



VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

Valtteri Autio

# Älykkään automaation testauksen ja sen johtamisen kehittäminen osastotasolla

Tekniikka  
2023

VAASAN AMMATTIKORKEAKOULU  
Tietotekniikka

## TIIVISTELMÄ

|                    |  |
|--------------------|--|
| Tekijä             | Valtteri Autio   |
| Opinnäytetyön nimi | Älykkään automaation testauksen ja sen johtamisen kehittäminen osastotasolla |
| Vuosi              | 2023   |
| Kieli              | suomi  |
| Sivumäärä          | 43   |
| Ohjaaja            | Kenneth Norrgård   |

---

Kehittämishanke toteutettiin erään yrityksen Kehitys-osastolle, joka vastaa älykkään automaation arkkitehtuurista, suunnittelusta, kehityksestä ja testauksesta. Kehittämishankkeen tavoitteena oli toteuttaa Kehitys-osastolle strukturoitu ja läpinäkyvä testausmalli sekä ottaa käyttöön sitä tukevat työkalut ja johtamistavat. Kehittämishankkeen aihe valittiin, sillä osaston testauksessa ja sen johtamisessa oli havaittu olevan puutteita, jotka vaikuttivat suoraan tuotannossa olevien ohjelmien laatuun. Lisäksi testauksen johtamisen puutteet hankaloittivat ja pitkittivät testauksen eri vaiheita.

Kehittämishankkeen kehitysmenetelmäksi valikoitui keskustelua edistävä fokusryhmä ja menetelmää tukivat erilaiset fokusryhmän workshopit. Valittu kehitysmenetelmä näkyi fokusryhmän tapaamisissa avoimena keskusteluna, jossa jokaiselle osallistujalle luotiin mahdollisuus osallistua keskusteluun ja tuoda omat ajatuksensa ja mielipiteensä julki. Fokusryhmä pyrki herättämään keskustelua myös fokusryhmän palaverien ulkopuolella koko Kehitys-osaston yhteisissä tapaamisissa.

Kehittämishankkeen tuloksena saatiin Kehitys-osastolle luotua läpinäkyvä ja selkeästi strukturoitu testausmalli. Lisäksi kehittämishankkeen myötä osastolla otettiin käyttöön testausta, sekä sen mittaamista tukevat dokumentointi- ja raportointityökalut. Kehittämishankkeen tulosta voidaan jatkokehittää ja laajentaa myös yrityksen muille osastoille.

---

Avainsanat                      Testaaminen, johtaminen, kehittäminen, älykäs automaatio

VAASAN AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES  
Tietotekniikka

## ABSTRACT

|                    |   |
|--------------------|---|
| Author             | Valtteri Autio  |
| Title              | Developing the testing of intelligent automation and its management at the department level |
| Year               | 2023  |
| Language           | Finnish   |
| Pages              | 43  |
| Name of Supervisor | Kenneth Norrgård  |

---

The development project was implemented in the Development Department of a company, responsible for the architecture, design, development, and testing of intelligent automation. The objective of the development project was to establish a structured and transparent testing model for the department and to implement supporting tools and management practices. The chosen topic for the development project was based on identified deficiencies in testing and its management within the department, which directly impacted the quality of the software in production. Furthermore, these deficiencies complicated and prolonged the different phases of testing.

The development methodology selected for the project was a discussion-based focus group, supported by various workshops. This chosen approach was reflected in the open discussions during focus group meetings, where each participant was given the opportunity to contribute to the conversation and express their thoughts and opinions. The focus group also aimed to stimulate discussions outside of their meetings, within the department's collective gatherings.

As a result of the development project, a transparent and clearly structured testing model was established within the department. Additionally, the project introduced documentation and reporting tools to support testing and its measurement. The outcomes of the development project can be further developed and extended to other departments within the company.

---

|          |  |
|----------|--|
| Keywords | Testing, leadership, development, intelligent automation |
|----------|--|

## SISÄLLYS

TIIVISTELMÄ

ABSTRACT

|     |   |    |
|-----|---|----|
| 1   | JOHDANTO.....   | 7  |
| 1.1 | Älykäs automaatio Kehitys-osastolla .....                       | 7  |
| 1.2 | Kehittämishankkeen tausta .....                                 | 8  |
| 1.3 | Kehittämishankkeen kehittämismenetelmä.....                     | 8  |
| 2   | TARVEANALYYSI JA NYKYTILAN KUVAUS.....                          | 10 |
| 2.1 | Testaus ja sen johtamisen tarpeet Kehitys-osastolla .....       | 10 |
| 2.2 | Nykytilan kuvaus .....  | 13 |
| 3   | TEORIATAUSTA .....  | 16 |
| 3.1 | Kehittämismenetelmät kehityshankkeessa .....                    | 16 |
| 3.2 | Ohjelmistotestaus ja testausmenetelmät .....                    | 17 |
| 3.3 | Testauksen tukena käytettävä projektinhallintatyökalu Jira..... | 19 |
| 3.4 | Confluence testauksen dokumentinhallintatyökaluna.....          | 20 |
| 3.5 | Testauksen ja projektin johtaminen .....                        | 20 |
| 3.6 | Älykäs automaatio .....   | 22 |
| 3.7 | SAFe®-viitekehys .....  | 23 |
| 4   | KEHITTÄMISPROSESSIN KUVAUS.....                                 | 25 |
| 4.1 | Ideointi .....  | 25 |
| 4.2 | Vaatimusten määrittely .....                                    | 26 |
| 4.3 | Suunnittelu.....  | 26 |
| 4.4 | Toteutus .....  | 27 |
| 4.5 | Käyttöönotto.....   | 30 |
| 4.6 | Jatkokehitys.....   | 30 |
| 5   | KEHITTÄMISTOIMINNAN KUVAUS .....                                | 32 |
| 5.1 | Tavoitteen asetanta .....                                       | 32 |
| 5.2 | Tarveanalyysi.....  | 33 |

|     |   |    |
|-----|---|----|
| 5.3 | Pilotointi ja arviointi.....              | 34 |
| 5.4 | Seuranta ja jatkokehitys.....             | 34 |
| 6   | KEHITTÄMISTOIMINNAN TULOKSEN KUVAUS ..... | 36 |
| 6.1 | Hyödyt.....                               | 36 |
| 6.2 | Haasteet .....                            | 37 |
| 7   | JOHTOPÄÄTÖKSET JA ARVIOINTI.....          | 39 |
| 7.1 | Luotettavuus .....                        | 39 |
| 7.2 | Käyttökelpoisuus.....                     | 39 |
| 7.3 | Siirrettävyys .....                       | 40 |
| 7.4 | Johtopäätökset .....                      | 40 |
|     | LÄHTEET .....                             | 42 |

**KUVIO- JA TAULUKKOLUETTELO**

|  |    |
|--|----|
| <b>Kuvio 1.</b> Kehittämistoiminnan menetelmiä ja välineitä (Salonen ja muut 2017, s. 55.) | 17 |
| <b>Kuvio 2.</b> Testauksen V-malli (Kasurinen 2013, s. 51)                                 | 18 |
| <b>Kuvio 3.</b> Testauksen elinkaari XRAY-lisäosaa käyttäessä (Docheva 2023)               | 20 |
| <b>Kuvio 4.</b> Pitkäaikainen Agile Release Train (©Scaled Agile 2023.)                    | 24 |
| <b>Kuvio 5.</b> Kehittämishankkeen prosessin kuvaus  | 25 |
| <b>Kuvio 6.</b> Testausprosessin eteneminen Kehitys-osastolla                              | 29 |
| <b>Kuvio 7.</b> Kuvaus kehittämissuorituksen vaiheista                                     | 32 |
| <br>   |    |
| <b>Taulukko 1.</b> Testauksen ja sen johtamisen tarpeet Kehitys-osastolla                  | 12 |

## 1 JOHDANTO

Kehittämishanke toteutetaan erään yrityksen kehittämisestä ja muutoksesta vastaavan yksikön Kehitys-osastolle. Kehitys-osasto vastaa kehitystoiminnan johtamisesta ja toteutuksista. Osaston työ käsittää kehityksen suunnittelun, toteutuksen ja testauksen yhdessä loppukäyttäjäorganisaatioiden kanssa. Kehitystiimin asiakaina toimivat niin sisäiset sidosryhmät kuin myös ulkoiset asiakkaat. Kehitys-osaston sisällä työ tehdään jatkuvan suunnittelun ja ketterien kehitysmenetelmien (mm. Scrum, Kanban, SAFe®) periaatteilla. Osaston vastuulla on varmistaa liiketoimintaedellytykset älykkään automaation toteutukseen sekä johtaa nopeita toteutuslähtöjä eri liiketoimintojen tilauksesta.

### 1.1 Älykäs automaatio Kehitys-osastolla

Kehitys-osasto toimii SAFe®-viitekehyksen mukaan ja kehyksen pohjana on ”Ketterä kehitysjunan” (Agile Release Train, ART). ARTia johtaa nimetty RTE (Release Train Engineer), joka vastaa ”Ketterän kehitysjunan” valmennuksesta, ja sujuvasta etenemisestä. Lisäksi Kehitys-osastolla on kolme PO:ta (Product Owner), jotka vastaavat kukin omista tuotealueistaan. Jokaisella tuotealueella on lisäksi omat Scrum Masterinsa, joiden tehtävänä on varmistaa, että kehitystyö etenee ketterästi. Lisäksi ARTilla on oma System Team, joka koostuu Tech Leadeista, testauspäälliköstä ja testaajasta. System Teamin vastuulla on muun muassa järjestelmäarkkitehtuuri, suorituskyky ja testaus. Loput ARTin asiantuntijat vastaavat älykkään automaation suunnittelusta ja kehityksestä. Lisäksi ARTia tukee ylläpidosta ja viankorjauksesta vastaava tiimi. Älykkään automaation kulmakivinä Kehitys-osastolla ovat lukuisat automaatio-, ohjelmisto- ja työpöytärobotit, chatbotit ja tekoäly, jotka helpottavat asiakaspalvelutyötä ja mahdollistavat kustannustehokkaan ja varman tavan palvella Kehitys-osaston loppuasiakkaita.

