäytetyön tavoitteena oli kehittää HUS Helsingin yliopistollisen sairaalan HR-järjestelmien kvartaalitestauksen prosessia hyödyntämällä Scrumban-menetelmää. HR-järjestelmien toimivuus on kriittistä henkilöstöhallinnon, palkanmaksun ja organisaation päätöksenteon kannalta ja järjestelmiin tehtävien säännöllisten kvartaalipäivitysten testauksella on merkittäviä vaikutuksia myös muihin tietojärjestelmiin ja organisaation päivittäisiin toimintaprosesseihin. Kehittämistyön tavoitteena oli luoda yhtenäinen ja läpinäkyvä toimintamalli testauksen hallintaan ja seurantaan, parantaa testausprosessin ennakoitavuutta sekä osallistaa keskeiset sidosryhmät aiempaa aktiivisemmin testaukseen liittyvään yhteistyöhön. Lisäksi tavoitteena oli yhdenmukaistaa toistuvat työvaiheet, selkeyttää vastuunjakoja sekä vähentää hukkaa testauksen tehokkuuden ja luotettavuuden parantamiseksi.

Työn teoriaviitekehys muodostui ketteristä menetelmistä, Lean-ajattelusta sekä ohjelmistotestauksesta. Ketteristä menetelmistä tarkasteltiin erityisesti Scrum-, Kanban- ja Scrumban-malleja sekä niiden soveltuvuutta HR-järjestelmien testausympäristöön. Lisäksi korostettiin tiimityön, vuorovaikutuksen ja jatkuvan parantamisen merkitystä osana kehittämistyötä. Ohjelmistotestausta käsiteltiin testauksen periaatteiden, vaiheiden ja keskeisten käytäntöjen näkökulmasta ketterässä ohjelmistokehityksessä.

Työ toteutettiin toimintatutkimuksena, joka eteni kolmen syklin kautta. Ensimmäisessä syklissä kartoitettiin lähtötilanne ja suunniteltiin kehittämistoimenpiteet. Toisessa syklissä toteutettiin ensimmäiset käytännön muutokset testausprosessiin ja arvioitiin niiden toimivuutta. Kolmannessa syklissä kehitettyä toimintamallia tarkennettiin ja vakiinnutettiin osaksi käytäntöä. Aineisto kerättiin fokusryhmähaastattelun, havainnoinnin, kehittämistyöpajojen ja retrospektiivien avulla ja se analysoitiin laadullisella sisällönanalyysillä. Kehittämistyön tuloksena palaverikäytännöt selkeytyivät, testausaikataulu keskitettiin ja koko testausprosessin hallittavuus parani. Tulosten perusteella ketterien menetelmien hyödyntäminen lisäsi läpinäkyvyyttä, yhteistyötä ja toiminnan tehokkuutta, mikä tukee organisaation jatkuvaa kehittämistä ja toimintavarmuutta.

Abstract

Author: Peltomäki Jenni

Title of the Publication: Development of Quarterly Testing in the Public Sector – Agile Methods

Degree Title: Master of Business Administration, Information and Communication Technologies

Keywords: software testing, agile methods, Scrumban, test automation, quarterly testing, HR systems

The aim of this thesis was to develop the quarterly testing process for the HR systems at HUS Helsinki University Hospital using the Scrumban method. The functionality of HR systems is critical for human resources management, payroll, and organizational decision-making, and testing regular quarterly updates to the systems also has a significant impact on other information systems and the organization's daily operating processes. The goal of the development work was to create a consistent and transparent operating model for testing management and monitoring, improve the predictability of the testing process, and involve key stakeholders more actively in testing-related collaboration. Another goal was to harmonize recurring work phases, clarify the division of responsibilities, and reduce waste in order to improve the efficiency and reliability of testing.

The theoretical framework for the work consisted of agile methods, Lean thinking, and software testing. Among the agile methods, the Scrum, Kanban, and Scrumban models and their suitability for the HR system testing environment were examined in particular. In addition, the importance of teamwork, interaction, and continuous improvement as part of development work was emphasized. Software testing was discussed from the perspective of testing principles, phases, and key practices in agile software development.

The work was carried out as action research, which progressed through three cycles. In the first cycle, the initial situation was mapped out and development measures were planned. In the second cycle, the first practical changes to the testing process were implemented and their effectiveness was evaluated. In the third cycle, the developed operating model was refined and established as part of the practice. The data was collected through focus group interviews, observation, development workshops, and retrospectives, and it was analyzed using qualitative content analysis. As a result of the development work, meeting practices were clarified, the testing schedule was centralized, and the manageability of the entire testing process improved. Based on the results, the use of agile methods increased transparency, cooperation, and operational efficiency, which supports the organization's continuous development and operational reliability.

Sisällys

1	Johdanto	1
2	Ketterän kehittämisen taustat.....	3
2.1	Agile Manifesti.....	4
2.2	Lean-periaatteet ja historia	5
3	Ketterät menetelmät	7
3.1	Scrum.....	7
3.1.1	Scrum-tiimi.....	8
3.1.2	Scrumin tapahtumat	10
3.1.3	Scrumin tuotokset.....	12
3.2	Kanban.....	14
3.2.1	Työn kulun visualisointi.....	15
3.2.2	Läpimenoajan mittaaminen	16
3.2.3	Työn alla olevien tehtävien rajoittaminen.....	16
3.3	Scrumban.....	17
4	Ketterä tiimi ja organisaatio	20
4.1	Tiimin rakenne, johtaminen ja yhteistyö.....	20
4.2	Ketterän toiminnan periaatteet ja käytännöt	21
5	Ohjelmistotestaus ketterässä kehityksessä.....	23
5.1	Testauksen rooli	23
5.2	Testauksen vaiheet ketterässä kehityksessä.....	24
5.2.1	Ennen sprintin aloitusta	25
5.2.2	Sprintin aikana	27
5.2.3	Sprintin päättäminen	28
5.3	Regressiotestaus ja testiautomaatio	28
5.4	Testauksen dokumentointi.....	30
5.4.1	Testaussuunnitelma.....	31
5.4.2	Testitapaukset.....	33
5.4.3	Testiraportti	34
5.5	Testaustyökalut	35
6	Tutkimus- ja kehittämismenetelmät	37

6.1	Toimintatutkimus tutkimusstrategiana.....	37
6.2	Aineiston keruu	38
6.2.1	Havainnointi.....	39
6.2.2	Fokusryhmähaastattelu	39
6.2.3	Kehitystyöpaja.....	40
6.3	Aineiston analysointi	40
7	Toimeksiantaja ja toimintatutkimuksen rakenne.....	42
7.1	Toimeksiantajan kuvaus	43
7.2	Kehitystyön syklien kuvaus.....	44
7.2.1	Sykli 1 kuvaus: Lähtötilanteen kartoitus ja suunnittelu.....	45
7.2.2	Sykli 2 kuvaus: Ensimmäinen käytännön toteutus	46
7.2.3	Sykli 3 kuvaus: Prosessin vakiinnuttaminen	47
8	Syklien toteutus ja tulokset	48
8.1	Sykli 1 – Lähtötilanteen kartoitus ja suunnittelu.....	48
8.1.1	Tiedon hankinta	48
8.1.2	Kvartaalitestauksen lähtötilanteen aikataulu ja toteutus	49
8.1.3	Fokusryhmähaastattelun valmistelu.....	51
8.1.4	Fokusryhmähaastattelun tulokset	52
8.1.5	Fokusryhmähaastattelusta vedetyt johtopäätökset.....	54
8.2	Sykli 2 – Ensimmäinen käytännön toteutus	55
8.2.1	Scrumban-koulutus.....	55
8.2.2	Testaussuunnitelman kehitystyöpaja	56
8.2.3	Testitapausten päivittämisen kehitystyöpaja	56
8.2.4	E2E- ja tiimien välinen testauksen suunnittelu.....	57
8.2.5	Testausprosessin organisointi ja seuranta.....	58
8.2.6	Q2-testauksen retrospektiivin tulokset	59
8.3	Sykli 3 – Prosessin vakiinnuttaminen	60
8.3.1	Testausprosessin organisointi ja seuranta.....	60
8.3.2	Q3-testauksen retrospektiivin tulokset	60
8.3.3	Q4-testauksen retrospektiivin tulokset	63
9	Analyysi ja tulkinta.....	65
9.1	Fokusryhmähaastattelu.....	65
9.2	Retrospektiivit Q2–Q4	66

10	Pohdinta	68
10.1	Luotettavuuden ja eettisyyden arviointi.....	70
10.2	Oman osaamisen kehittyminen suhteessa Arenen kompetensseihin.....	70
10.3	Jatkokehitysmahdollisuudet	73
	Lähteet	74
	Liitteet	

1 Johdanto

Tietojärjestelmät ovat keskeisessä roolissa organisaatioiden toiminnassa, tämän vuoksi niiden toimintavarmuus ja tehokkuus ovat olennaisia päivittäisten prosessien sujuvuuden kannalta. Koska ohjelmistot ja järjestelmät kehittyvät jatkuvasti, ne vaativat säännöllisiä päivityksiä. Jotta päivitykset eivät vaarantaisi järjestelmän toimintaa, on ohjelmistotestauksen ensisijaisena tavoitteena varmistaa, että järjestelmä täyttää sille asetetut toiminnalliset ja ei-toiminnalliset vaatimukset sekä toimii suunnitellusti käyttötarkoituksessaan. (Kasurinen, 2013, 12.)

Ketterät menetelmät tarjoavat tehokkaita keinoja testausprosessin selkeyttämiseen ja parantamiseen. Ketterissä menetelmissä korostuvat joustavuus ja jatkuva kehittäminen. (Smith & Hayes, 2024.) Ketteryyden merkitys näkyy vahvasti suomalaisessa ohjelmistokehityksessä. Nitorin ja Helsingin yliopiston toteuttaman kyselyn mukaan valtaosa organisaatioista suhtautuu ketteriin menetelmiin myönteisesti ja suosituimpia menetelmiä ovat Scrum ja Kanban. (Laanti, 2019.)

HUS Helsingin yliopistollinen sairaala on Suomen suurin erikoissairaanhoidon toimija, ja sen henkilöstöjohdon HR-järjestelmät ovat keskeisessä roolissa organisaation päivittäisessä toiminnassa. HR-järjestelmien avulla hallinnoidaan muun muassa rekrytointia, työvuorosuunnittelua ja palkanlaskentaa, joten niiden luotettava ja tehokas toiminta on välttämätöntä. HUSin henkilöstöhallinnon ja tietohallinnon kehityksessä on tunnistettu tarve parantaa kvartaalitestauksen prosesseja, jotta järjestelmän laatu, käytettävyys ja toimintanopeus voidaan optimoida. Lähtötilanteessa tiimillä ei ollut yhtenäistä toimintamallia kvartaalitestaukseen ja palaverikäytännöt olivat vähäisiä sekä vaihtelevia. Vaikka kvartaalitestauksella on laajempia vaikutuksia muihin järjestelmiin, yhteistyö muiden sidosryhmien kanssa oli melko vähäistä. Testitapausten luomiseen, ajamiseen ja seurantaan käytetään tehtävienhallintaohjelmistoa, mutta Kanban-taulua ei ollut käytössä ja testauksen dokumentointia ei ollut.

Tämän kehittämistyön tarkoituksena on kehittää HCM Core -järjestelmän kvartaalitestausprosessia hyödyntämällä Scrumban-menetelmää. Laadunvarmistuksen kehittäminen on keskeistä, sillä HR-järjestelmät tukevat organisaation henkilöstöhallinnon toimintaa ja päätöksentekoa, joka vaikuttaa suoraan koko organisaation toimivuuteen. Kvartaalipäivitysten testaamisella on myös laajempia vaikutuksia muihin järjestelmiin, ja testaamiseen osallistuu eri sidosryhmiä. Tavoitteena on kehittää yhtenäinen toimintamalli ketterän viitekehyksen avulla kvartaalitestauksen hallintaan ja seurantaan, osallistaa sidosryhmät aktiivisemmin testausprosessiin sekä yhdenmukaistaa ne työvaiheet, jotka pysyvät samoina kvartaalipäivityksestä riippumatta. Lisäksi työn tavoitteena on

selkeyttää työskentelyprosesseja ja vähentää hukkaa, jotta testauksen tehokkuus ja luotettavuus paranevat.

Kehittämistyö toteutettiin toimintatutkimuksena, jossa keskeistä oli pyrkimys pysyvään toiminnan kehittämiseen. Tutkimuksessa korostui koko tiimin osallistuminen kehitystyöhön, ja tutkija oli itse osa tutkimuskohteena olevaa tiimiä. Tutkimusmenetelminä käytettiin fokusryhmähaastattelua, havainnointia, kehitystyöpajoja sekä retrospektiiveistä nousseita kehitystarpeita. Aineiston analyysissä hyödynnettiin laadullista sisällönanalyysiä, joka sisälsi kerätyn datan teemoittelun ja johtopäätösten tekemisen.

Tutkimuskysymykset olivat:

1. Miten Scrumban-menetelmän käyttö vaikuttaa tiimityön tehokkuuteen ja yhteistyöhön kvartaalitestauksessa?
2. Miten Scrumban-menetelmän käyttö vaikuttaa kvartaalitestauksen prosessin tehokkuuteen ja hallittavuuteen?

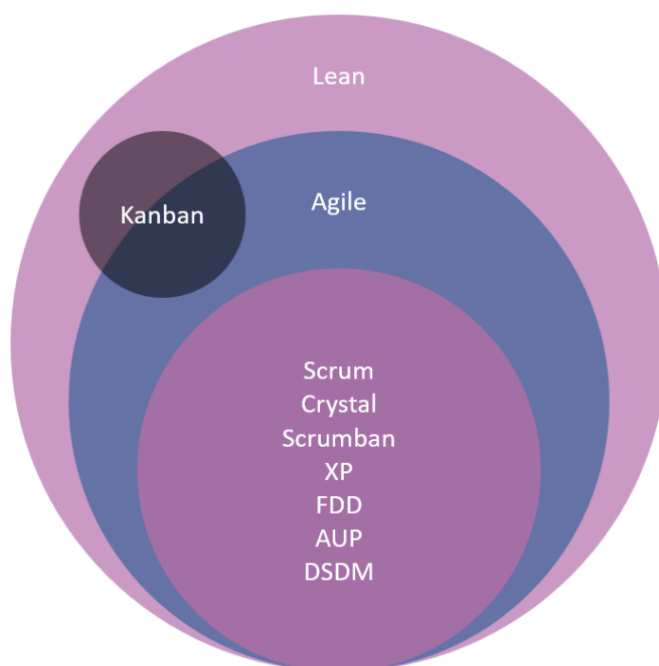
Teoriaviitekehys muodostuu viidestä pääluvusta, jotka käsittelevät ketteriä menetelmiä ja Lean-ajattelua, ohjelmistotestausta sekä tutkimusstrategiaa ja -menetelmiä. Ketterien menetelmien osuudessa tarkastellaan erityisesti Scrum-, Kanban- ja Scrumban-malleja sekä niiden käytännön soveltamista. Ohjelmistotestausta käsittelevässä luvussa kuvataan testauksen periaatteet, vaiheet ja keskeiset käytännöt ketterässä ohjelmistokehityksessä. Tutkimusmenetelmiä koskevassa osiossa esitellään toimintatutkimus tutkimusstrategiana sekä käytetyt aineistonkeruumenetelmät, joihin kuuluvat havainnointi, fokusryhmähaastattelu ja kehitystyöpajat, sekä aineiston analysoinnin periaatteet. Lisäksi työssä kuvataan toimeksiantajan konteksti ja toimintatutkimuksen rakenne sekä kehitystyön kolme sykliä, joiden varaan tutkimusprosessi rakentui. Lopuksi esitetään toimintatutkimuksen syklien toteutus ja tulokset sekä tutkimuksen analyysi, tulkinta ja pohdinta.

2 Ketterän kehittämisen taustat

Ketteriä menetelmiä voidaan pitää vastakohtana perinteiselle, kankealle vesiputousmallille. Niille on ominaista työn jakaminen sprintteihin eli pienempiin, hallittaviin kokonaisuuksiin. Ketterissä menetelmissä ohjelmiston vaatimuksia arvioidaan ja tarkennetaan jatkuvasti sprinttien aikana. (Juvonen, 2018, 17.)

Ketteryyden ytimessä on Agile Manifesti, joka julkaistiin vuonna 2001 (Haikala & Mikkonen, 2011, 8–9). Se toi mukanaan neljä keskeistä arvoa ja 12 periaatetta, jotka ohjaavat ketterää ajattelua (Beck ym., 2001). Tunnetuimpia ketteriä menetelmiä ovat Scrum, Kanban, Extreme Programming ja Lean Software Development (Agile Practice Guide, 2017).

Lean-ajattelu toimii usein kattoterminä ketterille menetelmille. Alun perin Toyotan tuotantofilosofiasta lähtöisin oleva Lean keskittyy hukkan minimointiin, asiakasarvon tuottamiseen ja jatkuvaan kehittämiseen. Ketterät menetelmät ovat omaksuneet monia Leanin periaatteita, kuten iteraatiivisen kehityksen, tiimien itseohjautuvuuden ja asiakasarvon korostamisen. Yhdessä Lean ja ketterät menetelmät muodostavat vahvan perustan nykyaikaiselle, ketterälle organisaatiolle, joka kykenee reagoimaan nopeasti muutoksiin ja tuottamaan arvoa tehokkaasti. (Agile Practice Guide, 2017.) Agile Practice Guide (2017) Kuva 1 havainnollistaa, että Agile ja Kanban nähdään Lean-menetelmien alaryhminä.



Kuva 1. Agile toimii kattoterminä useille ketterille menetelmille (Agile Practice Guide, 2017)

2.1 Agile Manifesti

Vuonna 2001 Utahissa kokoontui joukko kevyiden menetelmien kehittäjiä ja kannattajia keskustelemaan nykyisistä käytännöistään. He halusivat vastapainoa raskaammille kehitysprosesseille ja päättivät nimetä nämä ketteriksi (Agile) menetelmiksi. Kahden päivän tapaamisessa sovittiin ketteriä menetelmiä yhdistävistä periaatteista ja perustettiin Agile Alliance -järjestö edistämään näitä menetelmiä. (Haikala & Mikkonen, 2011, 8–9.)

Vuonna 2001 julkaistu Agile Manifesti pitää sisällään neljä pääarvoa ja 12 peruseriaatetta.

Pääarvot ovat:

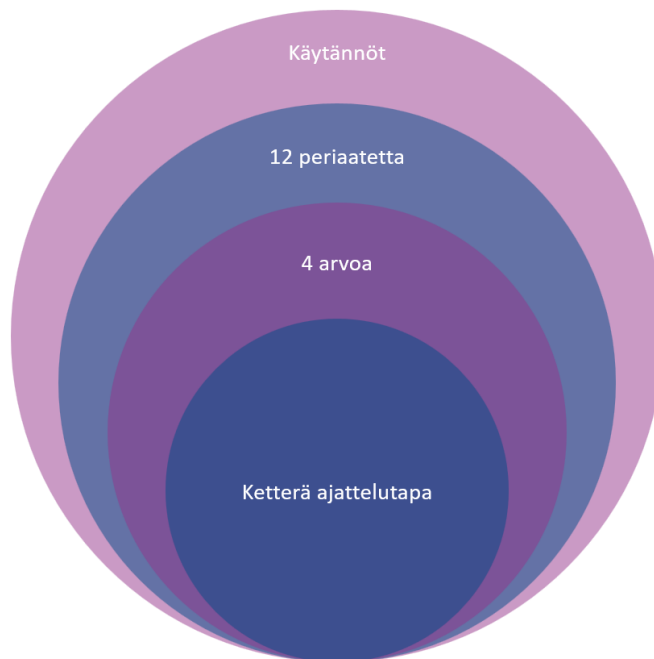
1. Yksilöitä ja vuorovaikutusta enemmän kuin prosesseja ja työkaluja.
2. Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota.
3. Yhteistyötä asiakkaan kanssa enemmän kuin sopimuksen neuvottelua.
4. Muutoksiin reagointia enemmän kuin suunnitelman seuraamista. (Beck ym., 2001.)

Agile Manifestin 12 peruseriaatetta:

1. Asiakastyytyväisyys on tärkein tavoitteemme aikaisen ja jatkuvan ohjelmistotoimituksen arvon avulla.
2. Toivotamme tervetulleiksi muuttuvat vaatimukset myös projektin myöhäisessä vaiheessa, koska ne auttavat tekemään tuotteesta entistä kilpailukykyisemmän asiakkaalle.
3. Toimitamme säännöllisesti toimivaa ohjelmistoa, muutaman viikon tai kuukauden välein, mahdollisimman tiiviillä aikavälillä.
4. Yhteistyö liiketoiminnan ja kehityksen välillä on päivittäistä koko projektin ajan.
5. Tiimi koostuu motivoituneista osaajista. Rakennamme projektit sitoutuneiden ammattilaisten ympärille ja tarjoamme heille tarvittavan tuen sekä luottamuksen tehtävien suorittamiseen.
6. Kasvokkain kommunikointi on tehokkain ja selkein tapa jakaa tietoa kehitystiimissä.
7. Kehityksen tärkeimpiä mittareita ovat toimiva ohjelmisto.
8. Ketterät menetelmät tukevat kestävästä kehitystä. Projektin eri osapuolten tulisi pystyä työskentelemään tasaisella tahdilla määrättömän ajan.
9. Erinomainen tekninen laatu ja hyvä suunnittelu lisäävät ketteryyttä.
10. Yksinkertaisuuteen keskittyminen eli pyrkimys maksimoida työn määrä, joka jätetään tekemättä.
11. Parhaat suunnitelmat ja arkkitehtuurit syntyvät itseohjautuvissa tiimeissä.

12. Tiimi arvioi säännöllisesti omaa toimintaansa ja tekee parannuksia kehittäkseen työkentelyään entistä tehokkaammaksi. (Beck ym., 2001.)

Ketteryyden ajattelutapa, arvot ja periaatteet muodostavat perustan ketterille toimintatavoille. Nykyisin käytössä olevien ketterien menetelmien taustalla on tämän ajattelutavan, arvojen ja periaatteiden kanssa. Tämä yhteys on esillä Kuva 2. (Agile Practice Guide, 2017.)



Kuva 2. Agile Manifestin arvojen, periaatteiden ja yleisten käytäntöjen välinen suhde (Agile Practice Guide, 2017)

2.2 Lean-periaatteet ja historia

Lean on johtamismalli, jonka tavoitteena on maksimoida arvo asiakkaalle virtauksen maksimoinnilla samalla minimoiden hukkaa (Mäenpää, N.d.). Hukkaa ovat mm. odottelu, turha käsittely, virheet ja käyttämättä jätetty työntekijän luovuus (Vuorinen, 2013, 72). Vaikka Lean usein mielletään pelkästään hukan poistoon keskittyväksi menetelmäksi, sen varsinainen tarkoitus on parantaa prosessien nopeutta lyhentämällä läpimenoaikaa. Tämä on ratkaisevaa, sillä ilman läpimenoajan selkeää lyhenemistä taloudellisia hyötyjä ei yleensä synny. (Gebeyehu ym., 2022)

Leanin juuret ovat Toyotan tuotantojärjestelmässä, jonka kehitti Taichi Ohno toisen maailmansodan jälkeisessä Japanissa. Ohno yhdisti Fordilta, amerikkalaisilta supermarketeilta ja saksalaisilta

tuotantolinjoilta oppimiaan ideoita luodakseen tehokkaampia tuotantotapoja, kuten Kanban. Toyota otti oppia myös W.E. Demingin ja J.M. Juranin laatuopetuksista, mikä auttoi yritystä rakentamaan tehokkaan ja laadukkaan tuotantokulttuurin. (Sixsigma, n.d.)

Lean-johtamisen ytimessä on työn sujuvan virtauksen mahdollistaminen. Kolme keskeistä virtauksen vihollista ovat vaihtelu, ylikuormitus ja hukka. Näiden poistaminen on väline, jolla virtauksen päämäärään voidaan päästä. (Torkkola, 2015, 23.) Vaihtelu on keskeisin näistä kolmesta tekijästä, sillä se luo pohjan ylikuormitukselle ja hukalle. Esimerkiksi asiantuntijatyössä vaihtelua ilmenee eri osaamistasoissa, epätasaisessa työkuormassa ja organisaation omien toimintamallien synnyttämissä kiirejaksoissa. Vaihtelu voidaan tunnistaa mittaamalla läpimenoaikoja tai odotusaikoja. Ylikuormitus puolestaan viittaa siihen, että laite, järjestelmä tai henkilö joutuu työskentelemään yli kapasiteettinsa. Tämä johtaa uupumiseen, poissaoloihin ja kehittämismahdollisuuksien heikkenemiseen. Ylikuormituksen mittaamiseen sopii työn saapumisnopeuden ja valmistumisnopeuden vertailu. Hukka tarkoittaa kaikkia niitä toimintoja, jotka eivät tuota asiakkaalle lisäarvoa. (Torkkola, 2015, 23–25.) Leanin onnistunut soveltaminen edellyttää valmentavaa johtamista, jatkuvaa oppimista, toiminnan seuraamista mittareiden avulla, tiimityötä, viestintää visuaalisilla tekniikoilla sekä organisaation kulttuurin arvojen ja piirteiden tunnistamista. (Maijala, 2019)

3 Ketterät menetelmät

Ketterän kehyksen viitekehyksiä on useita, mutta tässä työssä tarkastellaan kolmea keskeistä lähestymistapaa: Scrum, Kanban ja näiden yhdistelmää Scrumban. Nämä menetelmät valittiin mukaan erityisesti testauksen näkökulmasta. Scrumista tarkastellaan erityisesti palaverikäytäntöjä, jotka tukevat tiimityötä ja jatkuvaa palautetta. Kanbanin osalta keskitytään työn visualisointiin ja läpimenoaikojen mittaamiseen, jotka auttavat seuraamaan testauksen etenemistä ja tunnistamaan mahdollisia pullonkauloja. Scrumban puolestaan yhdistää näiden kahden menetelmän vahvuudet, tarjoten joustavan mutta rakenteellisen tavan hallita työtä.

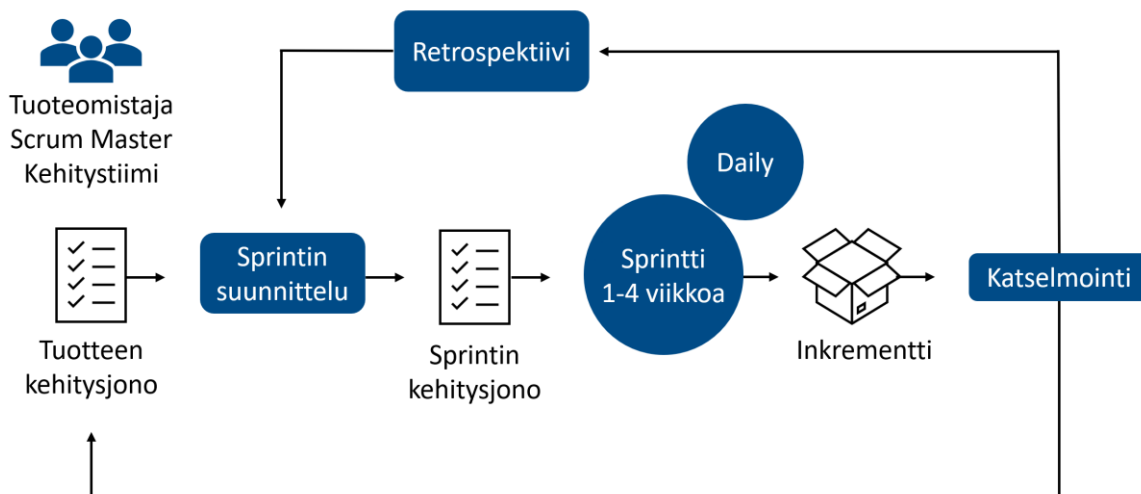
3.1 Scrum

Scrum on ketterä projektinhallintamenetelmä, joka on saavuttanut laajaa suosiota sekä ohjelmistoprojekteissa että liike-elämässä sen yksinkertaisen ja tehokkaan luonteensa ansiosta (Kasurinen, 2013, 28). Scrumin perustana on empirismi ja Lean-ajattelu. Empirismin mukaan tieto perustuu kokemukseen ja Lean hukun vähentämiseen. Scrumille on ominaista iteratiivinen ja inkrementaalinen toimintatapa. (Schwaber & Sutherland, 2020, 4.)

Scrum ei ole pelkästään projektinhallintamenetelmä, vaan myös tapa kehittää tiimin toimintaa ja vuorovaikutusta. Toistuvat sprintit ja retrospektiivit tarjoavat rakenteen, jossa tiimi voi jatkuvasti oppia ja parantaa työskentelytapojaan. Tämä tukee sekä yksilöiden että koko tiimin kehittymistä. Scrumin avulla voidaan luoda avoin ja läpinäkyvä työympäristö, jossa palautteen antaminen ja vastaanottaminen on osa arkea. Tällainen kulttuuri edistää luottamusta ja yhteistä vastuunkantoa, mikä näkyy usein myös parempina tuloksina. (Schwaber & Sutherland, 2020, 3–4.)

Kuva 3 esitetään Scrumin keskeiset osatekijät: tiimin roolit, prosessin tapahtumat ja tuotokset. Kuvan avulla hahmotetaan, miten Scrum rakentuu kolmesta roolista: tuoteomistajasta (Product Owner), Scrum Masterista ja kehitystiimistä (Development Team), jotka yhdessä vastaavat tuotteen kehittämisestä. Prosessi etenee toistuvissa sprinteissä, jotka sisältävät tapahtumat: suunnittelupalaverin, päivittäispalaverin Dailyn, katselmoinnin ja retrospektiivin. Kuvassa näkyvät myös Scrumin tuotokset: tuotteen kehitysjojo (Product Backlog), sprintin kehitysjojo (Sprint Backlog) ja inkrementin (Increment), jotka ohjaavat yhteistyötä ja varmistavat läpinäkyvyyden. Visuaalinen

esitys auttaa ymmärtämään, miten nämä elementit muodostavat yhtenäisen ja iteratiivisen kehitysmallin, jossa tiimi voi jatkuvasti oppia ja parantaa toimintaansa. (Niemi & Hietaniemi, 2020, 20.)



Kuva 3. Scrum-viitekehys (The Home of Scrum, 2023)

Seuraavissa alaluvuissa tarkastellaan Scrumin rakennetta tarkemmin. Ensiksi esitellään tiimin roolit ja vastuut, sen jälkeen käydään läpi menetelmän keskeiset tapahtumat ja lopuksi kuvataan Scrumin tuotokset, jotka ohjaavat kehitystyötä ja varmistavat sen läpinäkyvyyden.

3.1.1 Scrum-tiimi

Scrum-tiimi koostuu kolmesta roolista, jotka ovat kehitystiimi (Development Team), tuoteomistaja (Product Owner) sekä Scrum Master. Tämä tiimi on hierarkiaton ja itseohjautuva enintään 10 jäsenen tiimi, joka keskittyy yhteen tavoitteeseen kerrallaan. Tämä tavoite tunnetaan nimellä tuotteen tavoite. (Schwaber & Sutherland, 2020, 6.)

Kehitystiimi on Scrum-kehityksen keskeinen osa, jonka vastuulla on käyttökelpoisen inkrementin tuottaminen jokaisessa sprintissä. Kehittäjiltä edellytetään monipuolista osaamista, joka vaihtelee toimialoittain ja hankkeittain. Heidän vastuuseen määritellään sprintin suunnitelman laatiminen, laadun varmistaminen valmiin määritelmää noudattaen, suunnitelman mukauttaminen sprintin tavoitteen saavuttamiseksi sekä ammatillisen vastuun kantaminen tiimin sisällä. (Schwaber & Sutherland, 2020, 6.)

Kehitystiimi toimii itseohjautuvasti ja määrittelee itsenäisesti, miten asetetut tavoitteet saavutetaan. Tiimin toiminta ei kuitenkaan tapahdu eristyksissä, vaan yhteistyö sidosryhmien, kuten loppukäyttäjien, muiden organisaatioiden ja kehitystiimien kanssa on olennainen osa prosessin onnistumista. (Niemi & Hietaniemi, 2020, 21.)

Tuoteomistaja vastaa siitä, että Scrum-tiimin työ tuottaa mahdollisimman paljon arvoa organisaatiolle ja sen sidosryhmille. Hän määrittelee tuotteen vision ja kehityssuunnan sekä hallinnoi kehitysjonoa siten, että se tukee liiketoiminnan tavoitteita. Keskeisiin tehtäviin sisältyy tuotteen tavoitteen määrittely ja sen viestiminen, kehitysjonon sisällön valmistelu, järjestäminen ja ylläpito sekä sen läpinäkyvyyden varmistaminen. (Niemi & Hietaniemi, 2020, 21; Schwaber & Sutherland, 2020, 7.)

Vaikka tehtäviä voidaan delegoida, vastuu niiden toteuttamisesta säilyy tuoteomistajalla. Organisaatioiden odotetaan tukevan häntä ja kunnioittavan hänen päätöksiään, sillä niiden avulla ohjataan kehitystyötä ja vaikutetaan sprinttien tuloksiin. Lisäksi tuoteomistaja toimii linkkinä Scrum-tiimin ja ulkoisten sidosryhmien välillä varmistaen, että kehitystyö tuottaa lisäarvoa loppukäyttäjille ja muille osallisille. (Schwaber & Sutherland, 2020, 7.)

Scrum Master on vastuussa Scrumin toteutuksesta ja sen periaatteiden oikeaoppisesta soveltamisesta. Hänen roolinsa perustuu palvelujohtajuuteen, jossa tiimiä tuetaan esimerkiksi esteiden poistamisella, itseohjautuvuuteen valmentamisella ja Scrumin tapahtumien tarkoituksenmukaisella fasilitoinnilla. Näin varmistetaan tiimin keskittyminen arvon tuottamiseen ja valmiin määritelmää noudattavien inkrementtien kehittämiseen. (Niemi & Hietaniemi, 2020, 21; Schwaber & Sutherland, 2020, 7–8.)

Lisäksi Scrum Master tukee tuoteomistajaa muun muassa tuotteen tavoitteen määrittelyssä, kehitysjonon sisällön selkeyttämisessä ja sidosryhmäyhteistyön edistämisessä. Organisaatiotasolla hän toimii kouluttajana ja neuvonantajana Scrumin käyttöönotossa ja kehittämisessä sekä vahvistaa havaintoihin perustuvaa työskentelytapaa kompleksisessa ymärästössä. Roolin kautta varmistetaan myös, että tiimin työskentely sujuu tehokkaasti ja että ongelmat ratkaistaan nopeasti, samalla kun koko organisaatiossa edistetään ketterän toimintakulttuurin vahvistumista. (Schwaber & Sutherland, 2020, 7–8.)

3.1.2 Scrumin tapahtumat

Scrumin tapahtumat muodostavat menetelmän rakenteen, jonka avulla tiimi voi toimia tavoitteellisesti ja ketterästi. Viisi toistuvaa tapahtumaa tarjoavat rakenteen, jossa työ suunnitellaan huolellisesti, edistymistä seurataan päivittäin ja tuloksia arvioidaan sprintin päätteeksi. Tämä rytmi tukee tiimin oppimista ja jatkuvaa kehittymistä. (Niemi & Hietaniemi, 2020, 20.)

Sprintti on Scrumin keskeinen aikarajattu kehitysjakso, jonka aikana Scrum-tiimi tuottaa valmiin ja käyttökelpoisen inkrementin. Sprintin pituus on vakio ja enintään neljä viikkoa, ja uusi sprintti alkaa välittömästi edellisen päätyttyä. Sprintti sisältää kaikki tarvittavat tapahtumat tuotteen tavoitteen saavuttamiseksi: sprintin suunnittelun, päivittäispalaverit, sprintin katselmoinnin ja retrospektiivin. Sprintin aikana ei saa tehdä muutoksia, jotka vaarantavat sprintin tavoitteen, eikä laadusta tingitä. Tuotteen kehitysjonoa voidaan kuitenkin jalostaa ja sprintin sisältöä tarkentaa yhteistyössä tuoteomistajan kanssa ymmärryksen karttuessa. (Schwaber & Sutherland, 2020, 8–9.)

Sprintit tukevat ennustettavuutta ja jatkuvaa oppimista, sillä etenemistä tarkastellaan ja mukautetaan vähintään kerran kuukaudessa. Lyhyemmät sprintit vähentävät riskejä ja mahdollistavat nopeamman palautteen saamisen. Jokainen sprintti voidaan nähdä pienenä projektina, jolla on oma tavoite ja toteutussuunnitelma. Vaikka etenemistä voidaan seurata erilaisilla työkaluilla, kuten kertymäkuvaajilla, Scrum perustuu empirismiin: päätöksenteko perustuu siihen, mitä on tapahtunut. Sprintti voidaan keskeyttää vain, jos sen tavoite muuttuu tarpeettomaksi, ja tällöin ainoastaan tuoteomistajalla on oikeus tehdä keskeytyspäätös. (Niemi & Hietaniemi, 2020, 20–21; Schwaber & Sutherland, 2020, 9.)

Sprintin suunnittelu käynnistää jokaisen sprintin ja määrittää sen tavoitteen, sisällön ja toteutustavan. Koko Scrum-tiimi osallistuu suunnitteluun ja tuoteomistaja varmistaa, että osallistujat ovat valmistautuneet keskustelemaan tuotteen kehitysjonon tärkeimmistä kohdista suhteessa tuotteen tavoitteeseen. Tarvittaessa suunnitteluun voidaan kutsua myös muita asiantuntijoita. Sprintin suunnittelussa vastataan kolmeen keskeiseen kysymykseen: miksi sprintti on arvokas, mitä voidaan saada valmiiksi ja miten työ saadaan tehtyä. (Schwaber & Sutherland, 2020, 8–9.)

Ensimmäisessä vaiheessa tuoteomistaja ehdottaa, miten tuotteen arvoa voidaan kasvattaa sprintin aikana ja tiimi määrittelee yhdessä sprintin tavoitteen. Tämän jälkeen kehittäjät valitsevat kehitysjonosta ne kohdat, jotka voidaan toteuttaa sprintin aikana ja suunnittelevat työn toteutuksen siten, että se täyttää valmiin määritelmän. Työn suunnittelutapa on kehittäjien päätettävissä

eikä ulkopuolisilla ole oikeutta sanella toteuttamistapaa. Sprintin tavoite, valitut kehitysjohto-kohdat ja suunnitelma niiden toteuttamiseksi muodostavat yhdessä sprintin kehitysjonon. Sprintin suunnittelun enimmäiskesto on kahdeksan tuntia kuukauden mittaisessa sprintissä ja lyhyemmissä sprinteissä suunnitteluun käytettävä aika on vastaavasti lyhyempi. (Niemi & Hietaniemi, 2020, 20–21; Schwaber & Sutherland, 2020, 9.)

Päivittäispalaveri on 15 minuutin mittainen joka arkipäivä toistuva tapahtuma, jossa kehittäjät tarkastelevat etenemistään kohti sprintin tavoitetta ja mukauttavat suunnitelmaansa tarvittaessa. Palaveri pidetään aina samaan aikaan ja samassa paikassa sprintin aikana mikä vähentää kompleksisuutta ja tekee rutiinia. Vaikka tapahtuma on tarkoitettu ensisijaisesti kehittäjille, myös tuoteomistaja ja Scrum Master voivat osallistua siihen kehittäjän roolissa, mikäli he osallistuvat aktiivisesti sprintin toteutukseen. (Schwaber & Sutherland, 2020, 10.)

Kehittäjät saavat itse päättää päivittäispalaverin rakenteen ja toteutustavan, kunhan se keskittyy sprintin tavoitteen saavuttamiseen ja tuottaa selkeän suunnitelman tulevalle työpäivälle. Tapahtuma edistää tiimin keskittymistä, lisää itseohjautuvuutta ja parantaa vuorovaikutusta. Päivittäispalaveri auttaa tunnistamaan esteitä ja tukee nopeaa päätöksentekoa, mikä vähentää muiden kokousten tarvetta. Vaikka palaveri on tärkeä suunnittelun väline, kehittäjät voivat käydä päivän aikana myös tarkempia keskusteluja työn etenemisestä ja suunnitelmien mukauttamisesta. (Kasurinen, 2013, 28; Schwaber & Sutherland, 2020, 10.)

Sprintin katselmointi on sprintin lopussa järjestettävä työpajamainen tapahtuma, jonka tarkoituksena on tarkastella sprintin aikana saavutettuja tuloksia ja arvioida tuotteen kehityksen suuntaa. Scrum-tiimi esittelee työnsä tulokset tärkeimmille sidosryhmille ja yhdessä keskustellaan edistymisestä kohti tuotteen tavoitetta. Katselmoinnissa tarkastellaan myös mahdollisia muutoksia tuotteen toimintaympäristössä ja näiden pohjalta voidaan tehdä päätöksiä kehitysjärjestyksen muuttamisesta tai uusien mahdollisuuksien hyödyntämisestä. (Schwaber & Sutherland, 2020, 7–8.)

Sprintin katselmointi ei ole pelkkä esittelytilaisuus vaan vuorovaikutteinen keskustelu, jonka tavoitteena on lisätä läpinäkyvyyttä ja ohjata tuotteen jatkokehitystä. Tapahtuman enimmäiskesto on neljä tuntia kuukauden mittaisessa sprintissä ja lyhyemmissä sprinteissä myös katselmoinnin kesto on yleensä lyhyempi. Katselmointi tarjoaa mahdollisuuden saada palautetta aidossa käytöympäristössä ja ohjata kehitystä kohti sidosryhmien tarpeita. (Niemi & Hietaniemi, 2020, 20; Schwaber & Sutherland, 2020, 8.)

Sprintin retrospektiivi on sprintin päättävä tapahtuma, jonka tarkoituksena on tunnistaa keinoja tiimin laadun ja tehokkuuden parantamiseksi. Scrum-tiimi tarkastelee sprintin kulkua kokonaisvaltaisesti huomioiden ihmiset, yhteistyön, prosessit, työkalut sekä valmiin määritelmän toteutumisen. Tavoitteena on tunnistaa onnistumiset, kohdatut haasteet ja ratkaisematta jääneet ongelmat. Lisäksi retrospektiivissä pyritään tunnistamaan harhaanjohtavat oletukset ja niiden taustalla olevat syyt. (Schwaber & Sutherland, 2020, 10.)

Retrospektiivin lopputuloksena Scrum-tiimi valitsee vaikuttavimmat parannusehdotukset, jotka pyritään toteuttamaan mahdollisimman pian, usein jo seuraavan sprintin aikana. Näitä tapauksia voidaan lisätä suoraan sprintin kehitysjonoon. Tapahtuman enimmäiskesto on kolme tuntia kuukauden mittaisessa sprintissä ja lyhyemmissä sprinteissä retrospektiivin kesto on vastaavasti lyhyempi. Retrospektiivi tukee jatkuvaa oppimista ja tiimin itseohjautuvuuden kehittymistä ja se on keskeinen osa Scrumin iteratiivista ja inkrementaalista luonnetta. (Niemi & Hietaniemi, 2020, 20; Schwaber & Sutherland, 2020, 10.)

3.1.3 Scrumin tuotokset

Scrumin tuotoksia ovat tuotteen kehitysjoono, sprintin kehitysjoono sekä inkrementti (Schwaber & Sutherland, 2020, 10; Scrum.org, 2023.c). Niiden ensisijaisena tarkoituksena on lisätä läpinäkyvyyttä, jotta kaikilla tuotoksia tarkastelevilla on yhtenäinen käsitys työn tilasta ja samat lähtökohdat niiden mukauttamiseen. (Schwaber & Sutherland, 2020, 10.)

Tuotteen kehitysjoono (Product Backlog) määritellään järjestetyksi ja jatkuvasti kehittyväksi listaksi kaikista asioista, joita tarvitaan tuotteen parantamiseksi (Scrum.org, 2023b). Se toimii ainoana lähteenä Scrum-tiimin työtehtäville ja sisältää sekä pitkän aikavälin tavoitteita että konkreettisia kehityskohtia. Kehitysjonon kohdat, jotka voidaan toteuttaa yhden sprintin aikana valmiin määritelmän mukaisesti, katsotaan valmistelluiksi ja ne voidaan valita sprintin suunnittelussa toteutettaviksi. Kehitysjonon jalostamista pidetään jatkuvana toimintana, jossa kohteita pilkotaan, tarkennetaan ja arvioidaan yhteistyössä tuoteomistajan kanssa. Työmäärän arvioinnista vastaavat kehittäjät, kun taas tuoteomistajan tehtävänä on tarjota päätöksenteon tueksi vaihtoehtoja ja näkökulmia. (Schwaber & Sutherland, 2020, 11.)

Tuotteen kehitysjoono on kiinteästi sidottu tuotteen tavoitteeseen (Product Goal), joka kuvaa tuotteen tulevaa tilaa ja toimii tiimin pitkän aikavälin päämääränä. Tavoite sisällytetään kehitysjonoon

ja se ohjaa sen sisältöä. Nykyinen tavoite on saavutettava tai hylättävä ennen kuin seuraava tavoite voidaan asettaa. Tuotteen tavoitteella tuetaan suunnittelua, fokusta ja arvon tuottamista ja se luo koko tiimille yhteisen suunnan. (Schwaber & Sutherland, 2020, 11.)

Sprintin kehitysjono koostuu kolmesta elementistä: sprintin tavoitteesta (Sprint Goal), sprinttiin valituista tuotteen kehitysjono-kohdista sekä suunnitelmasta niiden toteuttamiseksi. Sprintin kehitysjono muodostaa kehittäjien itselleen laatiman suunnitelman ja toimii reaaliaikaisena kuvauksena sprintin aikana tehtävästä työstä. Sitä tarkennetaan jatkuvasti uuden ymmärryksen karttuessa, ja sen on oltava riittävän yksityiskohtainen, jotta tiimi voi arvioida edistymistään päivittäisissä Scrum-tapahtumissa. (Schwaber & Sutherland, 2020, 11–12.)

Sprintin kehitysjono on sidottu sprintin tavoitteeseen, joka määrittää sprintin suunnittelussa ja joka ohjaa tiimin toimintaa koko sprintin ajan. Tavoite luo keskittymistä ja johdonmukaisuutta sekä kannustaa tiimiä työskentelemään yhteisen päämäärän eteen. (Schwaber & Sutherland, 2020, 11–12.) Kehittäjät sitoutuvat sprintin tavoitteeseen, mutta samalla heille jätetään joustavuutta sen saavuttamiseen tarvittavan työn suhteen. Mikäli työ osoittautuu erilaiseksi kuin suunniteltiin, sprintin kehitysjonoa voidaan mukauttaa yhteistyössä tuoteomistajan kanssa, kunhan sprintin tavoite ei vaarannu. (Scrum.org, 2023d.)

Inkrementti on konkreettinen lisäys tuotteeseen, joka vie tiimiä lähemmäksi tuotteen tavoitetta. Jokainen inkrementti rakentuu aiempien päälle, ja niiden yhteistoimivuus on varmistettava kattavasti. Inkrementillä on arvoa vain, jos se on käyttökelpoinen ja täyttää valmiin määritelmän (Definition of Done). (Schwaber & Sutherland, 2020, 12.) Sprintin aikana voidaan tuottaa useita inkrementtejä, ja ne esitellään sprintin katselmoinnissa. Arvon julkaisemista ei kuitenkaan tarvitse odottaa katselmointiin asti, vaan inkrementti voidaan toimittaa sidosryhmille heti sen valmistuttua. (Scrum.org, 2023a.)

Inkrementti kytkeytyy kiinteästi valmiin määritelmään, joka on muodollinen kuvaus siitä, milloin työ voidaan katsoa valmiiksi. Valmiin määritelmä luo läpinäkyvyyttä ja yhteisen ymmärryksen tiimin sisällä. Jos kehitysjonon kohta ei täytä määritelmää, sitä ei voida julkaista eikä esitellä sprintin katselmoinnissa, vaan se palautetaan kehitysjonoon jatkokäsittelyä varten. Mikäli valmiin määritelmä on organisaation standardi, sitä on noudatettava kaikkien Scrum-tiimien toimesta. Muussa tapauksessa tiimin tulee itse määritellä ja hyväksyä yhteinen laatuvaatimus, johon sitoudutaan. (Schwaber & Sutherland, 2020, 12.)

3.2 Kanban

Kanban on visuaalinen työnhallintamenetelmä, joka on alun perin kehitetty Toyotan autotehtailla varaston hallintaan eri tuotantovaiheissa, mutta nykyään sitä sovelletaan laajasti myös tietotyössä, kuten ohjelmistokehityksessä ja palveluprosesseissa. Sen avulla pyritään optimoimaan tuotantoa välttämällä suuria varastoja ja varmistamalla työvaiheiden sujuvuus. (Niemi & Hietaniemi, 2020, 15; Juvonen, 2018, 23–24; Torkkola, 2015, 62.) Kanban perustuu Lean-ajatteluun ja sen keskeisiä periaatteita ovat työn kulun visualisointi, läpimenoajan mittaaminen ja työn alla olevien tehtävien rajoittaminen (Hietaniemi, 2020; Niemi & Hietaniemi, 2020, 18).

Kanbanin hyödyntäminen ohjelmistokehityksessä tarjoaa monia etuja. Menetelmä lisää työn läpinäkyvyyttä, sillä tiimi pystyy seuraamaan reaaliaikaisesti, mitkä tehtävät ovat työn alla ja mitkä valmiita. Lisäksi Kanban tehostaa työnkulkua rajoittamalla keskenteräisen työn määrää ja ohjaamalla virtauksia, mikä edistää prosessin sujuvuutta. Keskittyminen tehtävien loppuunsaattamiseen vähentää virheiden määrää ja parantaa näin työn laatua. Menetelmä tarjoaa myös joustavuutta, sillä tiimi voi nopeasti sopeutua muuttuviin vaatimuksiin ja asiakastarpeisiin. Kanban tukee lopulta yhteistä ymmärrystä prosessista: visuaaliset työvälineet ja selkeät säännöt auttavat tiimiä hahmottamaan työn etenemisen kokonaisuudessaan ja tunnistamaan kehityskohteet. (Torkkola, 2015, 66.)

Vaikka Kanbanissa ei ole etappeja, on hyödyllistä ajoittain pysähtyä arvioimaan työn merkityksellisyyttä, hyödyllisyyttä ja tehokkuutta. Pohtikaa tiimin kesken, mitä he ovat tehneet ja miten, ketä työ on hyödyttänyt, oletteko menossa oikeaan suuntaan pitkällä aikavälillä ja miten voitte parantaa prosesseja. (Niemi & Hietaniemi, 2020, 18.)

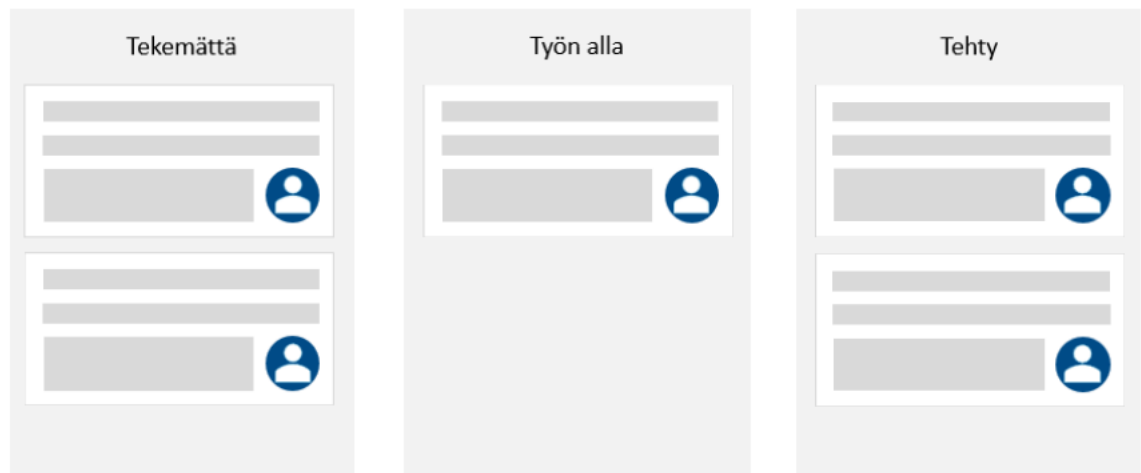
Kanbanin käyttöönotto voi vaikuttaa yksinkertaiselta, mutta käytännössä se vaatii huolellista suunnittelua. Jos esimerkiksi tehtäviä ei siirretä taululla ajantasaisesti, koko järjestelmä menettää merkityksensä. Myös roolien ja vastuiden puuttuminen voi aiheuttaa epäselvyyksiä, etenkin uusissa tai etätiimeissä. Siksi on tärkeää, että tiimi sopii yhdessä, miten taulua käytetään, kuka vastaa sen ylläpidosta ja miten prosessia arvioidaan säännöllisesti. (Torkkola, 2015, 67.)

Kanbanin hyötyjä ja haasteita ohjelmistokehityksessä on tarkasteltu yhdistämällä 20 empiiristä tutkimusta. Tulosten perusteella merkittävimmät hyödyt liittyivät työn näkyvyyden parantamiseen, projektin tehtävien hallinnan tehostumiseen, työn virtauksen sujuvoitumiseen sekä nopeampaan markkinoille pääsyyn. Haasteista keskeisimmäksi nousi organisaatiokulttuuri, joka vai-

kuttaa ratkaisevasti Kanbanin onnistuneeseen käyttöönottoon. Menetelmän soveltaminen edellyttää, että organisaation toimintakulttuuri tukee avoimuutta, jatkuvaa parantamista ja itseohjautuvuutta. Kulttuuriset tekijät voivat joko mahdollistaa tai estää Kanbanin juurtumisen osaksi arjen toimintatapoja, mikä korostuu erityisesti hierarkkisissa tai muutokselle vastahakoisissa rakenteissa. (Dos Santos ym., 2018.)

3.2.1 Työn kulun visualisointi

Työn kulun visualisointi on yksi Kanbanin keskeisistä periaatteista. Sen avulla tiimi saa kokonaiskuvan työn etenemisestä ja siitä, missä vaiheessa eri tehtävät ovat prosessissa. Visualisointi toteutetaan yleensä Kanban-taulun avulla, jossa työ jaetaan vaiheisiin, kuten *Tekemättä*, *Työn alla* ja *Valmis*. Jokainen tehtävä esitetään omana korttinaan, joka siirtyy sarakkeesta toiseen työn edessä. (Niemi & Hietaniemi, 2020, 18; Juvonen, 2018, 23–24.) Hietaniemen (2020) Kuva 4 havainnollistaa Kanban-taulun työnkulun visualisointia.



Kuva 4. Esimerkki Kanban-taulusta (Hietaniemi, 2020)

Kanban-taulu voi olla fyysinen, esimerkiksi valkotaulu ja post-it -laput tai digitaalinen. Tärkeintä on, että taulu on kaikkien tiimin jäsenten nähtävillä ja että sitä päivitetään reaaliaikaisesti. Visualisointi auttaa tunnistamaan ruuhkautumiset, pullonkaulat ja viiveet, jolloin tiimi voi reagoida nopeasti ja tehdä tarvittavia muutoksia työnkulkuun. (Juvonen, 2018, 23–24.)

Hyvin visualisoitu työnkulku tukee tiimin itseohjautuvuutta, lisää läpinäkyvyyttä ja parantaa yhteistyötä. Kun kaikki näkevät työn etenemisen yhdellä silmäyksellä, epäselvyydet vähenevät ja

työn laatu sekä tehokkuus paranevat. Visualisointi antaa tiimille mahdollisuuden tehdä laadukkaita ja itsenäisiä päätöksiä kokonaisuuden näkökulmasta. Lisäksi visuaalinen taulu nopeuttaa palavereja: kokoukset eivät ala tilanteen kertaamisella, vaan osallistujat näkevät tilanteen heti ja voivat keskittyä ongelmien ratkaisemiseen ja toimenpiteistä sopimiseen. (Torkkola, 2015, 49–50.)

3.2.2 Läpimenoajan mittaaminen

Läpimenoajalla tarkoitetaan aikaa, joka kuluu tehtävän aloittamisesta sen valmistumiseen. Kanban-menetelmässä läpimenoajan mittaaminen on keskeinen tapa seurata työn etenemisen sujuvuutta ja prosessin tehokkuutta. (Niemi & Hietaniemi, 2020, 18.) Kun tehtävien läpimenoaikoja tarkastellaan säännöllisesti, tiimi saa arvokasta tietoa työn virtauksesta, mahdollisista viiveistä ja prosessin pullonkaloista. Lyhyt ja tasainen läpimenoaika kertoo, että työ etenee hallitusti ja että resurssit on kohdennettu oikein. Pitkät tai vaihtelevat läpimenoajat puolestaan voivat viitata kuormitukseen, priorisointiongelmiin tai liialliseen keskeneräiseen työhön. (Atlassian, n.d.b.)

Läpimenoajan seuranta voidaan hyödyntää jatkuvassa parantamisessa. Analysoimalla mittaus tuloksia tiimi pystyy tunnistamaan kehityskohteet ja tekemään perusteltuja päätöksiä prosessin optimoinnista. (Niemi & Hietaniemi, 2020, 18.)

3.2.3 Työn alla olevien tehtävien rajoittaminen

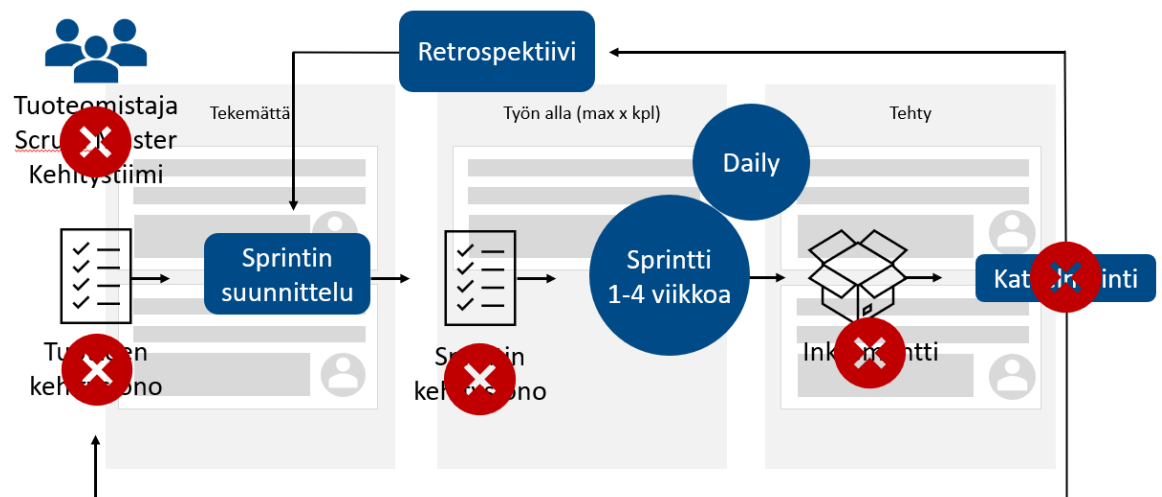
Työn alla olevien tehtävien rajoittaminen on yksi Kanbanin keskeisistä periaatteista. Käytännössä tämä tarkoittaa, että tiimi sopii, montako tehtävää voidaan tehdä kerrallaan. Rajoittamalla tehtävien määrää vältetään ylikuormitus ja varmistetaan, että työ etenee tasaisesti. (Hammarberg & Sundén, 2014, 95.)

Keskittymällä rajalliseen määrään tehtäviä tiimi voi varmistaa työn huolellisuuden ja nopeuttaa kokonaisuusien valmistumista. Tämä tukee laadun paranemista ja vähentää keskeneräisen työn määrää. Lisäksi WIP-rajojen käyttö auttaa tunnistamaan prosessin pullonkaulat: ruuhkautuneet vaiheet erottuvat visuaalisesti taulusta ja mahdollistavat nopean reagoinnin. (Hammarberg & Sundén, 2014, 115–123.)

Työn määrän rajoittaminen tukee Lean-ajattelua, jossa pyritään tekemään vähemmän, mutta arvokkaampaa. Se auttaa tiimiä keskittymään tärkeimpiin asioihin ja tekee työprosessista sujuvampaa. WIP-rajojen sopiminen vaatii yhteistä keskustelua ja seuranta, mutta tuloksena on usein selkeämpi ja hallitumpi työnkulku. (Hietaniemi, 2020; Niemi & Hietaniemi, 2020, 18).

3.3 Scrumban

Scrumban on ketterä kehittämistapa, joka alun perin suunniteltiin siirtymävaiheeksi Scrumista Kanbaniin. Kun uusia ketteriä viitekehyksiä ja menetelmiä syntyi, siitä kehittyi itsessään kehittyvä hybridikehys, jossa tiimit käyttävät Scrumia viitekehyksenä ja Kanbania prosessien parantamiseen. Scrumban yhdistää Scrumin rakenteen (roolit, sprintit ja tapahtumat) sekä Kanbanin joustavuuden (työnkulun visualisointi, WIP-rajoitukset), mikä tekee siitä tehokkaan menetelmän projekteihin, joissa vaatimukset muuttuvat usein. Menetelmä soveltuu erityisesti ylläpitoprojekteihin ja jatkuvaan kehitykseen, koska se tukee iteratiivista parantamista ja läpinäkyvyyttä. (Bhavsar ym., 2020.) Kuvassa 5 havainnollistettu hybridikehystä. (Project Management Institute, 2017, 108.)