Samalla kun älykkään automaation erilaisten sovellusten määrä yrityksessä on kasvanut, ja jo olemassa oleviin sovelluksiin on tullut tarve integroida uusia toiminnallisuuksia, on huomattu tarve testausmenetelmien ja työkalujen sekä testauksen

johtamisen kehittämiseksi. Älykkään automaation kehitystiimin testauksessa on kiinnitettävä huomiota kehityksen eri vaiheissa alun Ad-hoc-testauksesta järjestelmäintegraatiotestaukseen (SIT) ja aina hyväksymistestaukseen (UAT) saakka.

## **1.2 Kehittämishankkeen tausta**

Tarve älykkään automaation testausmenetelmien ja työkalujen sekä testauksen johtamisen implementoinnin kehitykseen perustuu haluun luoda malli, joka tehostaa ja sujuvoittaa testausta ja sen johtamista. Mallin tavoite on sitouttaa kehittäjät ja testaajat toteuttamaan yhdessä testaukseen valittuja periaatteita ja dokumentoimaan havaintoja sovittujen käytäntöjen mukaisesti.

Tavoitteena on myös luoda raamit testauksen johtamiselle siten, että testaukset suoritetaan luodun testausmenetelmän mukaisesti sekä valvoa, että testaussuunnitelmaa noudatetaan. Testauksesta vastaavan henkilön vastuulla on jatkossa myös tuottaa tarvittavat raportit sekä huolehtia, että Kehitys-osasto toteuttaa yrityksen testauspolitiikkaa ja -strategiaa.

Älykkään automaation kehityksessä ja tuotannossa käytettävät ympäristöt ovat monimutkaisia ja erilaisia taustarobotteja- ja järjestelmiä sekä kohdejärjestelmiä on lukuisia. Tämän johdosta hankkeessa päädyttiin soveltamaan V-mallia, joka on riittävän kattava malli älykkään automaatio testaamiseen. V-malli käsittää yksikkö-, integrointi-, ja järjestelmätestauksen joita seuraa hyväksymistestaus, jonka jälkeen tuote on valmis julkaistavaksi. (Kasurinen 2013, s. 51).

## **1.3 Kehittämishankkeen kehittämismenetelmä**

Kehittämishankkeen kehitysmenetelmäksi valikoitui keskustelua edistävä fokusryhmä, jonka koordinoituvastuu on kehityshankkeen johtajalla. Fokusryhmä koostuu eri toimintojen asiantuntijoista ja sen tavoitteena on kehittää malli, jolla osaston ohjelmistotestausta voidaan edelleen parantaa.



Fokusryhmä työskenteli pääosin online workshopeissa, mutta myös erilliset ad-hoc palaverit kuuluivat asiaan tarpeen mukaan. Valittu kehitysmenetelmä näkyi fokusryhmän tapaamisissa avoimena keskusteluna, jossa jokaiselle osallistujalle luotiin mahdollisuus osallistua keskusteluun ja tuoda omat ajatuksensa ja mielipiteensä julki. Fokusryhmä pyrki myös herättämään keskustelua myös fokusryhmän palaverien ulkopuolella koko kehitys-osaston yhteisissä tapaamisissa. Näin toimimalla saatiin kerättyä mahdollisimman laajalta joukolta kehitysideoita ja muutostarpeita toiminnan kehittämisen tueksi.

## 2 TARVEANALYYSI JA NYKYTILAN KUVAUS

Kehittämishankkeen alkuvaiheessa luotiin tarveanalyysi Kehitys-osaston testaukselle, sekä testaamisen johtamiselle ja kuvattiin millainen nykytilanne oli. Tarveanalyysi tehtiin pari- ja ryhmätyönä Kehitys-osaston sisällä. Tarveanalyysissä pohdittiin, mitä testauksessa sekä sen johtamisessa halutaan saavuttaa ja miksi tavoiteltavat asiat ovat tärkeitä. Tarveanalyysin aikana kuvattiin myös Kehitys-osaston testauksen nykytilanteen silloiset haasteet ja sellaiset asiat, jotka oli koettu toimiviksi ja kannatti pitää mukana myös jatkossa.

### 2.1 Testaus ja sen johtamisen tarpeet Kehitys-osastolla

Kehitys-osaston testauksen yli tiimi- ja osastorajojen tulee olla läpinäkyvää. Kun testauksesta saadaan läpinäkyvää, vältetään siltä, että päällekkäinen kehitys häiritsee testausta. Tällöin testauksesta tulee entistä sujuvampaa ja laadukkaampaa. Testauksen läpinäkyvyyttä voidaan parantaa testauksen tukena käytettävien projektointityökalujen laajamittaisemmalla hyödyntämisellä sekä paremmalla viestinnällä sidosryhmien suuntaan. Testauksen läpinäkyvyyden ansioista voidaan jatkossa myös paremmin arvioida, kuinka paljon aikaa testaukselle kannattaa varata.

Testausmallin tulee olla strukturoitua, jotta se tukee tavoitetta testauksen läpinäkyvyydestä. Testauksen tulee olla paremmin organisoitua, dokumentoitua ja raportoitua. Tämän avulla voidaan paremmin ylläpitää ja valvoa testauksen laatua ja sitä, että asiat etenevät sovitusti ja vaaditut asiat tulee testattua. Strukturoidusta testausmallin ylläpidosta on vastuussa testauspäällikkö, joka varmistaa, että asiat dokumentoidaan riittävällä tasolla.

Myös testauksen tulosten raportoinnin laatuun on jatkossa kiinnitettävä huomiota. Raportoinnin tulee olla selkeää ja kattavaa. Siitä tulee ilmetä mitä asioita on testattu ja mitä on vielä testaamatta. Raportoinnin nimeämistapa on oltava selkeä ja johdonmukainen. Raportointi täytyy olla helposti löydettävissä niin, että se

on kaikille tarpeellisille sidosryhmille saatavilla. Raportoinnin laatuun on kiinnitettävä huomiota sillä sen perusteella voidaan tulkita, mitkä tapaukset aiheuttavat ongelmia ja jatkossa näihin voidaan kiinnittää paremmin huomiota jo analyysi- ja kehitysvaiheessa. Näin toimien vastaavilta ongelmilta voidaan vastaisuudessa välttyä.

Jatkossa testaus tulee suorittaa ensisijaisesti testiympäristössä tuotantoympäristön sijaan. Jos testauksen yhteydessä on tarve käyttää asiakastietoja, niin käytetään testikannassa olevia asiakkaita, joiden asiakastiedot on anonymisoitu, eikä niistä voi tunnistaa henkilötietoja. Tämän avulla testaamisesta tulee turvallista ja välttyään GDPR-rikkomusten riskeiltä. Testaus myös nopeutuu, kun testikannassa olevien testiasiakkaiden tietoja ja palveluja voidaan muokata halutunlaisiksi.

Tiivis yhteistyö testaajien, kehitystiimien ja muiden sidosryhmien välillä on myös tarpeellista jatkossa. Sen avulla Kehitys-osastolla päästään laadukkaampaan testaukseen ja nopeampaan viankorjaukseen ja sitä kautta uudelleen testaukseen. Jatkossa on määriteltävä myös aikaisemmassa vaiheessa, minkä tahon vastuulla on testitapausten suunnittelu, jotta ne ovat riittävän kattavat ja vastaavat mahdollisimman hyvin oikeita käyttötapauksia.

Testauksen vauhdittamiseksi ja laadun parantamiseksi Kehitys-osastolla on selkeä tarve ammattitestaajalle, jonka tulisi tuntea jo entuudestaan tärkeimmät sidosryhmät ja taustajärjestelmät. Tämä sujuvoittaisi testauksen kulkua ja toisi joustoa testauksen aikataulutukseen. Ammattitestaaja voisi myös täysipainoisesti keskittyä testaamiseen sekä sen kehittämiseen ja sitä kautta parantaa testausprosesseja omalla panoksellaan.

Jatkossa testauspäällikön on vastattava testauksen johtamisesta, sen kehittämisestä, laadun valvonnasta ja oltavana tukena testitapausten suunnittelussa. Lisäksi testauspäällikön vastuulla on varmistaa, että testaus tapahtuu testausympäristöissä ja että kaikilla vaadituilla resursseilla on tarvittavat järjestelmät käytettä-

vissä testausta varten. Testauspäällikön on myös toimittava aktiivisesti eri sidosryhmien välillä ja hankittava lisää tietoutta testaamisesta ja sen teoriasta sekä mahdollisesti hankkia lisää osaamista erilaisten testaukseen liittyvien sertifikaattien muodossa. Nämä toimenpiteet kohottavat testauksen laatua, varmuutta ja selkeyttävät toimintatapoja.

Testauspäällikön tulee olla sisäinen resurssi, jonka avulla pyritään sitoutuneempaan suorittamiseen ja pitkäjänteiseenpään testauksen kehittämiseen. Samalla testauksesta saadaan myös kustannustehokkaampaa.

**Taulukko 1.** Testauksen ja sen johtamisen tarpeet Kehitys-osastolla

| Välttämättömät  | Hyödylliset  |
|---|--|
| Läpinäkyvä testaus tiimi- ja osastorajojen yli                                    | Laajamittainen projektinhallintatyökalujen hyödyntäminen testauksen tukena |
| Strukturoidun testausmallin käyttö testauksen läpinäkyvyyden tukemiseksi          | Tehostettu viestintä sidosryhmien suuntaan                                 |
| Organisoitu, dokumentoitu ja raportoitu testaus                                   | Laadukas raportointi   |
| Testaus pääasiassa suoritetaan testiympäristössä, ei tuotantoympäristössä         | Anonymisoidut asiakastiedot testauksessa                                   |
| Tiivis yhteistyö testaajien, kehitystiimien ja sidosryhmien välillä               | Selkeä vastuunjako testitapausten suunnittelussa                           |
| Ammattitestaaja, jolla on tuntemusta tärkeimmistä sidosryhmistä ja järjestelmistä | Testauspäällikkö tukemaan testitapausten suunnittelua                      |

## 2.2 Nykytilan kuvaus

Nykyisellään Kehitys-osastolla on paljon erilaisia kehitystiimejä, jotka kehittävät erilaisia älykkään automaation ratkaisuja hyödyntäen eri teknologioita ja kehitysympäristöjä. Eri tiimit vastaavat suurimmaksi osaksi itse testauksesta ja sen suunnittelusta. Myös testauksen johtaminen ja seuranta sekä raportointi on kehittäjien vastuulla. Poikkeuksena ovat isoimmat kokonaisuudet, jotka näkyvät suoraan loppukäyttäjille. Silloin mukana on testauspäällikkö, joka vastaa edellä mainituista asioista. Tämä johtaa siihen, että testaus saattaa olla vaillinaista eikä testauksessa oteta huomioon tarvittavan monia eri käyttötapauksia. Tällaisissa tapauksissa tuotantoon saattaa päästä ohjelma, joka aiheuttaa ongelmia.

Koska useimmissa tapauksissa testaus suoritetaan omissa tiimeissä ja testitapaukset ja raportointi ovat lokaalisti kehittäjien omilla tietokoneilla, aiheuttaa tämä informaatiokatkoksen ja läpinäkyvyyden eri sidosryhmien välillä heikkenee. Eri tiimit eivät tiedä, mitä testausta on menossa missäkin ympäristössä ja tämä saattaa aiheuttaa ongelmia testiajoissa. Voi olla tilanne, jossa toisen kehitystiimin testaus sotkee jonkin toisen tiimin testaukset, eikä välttämättä tiedetä keneltä lähteä selvittämään asiaa. Tämä ongelma saattaa myös johtaa turhaan vianselvitykseen jolloin kehittäjien aikaa menee hukkaan.

Kehitysosastolla on havaittu ongelmia testausresurssien saatavuudessa. Se johtaa testauksen aloittamisen viivästymiseen, joka johtaa taas siihen, että valmista ohjelmaa ei päästä julkaisemaan eikä siitä saada sille laskettuja hyötyjä riittävän nopeasti. Jos julkaisematonta ohjelmakoodia alkaa kertymään paljon, kasvaa myös regressiotestauksen tarve, mikä omalta osaltaan kasvattaa testaukseen käytettävää tuntimäärää. Taas toisaalta, jos testausresurssit ovat käytettävissä, niin voi olla tilanne, että kehittäjät ovat jo uuden työn parissa. Silloin kehittäjät eivät ehdi korjaamaan ohjelmasta mahdollisesti löytyviä vikoja. Myös tämä viivästyttää testauksen valmistumista.

Testitapauksien suunnitteluun ei ole olemassa mitään yhdessä sovittuja menettelyjä vaan ne ovat monesti täysin kehittäjien vastuulla. Valmista ohjelmakoodia testataan savutesteillä, minkä jälkeen se menee joko uudelleen kehitykseen tai julkaisuun. Testitapaukset eivät kuitenkaan ole aina riittävän kattavia ja tuotantoon saattaa päästä viallista ohjelmakoodia. Poikkeuksena on isoimmat kokonaisuudet, joissa on mukana useamman eri teknologian ja ympäristön riippuvuuksia. Tällöin testausta on tavanomaisesti johtanut testauspäällikkö, jonka vastuulla on ollut varmistaa, että testitapaukset on suunnitellut taho, joka ymmärtää ohjelmakoodin eri riippuvuudet ja käyttötapaukset. Myös testauspäällikön johtamat testaukset saattavat olla liian yksipuolisia tai kevyitä.

Useissa tapauksissa testaus suoritetaan tuotantoympäristössä, vaikka nykyään löytyy kaikki valmiudet testata testiympäristössä. Tämä johtuu siitä, että kun aiemmin ei ollut valmiuksia testata testiympäristössä, niin kaikkiin kehitysympäristöihin ei ole luotu rajapintoja testiympäristön suuntaan, vaan ainoastaan tuotantoon. Tuotantoympäristössä testaaminen on hankalaa ja aikaa vievää, kun jokaista eri vaihetta täytyy valvoa tarkasti. Onneksi tämän asian korjaaminen on kuitenkin enemmän aikataulu- ja resurssikysymys, kuin tekninen.

Isompien kokonaisuuksien testaamisen johtaminen on hyvällä tasolla. Testauksesta vastaava henkilö, yleensä testauspäällikkö, varmistaa että testitapaukset ovat riittävän kattavat ja ne testaavat oikeita asioita. Testauspäällikkö myös järjestää testauksen aloituspalaverin, jossa katsotaan testitapaukset vielä kertaalleen läpi testaajien, kehittäjien ja oleellisten sidosryhmien kanssa. Testitapauksia verrataan ohjelman spesifikaatioon. Tässä vaiheessa testisuunnitelmassa olevia testitapauksia yleensä vielä parannellaan ja niitä voidaan myös tarvittaessa lisätä. Testauksen aloituspalaverissa tarkistetaan myös testaajien aikataulut, sekä sovitaan testauksen aikainen tapa kommunikoida. Tämä käytäntö on todettu toimivaksi ja sitä kannattaa hyödyntää laajemminkin osastolla.

Aiemmin Kehitys-osaston testauksen johtamisesta ovat vastanneet ulkoiset konsultit. Tähän ratkaisuun ollaan aikoinaan päädytty, kun ollaan todettu tarve paremmin organisoidulle testaamiselle, mutta uuden henkilön palkkaaminen ei ollut vaihtoehto. Pätevän ammattilaisen avulla Kehitys-osastolla ollaan saatu ymmärrystä testaamisen eri vaiheista ja keinoista. Ulkoiset konsultit ovatkin osaltaan luoneet raamit tulevaisuuden testaukselle. Samalla Kehitys-osastolla on huomattu, että ulkoisten konsulttien sitoutumisessa on ollut vaihtelua, joka on vaikuttanut suoraan testauksen laatuun.

Kehitys-osastolla halutaan yhtenäistää testausmenetelmät- ja työkalut, sekä parantaa testauksen johtamismalleja. Tämän uskotaan tukevan älykkään automaation kehitystä jolloin myös tuotantoon menee enemmän ehjää ohjelmakoodia. Tämä johtaa siihen, että manuaaliryöön määrä vähenee ja älykäs automaatio pystyy tuottamaan pitkällä tähtäimellä parempaa arvoa.

### 3 TEORIATAUSTA

Vaadittavan tietoperustan hankkimiseksi kehittämishankkeelle tulee tutustua ohjelmistotestausta ja projektin johtamista käsittelevään kirjallisuuteen, sekä niihin liittyvien työkalujen teoriaan. Lisäksi kehittämishankkeen onnistumisen varmistamiseksi tulee perehtyä kehittämishankkeen toteuttamiseen liittyviin tietolähteisiin. Jotta tietoperusta olisi mahdollisimman kattava, tulisi perehtyä myös älykästä automaatiota koskevaan kirjallisuuteen.

#### 3.1 Kehittämismenetelmät kehityshankkeessa

Kehittämisen menetelmiä voidaan jaotella usein eri tavoin. Siitä, kuinka ne sopivat eri tutkimusmenetelmiin ja projektitoiminnassa sovellettaviin käytäntöihin, on erilaisia näkemyksiä. Joskus työskentelyä edistävät parhaiten tutkimusmenetelmät, kuten kysely tai haastattelu tarpeiden tunnistamisessa tai tuloksien arvioinnissa. Yleisesti menetelmissä on kuitenkin kyse käytännön keinoista ja tavoista tehdä työtä, jotta päästään kehittämistoiminnan tavoitteisiin. Kuitenkin se, mitä menetelmäksi kutsutaan, vaihtelee. Kehittämistoiminnan menetelmiä ovat esimerkiksi osallistavat menetelmät, esittävät menetelmät sekä keskustelua edistävät menetelmät, jotka on kuvattu **Kuvio 1**. (Salonen ja muut 2017, s. 55.)

Kirjallisuudessa ja käytännön työskentelyssä ei ole tarkkoja sääntöjä menetelmien, työkalujen ja välineiden välillä, joten ne ovat sopimuksenvaraisia ja tulkinnallisia. Kehittämisen menetelmiksi kutsutaan niin pienempiä kuin suurempiakin kehittämisen menetelmiä. Näitä ovat esimerkiksi Lean-ajattelu, palvelumuotoilu, tasapainotettu tulokortti BSC, työkonferenssi ja standardisoidut laatujärjestelmät. Jotkut näistä auttavat kehittämään uusia ideoita, toiset rohkaisevat keskustelemaan tai osallistumaan, kun jotkut taas soveltuvat kehittämistoiminnan kuvaamiseen ja esittämiseen. (Salonen ja muut 2017, s. 56.)