Kuva 5. Havainnollistava kuva Kanbanin ja Scrumin eroista Scrumbaniin (Atlassian, n.d.a.)

Scrumban yhdistää Scrum- ja Kanban-menetelmien parhaat puolet, mikä voi säästää aikaa, parantaa laatua ja vähentää hukkaa (Alqudah & Razali, 2018; Plengvittava & Sanpote, 2018). Kanbanin visuaaliset työkalut, kuten Kanban-taulut, auttavat parantamaan työn näkyvyyttä ja mahdollistavat projektin reaaliaikaisen seurannan. Tämä lisää läpinäkyvyyttä ja auttaa tiimejä pysymään ajan tasalla tehtävistään. (Paul & Eldhose, 2020; Plengvittava & Sanpote, 2018.)

Scrumbanissa työ organisoidaan pieniin sprintteihin, ja Kanban-tauluja hyödynnetään työn visualisointiin ja seurantaan. Tehtävät sijoitetaan Kanban-taululle, ja tiimi hallitsee työtään työn alla olevien tehtävien rajoitusten avulla. Päivittäispalavereja pidetään tiimin yhteistyön ylläpitämiseksi ja esteiden poistamiseksi. Suunnittelupalaveri järjestetään, jotta tiimi tietää, milloin seuraava suunnittelu on tarpeen, yleensä silloin, kun keskeneräisen työn taso on alle ennalta määrätyn rajan. Scrumbanissa ei ole ennalta määriteltyjä rooleja - tiimi säilyttää nykyiset roolinsa. (Project Management Institute, 2017, 108.)

Scrumban on joustava menetelmä, joka soveltuu erilaisiin projektitilanteisiin ja tiimikokoihin. Se mahdollistaa nopean sopeutumisen muuttuviin vaatimuksiin ja prioriteetteihin, mikä tekee siitä erittäin hyödyllisen dynaamisissa ympäristöissä. (Alqudah & Razali, 2018; Alqudah & Razali, 2017.) Lisäksi Scrumban voi parantaa resurssienhallintaa ja tuottavuutta, koska se edellyttää asiakkaan aktiivista osallistumista projektiin. Iteratiivinen kehitys, jota Scrumban tukee, mahdollistaa projektin vaatimusten ja prioriteettien säännöllisen tarkistamisen ja päivittämisen. (Paul & Eldhose, 2020.)

Kuitenkin Scrumbanin käyttöönotto voi olla haastavaa, koska se vaatii Scrum- ja Kanban-menetelmien yhdistämistä ja mukauttamista tiimin ja projektin tarpeisiin (Alqudah & Razali, 2018; Alqudah & Razali, 2017). Tiimin jäsenet saattavat tarvita lisäkoulutusta ja ohjausta Scrumbanin käytössä, mikä voi lisätä aloituskustannuksia ja -aikaa (Alqudah & Razali, 2018; Plengvittaya & Sarnpone, 2018). Jos Scrumbania ei toteuteta oikein, se voi johtaa tehottomuuteen, lisääntyneeseen hukkaan ja heikentyneeseen laatuun (Alqudah & Razali, 2018; Alqudah & Razali, 2017).

Roolien ja vastuiden epäselvyydet voivat myös aiheuttaa sekaannusta ja tehottomuutta tiimissä (Alqudah & Razali, 2018; Alqudah & Razali, 2017). Suurten tiimien hallinta voi olla haastavaa, koska se vaatii tarkkaa koordinaointia ja viestintää. Näiden haasteiden vuoksi on tärkeää, että Scrumban-menetelmä muodostetaan oikein, jotta vältetään tehottomuus ja laatuongelmat. (Buttym., 2022). Atlassian (n.d.a.) on koonnut yleiskatsauksen Scrumista, Kanbanista sekä näiden yhdistelmästä Scrumbanista, joka on kuvattu taulukossa 1.

Taulukko 1. Yleiskatsaus: Scrum, Kanban ja Scrumban (mukaillen Atlassian, n.d.a.)

	Scrum	Kanban	Scrumban
Menetelmä	Kiinteät sprinttien pituudet Kiinteät roolit Jatkuva toimitus	Työn alla olevien tehtävien rajoittaminen Tehtävien visuaalinen seuranta Työn jatkuva virtaus	Kiinteät sprinttien pituudet Työn alla olevien tehtävien rajoittaminen Tehtävien visuaalinen seuranta Työn jatkuva virtaus
Roolit	Tuoteomistaja Scrum Master Kehitystiimi		
Työväline	Tuotteen kehitysajonot Sprintin kehitysajonot Valmis inkrementti	Kanban taulu Kanban kortit	Scrumban taulu Scrumban kortit
Tapahtumat	Sprintin suunnittelu Päivittäispalaverit Demo Retrospektiivi	Kanban palaveri	Sprintin suunnittelu Päivittäispalaverit Retrospektiivi
Työnkulku	Tuotteen kehitysajonot Sprintin kehitysajonot Työn alla Katselmus Tehty	Tekemättä Työn alla Tehty	Tekemättä Työn alla Tehty

4 Ketterä tiimi ja organisaatio

Ketterä tiimityöskentely perustuu joustavaan ja itseohjautuvaan toimintamalliin, jossa tiimin jäsenet jakavat vastuuta ja tekevät päätöksiä yhdessä (Almadhoun & Hamdan, 2018; Dyba & Dingsoyr, 2015). Tehokkuutta tukevat optimaalinen tiimikoko, monipuolinen osaaminen, avoin viestintä ja palveleva johtajuus (Strode ym., 2022). Organisaatioiden välinen yhteistyö edellyttää luottamuksellista vuorovaikutusta ja kulttuurien ymmärtämistä, mikä mahdollistaa yhteisten toimintamallien kehittämisen ja kokemusten jakamisen (Halonen & Käpykangas, 2023). Ketterän toiminnan periaatteet, kuten asiakaskeskeisyys, jatkuva parantaminen ja T-muotoinen osaaminen, ohjaavat tiimin arkea. Toimintaa mitataan toimituksen, laadun ja arvon perusteella ja ketteryyden soveltaminen eri konteksteissa edellyttää käytäntöjen mukauttamista. Kun organisaatiot toimivat ketterästi yhdessä, ne voivat kehittää yhteisiä toimintamalleja, jakaa kokemuksia ja oppia toisiltaan, mikä vahvistaa koko verkoston ketteryyttä. (Figalist ym., 2019.)

4.1 Tiimin rakenne, johtaminen ja yhteistyö

Ketterä tiimityöskentely eroaa merkittävästi perinteisestä organisaatiomallista. Ketterät tiimit koostuvat eri alojen osaajista, jotka työskentelevät itseohjautuvasti ja jakavat vastuuta yhteisistä tavoitteista. Tiimin jäsenillä on kyky tehdä päätöksiä itsenäisesti, mikä lisää joustavuutta ja reagointikykyä muuttuvissa olosuhteissa. (Dyba & Dingsoyr, 2015.) Keskeisiä ominaisuuksia ovat mukautuva päätöksenteko, kollektiivinen omistajuus ja jatkuva oppiminen. (Herranen, 2020, 92).

Optimaalinen ketterän tiimin koko on 5–9 jäsentä, joilla on toisiaan täydentäviä taitoja (Sixsigma.us, 2024). Tiimin jäsenillä tulisi olla myös päällekkäisiä taitoja, jotta he voivat tarvittaessa tukea toisiaan (Strode ym., 2022). Liian suuret tiimit voivat kärsiä viestinnän pullonkauloista, koordinoituaakasta ja heikentyneestä yhteenkuuluvuudesta, mikä heikentää tiimin suorituskykyä (Almadhoun & Hamdan, 2018).

Johtaminen ketterissä tiimeissä ei ole yhden henkilön vastuulla, vaan se jakautuu tiimin jäsenten kesken. Palveleva johtajuus tukee tiimin kehitystä ja poistaa esteitä. Tehokas johtaminen edellyttää myös toimivia ihmissuhteita, selkeitä rakenteita ja jatkuvaa vuorovaikutusta. Avoin viestintä, aktiivinen kuuntelu ja palautteen antaminen ovat keskeisiä tekijöitä tiimin toimivuuden kannalta. (Työturvallisuuskeskus, n.d.)

Organisaatioiden välinen yhteistyö on tärkeä osa ketterää toimintakulttuuria. Kun eri organisaatiot tai tiimit yhdistävät osaamisensa ja resurssinsa, syntyy synergiaa, joka voi parantaa tehokkuutta, lisätä innovatiivisuutta ja edistää hyvinvointia. Yhteistyön onnistuminen edellyttää luottamuksellista vuorovaikutusta, jossa tieto ja osaaminen jaetaan aktiivisesti. (Halonen & Käpykangas, 2023.) Vuorovaikutuksen laatu, kulttuurierojen tunnistaminen ja viestintätäylin sopeuttaminen ovat ratkaisevia tekijöitä onnistuneessa yhteistyössä (Aira, 2012).

Ketterissä organisaatioissa yhteistyö ei rajoitu vain tiimin sisäisiin suhteisiin, vaan ulottuu myös tiimien ja organisaatioiden välisiin rajapintoihin (Schwaber & Sutherland, 2020, 4). Tällainen laaja-alainen yhteistyö mahdollistaa asiakaslähtöisen kehittämisen, nopean reagoinnin muutoksiin ja jatkuvan oppimisen. Kun organisaatiot toimivat ketterästi yhdessä, ne voivat kehittää yhteisiä toimintamalleja ja jakaa kokemuksia ja oppia toisiltaan, mikä lisää koko verkoston ketteryyttä. (Figuilist ym., 2019.)

4.2 Ketterän toiminnan periaatteet ja käytännöt

Ketterän tiimin tehokkuus perustuu joukkoon periaatteita, jotka ohjaavat tiimin arkea ja päätöksentekoa (Schwaber & Sutherland, 2020, 4). Näitä ovat muun muassa itseorganisoituminen, autonomia, ristikkäiset taidot, jatkuva parantaminen, avoin viestintä ja asiakaskeskeisyys. Tiimin jäsenillä tulisi olla syvällistä osaamista omalta erikoisalaltaan, mutta myös laaja ymmärrys toistensa työstä, niin sanottu T-muotoinen osaaminen. Tämä mahdollistaa joustavan työnjaon ja tukee tiimin kykyä reagoida nopeasti muuttuviin tilanteisiin. (Sixsigma.us, 2024.)

Samankaltaisia teemoja tarkastellaan myös älykkään organisaation näkökulmasta. Organisaation rakenteen tulisi tukea uusiutumista ja ketterää toimintaa poistamalla turhat siilot ja raja-aidat tiimien väliltä. Älykkäässä organisaatiossa yhteinen visio ja päämäärä luovat suunnan, mutta rakenteet ja toimintamallit voivat mukautua tilanteen mukaan. Johtamisen roolina on tasapainoilla pysyvyyden ja muutoksen välillä sekä luoda edellytykset jatkuvalle oppimiselle ja yhteistyölle. Näin organisaation ketteryys ja tiimien itseohjautuvuus vahvistuvat toisiaan tukevalla tavalla. (Sydänmaanlakka, 2024.)

Tiimin rakentaminen vaatii suunnitelmallisuutta ja systemaattista lähestymistapaa. Ensimmäinen askel on valita tiimiin henkilöt, joilla on sekä teknistä osaamista että hyvät yhteistyötaidot. Tämän jälkeen asetetaan selkeät tavoitteet, jotka tukevat organisaation strategisia päämääriä. Ketterät käytännöt, kuten sprintit, retrospektiivit ja päivittäiset palaverit otetaan käyttöön asteittain, jotta

tiimi ehtii omaksua ne luontevasti. Lopuksi tiimidynamiikkaa kehitetään valmennuksen, palautteen ja yhteisten kokemusten avulla. (Steeh ym., 2025.)

Ketterän tiimin suorituskykyä arvioidaan mittareilla, jotka keskittyvät toimitukseen, laatuun ja arvoon. Toimitusmittarit kuvaavat tiimin kykyä tuottaa valmista työtä, ja niihin kuuluvat esimerkiksi syklien kesto, läpimenoaika ja keskeneräisen työn määrä (WIP). Laatua arvioidaan virheiden määrän, teknisen velan ja tuotteen vakauden perusteella. Arvomittarit puolestaan keskittyvät asiakastyytyvyyteen, käyttäjäkokemukseen ja liiketoimintavaikutuksiin. On tärkeää, että mittarit tukevat tiimin tavoitteita kokonaisvaltaisesti. Jos painotus on liikaa nopeudessa, se voi johtaa laadun heikkenemiseen. Vastaavasti liian tiukka fokus yksittäisiin suorituskykymittareihin voi ohjata tiimiä pois asiakasarvon tuottamisesta. Parhaimmillaan mittarit toimivat tiimin oppimisen ja kehittämisen välineinä, eivät pelkästään kontrollin välineinä. (Medavarapu, 2023.)

Ketterä toimintamalli ei rajoitus vain ohjelmistokehitykseen vaan sen periaatteita voidaan soveltaa monilla eri toimialoilla. Iteratiivinen arvon tuottaminen, jatkuva palaute ja asiakaslähtöisyys ovat hyödyllisiä myös esimerkiksi markkinoinnissa, tuotekehityksessä ja palvelumuotoilussa. Ketteryyden soveltaminen edellyttää kuitenkin käytäntöjen räätälöintiä toimialan ja organisaation erityispiirteiden mukaan. (Dong ym., 2024.)

Etätiimeissä ketteryyden ylläpitäminen vaatii digitaalisia yhteistyövälineitä, muokattuja seremonioita ja selkeitä viestintäprotokollia. Esimerkiksi päivittäiset palaverit voidaan toteuttaa videoyhteyksin ja visuaaliset johtamistaulut voidaan siirtää sähköisiin alustoihin. Tärkeintä on säilyttää avoin viestintä ja yhteisöllisyys, vaikka fyysinen läsnäolo puuttuu. (Ojala, 2018, 139.) Suurissa organisaatioissa ketteryyden soveltaminen tuo mukanaan erityisiä haasteita. Tiimien välisen koordinaation vaikeudet, riippuvuudet ja strategisten tavoitteiden yhtensovittamisen haasteet. Hajautetuissa ympäristöissä tiedon jakaminen ja asiakasvuorovaikutuksen ylläpitäminen voivat olla puutteellisia, ellei niihin kiinnitetä erityistä huomiota. Ratkaisuina nähdään ketterien menetelmien mukauttaminen organisaation tarpeisiin, johdon tuki ja sitoutuminen sekä oppiminen kokemusten jakaminen tiimien välillä. (Paasivaara ym., 2018.)

5 Ohjelmistotestaus ketterässä kehityksessä

Ketterän ajattelun ytimessä on ohjelmiston kehittäminen yhteistyöhön perustuvalla, arvoa tuottavalla ja tehokkaalla tavalla (Baumgartner ym., 2021, 1). Tämä periaate näkyy erityisesti testauksessa, joka on olennainen osa ketterää ohjelmistokehitystä. Se ei ole erillinen vaihe, vaan integroitu osa kehitysprosessia, jossa virheet pyritään havaitsemaan mahdollisimman varhain. Ketterässä toimintamallissa korostuvat jatkuvuus, nopea palaute ja laadun varmistaminen jokaisessa kehityssyklissä. (Gregorio, 2012; Smith & Hayes, 2024; Wang ym., 2019). Laadunvarmistus ketterässä kehityksessä nähdään jatkuvana prosessina, joka tapahtuu jokaisessa sprintissä. Testaus ja QA integroidaan tiimin päivittäiseen työhön, ja automaatiolla on keskeinen rooli regressiotestauksessa ja jatkuvassa integraatiossa. Lisäksi dokumentointi on kevyempää kuin perinteisissä malleissa, mutta selkeät testitapaukset ja hyväksymiskriteerit ovat välttämättömiä laadun varmistamiseksi. (Fortunato ym., 2017.)

Tässä luvussa käsitellään testauksen roolia ketterässä kehityksessä, sen vaiheita sprintin eri osissa, regressiotestauksen ja testiautomaation merkitystä, dokumentoinnin käytäntöjä sekä keskeisiä testaustyökaluja.

5.1 Testauksen rooli

Ohjelmistotestauksella varmistetaan, että toteutettava ohjelmisto vastaa sille asetettuja vaatimuksia ja toimii suunnitellusti. Käytännössä tämä tarkoittaa vertailua toteutuksen ja suunnitelmien välillä sekä poikkeamien tunnistamista. (Kasurinen, 2013, 10.) Testaus on olennainen osa laadunvarmistusta, sillä se auttaa minimoimaan riskejä ja parantamaan ohjelmiston luotettavuutta (ISTQB, 2018, 12).

Ohjelmistotestaus kohdistuu pääasiassa ohjelmiston koodiin ja jaetaan yleisesti neljään tasoon: yksikkötestaus, integraatiotestaus, järjestelmätestaus ja hyväksymistestaus (Juvonen, 2018, 26). Näillä tasoilla käytetään erilaisia testausmenetelmiä, kuten musta laatikko-, valkoinen laatikko-, harmaa laatikko -testaus. Lisäksi voidaan hyödyntää suorituskykytestausta, savutestausta, käytettävyydestestausta, tutkivaa testausta, mallipohjaista testausta ja regressiotestausta. (Juvonen, 2018, 28–33.)

Testauksen rooli vaihtelee kehitysmenetelmien mukaan. Perinteisessä vesiputousmallissa testaus tapahtuu vasta kehityksen loppuvaiheessa, mikä voi johtaa virheiden myöhäiseen havaitsemiseen ja korkeisiin korjauskustannuksiin. (Smith & Hayes, 2024; Wang ym., 2019.) Ketterässä kehityksessä testaus on jatkuva prosessi, joka tapahtuu jokaisessa kehityssyklissä. Tämä mahdollistaa virheiden nopean havaitsemisen ja korjaamisen sekä parantaa ohjelmiston laatua. (Gregorio, 2012; Smith & Hayes, 2024.)

Vaikka ketterät menetelmät tukevat jatkuvaa testausta, testausprosessin kehittämisessä esiintyy edelleen haasteita. Ohjelmisto-organisaatiot lähestyvät testausprosessien kehittämistä usein reaktiivisesti, mikä tarkoittaa, että painopiste on ongelmien korjaamisessa sen sijaan, että prosesseja kehitettäisiin ennakoivasti. Testausresursseja on yleensä riittävästi, mutta niitä ei käytetä tehokkaasti. Reaktiivinen toimintatapa voi johtaa toistuviin ongelmiin, koska prosessia ei kehitetä systemaattisesti. Testausprosessien kehittäminen kohti ennakoivia ja hyviä käytäntöjä parantaisi ohjelmistojen laadunvarmistusta kokonaisvaltaisesti. (Kasurinen, 2011.)

Ketterässä kehityksessä testaus on koko tiimin vastuulla. Scrum-tiimeissä korostuu poikkitoiminnallisuus, jossa kehittäjät, testaajat ja liiketoiminnan edustajat työskentelevät tiiviissä yhteistyössä. Testauksen suunnittelu, toteutus ja arviointi tapahtuvat osana tiimin päivittäistä toimintaa ja päätöksenteko perustuu jaettuun vastuuseen. Avoin kommunikaatio ja jatkuva vuorovaikutus eri sidosryhmien kanssa tukevat laadunvarmistusta ja mahdollistavat nopean reagoinnin muutoksiin. (Heuer, 2014, 14; ISTQB, 2014, 32.)

5.2 Testauksen vaiheet ketterässä kehityksessä

Ohjelmistotestauksen vaiheet Scrum-viitekehityksessä kattavat koko kehitysprosessin alusta loppuun. Ennen varsinaisten sprinttien käynnistymistä toteutetaan nollasprintti, joka luo perustan kehitysprosessille ja valmistaa tiimin tulevaan työskentelyyn. (ISTQB, 2014, 32.) Nollasprintissä määritellään projektin laajuus ja kehitysjono, luodaan ensimmäinen järjestelmäarkkitehtuuri ja korkean tason prototyypit, sekä katselmoidaan vaatimukset (Tuomikoski & Tervonen, 2009; Van den Broek ym., 2014). Testaajan rooli on keskeinen testausympäristön suunnittelussa ja perustamisessa, testausstrategian laadinnassa sekä riskianalyysin suorittamisessa (Heuer, 2014, 12–13; Berlowski ym., 2016).

Sprintin aikana testaus on tiiviisti integroitu osaksi kehitystyötä. Päivittäispalaverit auttavat tiimiä seuraamaan edistymistä ja tunnistamaan esteitä. (Tuomikoski & Tervonen, 2009; Moe ym., 2015.)

Testausta suoritetaan jatkuvasti osana jatkuvaa integraatiota, mikä mahdollistaa sekä uusien ominaisuuksien testaamisen että regressiotestauksen aiemmille toiminnallisuuksille (Berlowski ym., 2016; Collins & de Lucena, 2012). Testauksen tehokkuus perustuu jatkuvaan seurantaan, selkeään raportointiin, virheiden priorisointiin, testiautomaatioon ja kattavuuden arviointiin (ISTQB, 2014, 24–25).

Sprintin lopussa järjestettävä demo tarjoaa tiimille mahdollisuuden esitellä valmiit ominaisuudet ja testitulokset sidosryhmille, mikä tuo kehitystyöhön läpinäkyvyyttä ja mahdollistaa välittömän palautteen saamisen (Tuomikoski & Tervonen, 2009; Van den Broek ym., 2014). Demoa seuraa retrospektiivi, jossa tiimi tarkastelee sprintin kulkua ja tunnistaa parannuskohteita (ISTQB, 2014; Moe ym., 2015; Tuomikoski & Tervonen, 2009). Testitulokset analysoidaan ja dokumentoidaan, ja palautetta kerätään varhaisessa vaiheessa (Berlowski ym., 2016; ISTQB, 2014, 23). Testauksen päättäminen varmistaa, että kaikki oleellinen on käsitelty ennen projektin tai testausvaiheen päättymistä, ja testausmateriaali viimeistellään ja arkistoidaan myöhempää käyttöä varten (ISTQB, 2018, 19–20).

5.2.1 Ennen sprintin aloitusta

Ennen varsinaisten sprinttien käynnistymistä toteutetaan nollasprintti, joka luo perustan koko kehitysprosessille ja valmistaa tiimin tulevaan työskentelyyn (ISTQB, 2014, 32). Tämän ensimmäisen iteraation aikana on tärkeää, että tiimi ja testaaja tekevät tiivistä yhteistyötä varmistaakseen, että kaikki tarvittavat valmistelut on suoritettu. Nollasprintissä määritellään projektin laajuus ja kehitysjono, eli backlog, jonka avulla varmistetaan, että kehitystyö etenee oikeaan suuntaan ja prioriteetit ovat selkeät. (Tuomikoski & Tervonen, 2009; Van den Broek ym., 2014.) Samalla luodaan ensimmäinen järjestelmäarkkitehtuuri ja korkean tason prototyypit, jotka tarjoavat teknisen pohjan seuraaville iteraatioille. (ISTQB, 2014, 32.)

Eriyistä huomiota kiinnitetään vaatimusten muotoon ja esitystapaan, kuten käyttäjätarinoihin tai käyttötapauksiin, ja ne katselmoidaan yhdessä koko tiimin kanssa. Tämä varmistaa, että vaatimukset ymmärretään samalla tavalla ja että ne ovat riittävän selkeitä testauksen ja kehityksen tueksi. Vaatimuksia ei pyritä määrittämään yksityiskohtaisesti etukäteen, vaan niitä tarkennetaan ja muokataan projektin edetessä, erityisesti käyttäjän saadessa kokemusta ohjelmiston varhaisista versioista. (Heuer, 2014, 12–13.)

Testaajan rooli tässä vaiheessa on keskeinen, erityisesti testausympäristön suunnittelussa ja perustamisessa. Tämä sisältää tarvittavien työkalujen, kuten testiautomaatiojärjestelmien ja jatkuvan integraation työkalujen asennus ja konfigurointi. (Berlowski ym., 2016.) Lisäksi testiautomaatio otetaan käyttöön jo tässä vaiheessa, tukemaan jatkuvaa testausprosessia ja nopeaa palautetta kehitykselle. Työkalujen tehokas hyödyntäminen edellyttää, että tiimille järjestetään tarvittaessa koulutusta niiden käytöstä, mikä varmistaa yhtenäisen osaamistason. (Heuer, 2014, 12–13.)

Nollasprintin aikana laaditaan testausstrategia, jossa määritellään testauksen laajuus, prioriteetit, tekniset riskit, testityypit ja kattavuustavoitteet. Strategian avulla varmistetaan, että testaus tukee projektin tavoitteita koko kehityksen ajan ja että sen suunta on selkeästi linjassa ohjelmiston laadullisten vaatimusten kanssa. Tämän lisäksi suoritetaan laaturiskianalyysi, jonka perusteella tunnistetaan mahdolliset projektia uhkaavat riskit sekä määritellään toimenpiteet niiden hallintaan. Samalla määritetään testimetriikat, joiden avulla voidaan seurata testauksen etenemistä, tuotteen laatua ja prosessin tehokkuutta. (ISTQB, 2014, 32.)

Testaaja ja muu tiimi sopivat myös yhteiset kriteerit sille, milloin testaus katsotaan valmiiksi ja missä tilanteissa se voidaan keskeyttää. Lisäksi arvioidaan, onko tiimillä riittävä osaaminen ja resurssit projektin onnistuneeseen toteuttamiseen, vai tarvitaanko lisäkoulutusta tai mahdollisesti uusia rekrytointia testauksen tueksi. Tämän lisäksi määritellään testauksen aikataulu ja priorisoidaan testattavat osa-alueet, jotta resurssit voidaan kohdentaa mahdollisimman vaikuttavasti. (Heuer, 2014, 12–13.)

Roolien ja vastuiden selkeyttäminen on myös keskeistä tässä vaiheessa. Testaaja, kehittäjä, Scrum Master ja Product Owner määrittelevät yhdessä vastuualueensa ja viestintäkäytännöt, jotta yhteistyö sujuu saumattomasti koko projektin ajan. Tämä luo pohjan tehokkaalle testausprosessille ja parantaa tiimin kykyä reagoida muutoksiin. (ISTQB, 2014, 32.) Testauksen suunnittelu alkaa julkaisusuunnittelutilaisuudessa, jossa tarkennetaan testattavat ominaisuudet ja laaditaan alustavat testitapaukset. Näitä suunnitelmia tarkastellaan ja päivitetään jokaisen sprintin alussa. Sprinttien aikana testauksen edistymistä seurataan aktiivisesti, ja mahdollisiin ongelmiin puututaan välittömästi, jotta testaus voidaan pitää jatkuvasti käynnissä ilman tarpeettomia viiveitä. (Heuer, 2014, 12–13.)

5.2.2 Sprintin aikana

Sprintin aikana testaus on tiiviisti integroitu osaksi kehitystyötä. Päivittaiset Scrum-palaverit (Dailyt) auttavat tiimiä seuraamaan edistymistä, tunnistamaan esteitä ja varmistamaan, että kaikki pysyvät ajan tasalla. (Tuomikoski & Tervonen, 2009; Moe ym., 2015.) Tehtävien etenemistä havainnollistetaan usein visuaalisilla työkaluilla, kuten Kanban-tauluilla, jotka tukevat tiimin itseohjautuvuutta ja reaktiokykyä (ISTQB, 2014, 23–24).

Kanbanin hyödyntämistä ohjelmistokehityksen eri ympäristöissä on tarkasteltu laajasti ja menetelmä soveltuu erityisen hyvin ylläpitoprojekteihin, joissa työtehtävät ovat nopeita ja vaihtelevia. Visuaalinen työnkulun hallinta sekä työn määrän rajoitukset (WIP) parantavat työn hallittavuutta ja tasapainottavat kuormitusta. Testausvaiheiden hallinta helpottuu merkittävästi, kun testauksen eteneminen ja mahdolliset viiveet saatiin näkyville Kanban-taululle, mikä mahdollistaa nopeammat ratkaisut ja tehokkaamman resurssien käytön. Kanbanin käyttöönottoon liittyy kuitenkin haasteita: menetelmän edellyttämä avoimuus ja jatkuva priorisointi eivät aina sovi organisaatioiden vakiintuneisiin toimintamalleihin, mikä voi johtaa muutosvastarintaan. Työn määrän rajoittaminen koettiin aluksi hankalaksi, ja rajoitukset saattoivat tuntua työn hidasteilta. Lisäksi Kanbanin yksinkertainen rakenne ei kaikissa tapauksissa tarjoa riittävästi tukea monimutkaisille projekteille. Johdon tuen puute sekä mittareiden ja seurannan opettelu vaikeuttavat menetelmän jalkauttamista käytännön ohjelmistokehitykseen. (Ahmad, 2016.)

Testausta suoritetaan jatkuvasti osana jatkuvaa integraatiota, mikä mahdollistaa sekä uusien ominaisuuksien testaamisen että regressiotestauksen aiemmille toiminnallisuuksille. Samalla se tukee nopeaa palautesykliä kehitystiimin ja testauksen välillä. (Berlowski ym., 2016; Collins & de Lucena, 2012.) Testit voivat olla manuaalisia tai automatisoituja ja niiden tulokset auttavat varmistamaan ohjelmiston laadun sprintin aikana (Heuer, 2014, 22).

Testauksen tehokkuus perustuu jatkuvaan seurantaan ja selkeään raportointiin, joiden avulla havaittuihin puutteisiin voidaan reagoida nopeasti. Virheiden priorisointi auttaa keskittymään kriittisimpiin ongelmiin. Testiautomaation ansiosta testit voidaan toistaa tehokkaasti muutosten jälkeen. Kattavuuden arviointi tukee laadunvarmistusta varmistamalla, että tärkeimmät toiminnallisuudet ja riskialueet on testattu riittävän laajasti. (ISTQB, 2014, 24–26.)

5.2.3 Sprintin päättäminen

Sprintin ja testauksen päättäminen muodostavat tärkeän osan ketterän kehityksen sykliä. Sprintin lopussa järjestettävä demo tarjoaa tiimille mahdollisuuden esitellä valmiit ominaisuudet ja testitulokset sidosryhmille. Tämä tuo kehitystyöhön läpinäkyvyyttä ja mahdollistaa välittömän palautteen saamisen, mikä auttaa varmistamaan, että kehitys vastaa vaatimuksia. (Tuomikoski & Tervonen, 2009; Van den Broek ym., 2014.) Demoa seuraa retrospektiivi, jossa tiimi tarkastelee sprintin kulkua: mitä onnistuttiin tekemään hyvin ja missä on parantamisen varaa. Tavoitteena on jatkuva kehittyminen tunnistamalla onnistumisia, haasteita ja mahdollisia toimenpiteitä tulevia sprinttejä varten. (ISTQB, 2014, 14; Moe ym., 2015; Tuomikoski & Tervonen, 2009.)