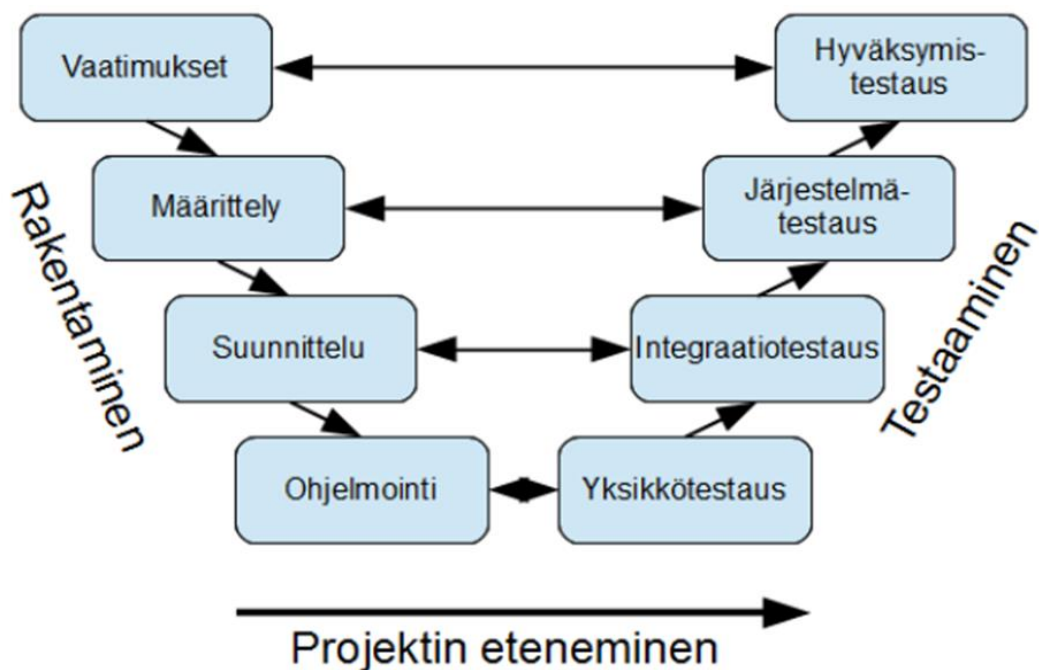


**Kuvio 1.** Kehittämistoiminnan menetelmiä ja välineitä (Salonen ja muut 2017, s. 55.)

### 3.2 Ohjelmistotestaus ja testausmenetelmät

Älykkään automaation kehityksessä ja tuotannossa käytettävät ympäristöt ovat monimutkaisia ja erilaisia taustarobotteja ja -järjestelmiä sekä kohdejärjestelmiä on lukuisia. Tämän johdosta päädyin soveltamaan hankkeessa V-mallia, joka on kuvattu **Kuvio 2.**, se on riittävän kattava malli älykkään automaatio testaamiseen. V-malli käsittää yksikkö,- integrointi,- ja järjestelmätestauksen, joita seuraa hyväksymistestaus, jonka jälkeen tuote on valmis julkaistavaksi. (Kasurinen 2013, s. 51).

V-mallissa testauksen ensimmäinen vaihe on yksikkötestaus. Siinä testataan ohjelmiston yksittäisiä yksiköitä tai komponentteja. Tarkoituksena on varmistaa, että jokainen ohjelmistokoodin yksikkö toimii suunnitellusti. Yksikkötestaus tehdään sovelluksen kehittämisen (koodausvaiheen) aikana kehittäjien toimesta. Yksikkötestit käsittelevät tiettyä osaa koodista ja varmistavat sen laadun. Yksikkö voi olla yksittäinen toiminto, menetelmä, proseduuri, moduuli tai objekti. (Hamilton 2023.)



**Kuvio 2.** Testauksen V-malli (Kasurinen 2013, s. 51)

Järjestelmätestauksessa varmistetaan, että kokonainen ja täysin integroitu ohjelmistotuote toimii. Järjestelmätestauksessa arvioidaan end-to-end-järjestelmän vaatimusten täyttämistä. Monesti testattava ohjelma on vain osa laajempaa kokonaisuutta, joka tullaan lopuksi liittämään muihin ohjelmisto-/laitteistojärjestelmiin. Järjestelmätestauksessa ajetaan useita testejä, joiden ainoana tarkoituksena on käyttää koko järjestelmää. (Hamilton 2023). Järjestelmäintegraatiotestaus (SIT) testaa eri järjestelmien välisiä vuorovaikutuksia ja se voidaan tehdä järjestelmätestauksen jälkeen. (Hamilton 2023).

Hyväksymistestaus (UAT) on loppukäyttäjän tai asiakkaan suorittama testaus, jolla varmistetaan ohjelmistojärjestelmän toiminta ennen ohjelmistosovelluksen siirtämistä tuotantoympäristöön. UAT tehdään testauksen viimeisessä vaiheessa toiminnallisen-, integraatio- ja järjestelmätestauksen jälkeen. (Hamilton 2023.)

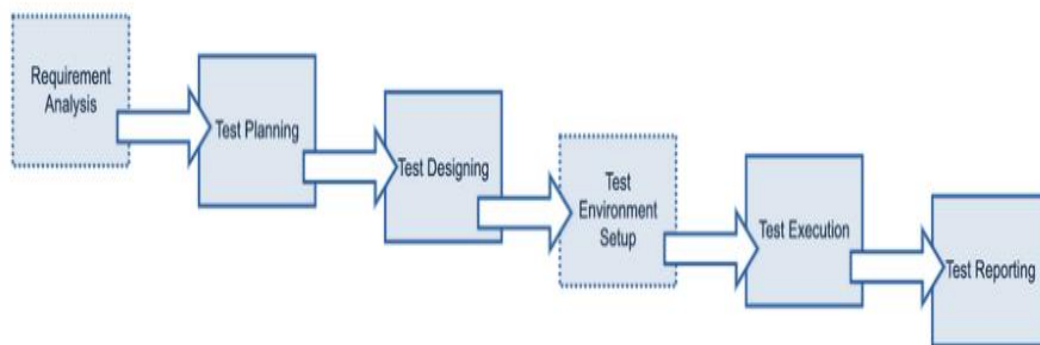
Toinen vaihtoehtoinen ja kevyempi testausmalli voisi olla esimerkiksi B-malli. B-malli sopii kuitenkin paremmin esimerkiksi verkkosivustojen testaamiseen, sillä niiden ominaisuuksia voidaan käydä läpi yksitellen (Mustafa & Khan 2007, s. 184).

### 3.3 Testauksen tukena käytettävä projektinhallintatyökalu Jira

Jira on ketterien kehitystiimien käyttöön suunniteltu työkalu, jonka avulla voidaan jakaa isot kokonaisuudet hallittavimpiin osiin. Näitä osia ovat esimerkiksi user storyt, issues ja taskit. Pienemmät osat voidaan jakaa eri tiimien kesken. Jiran avulla voi seurata projektien etenemistä, käydä keskustelua tiimien työstä sekä priorisoida osakokonaisuuksia. Jirassa on mahdollista myös ylläpitää julkaisulokia sekä raportointia. Sen lisäksi tiettyjä toimenpiteitä, kuten story-tikettien sulkemisia, voidaan automatisoida ilman koodausosaamista. (Atlassian. 2023b)

Jiraan on mahdollista lisätä XRAY-lisäosa, jonka avulla voidaan hallita manuaalisia ja automaattisia testejä Jira-issueina. XRAY-lisäosassa on mahdollista mukauttaa näyttöjä, kenttiä ja työnkulkua haluamakseen ja järjestää testit kansioihin ja testisarjoihin. Xray-lisäosan avulla voi luoda testisuunnitelmia ja testijoukon. Näiden avulla voi seurata suunniteltujen tai ad hoc -testien edistämistä. Lisäksi XRAY-lisäosan avulla voidaan tarkastella testien kattavuutta interaktiivisten kaavioiden avulla ja niitä voi analysoida version, testisuunnitelman ja suoritusympäristön mukaan. (Atlassian. 2023c)

XRAY-lisäosa käyttäessä tyypillinen testauksen elinkaari seuraa sarjaa vaiheita, jotka on kuvattu **Kuvio 3**. Vaiheita ei välttämättä tarvitse suorittaa peräkkäin. (Docheva, T. 2023)



**Kuvio 3.** Testauksen elinkaari XRAY-lisäosaa käyttäessä (Docheva 2023)

### 3.4 Confluence testauksen dokumentinhallintatyökaluna

Confluence on dokumenttien hallintaan suunniteltu työkalu, jonka avulla voi keskittää projekteihin liittyvän informaation yhteen paikkaan. Dokumentteja voi päivittää yhteistyössä verkon välityksellä. Confluencen avulla dokumenttien hallinta pysyy organisoituna ja ajantasaisena sekä helposti saavutettavana eri tiimien välillä. Tiimien ei tarvitse hakea oikeuksia toistensa dokumentteihin kun ne löytyvät Confluencesta. (Atlassian. 2023a)

Confluencessa voi luoda suunnitelmia, jotka sisältävät esimerkiksi tekstiä, taulukoita, kuvia, aikajanoja sekä koodinpätkiä. Projektisuunnitelmat voidaan myös integroida toimintakohteisiin integroimalla halutut projektinhallinta sovellukset Confluenceseen. Confluencessa voidaan myös koordinoita suunnitelmia ja tehdä päätöksiä. Tämä voidaan tehdä siten, että status-päivitykset yhdistetään projekti-suunnitelmiin, jolloin kaikki sidosryhmät pysyvät ajan tasalla. Projektisuunnitelmat ovat käytettävissä myös mobiilisovelluksen kautta. (Atlassian. 2023a)

### 3.5 Testauksen ja projektin johtaminen

Yleisesti ottaen ohjelmistoprojekteissa projektipäällikön vastuulla on varmistaa kaikkien projektiin liittyvien työvaiheiden eteneminen. Pienemmissä projektipäällikön vastuulla saattaa olla myös vastata testauksesta. Testausta ja laadunvarmistusta varten on kuitenkin monesti oma päällikkö, jota kutsutaan joko testaus- tai

laatupäälliköksi. Pienissä hankkeissa koko projektin laadun varmistus on saattaa olla yhden henkilön vastuulla, mutta suuremmissa hankkeissa käytetään yleisesti tarkastustestaajaa. (Kasurinen, J. 2013)

Testipäällikön vastuulla on testaukseen liittyvien jokapäiväisten rutiinien varmistaminen, ja testisuunnitelman ylläpito ja suunnittelu. Lisäksi testipäällikön vastuulla on varmistaa, että testauksen aikana noudatetaan testaus suunnitelmaa ja huolehtia, että testaus etenee. Testipäällikön vastuulla on myös huolehtia testaukseen liittyvistä väli- ja loppuraporteista. (Kasurinen, J. 2013)

Testaaminen on kannattaa jakaa yksittäisiin prosessivaiheisiin tehokkuuden ja vaikuttavuuden varmistamiseksi. Hyvä ja kattava testausprosessi koostuu seuraavista vaiheista: testauksen suunnittelu ja ohjaus, testianalyysi ja suunnittelu, testien toteutus ja suoritus, testien arviointi ja raportointi sekä testitoimintojen viimeistely. Jokaisessa vaiheessa suoritetaan ennakkoon määritellyt tehtävät, kuten resurssien varmistaminen, testitapausten suunnittelu, testien suorittaminen, tulosten arviointi ja kokemuksesta oppiminen. On tärkeää, että testausprosessi on hallittu ja määritelty, jotta testityökalujen käyttö voi tuoda hyötyä. Testityökalut voivat auttaa testaajia ja testauspäällikköä parantamaan testitoimintojen laatua kussakin prosessivaiheessa. (Spillner, A. & muut. 2007)

Erinomaiset projektinjohtajat ovat ihmiskeskeisiä ja osaavat lukea ihmisiä. He ovat taitavia luomaan hyviä suhteita, kuuntelemaan ja osoittamaan empatiaa. Tämä on tärkeää, koska projektien vaikutukset ulottuvat muiden ihmisten työhön. Projektijohtajien tulee olla tietoisia tekojensa seurauksista ja siitä, miten ne vaikuttavat muihin. Tutkimukset ovat osoittaneet, että ihmiskeskeisyys on tärkein kyky projektijohtajille. Niiden, jotka eivät osoita taitoa tällä alueella, on vaikeampaa luoda toimivia työsuhteita ja saada parasta mahdollista tulosta tiimeiltään. (Harrin, E. 2018)

Projektijohtajien menestys perustuu kykyyn havaita yhteistyön haasteet ja olla tietoisia ihmisten käyttäytymisestä ympärillään. Menestyminen edellyttää myös nautintoa tiimityöstä ja kykyä ammentaa energiaa tiimiltä. Projektitiimi on pääasiallinen työympäristö, ja projektijohtajien on otettava johtajuus konfliktien ratkaisemisessa, toimiston politiikan navigoinnissa sekä tiimin jäsenten henkilökohtaisten tilanteiden hienovaraisessa ja tyylikkäässä käsittelyssä. (Harrin, E. 2018)

Projektijohtajan tulee pystyä järjestämään ja valmistelemaan kokoukset menestyksekkäästi. Tämä sisältää esityslistan laatimisen, kokoustilan varaamisen ja osallistujien tiedottamisen. Kokouksen alussa projektinjohtajan tulee johdattaa osallistujat aiheeseen ja selittää kokouksen tarkoitus ja odotukset. Tärkeää on myös ohjata kokouksia tarkasti ja tehokkaasti, käsitellen kaikki aiheet asianmukaisessa yksityiskohtatasossa. Mahdolliset muut asiat käsitellään joko kokouksen aikana, tai tallennetaan myöhempää keskustelua varten. Projektinjohtajan vastuulla on kirjata muistiinpanot ja toimenpiteet, sekä jakaa ne osallistujille. Tärkeää on myös seurata toimenpiteitä ja varmistaa, että kaikki suorittavat heille annetut tehtävät. (Harrin, E. 2018)

### **3.6 Älykäs automaatio**

Älykäs automaatio (IA) on yhdistelmä tekoäly (AI)-, liiketoimintaprosessien hallinta (BPM)- ja robottiprosessiautomaatio (RPA)- teknologioita, jotka mahdollistavat nopean end-to-end liiketoimintaprosessien automatisoinnin. (Automation Anywhere. 2022)

Älykäs automaatio on tekoälyn (AI), robottiprosessiautomaation (RPA) sekä työnkulun ja pilvialustojen keskiössä. Älykkään automaation tarkoitus on automatisoida tietokoneiden end-to-end prosesseja. Älykästä automaatiota kutsutaan myös hyperautomaatioksi tai kognitiiviseksi automaatioksi. Älykäs automaatio toimii yhdessä ihmisten kanssa ihmisten, organisaatioiden ja prosessien yhdistämiseksi, jotta ihmiset voivat keskittyä työhön, joka tuo iloa. (Bornet, P. 2021)

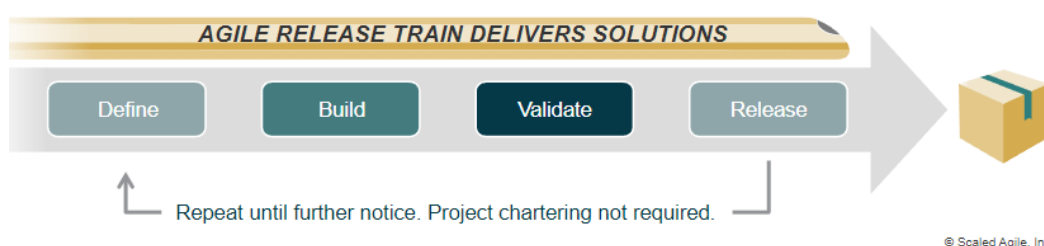
Älykäs automaatio voidaan jakaa neljään osaan, jotka ovat: Visio, Toteutus, Kieli ja Ajattelu ja Oppiminen. Visio eli tietokonenäkö, toimii automatisoidun työvoiman silminä. Toteutus, joka koostuu roboteista, toimii automatisoidun työvoiman "käsinä ja jalkoina". Automatisoidun työvoiman "suuna ja korvina" toimii kieli. Luonnollisen kielen käsittelyn (NLP) avulla tämä IA-yhtälön osa auttaa ohjelmistorobotteja puhumaan ja ymmärtämään keskusteluja kollegoiden ja asiakkaiden kanssa. Ajattelu ja oppiminen eli koneoppiminen toimii automatisoidun työvoiman "aivoina". (Bornet, P 2021)

### 3.7 SAFe®-viitekehys

Scaled Agile Framework® (SAFe®) on Dean Leffingwellin ja Drew Jemilon vuonna 2011 julkaisema viitekehys, jossa on kuvattuna organisaatio- ja työnkulkumalleja ketterän kehityksen toteuttamiseen yrityksissä. Se sisältää jäseneltyä ohjausta rooleista ja vastuista, työn suunnittelusta ja johtamisesta sekä arvoista, joita vaalia. SAFe® edistää yhdenmukaistamista, yhteistyötä ja toimitusta useissa ketterissä tiimeissä. Se muodostuu kolmen ensisijaisen tiedon ympärille: ketterä ohjelmistokehitys, kevyt tuotekehitys ja järjestelmäajattelu. (Piikkila, J. 2023.)

SAFe®:n neljä perusarvoa ovat yhteinen suunta, sisäänrakennettu laatu, läpinäkyvyys, ohjelman suorittaminen ja johtajuus. SAFe®-viitekehityksessä on lisäksi listattu yhdeksän periaatetta, jotka ovat: 1. ota taloudellinen näkemys, 2. hyödynnä systeemijattelua, 3. oleta vaihtelevuutta; säilytä vaihtoehdot, 4. rakenna lisäävästi, nopeita, integroituja oppimissyklejä hyödyntäen, 5. perusta välitavoitteet toimivan järjestelmän objektiiviselle arvioinnille, 6. visualisoi ja rajoita työprosessia (WIP), pienennä eräkokoja ja hallitse jonojen pituuksia, 7. hyödynnä rytmiä, synkronoi suunnittelussa yli kehitysalueiden, 8. vapauta tietotyöläisten sisäinen motivaatio, 9. hajauta päätöksenteko. Nämä ne kuvaavat kulttuuria, jota johtajuuden on vaalittava ja kuinka ihmisten tulisi käyttäytyä siinä kulttuurissa voidakseen käyttää viitekehystä tehokkaasti. (Piikkila, J. 2023.)

Agile Release Train (ART) on ketterien tiimien tiimi, joka kehittää, toimittaa ja käyttää yhtä tai useampaa ratkaisua asteittain arvovirrassa. Junat pyrkivät yhteiseen liiketoiminta- ja teknologiapäämäärään. Jokainen juna, joka koostuu tyypillisesti noin 50-125 henkilöstä, suunnittelee, sitoutuu, kehittää ja ottaa käyttöön yhdessä kehittämänsä ratkaisun. Junat on organisoitu yrityksen merkittävien kehitysarvovirtojen ympärille, ja ne ovat olemassa vain toteuttaakseen tämän arvon lupauksen rakentamalla ja toimittamalla ratkaisuja, jotka hyödyttävät asiakasta. Junat tuottavat arvoa alla olevan **Kuvio 4.** osoittamalla tavalla. (©Scaled Agile. 2023.)