Sprintin aikana kertyneet testitulokset analysoidaan ja dokumentoidaan. Tämä varmistaa, että havaitut virheet on korjattu ja että järjestelmä toimii odotusten mukaisesti. Testitulosten käsittely tukee päätöksentekoa ja jatkuvaa laadunvarmistusta koko kehityksen ajan. (Berlowski ym., 2016; ISTQB, 2014, 23.) Palautteen kerääminen varhaisessa vaiheessa, mahdollistettuna lyhyillä iteraatioilla ja jatkuvalla integraatiolla, auttaa tiimiä suuntaamaan työnsä liiketoiminnallisesti tärkeimpiin ominaisuuksiin sekä ehkäisee myöhäisiä ja kalliita korjaustarpeita (ISTQB, 2014, 10–11).

Testauksen päättäminen varmistaa, että kaikki oleellinen on käsitelty ennen projektin tai testausvaiheen päättymistä. Virheilmoitukset tarkistetaan ja dokumentoidaan tarvittavia jatkotoimia varten, ja testauksen yhteenveto laaditaan sidosryhmille. Testausmateriaali viimeistellään ja arkistoidaan myöhempää käyttöä varten sekä luovutetaan tarvittaessa ylläpidolle. Lisäksi testausprosessista kerättyjä kokemuksia hyödynnetään jatkuvaan kehittämiseen ja kypsyytason nostamiseen tulevia projekteja silmällä pitäen. (ISTQB, 2018, 19–20.)

5.3 Regressiotestaus ja testiautomaatio

Testausmenetelmät voidaan jakaa useisiin alaluokkiin sen mukaan, mihin tarkoitukseen niitä käytetään. Tässä opinnäytetyössä keskiössä ovat kvartaalitestauksen tueksi hyödynnetyt regressiotestaus ja testiautomaatio.

Regressiotestaus tarkoittaa käytännössä uudelleentestaamista, kun järjestelmää on muutettu. (Kasurinen, 2013, 63–64; Juvonen, 2018, 17) Se ei ole erillinen testauksen tyyppi, vaan yleisnimitys kaikelle testaukselle, jonka tarkoituksena on varmistaa muutosten jälkeinen toimivuus. Tätä

testausmuotoa tarvitaan esimerkiksi silloin, kun korjataan virheitä, lisätään uusia ominaisuuksia tai saavutetaan kehityksen osatavoite. (Kasurinen, 2013, 63–64.)

Oleellista regressiotestauksessa on varmistaa, että korjaukset pysyvät voimassa ja uudet muutokset eivät riko aiempia toimintoja. Käytännössä tämä tarkoittaa erityistä huomiota uusiin komponentteihin ja niiden vaikutuksiin koko järjestelmässä. (Juvonen, 2018, 18.) Muutoksiin suhtaudutaan ikään kuin koko kyseinen osa olisi rakennettu uudestaan (Kasurinen, 2013, 63–64).

Regressiotestaus yhdistyy usein testiautomaatioon, koska se sisältää paljon toistuvia testejä (ISTQB, 2018, 37). Esimerkiksi jokaisen version tai osatavoitteen saavuttamisen yhteydessä suoritetaan samat testitapaukset. Tämä tekee regressiotestauksesta automaatiolle ideaalin alueen, vähentäen manuaalisen työn määrää ja nopeuttaen prosessia. (Kasurinen, 2013, 63–64.)

Testiautomaatio tarkoittaa ohjelmistotestauksen osa-alueita, jossa toistuvat testitapaukset suoritetaan automaattisesti erillisten työkalujen avulla. Tavoitteena on vapauttaa testaajien aikaa rutiininomaisista tehtävistä, nopeuttaa palautteen saamista ja parantaa testauksen toistettavuutta ja luotettavuutta. (Butt ym., 2023; Kasurinen, 2013, 70.) Erityisen hyödyllistä automaatio on ympäristöissä, joissa ohjelmistosta tehdään usein uusia versioita, jolloin samojen testien manuaalinen suorittaminen olisi tehotonta. Automaatiotestit voidaan ajoittaa tapahtuvaksi esimerkiksi yöaikaan, jolloin kehittäjät voivat aamulla tarkastella tuloksia ja korjata mahdolliset virheet. (Kasurinen, 2013, 70.)

Ketterien menetelmien yleistyessä testiautomaatiosta on tullut olennainen osa nykyaikaista ohjelmistokehitystä. Sitä hyödynnetään erityisesti testausajan ja -kustannusten hallinnassa. On kuitenkin tärkeää, että automaatio otetaan käyttöön harkitusti ja osana suunnitelmallista kehitysprosessia. (Butt ym., 2023.)

Testiautomaation käyttöönottoon liittyy myös haasteita. Yleinen harhaluulo on, että automaatio korvaisi manuaalitestauksen täysin, mutta todellisuudessa se toimii sen täydentäjänä. (Kasurinen, 2013, 71.) Testiautomaatio voi tuoda merkittäviä etuja ohjelmistokehitykseen, mutta sen onnistunut käyttöönotto edellyttää huolellista suunnittelua ja jatkuvaa sitoutumista, sillä työkalujen tehokas hyödyntäminen vaatii jatkuvaa ylläpitoa ja prosessien mukauttamista muuttuviin tarpeisiin (ISTQB, 2018, 72). Automaatio soveltuu erityisesti regressiotestaukseen, jossa varmistetaan, ettei ohjelmiston uusi versio ole rikkonut aiemmin toimineita ominaisuuksia. Testien automatisointi on usein vaativaa kehitystyötä, joka vaatii sekä aikaa että ylläpitoresursseja. Yksittäisen testitapauksen automatisointi on yleensä kalliimpaa kuin sen manuaalinen suoritus, mutta pitkällä aikavälillä toistojen kautta saavutetaan kustannussäästöjä. (Kasurinen, 2013, 71–72.)

Testiautomaation hyötyjen maksimoimiseksi automatisoitavat testit tulee valita huolellisesti. Tärkeitä valintakriteerejä ovat testitapauksen toistuvuus, kriittisyys sekä ennakoitavuus. Testaus suunnitelmassa tulisi määritellä testien prioriteetit ja arvioida automaation tuoma hyöty suhteessa sen kustannuksiin. Lisäksi on tärkeää varata riittävästi aikaa testiautomaation ylläpidolle, sillä ilman jatkuvaa huoltoa testit vanhenevat nopeasti eivätkä enää palvele tarkoitustaan. (Kasurinen, 2013, 73–74.)

On tärkeää ymmärtää, että testiautomaatio on ennen kaikkea laadunvalvonnan työkalu – ei keino löytää uusia virheitä. Sen tehtävä on varmistaa, että jo toimivat ominaisuudet toimivat edelleen, kun järjestelmää kehitetään eteenpäin. Väärinkäytettynä testiautomaatio voi aiheuttaa enemmän haittaa kuin hyötyä, erityisesti jos sen tarkoitukset ja rajoitteet unohdetaan. (Kasurinen, 2013, 74.)

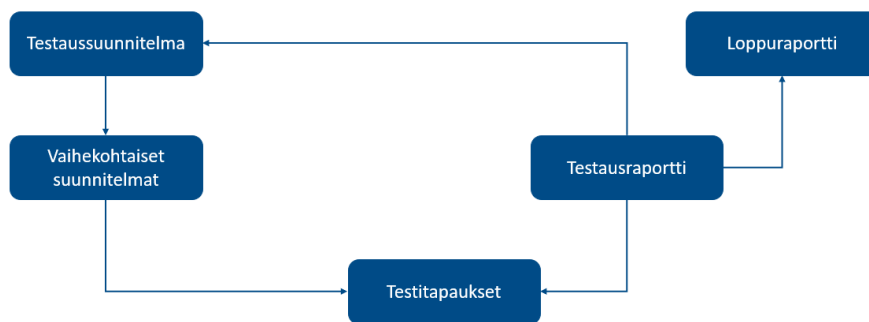
5.4 Testauksen dokumentointi

Ohjelmistotestauksen aikana syntyy runsaasti tietoa, jonka huolellinen dokumentointi on keskeistä testauksen toistettavuuden, virheiden jäljittämisen ja laadunvarmistuksen kannalta. Dokumentointi auttaa varmistamaan, että testauksen aikana tehdyt havainnot, suoritettut toimenpiteet ja mahdolliset virheet tallentuvat selkeästi ja systemaattisesti. Ilman dokumentaatiota testauksen tulokset voivat jäädä epäselviksi tai kokonaan hyödyntämättä, erityisesti silloin kun testauksesta vastaa useampi henkilö tai kun testauksen ja raportoinnin välillä kuluu aikaa. Dokumentaation avulla voidaan myös varmistaa testauksen jäljitettävyyden ja tukea tehokasta viestintää eri sidosryhmien välillä. Lisäksi se mahdollistaa testauksen arvioinnin ja jatkuvan kehittämisen osana ohjelmiston laadunvarmistusta. (ISTQB, 2018, 19–20; Kasurinen, 2013, 81.)

Ketterässä ohjelmistokehityksessä dokumentaation rooli on muuttunut. Dokumentaatiota ei hylätä, vaan sen määrää ja muotoa sopeutetaan projektin tarpeisiin. Dokumentaatio nähdään tiedontarpeen mukaan tuotettavana tukimateriaalina, jonka tarkoitus on palvella eri sidosryhmiä, kuten kehittäjiä, testaajia ja loppukäyttäjiä. Ketterän manifestin periaate ”Toimiva ohjelmisto on tärkeämpää kuin kattava dokumentaatio” ei tarkoita dokumentaation puuttumista, vaan sen tarkoituksenmukaista käyttöä. (Baumgartner ym., 2021, 143–144.)

Testauksen dokumentointia voidaan jäsentää useisiin eri osa-alueisiin, joista keskeisimpiä ovat testaus suunnitelma, testitapaukset ja testiraportti. Näiden avulla varmistetaan, että testaus ete-

nee suunnitellusti, testit suoritetaan kattavasti ja tulokset raportoidaan ymmärrettävästi. Dokumenttipohjien käyttö helpottaa tiedon kirjaamista ja yhtenäistää organisaation testauskäytäntöjä. Ne myös nopeuttavat raporttien koostamista ja helpottavat tiedon hakemista, sillä sisältö on aina tietyssä, ennalta määritellyssä muodossa. (ISTQB, 2018, 19–20.) Testausdokumentaatio voi sisältää yleisen testaussuunnitelman lisäksi vaihekohtaisia suunnitelmia ja raportteja, kuten yksikkö-, integraatio- ja järjestelmätestauksen dokumentteja. Jokaiselta testautasolta laaditaan tyypillisesti oma suunnitelma ja raportti, mutta projektin koon mukaan dokumentteja voidaan yhdistää suuremmiksi kokonaisuuksiksi. Pienemmissä projekteissa voi riittää yksi yleinen suunnitelma ja raportti, joka kattaa kaikki testauksen tasot. (Haikala & Mikkonen, 2011, 216–217.) Kuva 6 havainnollistetaan projektin testausdokumentaation keskinäisiä suhteita Kasurisen (2013) esittämällä tavalla.



Kuva 6. Projektin testausdokumentaation rakenne (Kasurinen 2013, 105)

5.4.1 Testaussuunnitelma

Testaussuunnitelma on keskeinen dokumentti, joka ohjaa ohjelmiston kehitys- ja ylläpitoprojekteissa toteutettavia testausmenetelmiä. Se määrittelee, mitä testataan, milloin ja millä menetelmillä. Suunnitelma perustuu organisaation testausstrategiaan ja sen sisältö määräytyy muun muassa testauksen tavoitteiden, riskien, resurssien, kehitysmallien ja testattavuuden mukaan. (ISTQB, 2018, 59; Kasurinen, 2013, 105.)

Testauksen suunnittelu ei ole kertaluonteinen vaihe, vaan jatkuva prosessi, joka kehittyy projektin edetessä. Suunnitelmaa päivitetään esimerkiksi testauksesta saadun palautteen ja muuttuvien riskien perusteella. Se voi ulottua myös projektin jälkeiseen ylläpitovaiheeseen. (ISTQB, 2018, 59–60.)

Testaussuunnitelma voi olla kokonaisvaltainen tai jakautua osa-aluekohtaisiin suunnitelmiin, kuten järjestelmä-, hyväksymis-, käytettävyy- tai suorituskykytestauksen suunnitelmiin. Lisäksi suunnitelmia voidaan laatia testitasojen mukaan, esimerkiksi yksikkö-, integraatio- tai suorituskykytestaukseen. Näin varmistetaan, että eri testausvaiheiden erityispiirteet ja vaatimukset tulevat huomioituiksi. (Kasurinen, 2013, 106–107.)

Ketterässä ohjelmistokehityksessä testaussuunnitelma ei välttämättä ole erillinen ja yksityiskohdainen dokumentti, vaan suunnittelu tapahtuu osana tiimin päivittäistä työskentelyä. Testaajat osallistuvat käyttäjätarinoiden määrittelyyn ja suunnittelevat testauksen rinnakkain kehityksen kanssa. Yleinen testaussuunnitelma voi silti olla hyödyllinen kokonaisuuden hallinnan, resurssien ja riskien ennakoinnin näkökulmasta. Tällainen suunnitelma voi kattaa esimerkiksi testauksen laajuuden, testityypit, testausympäristön ja aikataulut. Ketterässä kontekstissa suunnitelman tulee olla joustava ja mukautuva, ja sen laatimiseen osallistuu koko tiimi. (Heuer, 2014, 19–20.)

Testaussuunnitelman sisältö voi vaihdella projektin luonteen ja vaatimusten mukaan, mutta siihen sisältyy usein seuraavia osa-alueita:

- Testauksen laajuuden, tavoitteisen ja riskien määrittely.
- Testausstrategioiden ja lähestymistapojen valinta.
- Testaustehtävien integrointi kehitysprosessiin.
- Resurssien ja vastuiden määrittely.
- Aikataulutus eri kehitysmallien mukaisesti.
- Mittareiden ja budjetin suunnittelu.
- Dokumentaation rakenteen ja yksityiskohtaisuuden määrittely. (ISTQB, 2018, 59–60.)

Hyvin laadittu testaussuunnitelma toimii paitsi testauksen ohjausvälineenä myös viestinnän ja dokumentoinnin tukena koko projektin ajan. Se auttaa varmistamaan, että testaus on tavoitteellista, hallittua ja läpinäkyvää kaikille osapuolille. Suunnitelman huolellinen laatiminen ja päivittäminen edistää testauksen laatua ja tehokkuutta sekä tukee ohjelmiston kokonaisvaltaista laadunvarmistusta. (ISTQB, 2018, 59–60.)

5.4.2 Testitapaukset

Testitapaus kuvaa yksittäisen työvaiheen tai tapahtumaketjun, jonka seurauksena testattava järjestelmä suorittaa tietyn toiminnon. Siinä esitetään, mitä toimenpiteitä testissä suoritetaan ja millaista käyttäytymistä järjestelmältä odotetaan. Käytännön testausilanteissa tämä ei kuitenkaan useinkaan riitä, vaan testitapauksiin dokumentoidaan myös järjestelmän lähtötila, mahdolliset ehdot ja vaatimukset, yhteydet muihin testitapauksiin sekä testauksen tavoitteet ja perusteet. Näillä tiedoilla tuetaan niin testauksen toistettavuutta kuin sen arvioitavuutta. (Kasurinen, 2013, 109.)

Testauksen onnistumisen kannalta on olennaista, että testaaja tietää tarkalleen, miten järjestelmän pitäisi toimia. Tämä on keskeistä niin sanotun odotetun lopputuloksen määrittelyongelman ratkaisemiseksi: järjestelmä saattaa ulkoisesti näyttää toimivan oikein, vaikka sen sisäinen tila tai tiedon tallennus olisikin virheellinen. Esimerkiksi käyttöliittymässä näkyvä onnistumisilmoitus ei välttämättä vastaa tietokantaan tallentunutta tietoa. Tämän vuoksi testitapauksissa on määriteltävä täsmällisesti, mitä toimintoa järjestelmältä odotetaan. (Kasurinen, 2013, 110.)

Testitapaus sisältää lähtöehdot, syötteet, mahdolliset toiminnot, odotetut tulokset ja jälkiehdot, jotka perustuvat testiehtoon. Tällainen rakenne tekee testitapauksesta yksiselitteisen, toistettavan ja arvioitavan. Lisäksi testitapaukset tulee suunnitella siten, että ne tukevat testauksen läpinäkyvyyttä ja jäljitettävyyttä koko ohjelmistokehityksen ajan. (ISTQB, 2018, 41.)

Hyvin laaditun testitapauksen sisältö:

- **Vaikutusalue:** Osat järjestelmästä, joita testitapaus koskee.
- **Laatija:** Henkilö, joka on suunnitellut ja dokumentoinut testitapauksen.
- **Lisätiedot:** Lähteet, joista löytyy lisäinformaatiota testitapaukseen liittyen.
- **Muutosoikeudet:** Vastuutaho testitapauksen muuttamisesta tai hylkäämisestä.
- **Konteksti:** Testitapauksen suhde järjestelmän kokonaisuuteen ja muihin testitapauksiin.
- **Tavoite:** Toiminnallisuus tai ominaisuus, jota testillä pyritään arvioimaan.
- **Syötteet ja toimenpiteet:** Järjestelmään syötettävät tiedot ja testissä suoritettavat vaiheet.

- **Odotettu lopputulos:** Järjestelmän odotettu reaktio, tallennettavat tiedot ja lopputulos.
- **Testausympäristö:** Testin suoritusolosuhteet ja ympäristölle asetetut vaatimukset.
- **Erikoisehdot:** Poikkeus- ja virhetilanteet, joita testitapaus käsittelee.
- **Riippuvuudet:** Yhteydet muihin testitapauksiin.
- **Laatukriteerit:** Perusteet, joiden mukaan testitapausta pidetään onnistuneena.
- **Rajoitteet:** Tilanteet, joissa testitulokset voidaan jättää huomioimatta tai testitapaus ei ole merkityksellinen. (Kasurinen, 2013, 110.)

Testitapausten huolellinen dokumentointi on keskeinen osa laadukasta ohjelmistotestausta. Kun testitapaukset sisältävät selkeästi määritellyt tavoitteet, syötteet, odotetut lopputulokset ja kontekstin, ne tukevat testauksen luotettavuutta ja auttavat varmistamaan, että järjestelmä toimii suunnitellulla tavalla myös poikkeustilanteissa. (Kasurinen, 2013, 109–110.) Myös ISTQB (2018, 40–43) korostaa, että testitapaukset tulee suunnitella huolellisesti, jotta ne tukevat testauksen arvioitavuutta, toistettavuutta ja tehokkuutta.

Testitapausten ja testisuhteiden laatuominaisuuksia ohjelmistotestauksen ammattilaisten näkökulmasta on tunnistettu keskeisiä piirteitä. Tärkeimmiksi ominaisuuksiksi nousivat virheiden havaitseminen, käytettävyys, ylläpidettävyys, luotettavuus ja kattavuus. Sen sijaan resurssitehokkuuden ja uudelleenkäytettävyyden merkitys vaihteli vastaajien mukaan. Lisäksi on havaittu haasteita, kuten laatukriteereiden epäselvyys, mittareiden puute ja arviointikäytäntöjen hajanaisuus. Tulokset korostavat tarvetta selkeille prosesseille ja paremmalle tuelle laadukkaiden testitapausten tuottamisessa. (Tran ym., 2025.)

5.4.3 Testiraportti

Testiraportoinnin tehtävänä on koota ja välittää tietoa testauksen etenemisestä ja tuloksista erisidosryhmille. Raportointia tehdään sekä testauksen aikana että sen päätyttyä, ja sen muoto ja sisältö vaihtelevat projektin vaatimusten ja kohderyhmän mukaan. Testauksen aikana laadittavaa raporttia kutsutaan edistymisraportiksi, kun taas testauksen päätyttyä laaditaan yhteenvetoraportti, joka kokoaa yhteen testauksen keskeiset havainnot ja johtopäätökset. (ISTQB, 2018, 64.)

Edistymisraportti sisältää tyypillisesti tietoa testauksen tilasta suhteessa suunnitelmaan, mahdollisista esteistä, seuraavan jakson suunnitelluista toimenpiteistä sekä tuotteen laadusta. Yhteenvedoraportti puolestaan tarjoaa kokonaiskuvan suoritetusta testauksesta, poikkeamista suunnitelmaan, tuotteen laadusta suhteessa päätöskriteereihin sekä jäljellä olevista riskeistä. Molemmat raporttityypit voivat sisältää myös mittaritietoa esimerkiksi virheistä, testikattavuudesta ja resurssien käytöstä. Raporttien sisältöä tulisi mukauttaa paitsi projektin luonteen, myös kohdeyleisön tarpeiden mukaan, tekniselle tiimille suunnattu raportti voi poiketa merkittävästi johdolle laaditusta yhteenvedosta. (ISTQB, 2018, 64.) Hyvin laadittu raportti auttaa ymmärtämään testauksen kontekstia, päätöksenteon perusteita ja tuotteen laatua. Raportin rakenne ja sisältö tulisi suunnitella niin, että se tukee sekä teknistä analyysiä että sidosryhmien tiedontarpeita. (PractiTest Team, 2025; Winkler ym., 2019.)

5.5 Testaustyökalut

Testaustyössä käytettävät työkalut ovat usein samoja kuin ohjelmistokehityksessäkin. Testauksen hallintatyökalujen keskeisenä tehtävänä on mahdollistaa testausprosessin seuranta ja ohjaus. (Kasurinen, 2013, 82.)

Työkaluihin liittyvät keskeiset toiminnot liittyvät muun muassa seuraaviin asioihin:

- Toteutettujen testitapausten seuranta.
- Testien onnistumisasteen arviointi.
- Uusimmassa ohjelmistoversiossa havaittujen virheiden määrä.
- Virheiden vakavuus ja tyyppi.
- Testitapausten jakautuminen eri henkilöille sekä yksittäisten testaaajien työkuorman hallinta. (Kasurinen, 2013, 82.)

Kuten ohjelmistoprojekteissa yleensä, myös testauksen hallintatyökalujen tarve kasvaa projektin koon kasvaessa. Pienissä, noin viiden hengen tiimeissä erillisiä hallintatyökaluja ei välttämättä tarvita, sillä tiimin jäsenet voivat kommunikoida keskenään suoraan. Sen sijaan yli kymmenen hengen projekteissa erilliset hallinnan työkalut nousevat tärkeään rooliin. (Kasurinen, 2013, 82–83.)

Yksi keskeisimmistä testauksen hallintaan liittyvistä työkalutarpeista on testitapausten hallinta. Tällaiset työkalut auttavat projektin johtoa seuraamaan vikailmoituksia, virheiden korjaamisen etenemistä sekä arvioimaan testauksen laatua. Lisäksi testitapauksia voidaan lisätä järjestelmään projektin edetessä, mikä tukee testauksen organisointia ja työnjakoa. Usein testitapausten hallinta on integroitu osaksi vikahallintajärjestelmää tai projektinhallintatyökalua. (Kasurinen, 2013, 83.)

Laadunvalvontatyökaluilla tarkoitetaan järjestelmiä, joiden avulla voidaan koostaa seurantaraportteja ja arvioida järjestelmän tilaa reaaliaikaisesti. Näiden työkalujen avulla seurataan esimerkiksi sitä, eteneekö projekti suunnitellusti ja pysyykö kriittisten virheiden määrä hyväksyttävällä tasolla. (Kasurinen, 2013, 83).

Testauksen työvälineisiin kuuluvat myös viestintäkanavat, joiden merkitys korostuu erityisesti laajoissa ohjelmistoprojekteissa ja hajautetuissa tiimeissä. Testaajien on voitava kommunikoida sujuvasti kehittäjiä ja muiden sidosryhmien kanssa, jotta havaitut virheet ja epäselvyydet voidaan käsitellä nopeasti ja tehokkaasti. Viestintäkanavina käytetään esimerkiksi sähköpostia, pikaviestimiä, keskustelufoorumeita ja projektinhallintajärjestelmiä. Kun projektissa käytetään ulkopuolisia komponentteja tai tiimin jäsenet työskentelevät eri sijainneissa, viestintävälineiden rooli korostuu entisestään. Tiedon jakaminen ja työn koordinointi onnistuvat tehokkaasti, kun käytössä on tarkoituksenmukaiset työkalut, jotka tukevat sekä teknistä että organisatorista yhteistyötä. (Kasurinen, 2013, 84.)

Testausautomaatiotyökalujen valinta on keskeinen osa ohjelmistokehityksen laadunvarmistusta. Vaikka arviointikriteerit kuten yhteensopivuus, käyttöönoton helppous ja kustannukset ovat yleisesti tunnettuja, niiden soveltaminen käytännössä on usein epä johdonmukaista. Valintaprosessit puuttuvat tai ovat dokumentoimattomia, ja päätökset perustuvat usein kokemukseen tai mielikuviin. Lisäksi on havaittu, että vaikka yksimielisyys kriteereistä on olemassa, niiden käyttö on hajanaista. Molemmat tutkimukset korostavat tarvetta kehittää selkeitä ja läpinäkyviä valintakäytäntöjä, jotka tukevat tehokasta ja tarkoituksenmukaista työkalujen käyttöä. (Raulamo-Jurvanen, 2020.)

6 Tutkimus- ja kehittämismenetelmät

Tämän opinnäytetyön tutkimusstrategiaksi valittiin toimintatutkimus ja tutkimusotteeksi laadullinen lähestymistapa. Toimintatutkimus soveltuu erityisesti työelämän kehittämiseen, koska sen tavoitteena on parantaa käytännön toimintaa ja etsiä ratkaisuja yhdessä tutkimukseen osallistuvien kanssa (Heikkinen & Kaukko, 2023, 17–19). Laadullinen lähestymistapa mahdollistaa tutkitavan ilmiön syvällisen ymmärtämisen sekä osallistujien kokemusten, havaintojen ja merkitysten tarkastelun heidän omassa ympäristössään (Kananen, 2014, 109–110).

Aineistoa kerättiin useilla menetelmillä: fokusryhmähaastattelu, kehitystyöpajat ja havainnointi. Menetelmät valittiin, koska niiden avulla on mahdollista kerätä monipuolista aineistoa sekä tukea osallistavaa ja vuorovaikutteista kehittämisprosessia. Fokusryhmähaastattelu tuottaa ryhmässä syntyvää keskustelua ja yhteisiä näkökulmia (Mäntyranta & Kaila, 2008, 1), kun taas havainnointi tuo tietoa toiminnasta sen luonnollisissa tilanteissa (Puusa & Juuti, 2020, luku 8 Havainnointi organisaatiotutkimuksessa). Kehittämistyöpajat mahdollistavat osallistujien yhdessä tuottamia ratkaisuehdotuksia ja niiden arviointia (Pavelin ym., 2014).

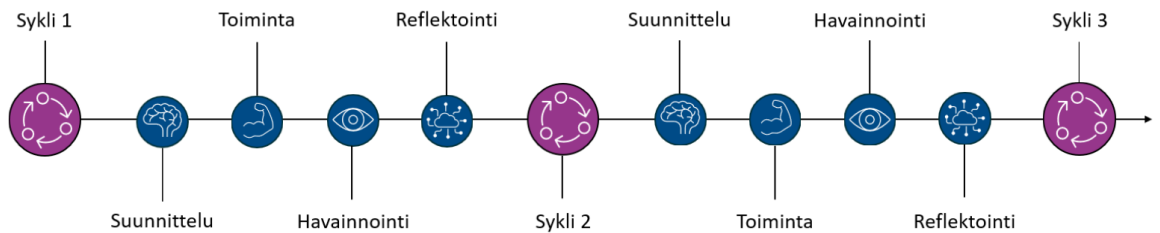
Aineiston analyysi eteni vaiheittain: fokusryhmähaastattelu litteroitiin ja koodattiin teemoiksi, retrospektiivien ja työpajojen tuotokset koottiin ja ryhmiteltiin pääsisällöiksi ja lähtötilannetta kuvaava aineisto järjestettiin taulukkomuotoon. Näiden analyysivaiheiden avulla eri menetelmillä kerätty tieto yhdistyi kokonaisuudeksi, joka tuki tutkimuskysymyksiin vastaamista ja kehittämistehtävän tavoitteiden saavuttamista.

6.1 Toimintatutkimus tutkimusstrategiana

Toimintatutkimus valittiin tämän opinnäytetyön tutkimusstrategiaksi, koska tutkimuksen tavoitteena oli kehittää tiimin kvartaalitestaukseen liittyvää toimintaa käytännössä ja tutkija työskentelee itse tutkimuksen kohteena olevassa tiimissä. Tämä mahdollisti aidon osallistumisen tutkimusprosessiin ja tiiviin vuorovaikutuksen muiden tiimin jäsenten kanssa.

Toimintatutkimuksen perusrakenne on syklinen, jossa jokainen sykli sisältää suunnittelun, toiminnan, havainnoinnin ja reflektoinnin, ja sykli voi toistua useita kertoja sitä mukaa, kun uusia kehittämistarpeita nousee esiin. (Heikkinen & Kaukko, 2023, 24–26; Jyrkämä, 2021) Näin toimintatutkimus toimii sekä tutkimus- että kehittämismenetelmänä, jossa tutkija ja osallistujat rakentavat

muutosta yhdessä. Toimintatutkimuksen vaihekaavioita on useita, Kuva 7 on esitelty Kananen (2014, 13) kuvaus toimintatutkimuksesta.



Kuva 7. Toimintatutkimuksen vaiheet ja sykli (Kananen 2014, 13)

Toimintatutkimuksella on pitkä historia, ja sen keskeisenä kehittäjänä pidetään Kurt Lewiniä, joka korosti työntekijöiden osallistumisen merkitystä toiminnan kehittämisessä ja päätöksenteossa (Kananen, 2014, 18). Lewinin mukaan muutosta rakennetaan vaiheittain, jokaisessa vaiheessa suunnitellen, toimien ja arvioiden tuloksia. Toimintatutkimuksen tavoitteena on liittää teoreettinen tieto käytännön toimintaan ja etsiä ratkaisuja todellisissa olosuhteissa (Puusa & Juuti, 2020, luku 17 Toimintatutkimus.)

Tässä opinnäytetyössä toimintatutkimuksen syklit toteutuivat tiimin arjessa Scrum-kehityksen mukaisissa toiminnoissa, joissa havainnointi ja reflektointi olivat osa päivittäistä, viikoittaista ja sprinttikohista kehittämistä.