**Kuvio 4.** Pitkäaikainen Agile Release Train (©Scaled Agile 2023.)

Junassa tärkeässä roolissa on Release Train Engineer (RTE), joka auttaa ARTin toteuttamisessa, esteiden poistamisessa, riskien ja riippuvuuden hallinnassa ja jatkuvassa parantamisessa. Tuotehallinta (PM) on vastuussa siitä, mitä kehitetään. PM:t työskentelevät asiakkaiden, tiimien ja tuotteiden omistajien kanssa ymmärtääkseen ja viestiäkseen heidän tarpeitaan ja osallistuakseen ratkaisun validointiin. Järjestelmäarkkitehti (SA) on henkilö tai tiimi, joka määrittelee järjestelmän kokonaisarkkitehtuurin. Yritysomistajat (BO) ovat ARTin keskeisiä sidosryhmiä, ja heillä on lopullinen vastuu junan liiketoiminnan tuloksista. (©Scaled Agile. 2023.)



## 4 KEHITTÄMISPROSESSIN KUVAUS

Fokusryhmä aloitti kehittämishankkeen ideoinnilla ja vaatimusten määrittelyllä. Seuraavaksi siirryttiin suunnitteluvaiheeseen, jossa luotiin raamit testausmenetelmien ja työkalujen, sekä testauksen johtamisen implementointiin älykkään automaation kehitykseen. Toteutusvaiheessa luotiin alustava malli, jota pilotoitiin. Käyttöönottovaiheessa testausmalli jalkautettiin koko Kehitys-osastolle. Testausmallin jatkokehitystä tehdään saatujen kokemusten ja palautteiden perusteella.



**Kuvio 5.** Kehittämishankkeen prosessin kuvaus

### 4.1 Ideointi

Kehittämispöessi aloitettiin syyskuussa 2022 kasaamalla fokusryhmä, johon osallistui asiantuntijoita kehittämishankkeen Kehitys-osastolta, sekä älykkään automaation hyödynsaajista eli loppukäyttäjäorganisaatioista. Ideointi käynnistettiin tunnistamalla tarve paremmin strukturoidusta ja johdetusta testausmallista älykkään automaatin kehityksessä. Tämän toivotaan johtavan laadukkaampaan lopputulokseen loppukäyttäjänäkökulmasta ja laskettujen hyötyjen toteutumiseen.

Ideota kerättiin muistiinpanoina sekä online-palavereissa että live-tapaamisissa käydyissä vapaamuotoisissa keskusteluissa, joita johti kehityshankkeen johtaja.

Ideat käytiin läpi yhdessä yksi idea kerrallaan ja ne arviointiin pohtien intuitiivisesti. Jatkoon valittiin ideat, jotka siinä hetkessä tuntuivat järkeviltä toteuttaa. Ne ideat, joita ei valittu jatkoon kirjattiin kuitenkin muistiinpanoihin myöhempää tarkastelua varten. Ideoinnin pohjalta luotiin kehittämishankkeen prosessin kuvaus ja alustava aikataulu, jotka on kuvattu **Kuvio 5**.

#### **4.2 Vaatimusten määrittely**

Kehittämishankkeen määrittelyvaiheessa varmistettiin ne tavoitteet ja tarpeet, jotka haluttiin saavuttaa kehittämishankkeen aikana. Vaatimusten määrittelyssä peilattiin ideointivaiheen parhaimmaksi arvioituja kehitysideoita ja luotiin niiden perusteella lista tarpeista ja tavoitteista, jotka tukevat Kehitys-osaston testausta ja sen johtamista. Määrittelyvaiheessa varmistettiin myös, että tavoitteet ovat linjassa yrityksen tavoitteiden kanssa.

Vaatimusten määrittelyvaihetta johti hankkeen johtaja ja siihen osallistui Kehitys-osastolta Scrum Mastereita, Release Train Engineer, Development Leadeja sekä muita oleellisia asiantuntijoita, kuten testaajia. Vaatimukset kirjattiin Confluenceen, jossa ne ovat kaikkien sidosryhmien nähtävillä. Tämä koettiin tärkeäksi, jotta kehittämishanke olisi mahdollisimman läpinäkyvä alusta asti.

#### **4.3 Suunnittelu**

Suunnittelu lähti liikkeelle siitä, että fokusryhmä tunnisti tarpeen älykkään automaation testauksen ja sen johtamisen kehittämiseksi. Palautteiden ja aikaisempien kokemusten perusteella arviotiin kehityskohteita ja mitä niihin puuttumalla Kehitys-osastolla voitaisiin saavuttaa. Fokusryhmä havaitsi, että Kehitys-osastolla on pikainen tarve testausmallin muutokselle, jotta testaamisen laatua saadaan parannettua.

Seuraavaksi fokusryhmä asetti tavoitteet, jotka kehittämishankkeen aikana on tarkoitus saavuttaa. Tavoitteet asetettiin siten, että ne ovat linjassa hankkeen tarpeisiin nähden. Tavoitteet suunniteltiin myös siten, että niitä on helppo mitata ja sen

myötä saada reaaliaikaista tietoa hankkeen edistymisestä. Hankkeen edistyminen on avainasemassa Kehitys-osaston tavoitteiden saavuttamisen vauhdittamiseksi.

Kehittämishankkeen yhtenä kulmakivenä oli testauksen ammattimaistaminen ja sitä varten Kehitys-osastolla alettiin kartoittamaan mahdollisuutta ammattitestaajan palkkaamiselle. Uuden henkilön rekrytointi ei ollut mahdollista, joten Kehitys-osastolla päädyttiin hyödyntämään työnkiertoa. Työnkiertoon tulevalla testausressursilla on oltava vankka kokemus asiakasrajapinnassa toimimisesta ja asiakaspalvelutyössä käytettävien järjestelmien tuntemus. Kehitys-osaston sisällä varmistettiin myös se, että tarvittavien sidosryhmien käytössä on kaikki ne työkalut, joita testauksen dokumentointiin ja raportointiin tarvitaan.

Suunnitteluvaiheen viimeisessä osiossa hankkeelle luotiin toimintasuunnitelma, johon määritettiin aikataulut, vastualueet, ja hankkeen toteuttamisen eri vaiheet (**Kuvio 5**). Lisäksi määritettiin, kuinka hanketta hallittaisiin ja kuinka sen etenemisestä raportoidaan tärkeimmille sidosryhmille.

#### **4.4 Toteutus**

Kehittämishankkeen toteutusvaiheessa pilotoitiin kehitettyä testaus- ja johtamismallia ja niitä tukevia työkaluja, sekä ammattitestaajan vaikutuksia testauksen laatuun. Ensiksi testaajat ja kehittäjät sekä muut olennaiset sidosryhmät perehdytettiin testauksen projektointi- ja dokumentinhallintatyökalujen käyttöön ja siihen, kuinka niitä oli tarkoitus jatkossa hyödyntää Kehitys-osastolla.

Työkaluihin tutustumisen aloitettiin esittelemällä Jiran tärkeimmät ominaisuudet. Lisäksi jokaiselle osallistujalle annettiin tehtäväksi luoda erityyppisiä Jira-tikettejä, jotta heillä karttuisi ymmärrys Jiran hierarkioista ja riippuvuuksista. Seuraavaksi siirryttiin dokumentinhallintatyökalu Confluencen perehdytykseen. Confluencen käyttö oli kohderyhmälle entuudestaan tuttua, joten perehdytyksessä keskityttiin pää- ja alitestisuunnitelman esittelyyn sekä siihen, kenen vastuulla niiden ylläpito ja päivitys on.

Varsinaisen testauksen pilottivaiheeseen valittiin pienemmän kokoluokan testauksen, mutta lisäksi aikaa ja ajatuksia panostettiin sen eri vaiheisiin, keräten kokemuksia testaajilta ja kehittäjiltä, sekä muilta sidosryhmiltä. Testaukset aloitettiin kirjaamalla Test Case -tiketit Jiraan ja liittämällä ne Test Planiin. Test Planistä luotiin Test Execution -tyypin tiketit, joita vasten testaus tapahtuu. Lisäksi kaikki luodut Test Case -tiketit liitettiin Jiran Test Repositoryyn, jotta ne ovat helposti löydettävissä ja uudelleen käytettävissä. Lisäksi struktuurin hallittavuuden ja muokattavuuden parantamiseksi käytettiin komponentteja, joiden avulla saadaan luotua haluttuja näkymiä.

Seuraavaksi laadittiin alitestisuunnitelma, jonne kirjattiin yleistä tietoa testaukseen liittyen. Näitä tietoja olivat esimerkiksi linkit olennaisiin asennusohjeisiin, testausympäristöt ja versiot, osallistujalista, aikataulut, testidatan käsittelyyn liittyvät ohjeet, muutoshistoria, asiakastiedon käsittelyyn liittyvät ohjeet, System Under Test -taulukko, kommunikointikanavat ja -tavat ja versiohistoria. Lisäksi alitestisuunnitelmaan lisättiin viittaukset Jiran Test Planiin, ja testaustuloksiin.

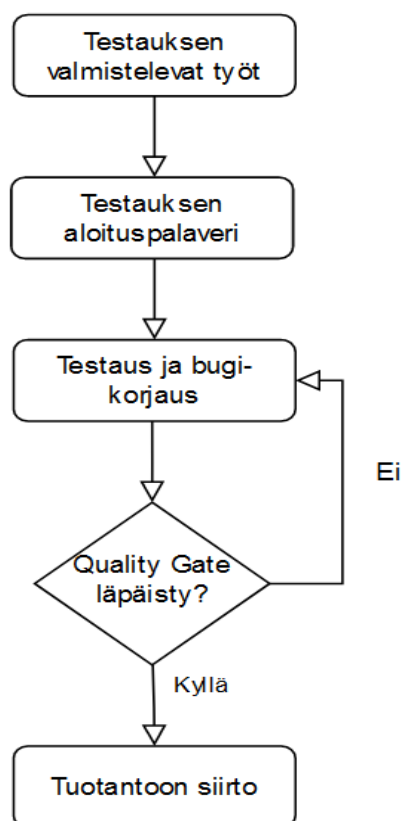
Alkuvalmistelujen jälkeen aloitettiin testaukset aloituspalaverilla, johon kutsuttiin uuden toiminteen luoneet kehittäjät sekä toiminnetta testaamaan valitut henkilöt, sekä muut oleelliset sidosryhmät. Palaverissa käytiin läpi testitapaukset ja niitä verrattiin testattavan toiminteen vaatimusmäärittelyihin ja tämän perusteella tehtiin vielä viimeisiä muokkauksia testitapauksiin. Seuraavaksi käytiin läpi alitestisuunnitelma ja siinä oleva System Under Test -taulukko päivitettiin. Lisäksi palaverissa sovittiin testauksen aikataulu ja kommunikointikanavat. Lopuksi varmistettiin, että bugien kirjausperiaatteet ovat selvät kaikille testaukseen osallistuville osapuolille.

Testauksen käynnistyttyä yhteyttä pidettiin testauksen aikaista kommunikointia varten perustetussa Teams-chatissä, jossa oli testaukseen liittyvät henkilöt. Näin toimimalla sujuvoitettiin testaukseen liittyvän viestinnän tavoittavuutta. Testauksen aikana havaitut bugit kirjattiin bug-tiketteinä Jiraan ja versiohistoriaan kirjattiin, mikä bugi on missäkin versiossa korjattuna ja uudelleen testattavana. Testaus

toteutettiin testiympäristössä tuotantoympäristön sijaan, jonka myötä testidataa oli aiempaa helpommin saatavilla ja tämä sujuvoitti testausta.

Kun testaukselle määritetty aikataulu läheni loppua, pidettiin Quality Gate -palaveri, jossa tarkasteltiin onko kehitetyllä toiminteella valmius tuotantonsiirtoon. Quality Gatessa käytiin läpi toiminteen avoimet bugit, luotettavuus, käytettävyys, käyttöasteraportointi ja se, toimiiko toiminne riittävällä nopeudella. Lisäksi sovittiin tarvittavien ohjeiden jalkauttamisesta käyttäjille. Lopuksi tehtiin ratkaisu, onko toiminne valmis tuotantoon vietäväksi vai ei.

**Kuvio 6.** on kuvattu karkeasti Testausprosessin etenemisen eri vaiheet Kehitys-osastolla.



**Kuvio 6.** Testausprosessin eteneminen Kehitys-osastolla

Koko kehitysprosessin ajan kerättiin huomioita ja parannusehdotuksia laaditun mallin kehittämiseksi.

#### **4.5 Käyttöönotto**

Pilotista saatujen kokemusten perusteella siirryttiin laajempaan käyttöönottovaiheeseen, jossa kehitetty malli implementoitiin osaksi koko Kehitys-osaston kehitystä. Jalkautus tapahtui esittelemällä testausmalli ja -työkalut, sekä käymällä läpi ohjeistus testaukseen tulevien kehitystikettien läpikäynnistä yhdessä testauspäällikön kanssa.

Kehitys-osastolle päätettiin palkata sisäinen testauspäällikkö ja ammattitestaaja, jotta testauksen kehittämiseksi saadaan sitoutuneisuutta ja jatkuvuutta sekä ketteryyttä. Jatkossa testauspäällikkö vastaa kaikesta Kehitys-osaston UAT-testauksesta, varmistaen että testeissä on tarvittavat testausresurssit. Lisäksi testauspäällikön vastuulla on testauksen dokumentointi ja jatkuva kehittäminen. Laajempi käyttöönotto käsittää kaiken Kehitys-osaston erilaisen kehitystoiminnan.

#### **4.6 Jatkokehitys**

Testausmallin jatkokehitys alkoi välittömästi käyttöönottovaiheen jälkeen. Kehitysideoita kerätään saatujen kokemusten perusteella ja niitä kirjataan Confluenceen ”Testauksen kehitys” -osioon. Kehitysideoita voi antaa kuka tahansa, joka keksii hyvän tavan, jolla testausmenetelmiä tai -työkaluja voidaan parantaa. Testauspäällikkö käy läpi kehitysehdotukset yhdessä testausryhmien ja Release Train Engineerin kanssa.

Tähän mennessä esille tuotu potentiaalinen kehitystoimi on esimerkiksi testauksen automatisointi. Automatisoinnilla voitaisiin kattaa tietyt regressiotestaustarpeet, mikä säästäisi testaajien aikaa uusien toiminnallisuuksien testaamiseen. Automatisoinnin avulla saataisiin myös luotua selkeitä ja kattavia raportteja testitulosista ja niiden perusteella voitaisiin helposti tulkita järjestelmän tilaa.

Vaikka nyt jo osa testauksesta saadaan toteutettua testiympäristössä, niin edelleen osa testeistä joudutaan suorittamaan tuotantoympäristössä. Tämä on seikka, joka vaatii vielä jatkokehitystä. Tavoitetilä on, että kaikki Kehitys-osaston testaaminen tapahtuu testiympäristössä.

Jatkon kannalta on tärkeää, että sidosryhmiä rohkaistaan antamaan palautetta testausmallista ja testaukseen käytettävien työkalujen käytöstä, jotta niiden käyttöä voidaan jatkojalostaa.

## 5 KEHITTÄMISTOIMINNAN KUVAUS

Tässä osiossa kuvaan kehittämistoiminnan eri vaiheet, jotka olivat tavoitteen asenta, tarveanalyysin teko, varsinainen kehitystyö, kehittämistyön tuotoksen testaaminen ja testaustulosten arviointi. Lisäksi viimeisessä kappaleessa kuvaan kuinka uutta testausmallia tullaan seuraamaan ja jatkojalostamaan. **Kuvio 7.** Kuvaus kehittämistoiminnan vaiheista vaiheet on kuvattu **Kuvio 7.**



**Kuvio 7.** Kuvaus kehittämistoiminnan vaiheista

### 5.1 Tavoitteen asetanta

Kehittämistoiminnan tavoitteet asetettiin yhdessä Kehitys-osaston johtajan sekä RTE:n kanssa. Tavoite asetettiin siten, että se tukee Kehitys-osaston kehityksen muita tavoitteita. Tavoitteeksi asetettiin, että uusittu tapa johtaa testausta saataisiin implementoitua kehitykseen alkuvuodesta 2023. Tämä tarkoitti sitä, että jatkossa kaikki testaaminen tapahtuu sovittujen mallien mukaisesti, sovittuja työkaluja käyttäen ja niitä johtaa testauspäällikkö.



Testaustyökalujen ja osalta tavoitteeksi asetettiin, että kaikki tiimit tulevat siirtymään uuden mallin mukaisesti käyttämään sovittuja käytäntöjä. Testauksen dokumentoinnin laatua pyritään myös lisäämään siirtämällä kaikki testaukseen liittyvä dokumentointi Jiraan tai Confluenceen tapauskohtaisesti. Näillä toimenpiteillä pyritään tekemään testauksesta strukturoidumpaa ja lisäämään läpinäkyvyyttä.

Testausympäristön osalta tavoitteeksi asetettiin, että jatkossa testaus suoritettaisiin testiympäristössä tuotantoympäristön sijaan. Tämä helpottaa testauksen suorittamista ja vähentää riskiä tuotannon virhetilanteille. Lisäksi testidata on helpommin manipuloitavissa halutunlaiseksi.

Lisäksi tavoitteeksi asetettiin, että jatkossa Kehitys-osastolla olisi vähintään yksi ammattitestaaja ja sen lisäksi yksi testauspäällikkö. Tämä tavoite tukee testauksen laadun parantamista ja on kustannustehokkaampi tapa kuin ulkoisen konsultin palkkaaminen johtamaan ja kehittämään testausta tai tekemään varsinaista testaustyötä.

Yllä olevien tavoitteiden todettiin olevan saavutettavissa ja niihin sitouduttiin, sillä ne tukisivat Kehitys-osaston tavoitteita.

## **5.2 Tarveanalyysi**

Tarveanalyysi tehtiin fokusryhmän kanssa pohtimalla vastauksia siihen, mitä jatkossa testauksessa halutaan saavuttaa ja kuinka testausta halutaan johdettavan. Pohdinnassa oli myös mitkä teknologiset seikat tukevat testausta ja sen johtamista, sekä mitkä muut tavat toimia koetaan tärkeäksi kehitystyölle.

Kehitys-osaston testauksen ja testauksen johtamisen nykytilaa ja sen vaikutuksia kehitettäviin järjestelmiin pohdittiin. Nykytilannetta pohtiessa kiinnitettiin huomiota siihen, miten nykyisin testausta on johdettu ja mitkä työkaluja sen raportointiin tai dokumentointiin on käytetty. Lisäksi pohdittiin myös sitä, millä kokoonpanolla testausta on tehty.

### 5.3 Pilotointi ja arviointi

Uutta testausta ja sen johtamismallia lähdettiin testaamaan pilotoimalla testaukseen, sen johtamiseen, raportointiin ja dokumentointiin liittyviä työkaluja ja toimintatapoja käytännön testauksen yhteydessä. Pilottiin osallistui testauspäällikkö, testausryhmä, sekä Product Owner. Pilotin aikana painotettiin sovittujen toimintatapojen noudattamista ja työkalujen oikeaoppista hyödyntämistä.