6.2 Aineiston keruu

Laadullisen tutkimuksen tavoitteena ei ole tilastollinen yleistys, vaan tutkittavan ilmiön mahdollisimman syvälle kuvaaminen ja merkityksen ymmärtäminen (Kananen, 2014, 20–22). Aineistonkeruun ja analyysin vaiheet vuorottelivat tutkimusprosessin aikana, mikä mahdollisti tulosten jatkuvan hyödyntämisen tutkimuksen suunnittelussa.

Aineisto kerättiin havainnoinnin, fokusryhmähaastattelun ja kehittämistyöpajojen avulla. Menetelmät valittiin, koska ne soveltuvat hyvin toimintatutkimuksen osallistavaan ja vuorovaikutteiseen lähestymistapaan. Havainnointi toteutui käytännön työssä koko tutkimusjakson ajan erityisesti Scrum-tapahtumien aikana, kuten testauksen suunnittelupalaverissa, päivittäisissä palave-

reissa ja retrospektiiveissa. Fokusryhmähaastattelu toteutettiin tiimin jäsenille, tavoitteena kartoittaa nykytilanne ja osallistujien näkemykset testauksen nykytilasta. Kehittämistyöpajat toimivat osallistavina menetelminä, joissa ideoitiin ja arvioitiin uusia toimintatapoja.

6.2.1 Havainnointi

Havainnointi on aineistonkeruumenetelmä, jossa tutkija systemaattisesti kerää tietoa tutkittavasta ilmiöstä tarkkailemalla sitä suoraan (Kajander & Lappi, 2022). Sen etuna on mahdollisuus seurata toimintaa reaaliajassa ja huomioida myös nonverbaalista viestintää, sosiaalista vuorovaikutusta ja eroja puheen ja toiminnan välillä (Puusa & Juuti, 2020, luku 8 Havainnointi organisaatiotutkimuksessa).

Havainnoinnista on olemassa monia eri muotoja ja luokitteluja, kuten suora ja epäsuora havainnointi, strukturoidut ja strukturoimattomat sekä inhimilliset ja mekaaniset havainnointitavat (Kananen, 2014, 80).

Tässä tutkimuksessa hyödynnettiin osallistuvaa havainnointia. Tutkija oli osa tiimiä ja osallistui luonnollisiin työtilanteisiin, mikä nopeutti ilmiön ymmärtämistä. Samalla tämä edellytti tutkijalta reflektiivisyyttä ja tietoisuutta omasta roolistaan.

6.2.2 Fokusryhmähaastattelu

Fokusryhmähaastattelu on laadullinen tutkimusmenetelmä, jossa tutkittavia haastatellaan ryhmässä tutkijan ohjaaman keskustelun avulla. Tavoitteena on kerätä monipuolista ja rikasta aineistoa osallistujien kokemuksista ja näkemyksistä. (Mäntyranta & Kaila, 2008, 1.) Menetelmä soveltuu hyvin esimerkiksi tutkimuksen alkuvaiheeseen ilmiön jäsentämiseksi (Hirsjärvi & Hurme, 2022, luku 5.3 Yksilö- vai ryhmähaastattelu).

Haastattelun onnistumiseen vaikuttavat ryhmän koko (yleensä 4–10 osallistujaa), ryhmän kokoonpano sekä haastattelijan ohjaustaidot. Haastattelijan tulee luoda turvallinen ilmapiiri, huolehtia, että kaikki osallistujat tulevat kuulluiksi (Mäntyranta & Kaila, 2008, 3.) ja ohjata keskustelua teemojen mukaisesti (Gundumogula, 2020, 2).

Haastattelu tallennettiin ja litteroitiin, koska ryhmämäisestä keskustelusta syntyvä aineisto on laaja ja rikas, eikä sitä ole mahdollista kirjata luotettavasti muistiin vain havaintojen avulla (Mäntyranta & Kaila, 2008, 4–5).

6.2.3 Kehitystyöpaja

Kehitystyöpaja on järjestelmällinen kokonaisuus, jossa osallistujaryhmä työskentelee yhdessä tietyn ongelman ratkaisemiseksi tai ennalta asetetun tavoitteen saavuttamiseksi. Työpaja eroaa perinteisestä kokouksesta pidemmän keston (vähintään 2–3 tuntia) ja osallistavien, luovuutta tukevien menetelmien käytön vuoksi. Työpajatyöskentelyn onnistuminen edellyttää osallistujien huolellista valintaa, sillä mukana tulisi olla kaikki käsiteltävään aiheeseen liittyvät tahot ja ryhmän moninaisuus rikastaa keskustelua ja näkökulmia (Pavelin ym., 2014).

Työpaja vaatii myös perusteellisen ennakkosuunnittelun: tavoitteet, tehtävät, rakenne ja tilajärjestelyt tulee valmistella etukäteen, jotta osallistujien asiantuntemus voidaan hyödyntää mahdollisimman tehokkaasti (Forsström, 2021). Yhteinen työpajatyöskentely mahdollistaa eri toimijoiden ideoiden ja tarpeiden kokoamisen samaan tilaisuuteen, mikä tekee menetelmästä erityisen toimivan kehittämishankkeissa (Lippo ym., 2022).

Fasilitaattori vastaa työpajan ohjaamisesta ja siitä, että kaikkien osallistujien näkemykset tulevat esille. Hän rakentaa työskentelylle selkeän prosessin ja auttaa pitämään keskustelun tavoitteellisena. (Forsström, 2021; Nummi, 2018, 31.) Riittävä ennakkovalmistelu on erityisen tärkeää, sillä heikko suunnittelu voi heikentää osallistujilta saatavan tiedon laatua (Forsström, 2021).

Tämä menetelmä valittiin, koska työpaja mahdollistaa osallistuvan ja yhteisöllisen työskentelyn, jossa tiimin jäsenet voivat tuottaa konkreettisia kehittämissuhteita yhdessä ja arvioida niitä systemaattisesti. Tämä tukee toimintatutkimuksen tavoitteita sekä käytännönläheistä kehittämistä.

6.3 Aineiston analysointi

Laadullisen aineiston analyysin tavoitteena on muuttaa kerätty tieto ymmärrettäväksi kokonaisuudeksi, joka vastaa tutkimuskysymyksiin (Kananen, 2014, 104–105). Analyysiprosessi koostui

useista vaiheista, jotka vaihtelivat aineiston tyyhin mukaan, mutta noudattivat yhteistä periaatetta: aineisto järjestettiin, pelkistettiin ja tulkittiin osaksi tutkimuksen kokonaisuutta.

Fokusryhmähaastattelun aineisto analysoitiin perinteisen laadullisen analyysin mallin mukaisesti. Ensimmäisessä vaiheessa äänitetty keskustelu litteroitiin tekstimuotoon, mikä mahdollisti sen systemaattisen tarkastelun (Hirsjärvi & Hurme, 2022, luku 7.2.1 Aineiston litterointi). Litteroinnin jälkeen aineisto koodattiin. Koodaus eteni raakakoodien muodostamisesta niiden ryhmittelyyn laajemmiksi luokiksi ja teemoiksi, kuten laadullisessa analyysissä on tyypillistä (Kananen, 2014, 107–109). Koodauksen pohjalta aineisto tulkittiin deduktiivisesti tutkimuksen teoreettisen viitekehyksen kautta, mutta analyysissä hyödynnettiin myös aineistolähtöistä tarkastelua (Kananen, 2014, 109–110).

Retrospektiivien aineisto kerättiin virtuaaliselle valkotalulle, johon osallistujat tuottivat kehitysideoita. Aineisto oli luonteeltaan tiivistä ja valmiiksi jäsenneiltyä, minkä vuoksi analyysi kohdistui sen kokoamiseen ja pelkistämiseen. Kehitysehdotuksista muodostettiin sisällöllisesti samankaltaisia kokonaisuuksia, joista koottiin tutkimusta varten keskeiset pääteemat. Prosessi muistutti laadullisen analyysin teemoittelua, mutta oli luonteeltaan kevyempi, koska aineiston määrä oli rajallinen ja sen sisältö ytimekästä.

Lisäksi tutkimuksessa hyödynnettiin lähtötilannetta kuvaavaa aineistoa, joka koostui projektin aikatauluista ja toteutuksesta. Aineisto kerättiin organisaation kalenterikutsuista ja koottiin taulukkomuotoon. Tämän aineiston analyysi perustui aineiston järjestämiseen ja vertailuun, ja sen avulla luotiin kokonaiskuva kvartaalitestauksen alkuvaiheen rakenteista ja ajallisesta etenemisestä. Vaikka aineisto oli määrällisesti vähäinen ja selkeä, sen yhtenäinen esittäminen oli tärkeää tutkimuksen kontekstin ymmärtämiseksi.

Näiden analyysivaiheiden kautta tutkimuksen eri aineistoista muodostettiin toisiaan täydentävä kokonaisuus, joka tuki sekä tutkimuskysymyksiin vastaamista että kehittämistyön etenemistä.

7 Toimeksiantaja ja toimintatutkimuksen rakenne

Kehittämistehtävän tarkoituksena oli parantaa HUS Muiden HR-järjestelmien tuki -tiimin kvartaalitestausprosessia hyödyntämällä Scrumban-menetelmää. Tavoitteena oli ottaa sidosryhmät aktiivisesti mukaan ja yhdenmukaistaa työvaiheet, jotta työskentelyprosesseja voitaisiin tehdä selkeämmiksi ja vähentää hukkaa.

Kehittämistehtävällä oli kaksi tutkimuskysymystä. Ensimmäinen kysymys antoi mahdollisuuden tutkia, miten Scrumban voi vaikuttaa tiimin sisäistä yhteistyötä ja kommunikaatiota, mikä voi johtaa tehokkaampaan testausprosessiin. Toisen kysymyksen avulla tutkittiin, miten Scrumban voi vaikuttaa testausprosessin sujuvuuteen ja työn hallintaan.

Tutkimuskysymykset ovat:

1. Miten Scrumban-menetelmän käyttö vaikuttaa tiimityön tehokkuuteen ja yhteistyöhön kvartaalitestauksessa?
2. Miten Scrumban-menetelmän käyttö vaikuttaa kvartaalitestauksen tehokkuuteen ja hallittavuuteen?

Tutkimus rajattiin kvartaalipäivityksiin, jotka toistuvat neljä kertaa vuodessa ja vaikuttavat laajasti muihin järjestelmiin ja sidosryhmiin. Opinnäytetyö toteutettiin toimintatutkimuksena, jossa korostuu tiimin osallistuminen kehitystyöhön. Tutkija toimi itse tiimin jäsenenä, mikä mahdollisti syvällisen ymmärryksen prosessista ja nopean reagoinnin havaittuihin haasteisiin.

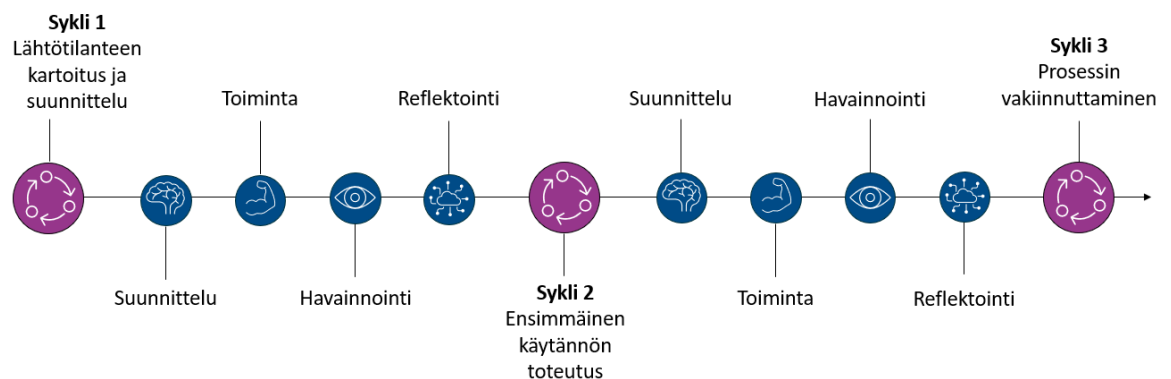
Tutkimus eteni toimintatutkimukselle tyypilliseen tapaan iteratiivisesti kolmessa syklissä, joissa jokaisessa tehtiin jatkuvaa kehitystyötä ja parannusta. Iteratiivinen lähestymistapa valittiin, koska se tukee ketterää kehittämistä ja mahdollistaa oppimisen jokaisessa vaiheessa. Lisäksi lähestymistapa mahdollisti kehittämistoimenpiteiden joustavan mukauttamisen tutkimuksen aikana esiin nousseiden havaintojen perusteella.

Tämä malli soveltui erityisen hyvin kvartaalitestauksen kehittämiseen, koska se mahdollisti vaiheittaisen etenemisen ja jatkuvan palautteen hyödyntämisen. Jokaisessa syklissä suunnittelu, toiminta, havainnointi ja reflektointi toistuivat, mikä loi rakenteen systemaattiselle kehittämiselle.

Kuva 8 esitetään opinnäytetyön eteneminen kolmen syklin mallissa:

1. Sykli: Lähtötilanteen kartoitus ja suunnittelu
2. Sykli: Ensimmäinen käytännön toteutus
3. Sykli: Prosessin vakiinnuttaminen

Jokainen sykli sisälsi suunnittelun, toiminnan, havainnoinnin ja reflektoinnin, mikä mahdollisti jatkuvan parantamisen.



Kuva 8. Kolmen syklin kehitysprosessi. Iteratiivinen eteneminen kolmessa vaiheessa: lähtötilanteen kartoitus, käytännön toteutus ja prosessin vakiinnuttaminen.

7.1 Toimeksiantajan kuvaus

Tämä kehittämistyö on tehty HUS Helsingin yliopistolliselle sairaalalle, joka on Suomen suurin terveydenhuoltoalan toimija. HUS vastaa Suomen vaativimmasta erikoissairaanhoidosta ja on keskitetty valtakunnallisesti useiden harvinaisten sekä haastavien sairauksien hoidosta ja tutkimuksesta. HUSissa hoidetaan vuosittain noin 700 000 potilasta ja työllistää noin 27 500 työntekijää. (HUS, 2025a.; HUS, 2025b.) Toimintaa ohjaavat arvot ovat kohtaaminen, edelläkävijyys ja yhdenvertaisuus. (HUS, 2025b.)

Toimeksiantajana tässä kehittämistyössä toimii HUS Konsernipalvelut, joka on yksi HUSin tulosalueista. HUSin Konsernipalveluihin kuuluu kuusi tulosyksikköä: Konsernihallinto, Apteekki, Runkopalvelut, Tietohallinto, Kiinteistö- ja toimitilapalvelut sekä Työterveys. Henkilöstö ja johtami-

nen vastuualue on osa Konsernihallinnon tuottamia palveluja, jossa vastataan HUS-yhtymän henkilöstöjohtamisen ohjauksesta ja kehittämisestä, työnantajatoiminnasta sekä henkilöstöpolitiikan kehittämisestä. (Auranne, 25.9.2024.)

Kehittämistyö keskittyy Henkilöstö ja johtaminen vastuualueeseen kuuluvan Muiden HR-järjestelmien tuki vastuuyksikköön. HR-järjestelmien pääkäyttäjät vastaavat niiden ylläpidosta ja kehittämisestä yhteistyössä Tietohallinnon ja toimittajien kanssa. (Auranne, 25.9.2024.)

7.2 Kehitystyön syklien kuvaus

Kvartaalitestauksen kehitystyö eteni iteratiivisesti toimintatutkimukselle tyypillisten syklien kautta. Kukin sykli sisälsi suunnittelun, toiminnan, havainnoinnin ja reflektoinnin vaiheet, joiden avulla testausprosessia kehitettiin iteratiivisesti saatujen havaintojen ja palautteiden perusteella. Kolmen syklin malli valittiin, koska se mahdollisti vaiheittaisen etenemisen ja jatkuvan oppimisen, mikä on keskeistä ketterässä kehittämisessä.

Iteratiivinen lähestymistapa tuki tutkimuskysymyksiä siten, että jokaisessa syklissä voitiin arvioida sekä tiimityön sujuvuutta että prosessin hallittavuutta käytännön toimenpiteiden avulla. Näin varmistettiin, että kehitystyö ei jää teoreettiseksi, vaan tuotti konkreettisia parannuksia jokaisessa vaiheessa.

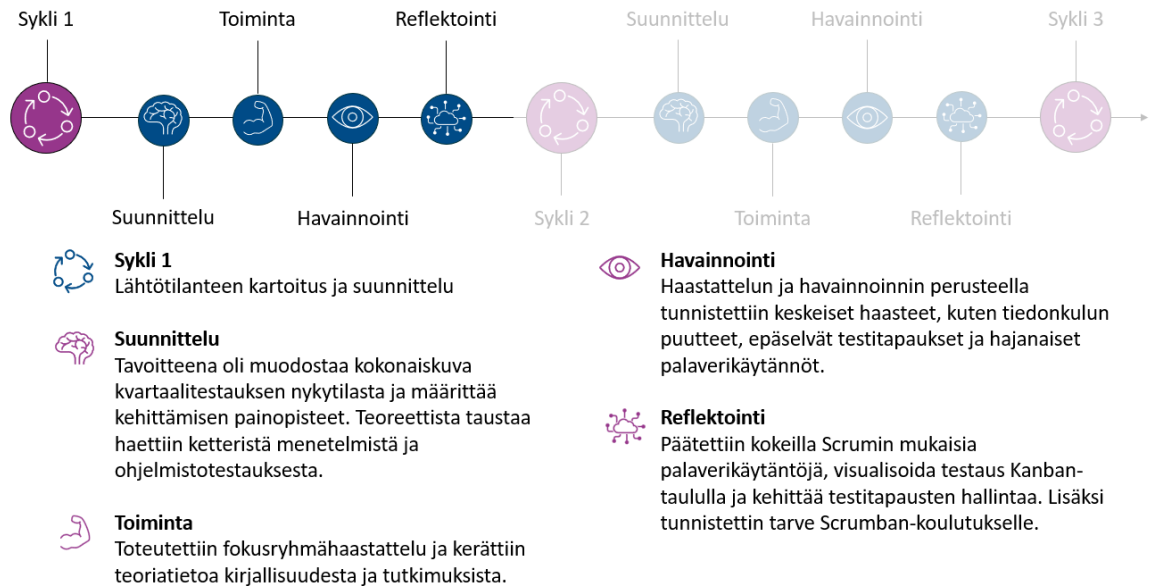
Tutkimusongelmaa lähestyttiin kolmen syklin kautta:

1. Sykli: Lähtötilanteen kartoitus ja suunnittelu
2. Sykli: Ensimmäinen käytännön toteutus
3. Sykli: Prosessin vakiinnuttaminen

Tässä luvussa esitetään kolmen kehityssyklin tiivistetyt kuvaukset. Tarkoituksena on antaa kokonaiskuva prosessin etenemisestä. Varsinaiset tulokset käsitellään luvussa 8 ja tulosten analyysi luvussa 9.

7.2.1 Sykli 1 kuvaus: Lähtötilanteen kartoitus ja suunnittelu

Ensimmäisen syklin tavoitteena oli ymmärtää HCM Core -kvartaalitestauksen lähtötilanne keskit-
tyen palaverikäytäntöihin, tiedonkulkuun sekä testauksen suunnitteluun. Koko syklin eteneminen
on esitetty tiivistetysti Kuva 9.



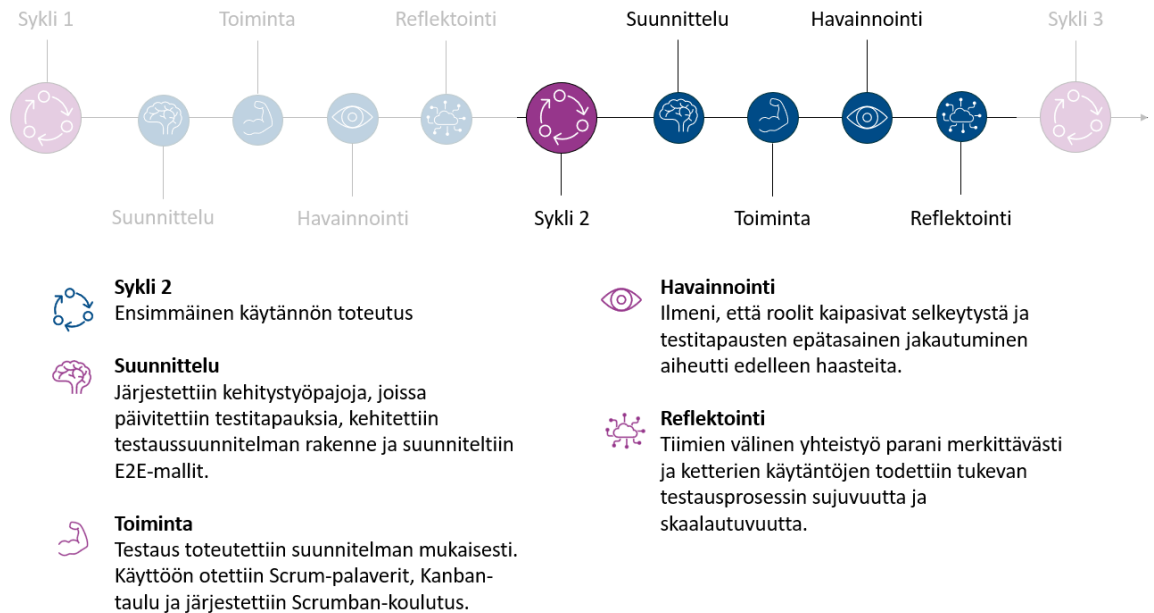
Kuva 9. Sykli 1: Lähtötilanteen kartoitus ja suunnittelu

Toteutus sisälsi lähtötilanteen kartoituksen fokusryhmähaastattelulla, johon osallistui kahdeksan eri sidosryhmien edustajaa. Haastattelun avulla täsmennettiin tutkimusongelma ja tutkimuskysymykset. Teoreettista taustaa haettiin ketteristä menetelmistä ja ohjelmistotestauksesta.

Keskeiset havainnot osoittivat merkittäviä haasteita tiedonkulussa, palaverikäytännöissä ja testitapausten hallinnassa. Testauksen aikataulua ehdotettiin lyhennettäväksi ja Jira Xray-työkalun käyttö koettiin hajanaiseksi. Merkityksenä oli, että reflektiovaiheessa päätettiin kokeilla Scrumin mukaisia palaverikäytäntöjä, visualisoida testaus Kanban-taululla, laatia tarkempi testausuunnitelma ja päivittää testitapaukset. Lisäksi tunnistettiin tarve Scrumban-koulutukselle.

7.2.2 Sykli 2 kuvaus: Ensimmäinen käytännön toteutus

Toisen syklin tavoitteena oli viedä kvartaalitestaustusprosessi ensimmäistä kertaa käytäntöön hyödyntäen ensimmäisen syklin havaintoja ja teoriaa. Koko syklin eteneminen on esitetty tiivistetysti Kuva 10.

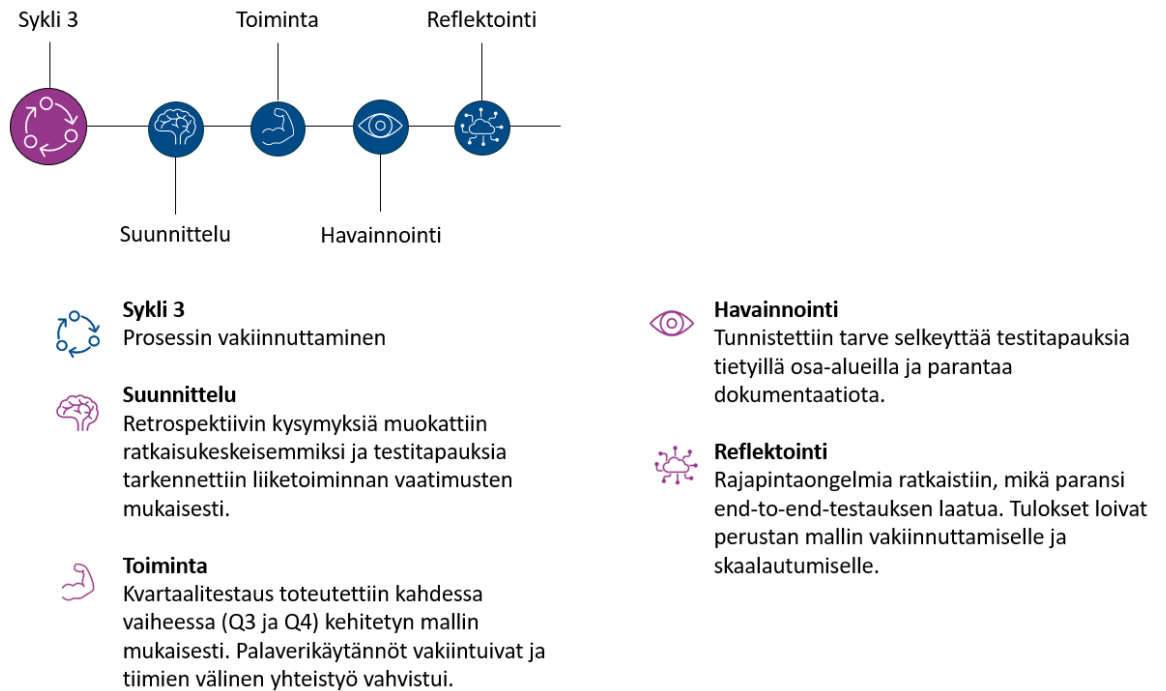


Kuva 10. Sykli 2: Ensimmäinen käytännön toteutus

Toteutus sisälsi kehitystyöpajoja, joissa päivitettiin testitapauksia, kehitettiin testaussuunnitelman rakennetta ja suunniteltiin E2E-testauksen toimintamalleja yhdessä sidosryhmien kanssa. Lisäksi toteutettiin Scrumban-koulutus tiimin ketterän osaamisen vahvistamiseksi. Testauksen toteutuksessa hyödynnettiin Scrumin palaverikäytäntöjä, kuten suunnittelupalaveria, päivittäispalavereja ja retrospektiiviä. Kanban-taulu otettiin käyttöön testauksen visualisointiin ja tehtävien hallintaan. Keskeiset havainnot osoittivat, että roolien määrittely kaipasi tarkennusta ja testitapausten epätasainen jakautuminen aiheutti edelleen haasteita. Merkityksenä oli, että tiimien välinen yhteistyö parani merkittävästi ja ketterät käytännöt todettiin toimiviksi, mikä loi vahvan pohjan kolmannelle syklille.

7.2.3 Sykli 3 kuvaus: Prosessin vakiinnuttaminen

Kolmannen syklin tavoitteena oli testausprosessin kehittäminen ja mallin vakiinnuttaminen. Koko syklin eteneminen on esitetty tiivistetysti Kuva 11.



Kuva 11. Sykli 3: Prosessin vakiinnuttaminen

Toteutus sisälsi kaksi erillistä kvartaalitestausta (Q3 ja Q4), jotka perustuivat aikaisemmista vaiheista saatujen havaintojen ja kehitettyjen käytäntöjen hyödyntämiseen. Retrospektiivin kysymyksiä muokattiin ratkaisukeskeisemmiksi ja testitapauksia tarkennettiin liiketoiminnan vaatimusten mukaisesti. Kvartaalitestausta toteutettiin kehitetyn mallin mukaisesti kahdessa vaiheessa. Palaverikäytännöt vakiintuivat ja tiimien välinen yhteistyö vahvistui.

Keskeiset havainnot osoittivat tarpeen selkeyttää testitapauksia tietyillä osa-alueilla ja parantaa dokumentaatiota. Rajapintaongelmia ratkaistiin, mikä paransi E2E-testauksen laatua. Merkityksenä oli, että tulokset loivat perustan mallin vakiinnuttamiselle ja skaalautumiselle.

8 Sykliä toteutus ja tulokset

Kvartaalitestauksen kehittämistyö toteutettiin toimintatutkimuksena, jossa prosessia tarkasteltiin ja kehitettiin iteratiivisesti kolmen syklin kautta. Kukin sykli sisälsi suunnittelun, toiminnan, havainnoinnin ja reflektoinnin vaiheet, joiden avulla testausprosessia pyrittiin systemaattisesti parantamaan saatujen havaintojen ja palautteiden perusteella. Ensimmäinen sykli keskittyi lähtötilanteen kartoittamiseen ja teoreettisen perustan luomiseen, toinen käytännön toteutukseen ja kolmas prosessin vakiinnuttamiseen. Tässä luvussa kuvataan kunkin syklin toteutus ja niissä syntyneet tulokset.

8.1 Sykli 1 – Lähtötilanteen kartoitus ja suunnittelu

Ensimmäisessä syklissä kartoitettiin HCM Core -kvartaalitestauksen lähtötilanne fokusryhmähaastattelun avulla. Nykytilannetta tarkasteltiin myös aiempien palaverikutsujen perusteella. Teoreettista taustaa kerättiin ketteristä menetelmistä ja ohjelmistotestauksesta, ja niiden soveltuvuutta arvioitiin kvartaalitestauksen kehittämisessä.

8.1.1 Tiedon hankinta

Ensimmäisen syklin ja koko opinnäytetyön aikana teoretietoa hyödynnettiin monipuolisesti eri lähteistä, kuten kirjallisuudesta, väitöskirjoista, tieteellisistä artikkeleista ja alan julkaisuista. Tavoitteena oli saada kattava ymmärrys ketteristä menetelmistä, kuten Scrum, Kanban ja Scrumban, sekä ohjelmistotestauksen käytännöistä. Näiden tietojen pohjalta pyrittiin arvioimaan, miten ketterät menetelmät ja Lean-ajattelu voisivat hyödyttää kvartaalitestausprosessin kehittämistä ja parantamista. Fokusryhmähaastattelun tulokset vahvistivat ennako-oletuksen siitä, että erityisesti Scrumbanin yhdistäminen kvartaalitestaukseen sopisi erinomaisesti. Lisäksi ohjelmistotestauksen teoriasta saatiin vahvistusta siihen, miten testauksen hallintaa ja yhteistyötä voisi kehittää. Myös Lean-ajattelun vahvistaminen lisää yhteistyötä tiimin sekä sidosryhmien välillä.

Ketterät menetelmät, kuten Scrum ja Kanban tarjoavat joustavia ja iteratiivisia lähestymistapoja kvartaalitestaukseen. Scrum keskittyy tiimityöhön, säännöllisiin sprintteihin ja palaverikäytäntöi-

hin sekä jatkuvaan parantamiseen, kun taas Kanban korostaa työn kulun visualisointia, läpime-
noajan mittaamista sekä työn alla olevien tehtävien rajoittamista. Scrumban yhdistää näiden kah-
den menetelmän parhaat puolet, tarjoten joustavuutta ja tehokkuutta. Lean-ajattelu puolestaan
keskittyy arvon tuottamiseen ja hukan minimointiin mikä voi parantaa testausprosessin tehok-
kuutta.

Ohjelmistotestauksen teoriasta etsittiin parhaita käytäntöjä ja menetelmiä, jotka voisivat paran-
taa ja kehittää kvartaalitestauksia. Hyvin määritellyt testausstrategiat- ja suunnitelmat auttavat
varmistamaan, että kaikki tärkeät testitapaukset- ja skenaariot tulevat katetuiksi. Testausstrate-
gioiden avulla voidaan myös priorisoida testausresursseja ja -aikaa tehokkaammin.

Regressiotestaus on tärkeä osa testausprosessia, sillä sen avulla varmistetaan, että uudet muu-
tokset eivät riko olemassa olevaa toiminnallisuutta. E2E-testaus puolestaan varmistaa, että koko
järjestelmä toimii odotetusti eri järjestelmien välillä. Näiden testausmenetelmien avulla voidaan
parantaa ohjelmiston laatua ja luotettavuutta.

Monitiimitestaus korostaa eri tiimien välistä yhteistyötä ja tiedon jakamista, mikä voi parantaa
testauksen kattavuutta ja tehokkuutta. Testiautomaatio puolestaan voi merkittävästi parantaa
testauksen tehokkuutta ja kattavuutta, erityisesti toistuvissa ja rutiininomaisissa testeissä. Auto-
matisoidut testit voivat myös vähentää inhimillisten virheiden riskiä ja vapauttaa resursseja mo-
nimutkaisempien testien suorittamiseen.

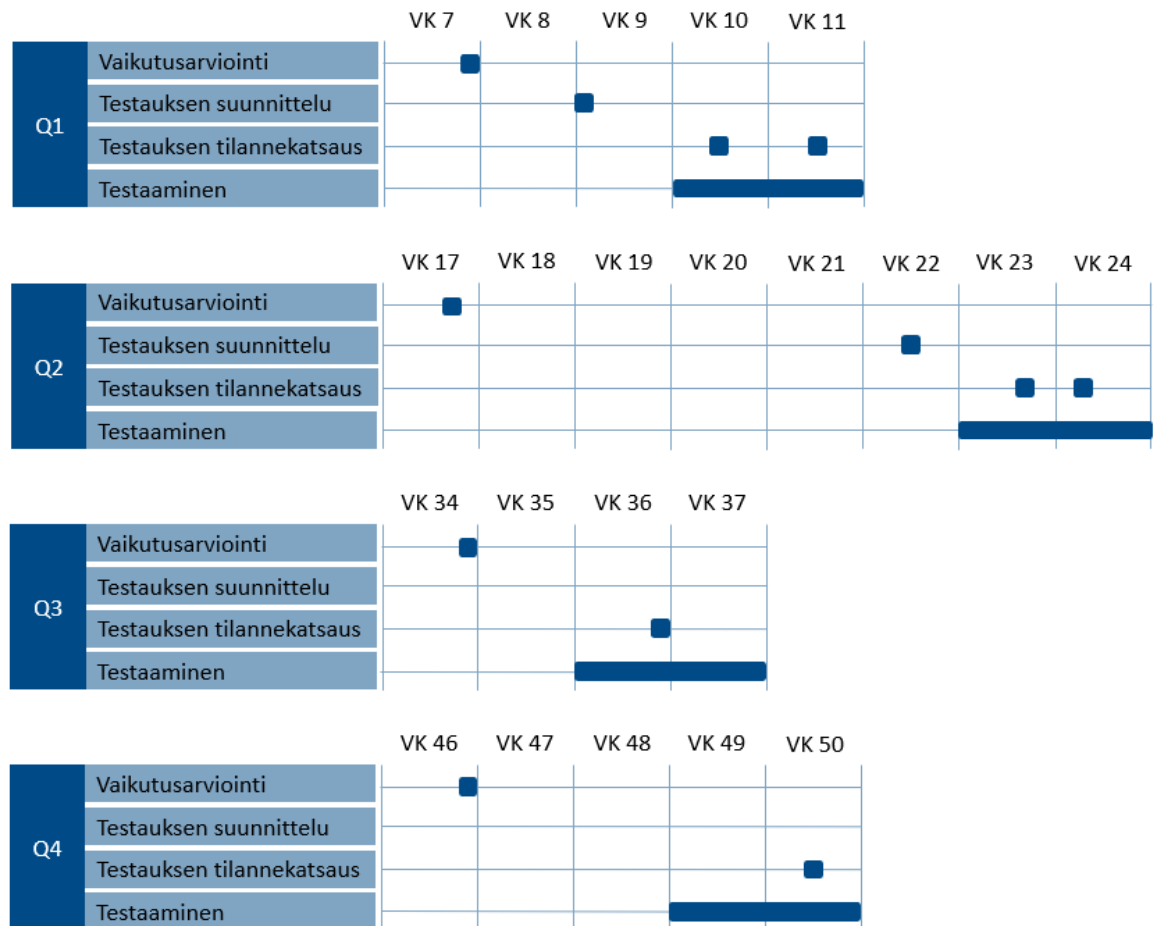
Näiden tietojen ja menetelmien avulla pyrittiin kehittämään kvartaalitestausprosessia siten, että
se olisi joustavampi, tehokkaampi ja paremmin hallittavissa. Ketterien menetelmien ja ohjelmis-
totestauksen parhaiden käytäntöjen yhdistäminen tarjoaa vahvan perustan jatkuvalla parantami-
selle ja laadukkaalle testaamiselle.

8.1.2 Kvartaalitestauksen lähtötilanteen aikataulu ja toteutus

Lähtötilannetta tarkasteltiin aiempien kvartaalitestauksen palaverikutsujen ja niihin perustuvan
aikataulun avulla. Tarkastelun tavoitteena oli muodostaa kokonaiskuva testausprosessin toistu-
vuudesta ja suunnitelmallisuudesta.

Kuva 12 on esitetty kvartaalitestauksen palaverien aikataulu, joka on koostettu aiempien palave-
rikutsujen perusteella. Taulukko havainnollistaa, miten vaikutusarvioinnit, suunnittelupalaverit ja

tilannekatsaukset sijoittuvat eri tavoin kunkin kvartaalin sisällä. Esimerkiksi kvartaalissa Q1 vaikutusarviointi toteutettiin viikolla 7 ja tilannekatsaukset viikoilla 10 ja 11, kun taas kvartaalissa Q3 vastaavat vaiheet ajoittuivat viikolle 34–37. Aikataulujen vertailu osoittaa, ettei kvartaalien välillä ole muodostunut yhtenäistä rakennetta tai toistuvaa rytmiä, mikä korostaa lähtötilanteen hajainaisuutta.



Kuva 12. Lähtötilanteen palaverikäytännöt ja testauksen aikataulu

Aineisto osoitti, että vaikka kvartaalien (Q1–Q4) rakenteessa toistuvat samat päävaiheet, kuten vaikutusarviointi, testauksen suunnittelu, testauksen tilannekatsaukset ja testaaminen, toteutus-tapa ja ajoitus vaihtelivat huomattavasti eri kvartaaleissa.

Vaikutusarviointien ja testauksen suunnittelun ajankohdat erosivat toisistaan, ja tilannekatsaus-ten määrä sekä aikaväli vaihtelivat ilman selkeää yhteistä rytmiä. Joissakin kvartaaleissa tilannekatsauksia järjestettiin kaksi, toisissa vain yksi. Myöskään palaverien sisällöissä tai rakenteissa ei ollut yhtenäistä mallia, vaan palaverirunko ja käsiteltävät asiat vaihtelivat kerrasta toiseen.

Kokonaisuutena lähtötilanteen kvartaalitestauksen kokonaisuus näyttäytyi hajanaisena ja osin epäyhtenäisenä. Prosessin toistettavuus ja ennakoitavuus olivat rajallisia, mikä vaikeutti kokonaisuuden hallintaa ja testausprosessin kehittämistä.

8.1.3 Fokusryhmähaastattelun valmistelu

Fokusryhmähaastattelun tavoitteena oli selvittää kvartaalitestausprosessin lähtötilanne sekä siihen liittyviä haasteita ja kehitysmahdollisuuksia. Erityisesti haluttiin ymmärtää, miten tiedonkulku, testauksen aikataulu, testitapausten hallinta ja palaverikäytännöt vaikuttavat tiimin tehokkuuteen ja työn sujuvuuteen. Lisäksi haastattelussa kartoitettiin, millaisia muutoksia tiimi itse kokee tarpeellisiksi kvartaalitestauksen kehittämiseksi.

Haastattelu toteutettiin Teamsin välityksellä, ja siihen osallistui kahdeksan tutkimuksen kohteena olevaa tiimin jäsentä eri sidosryhmistä.

Haastattelukutsu lähetettiin osallistujille kaksi viikkoa etukäteen ja kutsussa kerrottiin haastattelun tavoitteista ja siitä, että vastauksia käsitellään anonyymisti tiimitasolla. Lisäksi kerrottiin, että osallistuminen haastatteluun on vapaaehtoista. Kutsussa oli myös ennakkoon haastattelukysymykset:

- Mitä haasteita nykyisissä kvartaalitestauspalavereissa esiintyy?
- Tukeeko kvartaalitestauspalavereista saatu informaatio testausprosessia?
- Mitä parannuksia toivoisit kvartaalitestauspalavereihin?
- Ovatko Jira Xrayn testitapaukset riittävän kattavat?
- Mitä haasteita Jira Xrayn testitapausten ajamisessa esiintyy?
- Mitä haasteita on Jira Xray testitapausten jakamisessa ja priorisoinnissa?
- Miten testausprosessia voisi selkeyttää?
- Mitä haasteita testausprosessissa esiintyy?
- Miten testausprosessia voisi selkeyttää/parantaa?

Fokusryhmähaastattelun luotettavuutta varmistettiin tallentamalla keskustelut osallistujien suostumuksella. Tallenteet tallennettiin organisaation ohjeistaman tietoturvallisen alustan kautta, ja ne poistetaan tutkimuksen päätyttyä. Haastatteluaineisto litteroitiin ja analysoitiin ilman, että kysymysasettelua tai vastauksia muutettiin, jotta tutkimuksen tulokset heijastaisivat mahdollisimman tarkasti osallistujien näkemyksiä.

8.1.4 Fokusryhmähaastattelun tulokset

Fokusryhmähaastatteluun osallistui kahdeksan henkilöä eri sidosryhmistä, joilla kaikilla oli kokemusta HCM Core-testauksesta. Mukana oli esihenkilö, testauskoordinaattori ja kuusi testaajaa, joista yksi toimi myös testausvastaavana. Haastattelu aloitettiin käymällä kysymykset yksitellen läpi, jolloin jokainen sai ilmaista oman mielipiteensä. Melko nopeasti keskustelu muuttui vapaammaksi, ja fasilitaattori huolehti siitä, että kaikki kysymykset käsiteltiin. Vapaampi keskustelutyylisopi ryhmälle paremmin, ja saatiin aikaan hedelmällistä ja kunnioittavaa keskustelua.

Haastatteluaineisto litteroitiin ja jaettiin teema-alueittain analysointia varten. Yksittäisten osallistujien näkemykset yleistettiin tiimitasolle, ja aineistosta poistettiin vastaukset, jotka eivät suoraan liittyneet tutkimuskysymyksiin. Tulosten esittämisessä on käytetty valikoituja kommentteja, jotka edustavat tyypillisiä näkemyksiä ja havainnollistavat keskeisiä teemoja. Kaikkia yksittäisiä vastauksia ei sisällytetty raporttiin, jotta kokonaisuus pysyy selkeänä ja tutkimuskysymyksiin keskittyvänä. Valinnassa on huomioitu monipuolisuus ja eri sidosryhmien äänen kuuluminen. Karsittua aineistoa voidaan hyödyntää erikseen tiimin sisäisessä kehittämisessä.

Tulokset on jaoteltu keskeisiin teemoihin, jotka nousivat esiin haastatteluiden perusteella. Teemoittelun esimerkki on esitetty liitteessä 4. Alla on esitetty haastatteluiden tulokset teemoittain.

1. Pääteema: Palaverikäytännöt ja tiedonkulku

a. Alateema: Toimittajan palaverin hyödyttömyys

- i. Toimittajan palaveri ei tuo lisäarvoa, koska sama PowerPoint on jo nähtävissä Jira-tiketillä.

”En oikein näe hyötyä palaverissa, kun sama PowerPoint löytyy jo Jirasta.”

b. Alateema: Tarve testauksen suunnittelupalaverille

- i. Testauksen suunnittelupalaveria toivotaan takaisin, jotta tiimi voi itse käydä vaikutukset läpi ennen testauksen aloitusta.

"Kaipaam palaveria, jossa yhdessä käydään läpi testauksen alkua ja jaetaan testitapaukset."

c. Alateema: Tiedonkulun haasteet

- i. Tiedonkulku testaamisen etenemisestä tiimin kesken on puutteellista.

"Ei ole selkeää, missä vaiheessa testaus menee, kun tieto ei kulje tiimin sisällä."

- ii. Sidosryhmille ei välity tieto E2E-testauksesta.

"E2E-testauksesta ei tule tietoa ajoissa sidosryhmille."

- iii. Loppukäyttäjille ei mene tieto tulevista muutoksista.

"Loppukäyttäjille ei kerrota tulevista muutoksista."

2. Pääteema: Testausprosessin tehokkuus ja aikataulu

a. Alateema: Testausaikataulun haasteet

- i. Nykyinen testaamisaika on kaksi viikkoa, mutta se pitäisi lyhentää viikoon, jotta mahdollisten virheiden korjaukseen jää riittävästi aikaa.

"Testaus venyy yleensä ihan viimeisille päville ja mahdollisille korjauksille ei jää aikaa."

"Olisi parempi, jos testaus olisi viikko ja toinen viikko varattu korjauksille."

b. Alateema: Testitapausten jakautuminen ja selkeys

- i. Testitapaukset eivät jakaudu tasaisesti, osa huomattavasti haastavampia ja pidempiä.

”Osa testitapauksista on todella pitkiä ja haastavia, kun taas toiset ovat nopeita.”

”Testitapaukset eivät jakaudu tasaisesti.”

- ii. Testitapaukset ovat epäselviä, eikä kaikkia osata suorittaa niiden perusteella.

”Testitapaukset eivät ole ajan tasalla.”

- iii. Testitapauksista puuttuu tietoa siitä, millä roolilla testataan.

”Osassa testitapauksista puuttuu tieto rooleista, mikä hankaloittaa testausta.”

c. Alateema: Testauksen työkalut ja dokumentaatio

- i. Jira Xrayssa testitapaukset ovat sekaisin, mikä vaikeuttaa testiplanin koostamista.

”Testitapaukset Xrayssa pitäisi järjestää selkeämmin.”

- ii. Osa manuaalitestitapauksista ovat samoja kuin testiautomaatiotapauksissa, mikä aiheuttaa päällekkäisyyttä.

”Päällekkäisyys manuaali- ja automaatiotesteissä vie turhaan aikaa.”

8.1.5 Fokusryhmähaastattelusta vedetyt johtopäätökset

Fokusryhmähaastattelun perusteella testausprosessissa esiintyy useita kehityskohteita, joista keskeisimmät liittyvät tiedonkulkuun, palaverikäytäntöihin sekä testauksen suunnitteluun ja toteutukseen.

Tiedonkulku osoittautui merkittäväksi haasteeksi sekä tiimin sisällä että eri sidosryhmien välillä. Erityisesti testaamisesta ei ole riittävää tiedonvaihtoa tiimin kesken, ja loppukäyttäjille ei välity ajantasainen tieto tulevista muutoksista. Lisäksi toimittajan palaverin koetaan olevan tehoton, koska sen sisältö on jo saatavilla Jira-tiketillä. Sen sijaan testauksen suunnittelupalaveria toivotaan takaisin, jotta tiimi voisi paremmin valmistautua testaukseen ja arvioida sen vaikutuksia.

Testausprosessin aikataulu ja organisointi vaativat myös kehittämistä. Tällä hetkellä testaamiseen varattu aika (kaksi viikkoa) on liian pitkä, ja sitä tulisi lyhentää viikkoon, jotta mahdollisille virheiden korjaamiselle jää enemmän aikaa. Lisäksi testitapaukset eivät jakaudu tasaisesti: osa on huomattavasti työläämpiä kuin toiset, ja niiden epäselvyys vaikeuttaa suorittamista. Myös testitapausten hallinta ja dokumentaatio kaipaavat selkeyttämistä, esimerkiksi Xray-työkalussa testitapaukset ovat hajallaan, mikä hankaloittaa testauksen suunnittelun koostamista.

8.2 Sykli 2 – Ensimmäinen käytännön toteutus

Toisessa syklissä kvartaalitestausprosessi vietiin ensimmäistä kertaa käytäntöön. Toteutus käynnistyi Scrumban-koulutuksella, jonka tavoitteena oli vahvistaa tiimin ketterän testauksen osaamista. Tämän jälkeen järjestettiin useita kehitystyöpajoja, joissa kehitettiin testaussuunnitelman rakennetta, päivitettiin testitapauksia ja suunniteltiin E2E-testauksen toimintamalleja yhdessä sidosryhmien kanssa.

Testauksen toteutuksessa hyödynnettiin Scrumin palaverikäytäntöjä, kuten suunnittelupalaveria, päivittäispalavereja ja retrospektiiviä. Kanban-taulu otettiin käyttöön testauksen visualisointiin ja tehtävien hallintaan.

8.2.1 Scrumban-koulutus

Kvartaalitestauksista kehitettäessä keskeisessä osassa oli Scrumin ja Kanbanin käyttöönotto sekä tiimityön parantaminen. Ketterät menetelmät, kuten Scrum ja Kanban, tarjoavat joustavia ja tehokkaita työkaluja testauksen hallintaan, mutta tiimimme ei ollut aikaisemmin käyttänyt näitä lähestymistapoja. Tämän vuoksi koimme tärkeäksi järjestää Scrumban-koulutuksen, jotta tiimimme voisi saada tarvittavat taidot ja ymmärryksen ketteristä menetelmistä.

Koulutuksen järjestäjäksi saimme toisen sidosryhmän henkilön, joka toi mukanaan arvokasta asiantuntemusta. Koulutus oli kattava, kattaen aiheita kuten Agile-manifesti, Scrum-prosessin vaiheet, Kanban-tilojen käyttö ja Lean-ajattelun periaatteet. Osallistujat saivat käytännönläheisiä esimerkkejä siitä, miten ketteriä menetelmiä voidaan soveltaa päivittäisessä työssä, mikä auttoi tiimiämme ymmärtämään niiden merkityksen ja hyödyt.

Koulutukseen osallistui myös muutamia sidosryhmien edustajia, joiden kanssa tehdään yhteistyötä E2E-testauksessa.

8.2.2 Testaussuunnitelman kehitystyöpaja

Testaussuunnitelman kehitystyöpajassa keskityttiin kehittämään kvartaalitestauksessa käytettävää testaussuunnitelmapohjaa. Työpajaan osallistui yhteensä neljä henkilöä kahdesta eri tiimistä, joiden tehtävänä oli suunnitella ja koordinoita testauksen toteutusta. Lähtötilanteessa HCM Core -järjestelmän kvartaalitestaukselle ei ollut olemassa erillistä, systemaattista testaussuunnitelmaa, mikä hankaloitti testauksen suunnittelua ja organisointia.

Työpajan aluksi selvitettiin käytössä oleva HUSin testaussuunnitelman pohja, jonka sisältö ja rakenne käytiin yksityiskohtaisesti läpi yhdessä osallistujien kesken. Tarkoituksena oli arvioida pohjan soveltuvuutta HCM Core -järjestelmän tarpeisiin sekä kartoittaa mahdolliset kehitystarpeet nykyisessä suunnitelmassa. Keskustelujen yhteydessä tunnistettiin useita kohtia, ja niistä tehtiin työpajan aikana muistiinpanoja jatkotyöstöä varten.

Työpajan lopputuloksena testaussuunnitelman pohjaa päivitettiin siten, että se tukee HCM Core -järjestelmän kvartaalitestaukselta entistä paremmin. Päivityksissä huomioitiin muun muassa testauksen laajuus, eri testausvaiheiden aikataulut, vastuuhenkilöiden määrittely sekä testauksen hyväksymiskriteerit. Näin varmistettiin, että testaussuunnitelma tarjoaa selkeän ja yhtenäisen rakenteen testauksen suunnittelulle ja toimii tehokkaana työkaluna testauksen toteutuksen ohjauksessa. Testaussuunnitelman sisällysluettelosta kuva liitteessä 3.

8.2.3 Testitapausten päivittämisen kehitystyöpaja

Testitapausten päivittämisen kehitystyöpajassa keskityttiin parantamaan testitapausten laatua ja hallintaa Jira Xray -testienhallintajärjestelmässä. Kehitystyöpajaan osallistui yhteensä neljä henkilöä kahdesta eri tiimistä, joiden vastuulla oli testitapausten ylläpito ja käyttö. Työpajan tarkoituksena oli käydä systemaattisesti läpi kaikki järjestelmään tallennetut nykyiset testitapaukset ja kehittää niiden rakennetta ja sisältöä.

Lähtötilanteessa testitapaukset olivat hajautuneet useisiin eri kansioihin, mikä vaikeutti niiden hallintaa ja test planin koostamista. Tämä hidasti testauksen suunnittelua ja suorittamista, sillä

tarvittavia testitapauksia oli työlästä etsiä ja koota yhteen. Lisäksi testitapausten sisältö oli osittain puutteellista, eikä vastannut vaadittua laatutasoa tai testauksen kattavuusvaatimuksia.

Työpajassa käytiin läpi testitapausten sisältö, rakenne ja ajantasaisuus. Erityistä huomiota kiinnitettiin siihen, että osa testitapauksista oli päällekkäisiä testiautomaation kanssa, jolloin nämä tapaukset päätettiin poistaa manuaalisesta testauksesta. Tämän lisäksi testitapauksia päivitettiin kattavamiksi ja selkeämmiksi siten, että kuka tahansa tiimin jäsen voisi suorittaa ne ilman lisäohjeistusta. Tavoitteena oli yhdenmukaistaa testitapausten rakenne ja parantaa niiden käytettävyyttä osana testausprosessia.

8.2.4 E2E- ja tiimien välinen testauksen suunnittelu

E2E- ja tiimien välisen testauksen suunnittelua varten järjestettiin kehitystyöpaja, johon osallistui kaksi eri sidosryhmää, joita kvartaalitestauksen kehittäminen koski. Työpajan tavoitteena oli kehittää E2E-testauksen toimintamalleja sekä kartoittaa tarvittavat testitapaukset eri osapuolten näkökulmasta. Lisäksi pyrittiin vahvistamaan sidosryhmien välistä yhteistyötä testausprosessin aikana.

Kehitystyöpajassa käytiin yhdessä läpi, miten E2E-testausta voitaisiin parantaa ja millaisia testitapauksia kyseiseen testaukseen tarvitaan. Käytiin yhdessä läpi kaikki testitapaukset. Molempien sidosryhmien kanssa tarkennettiin heidän tarpeitaan vastaavat testitapaukset, jotka haluttiin sisällyttää E2E-testeihin.

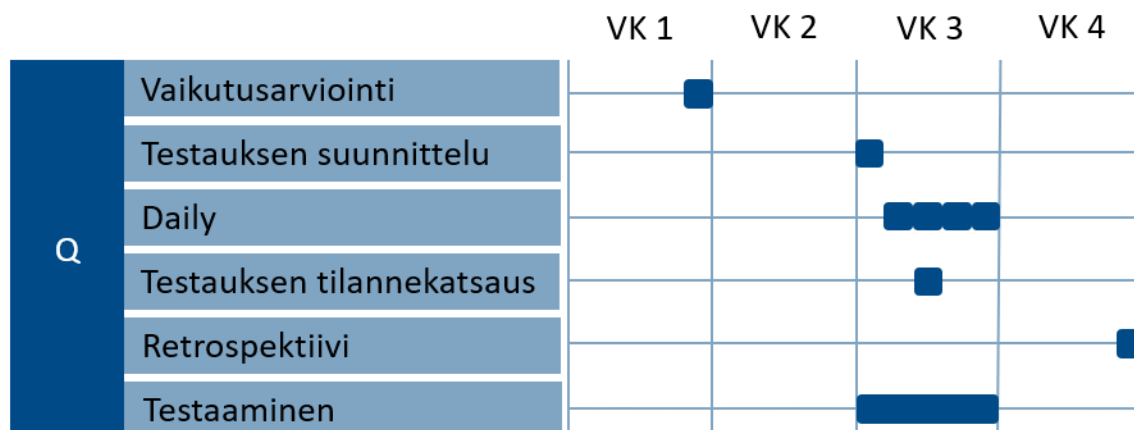
Lisäksi sovittiin käytännön toimintatavoista testauksen toteuttamisessa ja kommunikoinnissa:

- Ensimmäisen sidosryhmän kanssa sovittiin, että heidän tarvitsemansa testitapaukset siirretään Jira Xray -järjestelmässä heidän käyttöönsä.
- Toisen sidosryhmän kanssa sovittiin kommunikoinnista erillisen Teams-ryhmän kautta, sillä heillä ei ole käytössään Jira Xray -järjestelmää.

Näin varmistettiin, että molempien sidosryhmien tarpeet huomioidaan ja testauksen suunniteluun sekä toteutukseen liittyvä viestintä sujuu sujuvasti.

8.2.5 Testausprosessin organisointi ja seuranta

Kvartaalitestauksen toteutus perustui Scrumban-viitekehykseen, joka yhdistää Scrumin palaverikäytännöt ja Kanbanin visuaalisen työhallinnan. Mallin avulla testausprosessi jäsennetään selkeisiin vaiheisiin, mikä tuki suunnittelua ja läpinäkyvyyttä. Kuva 13 esitetään, miten palaverikäytännöt ja testaus ajoittuivat kvartaalisyklin aikana.



Kuva 13. Kvartaalitestauksen palaverikäytännöt ja testauksen jaksotus

Vaikutusarviointi tapahtuu noin kaksi viikkoa ennen testauksen aloitusta, eikä sen aikatauluun voida vaikuttaa, sillä ohjelmistotoimittaja määrittää sen. Testauksen suunnittelu sijoittuu tämän jälkeen, ja sen rinnalla käynnistyvät päivittäiset Scrum-palaverit (Daily), jotka tukevat tiimin sisäistä koordinaointia.

Scrumin palaverien lisäksi malliin sisältyy testauksen tilannekatsaus, joka pidetään sidosryhmien kesken. Tämä ei kuulu Scrumin malliin, mutta se on vakiintunut käytäntö, jota halutaan säilyttää tiedonkulun varmistamiseksi. Retrospektiivi toteutetaan testauksen päätyttyä, ja sen avulla kerätään kehitysehdotuksia seuraavia syklejä varten.

Testausaikaa lyhennettiin alkuperäisestä kahdesta viikosta yhteen viikkoon, jotta toinen viikko jäisi aikaa mahdollisille virheiden korjaustoimenpiteille.

Testausprosessin visualisointia varten Jiraan luotiin erillinen Kanban-taulu, jonka tarkoituksena oli parantaa työnkulun läpinäkyvyyttä ja helpottaa tehtävien hallintaa. Kanban-taulu tarjosi visuaalisen näkymän testauksen etenemisestä, mutta käytännössä sen käyttö jäi vähäiseksi, vaikka sen käyttöön annettiin koulutusta ja sitä esiteltiin päivittäispalavereissa. Yksi keskeinen syy oli se, että tiimille tuli samanaikaisesti omaksuttavaksi uusia palaverikäytäntöjä, mikä lisäsi muutoksen

kokonaiskuormaa. Käyttäjät valitsivat tutun ja koetun toimintatavan eli Jira Xrayn luettelonäkymän, koska se tarjosi riittävän tiedon ilman uuden työkalun opettelua.

8.2.6 Q2-testauksen retrospektiivin tulokset

Retrospektiivi järjestettiin testausviikon lopuksi ja siihen osallistui kahdeksan henkilöä eri sidosryhmistä, joilla kaikilla oli kokemusta HCM Core -testauksesta. Osallistujajoukkoon kuului esihenkilö, tuoteomistaja, testauskoordinaattori, testiautomaation asiantuntija sekä kolme testaajaa. Retrospektiivi toteutettiin Teamsin kautta ja aloitettiin käymällä läpi seuraavat kysymykset:

- Mikä meni hyvin?
- Mikä ei mennyt hyvin?
- Mitä voimme parantaa?

Jokaisella oli mahdollisuus tuoda esiin oma näkemyksensä. Lopuksi oli vapaampaa keskustelua aiheesta. Keskustelu pysyi koko ajan rakentavana ja kunnioittavana, ja sen myötä syntyi hedelmällistä vuoropuhelua.

Retrospektiivin aineisto tallennettiin Teamsin kautta ja litteroitiin tekstimuotoon. Vastaukset käytiin läpi kysymysten pohjalta (*Mikä meni hyvin? Mikä ei mennyt hyvin? Mitä voimme parantaa?*) ja niistä koottiin keskeiset havainnot. Tuotokset ryhmiteltiin sisällöllisesti samankaltaisiin kokonaisuuksiin, joista muodostettiin pääteemat. Prosessi muistutti laadullisen analyysin teemoittelua, mutta oli luonteeltaan kevyempi, koska aineiston määrä oli rajallinen ja sisältö tiivistä. Analyysin tavoitteena oli muodostaa kokonaiskuva retrospektiivissä esiin nousseista näkökulmista ja tunnistaa kehittämistyön kannalta keskeiset havainnot. Teemoittelun avulla pyrittiin selkeyttämään aineistoa ja tukemaan jatkokehittämisen suunnittelua. Näin retrospektiivin tuotoksia voitiin hyödyntää systemaattisesti osana toimintatutkimuksen seuraavia vaiheita.

Testitapausten kehittäminen

Haastateltavat kokivat testitapausten kehittämisen tärkeäksi osa-alueeksi. Erityisesti korostettiin tarvetta lisätä testitapauksia, jotka kattavat myös poikkeustilanteet eli ns. ei-happy-caset. Testitapauksia haluttiin hioa edelleen, jotta ne muodostaisivat kattavan ja laadukkaan kokonaisuuden. Raja-arvotestauksen merkitys nousi myös esiin kehityskohteena.

Positiivinen kokemus ja oppiminen

Useat osallistujat kokivat testausviikon sujuneen hyvin ja ilmaisivat tyytyväisyyttä prosessin etenemiseen. Päivittäiset Dailyt koettiin hyödyllisiksi ja niiden koettiin tukevan tiimin yhteistä työskentelyä. Lisäksi esiin nousi oppimiskokemuksia, kuten uusien testitapausten tekeminen ja järjestelmän osaamisen laajentuminen.

Viestintä ja kokoonpano

Tiimin kokoonpanoa ja osallistujien rooleja palaverissa pohdittiin. Tiedonkulun haasteet nousivat esiin erityisesti tilanteissa, joissa havaintoja ei saatu jaettua riittävän nopeasti. Tämä nähtiin kehityskohteena, joka vaatii jatkossa huomiota.

8.3 Sykli 3 – Prosessin vakiinnuttaminen

Kolmannessa syklissä keskityttiin testausprosessin kehittämiseen ja mallin vakiinnuttamiseen. Toimitus sisälsi kaksi erillistä kvartaalitestauksia (Q3 ja Q4), jotka perustuivat aikaisemmista vaiheista saatujen havaintojen ja kehitettyjen käytäntöjen hyödyntämiseen.

8.3.1 Testausprosessin organisointi ja seuranta

Lopullisessa malli säilyi pääosin samana kuin toisessa syklissä. Palaverikäytännöt ja testauksen aikataulu pysyivät ennallaan, sillä ne koettiin toimiviksi ja tukivat tiimin koordinaatiota. Merkittävin muutos oli Kanban-tilin poistuminen käytöstä. Vaikka taulu oli toteutettu Jiraan ja sen käyttöön annettiin koulutusta, tiimi jatkoi testauksen seurantaan Jira Xrayn luettelonäkymässä, joka koettiin tutuksi ja riittäväksi.

8.3.2 Q3-testauksen retrospektiivin tulokset

Toisen kvartaalitestauksen retrospektiivi järjestettiin testausviikon lopuksi ja siihen osallistui kahdeksan henkilöä eri sidosryhmistä, joilla kaikilla oli kokemusta HCM Core -testauksesta. Osallistujajoukkoon kuului esihenkilö, tuoteomistaja, testauskoordinaattori sekä neljä testaajaa.

Retrospektiivi toteutettiin Teamsin kautta ja aloitettiin käymällä läpi seuraavat kysymykset:

- Mikä meni hyvin?
- Mikä ei mennyt hyvin?
- Mitä voimme parantaa?

Palautteen keräämiseen käytettiin digitaalista valkotalua. Osallistajat kirjasivat näkemyksiään siitä mitä testauksessa sujui hyvin, mikä ei mennyt odotetusti ja mitä voitaisiin parantaa tulevissa testauskierroksissa. Kuvissa Kuva 14, Kuva 15, Kuva 16 on esitetty digitaalisen valkotaulun sisältö retrospektiivistä. Valkotaulun käyttö mahdollisti reaaliaikaisen ja avoimen palautteenannon ja keskustelu oli aktiivista sekä kunnioittavaa. Retrospektiivin rakenne tuki osallistujien reflektiota ja toi esiin konkreettisia kehitysehdotuksia.

Retrospektiivin aineisto tallennettiin Teamsin kautta. Vastaukset käytiin läpi kysymysten pohjalta (*Mikä meni hyvin? Mikä ei mennyt hyvin? Mitä voimme parantaa?*) ja niistä koottiin keskeiset havainnot. Tuotokset ryhmiteltiin sisällöllisesti samankaltaisiin kokonaisuuksiin, joista muodostettiin pääteemat. Prosessi muistutti laadullisen analyysin teemoittelua, mutta oli luonteeltaan kevyempi, koska aineiston määrä oli rajallinen ja sisältö tiivistä.

Palautteen perusteella testaus eteni aikataulussa ja päivittäiset Dailyt koettiin toimiviksi. Testitapaukset oli jaettu tasaisesti tiimin kesken, mikä edisti sujuvaa työskentelyä. Haasteita ilmeni erityisesti testauksen esivalmisteluissa, kuten testitapausten epäselvissä vaiheissa ja QA-ympäristön toimivuudessa. Lisäksi osa testitapauksista ei vastannut täysin todellisia prosesseja mikä aiheutti epävarmuutta testauksen aikana.

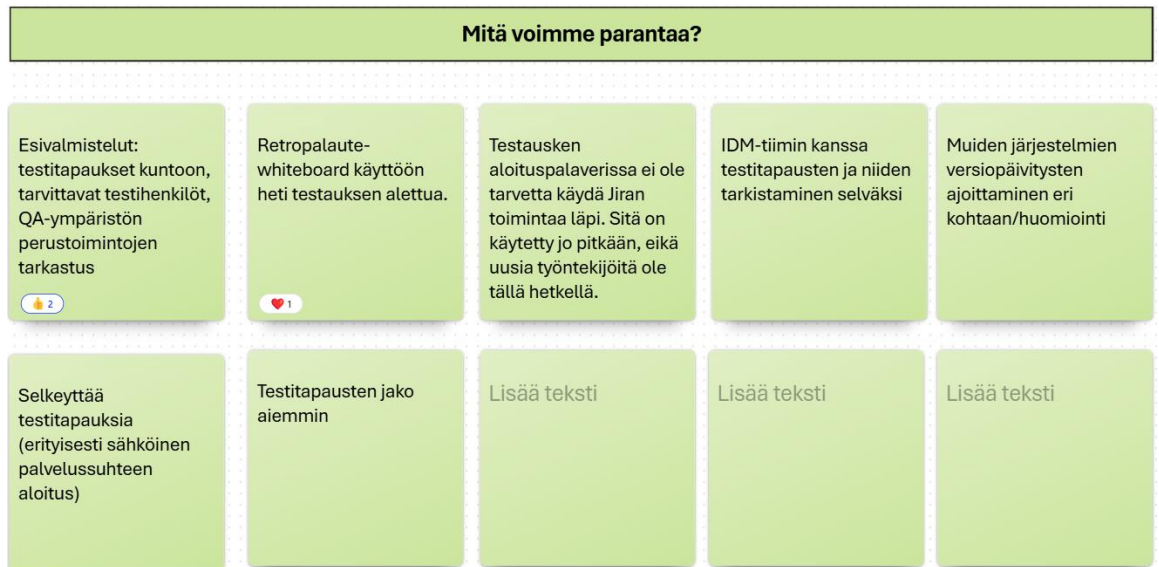
Parannusehdotuksissa korostui tarve selkeyttää testitapauksia, erityisesti sähköisen palvelussuhteen aloituksen osalta. Esivalmisteluihin toivottiin enemmän huomiota: testihenkilöiden varmistaminen, ympäristön tarkistus ja testitapausten läpikäynti ennen testauksen alkua. Retropalautteen keruuta ehdotettiin aloitettavaksi heti testauksen alussa, jotta palautetta saataisiin jatkuvasti. Lisäksi esitettiin, että testauksen aloituspalaverissa ei ole tarpeen käydä Jira Xrayn käyttöä läpi, mikäli mukana ei ole uusia työntekijöitä. Yhteistyötä IDM-tiimin kanssa haluttiin selkeyttää ja muiden järjestelmien versiopäivitysten ajoitus nähtiin tärkeänä huomioitavana tekijänä.

Mikä meni hyvin?				
Pysyttiin hyvin aikataulussa	Dailyt toteutui hyvin.	Testitapaukset oli jaettu sopivasti kaikkien kesken.	Lisää teksti	Lisää teksti
Lisää teksti	Lisää teksti	Lisää teksti	Lisää teksti	Lisää teksti

Kuva 14. Retrospektiivin palaute Q3 Mikä meni hyvin?

Mikä ei mennyt hyvin?				
Testaamisen esivalmisteluissa paranneltavaa	Testitapauksilla steppejä, jotka eivät täysin mätsää	Lisää teksti	Lisää teksti	Lisää teksti
Lisää teksti	Lisää teksti	Lisää teksti	Lisää teksti	Lisää teksti

Kuva 15. Retrospektiivin palaute Q3 Mikä ei mennyt hyvin?



Kuva 16. Retrospektiivin palaute Q3 Mitä voimme parantaa?

8.3.3 Q4-testauksen retrospektiivin tulokset

Kolmannen kvartaalitestauksen retrospektiivi toteutettiin Teamsin kautta ja palautteen keräämiseen hyödynnettiin digitaalista valkotalua. Tällä kertaa retrospektiivissä keskityttiin kahteen kysymykseen:

- Mikä meni hyvin?
- Mitä voimme parantaa?

Aiemmissa kierroksissa käytetty kysymys *mikä ei mennyt hyvin* koettiin osin päällekkäiseksi parannusehdotusten kanssa, minkä vuoksi se jätettiin pois. Kuvassa Kuva 17 on esitetty digitaalisen valkotalun sisältö retrospektiivistä. Muutoksella haluttiin selkeyttää keskustelua.

Retrospektiivin aineisto tallennettiin Teamsin kautta. Vastaukset käytiin läpi kysymysten pohjalta (*Mikä meni hyvin? Mitä voimme parantaa?*) ja niistä koottiin keskeiset havainnot. Tuotokset ryhmiteltiin sisällöllisesti samankaltaisiin kokonaisuuksiin, joista muodostettiin pääteemat. Prosessi muistutti laadullisen analyysin teemoittelua, mutta oli luonteeltaan kevyempi, koska aineiston määrä oli rajallinen ja sisältö tiivistä.

Palautteen perusteella testaus eteni suunnitellusti ja aikataulussa. Kommunikaatio tiimin sisällä koettiin toimivaksi, mikä tuki testauksen sujuvuutta ja yhteistyötä. Hyvin organisoitu viestintä auttoi pitämään kaikki osapuolet ajan tasalla ja mahdollisti nopean reagoinnin havaintoihin.

Kehityskohteina nostettiin esiin erityisesti testitapausten tarkennustarpeet. SPA- ja IDM-alueiden testitapaukset kaipaavat tarkempaa läpikäyntiä ja selkeyttämistä. Ehdotettiin, että IDM-tarkastukseen osallistettaisiin nimetty henkilö palaveriin, jotta vastuut ja sisältö olisivat selkeämmät. Lisäksi Prima-järjestelmän tarkistettavat tiedot toivottiin koottavan Exceliin ja Prima ja Titaniaa koskeva yhteinen palaveri nähtiin tarpeellisena, johon tulisi kutsua oikeat asiantuntijat mukaan. Nämä ehdotukset osoittavat, että testauksen koordinoitua ja valmistelua halutaan kehittää entistä systemaattisemmaksi.



Kuva 17. Retrospektiivin palaute Q4

9 Analyysi ja tulkinta

Tässä luvussa käsitellään opinnäytetyön empiirisen osan keskeiset havainnot ja niiden analyysi suhteessa teoriaan ja tutkimuskysymyksiin. Aineisto koostuu fokusryhmähaastattelusta sekä retrospektiiveistä, jotka toteutettiin kvartaalitestauksen yhteydessä. Näiden menetelmien avulla kerättiin tietoa testausprosessin nykytilasta, kehityskohteista ja tiimin kokemuksista. Luvun alussa esitetään fokusryhmähaastattelun tulokset ja niistä tehdyt johtopäätökset, minkä jälkeen tarkastellaan retrospektiivien havaintoja yhtenä kokonaisuutena. Analyysi nostaa esiin keskeiset teemat, niiden merkityksen ja kehittämisehdotukset ketterien menetelmien näkökulmasta. Tutkimuskysymykset on määritelty Johdanto-osiossa.

9.1 Fokusryhmähaastattelu

Fokusryhmähaastattelun tavoitteena oli kartoittaa tiimin näkemyksiä kvartaalitestauksen nykytilasta, tunnistaa keskeiset haasteet ja arvioida ketterien menetelmien soveltuvuutta prosessin kehittämässä. Haastattelun tulokset ja analyysi jaettiin kolmeen pääteemaan: palaverikäytännöt ja tiedonkulku, testausprosessin tehokkuus ja aikataulu sekä testauksen työkalut ja dokumentaatio.

Haastattelun perusteella ohjelmistotoimittajan palaveri koetaan päällekkäiseksi, sillä sama tieto on jo saatavilla Jira-tiketeillä. Lisäksi tiedonkulku tiimin sisällä ja sidosryhmille on puutteellista, eikä loppukäyttäjille välity tieto tulevista muutoksista. Tämä on ristiriidassa ketterien menetelmien periaatteiden kanssa, joissa korostetaan läpinäkyvyyttä ja jatkuvaa kommunikointia (Beck ym., 2001). Laadunvarmistusta tukee avoin tiedonkulku ja aktiivinen vuorovaikutus eri sidosryhmien välillä, sillä ne luovat edellytykset nopeille korjaustoimille. (Heuer, 2014, 14; ISTQB, 2014, 32). Organisaation tulisi harkita ohjelmistotoimittajan palaverin korvaamista tai tiivistämistä ja palauttaa testauksen suunnittelupalaveri, joka tukee valmistautumista ja arvontuottavuutta. Tiedonkulkua voidaan parantaa myös Scrumin mukaisilla Dailyilla. Lisäksi tiedonkulkua voidaan parantaa käytännöillä, kuten Jira-dashboardeilla, jotta tieto tavoittaa myös loppukäyttäjät.

Nykyinen kahden viikon testausaika koetaan liian pitkäksi, ja ehdotetaan lyhentämistä viikkoon, jotta virheiden korjaukselle jää aikaa. Lisäksi testitapaukset eivät jakaudu tasaisesti, ja osa niistä on epäselviä. Pitkä testausaika ei tue ketteryyden periaatetta nopeasta palautteesta. Kanbanin ja

Scrumbanin mukaan lyhyemmät syklit parantavat virtausnopeutta ja läpimenoaika (Niemi & Hietaniemi, 2020, 18). Epätasainen testitapausten jakautuminen ja epäselvyys lisäävät hukkaa, mikä on vastoin Lean-ajattelua. Testausaikataulua tulisi lyhentää ja varata erillinen korjausviikko. Lisäksi testitapausten selkeyttä ja tasapainoista jakautumista voidaan parantaa standardoimalla ohjeet ja roolit, mikä tukee tehokkuutta ja vähentää virheitä.

Jira Xrayssa testitapaukset ovat hajallaan, mikä vaikeuttaa testplanin koostamista. Lisäksi päällekkäisyys manuaali- ja automaatiotesteissä vie turhaan aikaa. Hajanaisuus ja päällekkäisyys heikentävät prosessin läpinäkyvyyttä ja lisäävät kognitiivista kuormaa. Teorian mukaan työkalujen selkeys ja visuaalisuus tukevat virtausta ja vähentävät hukkaa (Juvonen, 2018, 23–24). Testitapaukset tulisi järjestää Xrayssa selkeämmin ja poistaa päällekkäisyydet. Tämä parantaa suunnittelua ja vähentää turhaa työtä, mikä tukee ketterän kehittämisen periaatteita.

Haastattelun perusteella keskeiset kehityskohteet liittyvät tiedonkulkuun, palaverikäytäntöihin, testausaikatauluun ja dokumentaation selkeyteen. Havainnot osoittavat, että Scrumban-menetelmän käyttö vaikuttaa tiimityön tehokkuuteen ja yhteistyöhön erityisesti parantuneen kommunikoinnin ja selkeämpien palaverikäytäntöjen kautta. Lisäksi Scrumbanin visuaaliset työkalut ja työn rajoittaminen tukevat kvartaalitestauksen prosessin tehokkuutta ja hallittavuutta, kun testausaikatauluja voidaan lyhentää ja dokumentaatiota selkeyttää.

9.2 Retrospektiivit Q2–Q4

Retrospektiivien (Q2–Q4) analyysi täydentää fokusryhmähaastattelun havaintoja ja tuo esiin tiimin kokemukset prosessin kehittämisestä ketterien menetelmien näkökulmasta. Tarkastelu keskittyy neljään pääteemaan: testitapausten kehittäminen ja selkeys, viestintä ja roolit, yhteistyö ja skaalautuva ketteryys sekä positiiviset käytännöt.

Testitapausten kattavuudessa ja selkeydessä havaittiin merkittäviä puutteita. Q2:ssa korostui tarve laajentaa testitapauksia poikkeustilanteisiin ja raja-arvoihin, mikä tukee periaatteita kattavasta ja riskiperusteisesta testauksesta (Kasurinen, 2013). Q3:ssa nousi esiin epäselviä vaiheita ja QA-ympäristön ongelmia, jotka lisäävät virheiden riskiä ja viivästyksiä. Q4:ssa havaittiin SPA- ja IDM-alueiden testitapausten tarkennustarpeet, mikä hidastaa testausta ja heikentää laadunvarmistusta. Organisaation tulisi standardoida testitapausten ohjeistus, varmistaa QA-ympäristön toimivuus ennen testauksen alkua ja sisällyttää myös ei-happy-caset ja raja-arvot testisuunnitelmaan. Tämä tukee virtausnopeutta ja vähentää hukkaa Lean-ajattelun mukaisesti.

Viestinnän ja roolien osalta Q2:ssa havaittiin tiedonkulun haasteita ja roolien epäselvyyttä, mikä on ristiriidassa ketterän tiimin itseohjautuvuuden ja läpinäkyvyyden kanssa (Beck ym., 2001). Q4:ssa nousi esiin tarve roolien selkeyttämiseksi eri järjestelmäalueiden yhteistyössä. Selkeät roolit ja tehokas kommunikaatio ovat edellytys virtausnopeuden ja laadun parantamiselle (Niemi & Hietaniemi, 2020, 18). Organisaation tulisi tarkentaa roolit palaverissa ja luoda käytäntöjä havaintojen nopeaan jakamiseen, esimerkiksi Jira-komenttien avulla.

Yhteistyön ja skaalautuvan ketteryyden osalta Q3 ja Q4 osoittivat, että riippuvuuksien hallinta ei ole täysin toteutunut. Yhteistyön epäselvyydet IDM-tiimin kanssa ja muiden järjestelmien päivitysten ajoitus viittaavat siihen, että monitiimiympäristössä tarvitaan parempia käytäntöjä. Riippuvuuksien hallinta on kriittistä skaalautuvan ketteryyden onnistumiselle. (Figalist ym., 2019). Yhteistyön parantamiseksi voidaan ottaa käyttöön yhteiset backlogit ja palaverit, jotka tukevat läpinäkyvyyttä ja vähentävät viivästyksiä.

Positiivisina havaintoina kaikissa retrospektiiveissä korostui päivittäisten Daily-palaverien hyödyllisyys, mikä tukee Scrumin periaatteita läpinäkyvyydestä ja nopeasta reagoinnista. Lisäksi Q3:ssa havaittiin tasainen testitapausten jakautuminen, mikä parantaa tiimin tehokkuutta. Nämä käytännöt kannattaa säilyttää ja vahvistaa, sillä ne tukevat tiimin itseohjautuvuutta ja jatkuvaa parantamista.

Retrospektiivien perusteella keskeiset kehityskohteet liittyvät testitapausten kattavuuteen ja selkeyteen, viestinnän ja roolien tarkentamiseen sekä yhteistyön vahvistamiseen monitiimiympäristössä. Havainnot viittaavat siihen, että Scrumban-menetelmän käyttö tukee tiimityön tehokkuutta ja yhteistyötä selkeyttämällä vastuita ja vahvistamalla säännöllistä vuorovaikutusta, kuten päivittäisiä Daily-palavereita. Samalla Scrumban edistää kvartaalitestauksen prosessin tehokkuutta ja hallittavuutta parantamalla testitapausten hallintaa ja prosessien läpinäkyvyyttä. Positiiviset käytännöt, kuten Dailyt, tukevat ketterän toimintamallin periaatteita ja niitä kannattaa säilyttää.

10 Pohdinta

Tämän opinnäytetyön tarkoituksena oli kehittää HUSin HCM Core -järjestelmän kvartaalitestauksen prosessia ja arvioida Scrumban-menetelmän vaikutuksia tiimityöhön ja prosessin hallittavuuteen. Työ toteutettiin toimintatutkimuksena kolmessa syklissä, mikä mahdollisti iteratiivisen kehittämisen ja jatkuvan oppimisen. Aineistonkeruussa hyödynnettiin fokusryhmähaastattelua ja havainnointia Scrum-viitekehyksen tapahtumissa. Tavoitteena oli luoda selkeä ja toimiva malli, joka tukee testauksen suunnittelua, toteutusta ja jatkuvaa parantamista. Tutkimuskysymyksiksi asetettiin seuraavat: Miten Scrumban-menetelmän käyttö vaikuttaa tiimityön tehokkuuteen ja yhteistyöhön kvartaalitestauksessa? Miten Scrumban-menetelmän käyttö vaikuttaa kvartaalitestauksen prosessin tehokkuuteen ja hallittavuuteen?

Tulokset osoittivat, että Scrumban-menetelmä paransi tiimien välistä viestintää ja toi selkeyttä yhteistyöhön. Scrumin palaverikäytännöt, kuten suunnittelupalaverit, Dailyt ja retrospektiivit, loivat rakenteen, jossa tieto kulki säännöllisesti ja avoimesti. Tämä vähensi epäselvyyksiä ja paransi tiimien välistä kommunikaatiota, mikä vastaa ensimmäiseen tutkimuskysymykseen tiimityön tehokkuudesta. Dailyt auttoivat tiimiä reagoimaan nopeasti muutoksiin ja jakamaan esteitä, mikä lisäsi tehokkuutta ja loi yhteisen tilannekuvan. Retrospektiivit tukivat jatkuvaa parantamista ja tiimin itseohjautuvuutta, mikä on keskeinen osa ketterää toimintaa (Schwaber & Sutherland, 2020). Suunnittelupalaverit puolestaan toivat ennakoitavuutta ja selkeyttä testauksen aikatauluihin, mikä paransi prosessin hallittavuutta, mikä vastaa toiseen tutkimuskysymykseen. Kanbantaulu yhdessä palaverien kanssa teki työn etenemisestä näkyvää, mikä auttoi priorisoimaan kriittisiä tehtäviä ja tukee Lean-periaatetta läpinäkyvyydestä ja virtaustehokkuudesta (Bhavsar ym., 2020).

Ensimmäinen sykli keskittyi HCM Core -järjestelmän kvartaalitestauksen lähtötilanteen kartoittamiseen ja kehitystarpeiden tunnistamiseen. Fokusryhmähaastattelun avulla saatiin kattava kuva nykytilasta, jossa tiedonkulku osoittautui merkittäväksi haasteeksi tiimin sisällä ja sidosryhmien välillä. Testaamisesta ei ollut riittävästi tiedonvaihtoa ja loppukäyttäjille ei välittynyt ajantasainen tieto tulevista muutoksista. Toimittajan palaverin sisältö koettiin osin päällekkäiseksi Jira-tikettien tietosisällön kanssa ja testauksen suunnittelupalaveria toivottiin takaisin. Testauksen aikataulua edotettiin lyhennettäväksi kahdesta viikosta yhteen, jotta virheiden korjaukselle jäisi enemmän aikaa. Testitapaukset olivat epätasaisesti jakautuneita ja osin epäselviä sekä hallinta Jira Xrayssa koettiin hajanaiseksi. Havainnot ohjasivat ratkaisuehdotuksia: päätettiin kokeilla Scrumin palaverikäytäntöjä, visualisoida testauksen kulku Kanban-taululla, laatia tarkempi testaussuunnitelma

ja selkeyttää testitapaukset. Lisäksi tunnistettiin tarve Scrumban-koulutukselle, jotta tiimi saisi perusteellisen tietopohjan ketterille menetelmille.

Toisessa syklissä kvartaalitestausprosessi vietiin ensimmäistä kertaa käytäntöön hyödyntäen edellisen syklin havaintoja ja teoriaa. Suunnitteluvaiheessa luotiin testaussuunnitelma ja tarkennettiin testitapauksia. Toimintavaiheessa otettiin käyttöön Scrumin palaverikäytännöt ja Kanban-taulu, mikä toi selkeyttä viestintään ja testauksen seurantaan. Lisäksi tiimille järjestettiin Scrumban-koulutus ketterän testauksen osaamisen vahvistamiseksi. Havainnoissa korostui tarve tarkentaa testitapausten sisältöä ja tasapainottaa testitapausten jakautumista ja testauksen koordinaointi nähtiin kehityskohteenä. Reflektiossa todettiin, että tiimien välinen yhteistyö oli parantunut merkittävästi ja ketterät käytännöt tukivat prosessin sujuvuutta.

Kolmannessa syklissä keskityttiin prosessin toimivuuden parantamiseen ja mallin vakiinnuttamiseen. Retrospektiivin kysymyksiä muokattiin ratkaisukeskeisemmiksi, mikä ohjasi keskustelua parannuksiin. Testitapaukset tarkennettiin vielä paremmiksi ja palaverikäytännöt sujuvoituivat entisestään, vahvistaen tiimien välistä yhteistyötä. Havainnoissa nousi esiin tarve selkeyttää testitapauksia tietyillä osa-alueilla sekä parantaa dokumentaatiota. Reflektiossa todettiin, että työpajojen avulla ratkaistiin keskeisiä rajapintaongelmia, mikä paransi E2E-testauksen laatua ja tehokkuutta. Syklin tulokset loivat vahvan perustan prosessin vakiinnuttamiselle ja skaalautumiselle myös muihin järjestelmiin.

Sidosryhmien keskinäisen yhteistyön paraneminen vaikutti merkittävästi tiedon jakautumiseen ja läpinäkyvyyteen koko testausprosessissa. Yhteistyön tiivistyminen toi esiin monia kriittisiä asioita, jotka liittyvät testauksen onnistumiseen, kuten mitä elementtejä hyvä testitapaus sisältää ja millaisia ominaisuuksia testihenkilöltä vaaditaan. Keskusteluissa korostui, että selkeästi määritellyt testitapaukset, realistiset skenaariot ja käytännön vaatimusten huomioiminen ovat keskeisiä laadukkaan testauksen kannalta. Tiedonvaihto ei rajoittunut vain sidosryhmien välille, vaan myös tiimin sisällä tapahtui merkittävää osaamisen jakamista, mikä vahvisti yhteistä ymmärrystä ja paransi testauksen laatua. Kun testitapaukset jaettiin tasaisemmin ja niitä vaihdeltiin tiimin jäsenten kesken, syntyi uusia näkökulmia ja tiimi oppi uutta sekä järjestelmästä että testausmenetelmistä.

Onnistumisia olivat tiimien välisen yhteistyön paraneminen, palaverikäytäntöjen selkeytyminen ja testausdokumentaation tarkentuminen. Tulokset tukivat ketterien menetelmien ja Lean-ajattelun periaatteita: läpinäkyvyys, jatkuva parantaminen ja tiimien itseohjautuvuus vahvistuivat. Toimintatutkimuksen syklinen luonne osoittautui tehokkaaksi kehittämisen välineeksi. Haasteet

menetelmien juurtumisessa vahvistavat kirjallisuudessa esitetyn näkemyksen siitä, että kulttuurimuutos ja koulutus ovat keskeisiä onnistuneessa ketteryden käyttöönotossa.

10.1 Luotettavuuden ja eettisyyden arviointi

Työn luotettavuutta ja eettisyyttä on tarkasteltu koko tutkimusprosessin ajan. Tutkimusetiikka tarkoittaa periaatteita, joita noudatetaan suhteessa tutkittaviin, kollegoihin ja suureen yleisöön (Vilkkä, 2021). Opinnäytetyössä on noudatettu hyvää tieteellistä käytäntöä ja eettisiä menetelmiä, mikä takaa tutkimuksen hyväksyttävyyden ja luotettavuuden. Ennen tutkimuksen aloittamista varmistettiin tutkimuslupa toimeksiantajan kanssa ja huolehdittiin salassapidosta. Aineiston keruu, käsittely, säilytys ja hävittäminen toteutettiin eettisesti ja tulokset esitettiin todenmukaisesti. Aineiston hallintaa koskevat käytännöt on kuvattu aineistonhallintasuunnitelmassa (Liite 1). Luotettavuutta vahvistettiin kriittisellä ja arvioivalla otteella, mikä on keskeistä laadullisessa tutkimuksessa (Saaranen-Kauppinen & Puusniekka, 2006). Menetelmänä käytettiin toimintatutkimusta, jossa reflektointi jokaisessa syklissä tuki kehystoimien arviointia. Luotettavuuden arvioinnissa hyödynnettiin validiteettia ja reliabiliteettia (Kananen, 2019). Lisäksi käytettiin laadullisen tutkimuksen menetelmiä, kuten triangulaatiota, informanttien vahvistusta, member checkingiä ja sisäistä loogisuutta (Hirsjärvi & Hurme, 2022). Tutkijan rooli kohteen jäsenenä huomioitiin, sillä läsnäolo voi vaikuttaa osallistujien toimintaan ja vuorovaikutukseen (Sinivuo ym., 2012, 6). Tutkimukseen osallistuneille lähetettiin HUSin tutkimuslupaun sisältyvä tiedote (Liite 2), jossa kerrottiin tutkimuksen tarkoitus, aineiston käsittely ja vapaaehtoisuus. Aineisto tallennettiin tutkijan työkoneelle ja poistettiin opinnäytetyön valmistuttua. Pysyväksi tallenteeksi jää Theseukseen tallennettava opinnäytetyö. Tutkimuslupa haettiin HUSin ohjeiden mukaisesti ja myönnettiin 3.4.2024. Teoriaviitekehys perustuu monipuolisiin ja ajantasaisiin lähteisiin ja työ tarkistettiin plagiointiohjelmalla. Aineisto kerättiin triangulaatiota hyödyntäen fokusryhmähaastattelun ja havainnoinnin avulla Scrum-viitekehysten tapahtumissa. Saturatio katsottiin saavutetuksi, informanttien vahvistus varmistettiin ja tulokset vertailtiin aikaisempiin tutkimuksiin.

10.2 Oman osaamisen kehittyminen suhteessa Arenen kompetensseihin

Suomen ammattikorkeakouluissa tutkintojen osaaminen on määritelty koulutuskohtaisina ja yhteisinä kompetensseina, jotka muodostavat perustan työelämässä toimimiselle ja asiantuntijuuden kehittymiselle. Arenen kompetenssit ovat: oppimaan oppiminen, työelämässä toimiminen,

eettisyys, kestävä kehitys, kansainvälisyys ja monikulttuurisuus sekä ennakoiva kehittäminen. (Arene ry, 2022).

Oppimaan oppimisen kompetenssi korostaa valmiutta kehittää omaa ja yhteisön osaamista jatkuvasti, hyödyntää eri alojen tietoa sekä digitalisaation tarjoamia mahdollisuuksia. Valmistuva opiskelija osaa arvioida ja kehittää asiantuntijuuttaan eri elämänvaiheissa, hankkia ja kriittisesti arvioida tietoa eri näkökulmat huomioiden sekä edistää yhteisön tavoitteellista oppimista. (Arene ry, 2022.)

Opinnäytetyön tekeminen vaatii jatkuvaa tiedonhakua ja kriittistä arviointia, sekä kykyä soveltaa teoriaa käytännön tilanteisiin. Ketterien menetelmien ja ohjelmistotestauksen teoriaan perehtyminen muodostui laajaksi kokonaisuudeksi, jonka soveltaminen työssä edellytti tiedon rakentamista vaiheittain. Toimintatutkimuksen syklinen hyödyntäminen auttoi hahmottamaan, miten teoretieto ja käytännön työ voidaan yhdistää. Prosessi vahvisti taitoa tehdä perusteltuja valintoja ja löytää tasapaino teorian ja käytännön välillä.

Työelämässä toimiminen puolestaan tarkoittaa kykyä kehittää ja johtaa työyhteisöjä sekä uudistaa työelämän käytäntöjä. Tämä sisältää monialaisten tiimien ja työyhteisöjen johtamisen, viestintä- ja vuorovaikutuskulttuurin kehittämisen sekä työhyvinvoinnin tukemisen. Kompetenssi kattaa myös teknologian ja digitalisaation hyödyntämisen sekä työyhteisön resilienssin vahvistamisen. (Arene ry, 2022.)

Kolmen syklin toteuttaminen haastoi projektinhallinnan taitoja ja kykyä ohjata prosessia käytännön realiteeteista käsin. Työpajojen ja ryhmäkeskustelujen fasilitointi kehitti vuorovaikutustaitoja ja kykyä tukea yhteistä ajattelua. Opin sovittamaan yhteen eri näkökulmia ja ohjaamaan kehittämistä siten, että kokonaisuus etenee tavoitteiden mukaisesti eikä jää pelkästään oman työn suorittamisen tasolle.

Eettisyyden kompetenssi painottaa ammattialan arvojen ja eettisten periaatteiden huomioimista sekä tasa-arvon ja yhdenvertaisuuden edistämistä. Valmistuva opiskelija kykenee edistämään eettisesti kestävästä toiminnasta, huomioimaan yksilölliset, yhteisölliset ja yhteiskunnalliset näkökulmat päätöksenteossa, tukemaan moninaisuutta ja saavutettavuutta sekä soveltamaan hyvän tieteellisen käytännön periaatteita työyhteisössään. (Arene ry, 2022.)

Aineistonkeruun ja analyysin aikana jouduin tekemään käytännön eettisiä valintoja, kuten varmistamaan osallistujien anonymiteetin ja turvallisen ilmapiirin. Fokusryhmähaastattelun toteuttaminen vahvisti ymmärrystäni siitä, että eettisyys on jatkuva osa työskentelyä ja tulosten raportointia, ei erillinen vaihe.

Kestävä kehitys tarkoittaa vastuullisten toimintatapojen kehittämistä ja johtamista työyhteisössä sekä yhteiskunnassa laajemmin. Opiskelija osaa soveltaa kestävä kehityksen tietoa kokonaisvaltaisesti, analysoida monialaisten ongelmien systeemisiä riippuvuuksia ja johtaa kestävien ratkaisujen käyttöönottoa ja juurruttaa työyhteisöön. (Arene ry, 2022.)

Työskentely prosessien parissa konkretisoi, että kestävyteen liittyy myös pitkäaikaisen toimintakulttuurin huomioiminen. Vaikka opinnäytetyön tutkimus ei suoraan keskittynyt kestävyteen, opin tunnistamaan, miten pienetkin muutokset voivat vaikuttaa organisaation toimintatapojen pitkäjänteiseen kehitykseen.

Kansainvälisyys ja monikulttuurisuus korostavat kykyä toimia ja johtaa monikulttuurisissa ja kansainvälisissä toimintaympäristöissä. Valmistuva opiskelija osaa kehittää monikulttuurisuutta tukevia toimintatapoja, arvioida globaalien ilmiöiden vaikutuksia ja osallistua kansainväliseen vuorovaikutukseen ja toiminnan kehittämiseen omalla alallaan. (Arene ry, 2022.)

Toimintaympäristöni oli kansainvälinen, sillä ohjelmistotoimittaja toimii globaalisti. Työn aikana konkretisoitui, että tiimin on pystyttävä kommunikoimaan englanniksi ja toimimaan monikulttuurisessa ympäristössä. Tämä vahvisti ymmärrystäni kansainvälisen ja monikulttuurisen osaamisen merkityksestä arjen työssä.

Ennakoiva kehittäminen liittyy tulevaisuuteen suuntautuvien ratkaisujen kehittämiseen ja uuden tiedon tuottamiseen tutkimus- ja kehittämismenetelmiä hyödyntäen. Kompetenssi sisältää eri alojen osaamisen yhdistämisen, tutkimus- ja kehityshankkeiden toteuttamisen sekä asiakaslähtöisten, kestävien ja taloudellisesti kannattavien ratkaisujen suunnittelun ja käyttöönoton. Valmistuva opiskelija osaa analysoida nykytilannetta ja ennakoida alan ja toimintaympäristön muutoksia, mikä tukee organisaation strategista kehittymistä. (Arene ry, 2022.)

Työn edetessä opin tarkastelemaan ilmiötä laajemmasta näkökulmasta ja pohtimaan, mitä muutokset merkitsevät pidemmällä aikavälillä. Toimintatutkimuksen sykliisyys auttoi tunnistamaan kehittämiskohteita ja arvioimaan, millaiset toimet tukevat muutosten pysyvyyttä. Samalla vahvistui valmius arvioida toimintaympäristöä kokonaisvaltaisesti ja suunnitella kehittämistä strategisesti.

Prosessi vahvisti kokonaisvaltaista ymmärrystä siitä, millaista osaamista ylempi AMK-tason asiantuntijalta vaaditaan ja miten teoreettinen tieto voidaan kytkeä käytännön kehittämiseen. Työ lisäsi varmuutta ja valmiuksia toimia kehittäjänä, jonka päätökset pohjautuvat sekä tutkittuun tietoon että organisaation tarpeisiin.

10.3 Jatkokehitysmahdollisuudet

Työn hyödyntämismahdollisuudet ovat laajat. Kehitetty malli voidaan soveltaa HCM Core -järjestelmän lisäksi myös muiden järjestelmien testausprosesseihin. Jatkotutkimusaiheita voisivat olla testiautomaation ja manuaalitestauksen yhteensovittaminen sekä menetelmän skaalautuvuus laajempiin kehityskokonaisuuksiin. Lisäksi olisi hyödyllistä tutkia, miten ketterien menetelmien käyttöönottoa voidaan tukea organisaatiotasolla kulttuurimuutoksen ja koulutuksen avulla. Jos jatkaisin kvartaalitestauksen kehittämistä, tekisin sen retrospektiivien pohjalta kerättyjen kehitysideoiden avulla, sillä ne osoittautuivat tehokkaaksi keinoksi jatkuvaan parantamiseen.

Lähteet

Ahmad, M. (2016). Exploring Kanban in software engineering. Väitöskirja. University of Oulu <http://urn.fi/URN:ISBN:978-951-39-4708-8>

Aira, A. (2012). Toimiva yhteistyö: työelämän vuorovaikutussuhteet, tiimit ja verkostot. Väitöskirja. Jyväskylän yliopisto <http://urn.fi/URN:ISBN:978-951-39-4708-8>

Alqudah, M., & Razali, R. (2017). A comparison of scrum and Kanban for identifying their selection factors. 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI), 1–6. <https://doi.org/10.1109/ICEEI.2017.8312434>

Alqudah, M., & Razali, R. (2018). An Empirical Study of Scrumban Formation based on the Selection of Scrum and Kanban Practices. International Journal on Advanced Science, Engineering and Information Technology, 8(6), 2315–2322. <https://doi.org/10.18517/ijaseit.8.6.6566>

Almadhoun, W., & Hamdan, M. (2018). Optimizing the self-organizing team size using a genetic algorithm in agile practices. Journal of Intelligent Systems, 27(3), 469–482. <https://doi.org/10.1515/jisys-2018-0085>

Arene ry. (2022). Suositus ammattikorkeakoulujen yhteisiksi kompetensseiksi. Arene ry. Saatavilla 27.11.2025 https://www.arene.fi/wp-content/uploads/Raportit/2022/Kompetenssit/Suositus%20ammattikorkeakoulujen%20yhteisiksi%20kompetensseiksi.pdf?_t=1642539572

Atlassian. (N.d.a.). Scrumban: Mastering two Agile methodologies. Saatavilla 29.12.2024 <https://www.atlassian.com/agile/project-management/scrumban/>

Atlassian. (N.d.b.). 4 Kanban metrics you should be using right now. Saatavilla 26.10.2025 https://www.atlassian.com/agile/project-management/kanban-metrics?utm_source=chat-gpt.com

Auranne, T. (2024). Kehittämispäällikkö. Sähköposti 25.9.2024.

Baumgartner, M., Klonek, M., Mastnak, C., Pichler, H., Seidl, R., & Tanczos, S. (2021). *Agile Testing: The Agile Way to Quality*. Springer.

Beck, K., Beedle, M., Bennekum, A., Cunningham, W., Fowler, M., Martin, R., Mellor, S., Thomas, D., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Schwaber, K. & Sutherland, J. (2001). *Agile Manifesto*. Agile Alliance. Saatavilla 26.10.2024 <https://agilealliance.org/agile101/the-agile-manifesto/>

Berlowski, J., Chrusciel, P., Kasprzyk, M., Konaniec, I., & Jureczko, M. (2016). Highly automated agile testing process: An industrial case study. 69–87. <https://doi.org/DOI10.5277/e-Inf160104>

Bhavsar, K., Shah, Dr. V., & Gopalan, Dr. S. (2020). Scrumban: An Agile Integration of Scrum and Kanban in Software Engineering. *International Journal of Innovative Technology and Exploring Engineering*, 9(4), 1626–1634. <https://doi.org/10.35940/ijitee.D1566.029420>

Butt, S., Khan, S. U. R., Hussain, S., & Wang, W.-L. (2023). A conceptual model supporting decision-making for test automation in Agile-based Software Development. *Journal of Systems and Software*, 190, 110210. <https://doi.org/10.1016/j.datak.2022.102111>

Butt, S. A., Khalid, A., Ercan, T., Ariza-Colpas, P. P., Melisa, A.-C., Piñeres-Espitia, G., De-La-Hoz-Franco, E., Melo, M. A. P., & Ortega, R. M. (2022). A software-based cost estimation technique in scrum using a developer's expertise. *Advances in Engineering Software*, 171, 103159. <https://doi.org/10.1016/j.advengsoft.2022.103159>

Collins, E. F., & de Lucena, V. F. (2012). Software Test Automation practices in agile development environment: An industry experience report. 2012 7th International Workshop on Automation of Software Test (AST), 57–63. <https://doi.org/10.1109/IWAST.2012.6228991>

Dong, H., Dacre, N., Baxter, D., & Ceylan, S. (2024). What is Agile Project Management? Developing a New Definition Following a Systematic Literature Review. *Project Management Journal*, 55(6), 668–688. <https://doi.org/10.1177/87569728241254095>

Dos Santos, P. S. M., Beltrão, A. C., de Souza, B. P., & Travassos, G. H. (2018). On the benefits and challenges of using kanban in software engineering: a structured synthesis study. *Journal of Software Engineering Research and Development*, 6(1), 13. <https://doi.org/10.1186/s40411-018-0057-1>

Dyba, T., & Dingsoyr, T. (2015). Agile Project Management: From Self-Managing Teams to Large-Scale Development. 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, 945–946. <https://doi.org/10.1109/ICSE.2015.299>

Figalist, I., Elsner, C., Bosch, J., & Holmström Olsson, H. (2019). Scaling Agile Beyond Organizational Boundaries: Coordination Challenges in Software Ecosystems. In *Agile Processes in Software Engineering and Extreme Programming* (pp. 177–192). Springer. https://doi.org/10.1007/978-3-030-19034-7_12

Forsström, M. (11.2.2021). Fasilitointi tuo struktuuria ja tukee tavoitteellisuutta työpajassa. [blogikirjoitus]. Innokylä. Saatavilla 14.2.2025 <https://innokyla.fi/fi/ajankohtaista/fasilitointi-luostрукtuuria-ja-tukee-tavoitteellisuutta-tyopajassa>

Fortunato, C. A., Furtado, F., Selleri, F., de Farias Junior, I., & Leitão Júnior, N. (2017). Quality Assurance in Agile Software Development: A Systematic Review (pp. 142–148). https://doi.org/10.1007/978-3-319-55907-0_14

Gebeyehu, S. G., Abebe, M., & Gochel, A. (2022). Production lead time improvement through lean manufacturing. *Cogent Engineering*, 9(1), 2034255. <https://doi.org/10.1080/23311916.2022.2034255>

Gregorio, D. D. (2012). How the Business Analyst supports and encourages collaboration on agile projects. *SysCon 2012 - 2012 IEEE International Systems Conference, Proceedings*, 24–27. <https://doi.org/10.1109/SysCon.2012.6189437>

Gundumogula, M. (2020). Importance of Focus Groups in Qualitative Research. *The International Journal of Humanities & Social Studies*, 8 (11). <https://doi.org/10.24940/theijhss/2020/v8/i11/HS2011-082>

Haikala, I. & Mikkonen, T. (2011). Ohjelmistotuotannon käytännöt. Talentum.

Halonen, M. & Käpykangas, S. (2023). Menetelmä organisaatioiden välisen yhteistyön kehittämiseen. Saatavilla 2.2.2025 https://www.ttl.fi/ajankohtaista/uutinen/menetelma-organisaatioiden-valisen-yhteistyon-kehittamiseen?utm_source=chatgpt.com

Hammarberg, M. & Sundén, J. (2014). Kanban in Action. New York: Manning Publications Co.

Heikkinen, H. & Kaukko, M. (2023). Toimintatutkimus. Käytännön opas. Vastapaino.

Herranen, K. (2020). Ketterä kasvu. Alma Talent.

Heuer, F. (2014). Agile Testing: An Overview. CreateSpace Independent Publishing Platform

Hietaniemi, J. (18.3.2020) Mikä on Kanban? [blogikirjoitus]. Gofore. Saatavilla 14.2.2025 <https://gofore.com/mika-on-kanban/>

Hirsjärvi, S. & Hurme, H. (2022). Tutkimushaastattelu. Teemahaastattelun teoria ja käytäntö. Gaudemus Oy.

HUS. (2025.a.). Tietoa meistä. Saatavilla 19.4.2025 <https://www.hus.fi/tietoa-meista>

HUS. (2025.b.). Arvot ja strategia. Saatavilla 19.4.2025 <https://www.hus.fi/tietoa-meista/strategia-ja-vastuullisuus/arvot-ja-strategia>

ISTQB. (2018). Sertifioitu Testaaja. Perustason sertifikaattisisältö. Saatavilla 26.4.2025 <https://tivia-jasenyhdistykset.fi/fistb-testi/wp-content/uploads/sites/30/2020/12/CTFL-2018-Sertifikaattisisalto-20181010-1-Valmis.pdf>

ISTQB. (2014). Sertifioitu Testaaja. Perustason sertifikaattisisältö Ketterä testaaja. Saatavilla 26.4.2025 https://fistb.fi/wp-content/uploads/sites/30/2021/05/FND-Agile-Syllabus_FI.pdf

Juvonen, R. (2018). Ohjelmistoprojektin sudenkuopat ja miten ne vältetään. Books on Demand.

- Jyrkämä, J. (2021). Toimintatutkimus. Teoksessa Jaana Vuori (toim.) Laadullisen tutkimuksen verkkokäsikirja. Tampere: Yhteiskuntatieteellinen tietoarkisto <https://www.fsd.tuni.fi/fi/palvelut/menetelmaopetus/kvali/tutkimusasetelma/toimintatutkimus/>
- Kajander, K., & Lappi, T.-R. (2022). Kulttuurien tutkimuksen menetelmät (O. Fingerroos, Ed.). SKS Finnish Literature Society. <https://doi.org/10.21435/tl.274>
- Kananen, J. (2019). Opinnäytetyön ja pro gradun pikaopas. Avain opinnäytetyön ja pro gradun kirjoittamiseen. Jyväskylän ammattikorkeakoulu.
- Kananen, J. (2014). Toimintatutkimus kehittämistutkimuksen muotona. Miten kirjoitan toimintatutkimuksen opinnäytetyönä? Jyväskylän ammattikorkeakoulu.
- Kasurinen, J. (2013). Ohjelmistotestauksen käsikirja. Docendo.
- Kasurinen, J. (2011). Software Test Process Development. Väitöskirja. Lappeenranta University of Technology <https://urn.fi/URN:ISBN:978-952-265-144-4>
- Laanti, M. (30.1.2019). Ketteryydestä ei ole paluuta: Suomalaiset yritykset ovat ketteröityneet edelläkävijöiksi [blogikirjoitus]. Nitor. Saatavilla 5.3.2024 <https://nitor.com/fi/artikkelit/ketteryydesta-ei-ole-paluuta-suomalaiset-yritykset-ovat-ketteroityneet-edellakavijoiiksi>
- Lippo, A., Järvi, H., Poikolainen, T. (2022). Osallistava työpaja yhteiskehittämisen menetelmänä. Saatavilla 14.2.2025 <https://read.xamk.fi/2022/logistiikka-ja-merenkulku/osallistava-tyopaja-yhteiskehittamisen-menetelmana>
- Maijala, R. (2019). Lean terveydenhuollossa – Näkökulmina hukka ja johtaminen. Väitöskirja. Turun Yliopisto <https://urn.fi/URN:ISBN:978-951-29-7827-4>
- Medavarapu, S. V. (2023). Agile metrics and performance measurement. Journal of Scientific and Engineering Research, 10(4), 152–156. <https://jsaer.com/download/vol-10-iss-4-2023/JSAER2023-10-4-152-156.pdf>

Moe, N. B., Cruzes, D., Dyba, T., & Mikkelsen, E. (2015). Continuous Software Testing in a Globally Distributed Project. 2015 IEEE 10th International Conference on Global Software Engineering, 130–134. <https://doi.org/10.1109/ICGSE.2015.24>

Mäenpää, K. (N.d.) Mitä LEAN on? – Mitä LEAN ei ole?. Lean Thinking. Saatavilla 4.11.2025 <https://leanthinking.fi/mika-ja-mita-lean-on/>

Mäntyranta, T. & Kaila, M. (2008). Fokusryhmähaastattelu laadullisen tutkimuksen menetelmänä lääketieteessä. Lääketieteellinen Aikakauskirja Duodecim. 124 (13), 1507–13. Saatavilla 8.10.2024 <https://www.duodecimlehti.fi/duo97349>

Niemi, A. & Hietaniemi, J. (2020). Just sopivasti ketterä. Ketterällä kehityksellä kohti parempaa elämää. Art House.

Nummi, P. (2018). Fasilitoivan johtamisen käsikirja. 9 avainhetkeä. Alma Talent Oy.

Otala, L. (2018). Ketterä oppiminen. Keino menestyä jatkuvassa muutoksessa. Helsingin kamari Oy.

Paasivaara, M., Behm, B., Lassenius, C. et al. (2018). Large-scale agile transformation at Ericsson: a case study. *Empir Software Eng* 23, 2550–2596. <https://doi.org/10.1007/s10664-017-9555-8>

Paul, A. J., & Eldhose, S. (2020). Development of Framework for Residential Building Construction Using Agile Management, 725–737. https://doi.org/10.1007/978-3-030-26365-2_67

Pavelin, K., Pundir, S. & Cham, J. A. (2014). Ten Simple Rules for Running Interactive Workshops. *PLso Computational Biology*, 10 (2). Saatavilla 14.2.2025 <https://pub-med.ncbi.nlm.nih.gov/24586135/>

Plengvittaya, C., & Sanpote, D. (2018). Scrumban for teaching at undergraduate program: A case study from software engineering students, University of Phayao, Thailand. 2018 International Conference on Digital Arts, Media and Technology (ICDAMT), 109–114. <https://doi.org/10.1109/ICDAMT.2018.8376505>

Puusa, A. & Juuti, P. (2020). Laadullisen tutkimuksen näkökulmat ja menetelmät. Gaudeamus Oy.

PractiTest Team. (2025). Effective Test Reporting: Key Components & Best Practices. Saatavilla 8.11.2025 <https://www.practitest.com/resource-center/article/effective-test-reporting/>

Project Management Institute. (2017). Agile Practice Guide. Project Management Institute.

Raulamo-Jurvanen, P. (2020). Evaluating and selecting software test automation tools: synthesizing empirical evidence from practitioners. University of Oulu <https://urn.fi/URN:ISBN:9789526227665>

Saaranen-Kauppinen, A. & Puusniekka, A. (2006). KvaliMOTV – Menetelmäopetuksen tietovaranto. Yhteyskuntatieteellinen tietoaarkisto. Saatavilla 19.9.2024 <https://www.fsd.tuni.fi/menetelmaopetus/kvali/>

Schwaber, K., Sutherland, J. 2020. Scrum-opas – Scrumin määritelmä ja pelisäännöt. Saatavilla 23.3.2024. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Finnish.pdf>

Sinivuo, R., Koivula, M., & Kylmä, J. (2012). Havainnointi aineistonkeruumenetelmänä kliinisessä ympäristössä. *Hoitotiede*, 24(4), 291–301. <https://journal.fi/hoitotiede/article/view/128253/77376>

Scrum.org. (2023.a.). The Increment. Saatavilla 9.11.2025 <https://www.scrum.org/learning-series/increment/node/73824>

Scrum.org. (2023.b.). The Product Backlog. Saatavilla 9.11.2025 <https://www.scrum.org/learning-series/product-backlog/>

Scrum.org. (2023.c.). The Scrum Artifacts. Saatavilla 9.11.2025 <https://www.scrum.org/learning-series/scrum-artifacts/>

Scrum.org. (2023.d.). The Sprint Backlog. Saatavilla 9.11.2025 <https://www.scrum.org/learning-series/sprint-backlog/>

Sixsigma. (N.d.). Leanin historia. Saatavilla 18.11.2024 <https://sixsigma.fi/leanin-historia/>

Sixsigma.us (2024). What is an Agile Team? Structure, Principles, and Best Practices. Saatavilla 1.3.2024 <https://www.6sigma.us/six-sigma-in-focus/agile-team/>

Smith, J., & Hayes, W. (2024). Independent Verification & Validation (IV&V) for Agile Developed Projects. IEEE Aerospace Conference Proceedings. <https://doi.org/10.1109/AERO58975.2024.10521415>

Steegh, R., Van De Voorde, K., & Paauwe, J. (2025). Understanding how agile teams reach effectiveness: A systematic literature review to take stock and look forward. *Human Resource Management Review*, 35(1), 1–17. <https://doi.org/10.1016/j.hrmr.2024.101056>

Strode, D., Dingsøy, T., & Lindsjorn, Y. (2022). A teamwork effectiveness model for agile software development. *Empirical Software Engineering*, 27(2), 56. <https://doi.org/10.1007/s10664-021-10115-0>

Sydänmaanlakka, P. (2024). Innostu, innosta, innovoi. Alma Talent.

The Home of Scrum. (2023). What is Scrum? Saatavilla 28.12.2024 <https://www.scrum.org/resources/what-scrum-module>

Torkkola, S. (2015). Lean asiantuntijatyön johtamisessa. Talentum pro.

Tran, H. K. V., Ali, N. bin, Unterkalmsteiner, M., Börstler, J., & Chatzipetrou, P. (2025). Quality attributes of test cases and test suites – importance & challenges from practitioners' perspectives. *Software Quality Journal*, 33(1), 9. <https://doi.org/10.1007/s11219-024-09698-w>

Tuomikoski, J., & Tervonen, I. (2009). Absorbing Software Testing into the Scrum Method (pp. 199–215). https://doi.org/10.1007/978-3-642-02152-7_16

Työturvallisuuskeskus. N.d. Yhteistoiminta ja vuorovaikutus työpaikan arjessa. Saatavilla 2.2.2025 https://ttk.fi/julkaisu/yhteistoiminta-ja-vuorovaikutus-tyopaikan-arjessa/?utm_source=chatgpt.com

Van den Broek, R., Bonsangue, M. M., Chaudron, M., & van Merode, H. (2014). Integrating Testing into Agile Software Development Processes. Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development, 561–569. <https://doi.org/10.5220/0004877105610569>

Vilkka, H. (2021). Tutki ja kehitä. PS-Kustannus.

Vuorinen, T. 2013. Strategiakirja: 20 työkalua. Talentum Media Oy.

Wang, X.-L., Zeng, H.-W., & Lin, W.-W. (2019). Techniques for Regression Testing in Agile Development Environment. Jisuanji Xuebao/Chinese Journal of Computers, 2323–2338. <http://doi.org/10.11897/SP.J.1016.2019.02323>

Winkler, D., Meixner, K., Lehner, D., & Biffel, S. (2019). Test Reporting at a Large-Scale Austrian Logistics Organization: Lessons Learned and Improvement (pp. 53–69). https://doi.org/10.1007/978-3-030-35333-9_4

Opinnäytetyön aineistonhallintasuunnitelma

1. Aineistojen yleinen kuvaus

- Opinnäytetyön tutkimusaineisto koostuu pääasiassa haastatteluista ja havaintomuistiinpanoista, jotka on tallennettu tekstimuodossa (.docx- ja .one-tiedostoina), jotka ovat tallennettu toimeksiantajan sisäisiin järjestelmiin.

2. Aineiston dokumentaatio ja laatu

- Tutkimusaineisto dokumentoidaan vain valmiiseen opinnäytetyöhön, joka tallennetaan Theseukseen. Muita dokumentteja ei tulla säilyttämään.

3. Säilytys ja varmuuskopiointi

- Tutkimusaineisto dokumentoidaan vain valmiiseen opinnäytetyöhön, joka tallennetaan Theseukseen. Muita dokumentteja ei tulla säilyttämään.

4. Säilyttämiseen liittyvät eettiset ja laillisuuskysemykset

- Opinnäytetyön aikana ei kerätä henkilötietoja eikä aineistoa, josta voisi tunnistaa yksittäisiä henkilöitä.

5. Aineiston avaaminen ja pitkäaikaissäilytys

- Aineisto säilytetään opinnäytetyön teon ajan toimeksiantajan sisäisissä järjestelmissä ja opinnäytetyön valmistuttua aineisto tuhoetaan. Jäljelle jää vain Theseukseen tallennettu valmis opinnäytetyö, joka on julkisesti saatavilla.

Tiedote tutkittavalle

Hei,

Teemme Muiden HR-järjestelmien tuki -yksikössä tutkimusta kvartaalitestaustusprosessin selkeyttämisestä ketteriä menetelmiä hyödyntäen.

Sinut on kutsuttu tutkimukseen mukaan, koska osallistut yhteen tai useampaan tutkimukseen kuuluvaan palaveriin (Scrumban-koulutus, ryhmähaastattelu, kvartaalitestaustus tai työpaja). Osallistuminen on vapaaehtoista. Jos et halua osallistua tutkimukseen, ilmoita siitä minulle niin jätän osallistumisesi palaverissa tutkimuksen ulkopuolelle. Aineistonhankintamenetelmänä käytän ryhmähaastattelua ja havainnointia, vastaukset analysoidaan luottamuksellisesti allekirjoittaneen toimesta ja vastaukset jäävät vain minun haltuuni. Tulokset raportoidaan ja esitetään niin ettei niistä tunnisteta vastaajaa. Aineisto tuhoetaan opinnäytetyön hyväksynnän jälkeen.

Opiskelen ylempää ammattikorkeakoulututkintoa Kajaanin ammattikorkeakoulussa tietojenkäsittelyn ja liiketoimintaosaamisen koulutusohjelmassa ja tutkimus on osa opinnäytetyötäni.

Kiitos etukäteen osallistumisestasi tutkimukseen!

Jos sinulla herää kysyttävää, vastaan mielelläni!

Ystävällisin terveisin

Jenni Peltomäki

Testaussuunnitelman sisällysluettelo

Sisällys

Testaussuunnitelma.....	1
1 Dokumentin tarkoitus.....	4
2 Testaussuunnitelman yleistiedot.....	4
2.1 Testaussuunnitelman hyväksyntä.....	4
2.2 Testaussuunnitelman versio, tunnus ja tilatieto.....	4
3 Testauksen tavoitteet, kohde ja vastuuhenkilöt	4
3.1 Testauksen tavoitteet	4
3.2 Testauksen kohde	5
3.3 Testaustaso	5
3.4 Testauksen rajaaminen	5
3.5 Testauksen pohjana olevat dokumentit	5
3.6 Vastuuhenkilöiden roolit ja yhteystiedot	5
4 Testausmenetelmät ja -käytännöt	6
4.1 Projektissa käytettävä virheenkirjausprosessi.....	6
4.2 Testauksen työvälineet.....	6
4.3 Testauksen etenemisen ehdot ja kriteerit	6
4.4 Testauksessa löytyvien havaintojen vakavuusluokitus	8
4.5 Testauksen riskit	9
4.6 Raportointi ja tiedottaminen	10
4.7 Seurantakokoukset	11
5 Testauksen vaiheet, aikataulu, tehtävät ja tekijät.....	11
5.1 Testauksen vaiheet ja aikataulu	11
5.2 Testaajien opastaminen	12
5.3 Testaajat sekä työmääräarviot.....	13
6 Testauksen toteutus	13
6.1 Testiympäristöt.....	13
6.2 Testausapuvälineet ja aineisto.....	13
6.3 Testauksen dokumentointi.....	13

Esimerkki teemoittelusta

Alla oleva taulukko havainnollistaa, miten fokusryhmähaastattelun aineisto on analysoitu teemoittelun avulla. Haastatteluvastaukset on litteroitu, koodattu ja ryhmitelty pää- ja alateemoihin. Esimerkkikommentit ovat valikoituja sitaatteja, jotka edustavat tyypillisiä näkemyksiä ja havainnollistavat keskeisiä teemoja. Kaikkia alkuperäisiä vastauksia ei ole sisällytetty liitteeseen, jotta kokonaisuus pysyy selkeänä ja tutkimuskysymyksiin keskittyvänä.

Pääteema	Alateema	Esimerkkikommentti
Palaverikäytännöt ja tiedonkulku	Toimittajan palaverin hyödyttömyys	"En oikein näe hyötyä palaverissa, kun sama PowerPoint löytyy jo Jirasta."
Palaverikäytännöt ja tiedonkulku	Tarve testauksen suunnittelu-palaverille	"Kaipaam palaveria, jossa yhdessä käydään läpi testauksen alkua ja jaetaan testitapaukset."
Palaverikäytännöt ja tiedonkulku	Tiedonkulun haasteet	"Ei ole selkeää, missä vaiheessa testaus menee, kun tieto ei kulje tiimin sisällä."
Testausprosessin tehokkuus ja aikataulu	Testausaikataulun haasteet	"Olisi parempi, jos testaus olisi viikko ja toinen viikko varattu korjauksille."
Testausprosessin tehokkuus ja aikataulu	Testitapausten jakautuminen ja selkeys	"Osa testitapauksista on todella pitkiä ja haastavia, kun taas toiset ovat nopeita."
Testausprosessin tehokkuus ja aikataulu	Testauksen työkalut ja dokumentaatio	"Testitapaukset Xrayssa pitäisi järjestää selkeämmin."

Liitteessä esitetyt kommentit ovat valikoituja esimerkkejä, eivätkä sisällä kaikkia alkuperäisiä vastauksia.