Pilotin aikana kerättiin kokemuksia pilottiin osallistuneilta henkilöiltä niin testauksen johtamismallista, kuin työkalujen hyödyntämisestä testauksen tukena. Pilotin aikana saatiin paljon kokemuksia uudesta testausmallista. Pilotin päätyttyä kerättyjä havaintoja, palautteita ja kokemuksia arvioitiin yhdessä testauspäällikön, osaston johtajan ja Release Train Engineerin kanssa.

### 5.4 Seuranta ja jatkokehitys

Kehittämistoiminnan aikana luodun testausmallin implementointi koko Kehitysosaston kehitykseen on tehty ja siitä saatuja tuloksia seurataan mittaroimalla esimerkiksi testaukseen käytettävää aikaa ja testauksen aikana löydettyjen bugien määrää. Raportoinnilla nähdään jatkuvasti testauksien edistyminen ja kohdattuihin ongelmiin pystytään puuttumaan jo aikaisessa vaiheessa. Seuranta on testauspäällikön vastuulla, joka raportoi Release Train Engineerille ja osaston johtajalle.

Seurannan tuloksien perusteella pyritään jatkokehittämään testausmallia siten, että testausputki on ajallisesti lyhyt, mutta silti tehokas. Tämän avulla kehitettyjä sovelluksia saadaan nopeasti tuotantoon, jolloin ne alkavat tuottaa lisäarvoa tilaajille. Nopea testaaminen ei kuitenkaan tarkoita sitä, että testausta tehtäisiin huolimattomasti tai ohjelmakoodissa olevia virheitä piiloteltaisiin. Sen sijaan bugit raportoidaan nopeasti ja tiketit ohjataan oikealle kehitysryhmälle ja siirrytään seuraavaan testitapaukseen. Näin saadaan testausketjusta tehokas.

Jatkokehittäminen on testauspäällikön ja Release Train Engineerin vastuulla ja heitä tukee osaston johtaja. Jatkokehitysideoita kerätään ammattitestaajilta, testiryhmiltä, kehittäjiltä ja muilta sidosryhmiltä. Näiden sidosryhmien osallistaminen testauksen kehittämiseen takaa sen, että he sitoutuvat tuleviin kehitystoimiin ja sen myötä parempaan testauksen laatuun myös jatkossa.

## 6 KEHITTÄMISTOIMINNAN TULOKSEN KUVAUS

Kehittämistoiminnan aikana saadut tulokset voidaan jakaa karkeasti kahteen kategoriaan: hyötyihin ja haasteisiin. Kehittämistoiminnan suurin ja tärkein teema oli laadukkaampaa testausta paremmalla johtamisella. Tätä teemaa tukemaan valittiin työkalut, jotka auttavat testauksessa ja sen raportoinnissa, sekä dokumentoinnissa. Kehittämistoiminnan aikana törmättiin myös haasteisiin, jotka olivat sekä teknisiä että inhimillisiä.

### 6.1 Hyödyt

Kehittämishankkeen aikana käyttöön otetut testauksen tukena käytettävien projektointityökalujen laajamittaisempi hyödyntäminen auttoi testauksen läpinäkyvyyden lisäämiseen yli tiimi- ja osastorajojen. Tämän avulla välttyttiin päällekkäiseltä testaamiselta ja mahdollisilta virhetilanteilta. Myös ylimääräinen statusten selvittely väheni erinäisissä viestintäkanavissa, kun tiedot olivat helposti löydettävissä sovitusta paikoista.

Strukturoidumpi testausmalli toi selkeyttä testaamisen järjestämiseen ja hallintaan. Tämä helpotti myös uusien testaajien mukaantuloa, kun testaukselle on selkeä kaava jota noudattaa. Testaukseen liittyvästä alitestisuunnitelmasta löytyy tärkeimmät ohjeet ja testitapaukset ovat järjestetty hierarkioihin, jolloin niitä on helppo tutkia. Testauksen aikainen viestintätapa sovitaan aina yhdessä testaustiiimin ja muiden sidosryhmien kanssa, jolloin kaikki sitoutuvat siihen.

Jiran raportointityökalun avulla testauksen tulosten raportoinnin laatu parani ja raportit ovat saatavilla kaikille tahoille, jotka saattavat olla kiinnostuneita testauksen edistymisestä. Raportoinnin selkeys on tärkeää varsinkin testauspäällikön näkökulmasta, joka vastaa testauksen johtamisesta ja raportoi testauksesta edelleen muille sidosryhmille. Raporteilta saa kätevästi tiedon paljonko testaukseen on käytetty aikaa ja paljonko on meneillään päällekkäistä testaustekemistä ja mikä testauksen status on.

Kehittämishankkeen aikana saatiin luotua tiiviimpää yhteistyötä testaajien ja kehitystiimien ja muiden sidosryhmien välillä, jonka avulla saavutettiin laadukkaammat ja monipuolisemmat testitapaukset, jotka edesauttoivat laadukkaamman ohjelmakoodin saavuttamiseen tuotannossa. Tämän myötä ohjelmalle lasketut hyödyt toteutuivat aiempaa paremmin. Tiivis yhteistyö auttoi myös vauhdittamaan reagointia bugien korjaukseen, jolloin ohjelmakoodi saatiin nopeammin uudelleen testaukseen.

Kehittämishankkeen alussa todettiin Kehitys-osastolla olevan tarve ammattitestaajalle, joka toisi joustavuutta ja laatua testaamiseen ja samalle kehittäisi koko yrityksen testausta. Kehittämishankkeen tuloksena Kehitys-osastolle saatiin palkattua yksi ammattitestaaja ja sen tuloksena koko osaston testauksen taso ja tehokkuus on noussut.

Tavoitteena oli myös sisäisen testauspäällikön palkkaaminen, joka on edullisempi vaihtoehto kuin ulkoinen konsultti. Sen lisäksi sen toivottiin tuovan sitoutuneisuutta testauksen johtamiseen ja sen kehittämiseen. Kehittämishankkeen aikana saatiinkin palkattua Kehitys-osaston sisäinen testauspäällikkö, joka vastaa jatkossa testauksen johtamisesta, sen kehittämisestä, laadun valvonnasta. Sen lisäksi testauspäällikkö on tukena testitapausten suunnittelussa. Tämän avulla saavutettiin sitoutuneisuutta ja merkittäviä rahallisia säästöjä Kehitys-osastolla.

## **6.2 Haasteet**

Laadukkaan ja turvallisen testauksen takaamiseksi kehittämishankkeen yhdeksi tavoitteeksi asetettiin, että kaikki testaaminen tapahtuu jatkossa testausympäristössä. Resurssihaasteiden takia edelleen noin neljäsosa testauksesta joudutaan suorittamaan tuotantoympäristössä. Testausympäristöhaasteeseen on kuitenkin tulossa muutos, sillä kehitysympäristön muutoksen vuoksi myös viimeinen tiimi tulee saamaan käyttöönsä mahdollisuuden testata tuotantoympäristössä.

Tietyt testitapaukset vaativat ulkoisen kumppanin suorittamisen. Tämä koettiin haasteelliseksi, koska ulkoinen kumppani oli vaikeasti saavutettavissa, vaikka se nimellisesti olikin sitoutunut testaamaan. Tämä haaste kierrettiin asentamalla sisäiselle testausryhmälle Citrix-ympäristö, jonka avulla pystyttiin simuloimaan ulkoisen kumppanin testausta.

Eräänä haasteena tunnistettiin myös se, minkä kokoinen testaus jatkossa tulee testauspäällikön vetämäksi. Tätä haastetta pyrittiin ratkomaan työpajassa, jossa käytiin läpi minkälaisia ohjelmakoodia ollaan viemässä tuotantoon ja mitkä sen mahdolliset vaikutukset ovat loppukäyttäjille. Työpajassa sovittiin, että kaikista pienimmät ohjelmat, joilla ei arvioida olevan mahdollisuutta rikkoa tuotannossa olemassa olevaa ohjelmakoodia, eivät tule testauspäällikön vedettäväksi. Tällaisia pieniä ohjelmia ovat esimerkiksi täysin muista ohjelmistoroboteista irralliset robotit.

## **7 JOHTOPÄÄTÖKSET JA ARVIOINTI**

Kehittämishankkeen tuloksia arvioitaessa kiinnitettiin huomiota kolmeen seikkaan: kehittämishankkeen tuloksen luotettavuuteen, käyttökelpoisuuteen ja siirrettävyyteen. Lisäksi kehittämishankkeen lopputuloksesta ja sen vaikuttavuudesta Kehitys-osaston tavoitteisiin tehtiin johtopäätös.

### **7.1 Luotettavuus**

Kehittämishankkeen tulosten luotettavuus perustuu määrälliseen tekijään. Kehitetyn testausmallin jalkautus tehtiin kehittämishankkeen aikana koko Kehitys-osastolle ja palautteita kerättiin monissa eri kanavissa. Palautetta saatiin kiitettävästi, mikä tukee kehittämishankkeen lopputuloksen luotettavuutta.

### **7.2 Käyttökelpoisuus**

Kehittämishankkeen aikana Kehitys-osastolla käytetyille dokumentointi- ja projektihallintatyökaluille on koko maaorganisaation tuki, joten se takaa että ne ovat myös jatkossa IA-kehityksen käytettävissä. Kehittämishankkeen aikana luodut testausdokumentaatiota tukeva päättestisuunnitelma löytyy Kehitys-osaston dokumenteista ja alitestisuunnitelmasta on luotu mallipohja, jota täydentämällä testauksen johtaminen pysyy strukturoituna.

Testauspäällikön ja ammattitestaajan hyödyntäminen testauksessa on kannattavaa, sillä se tuo ammattimaisuutta ja laajaa ymmärrystä testauksen eri vaiheisiin. Näin testauksesta saadaan laadukasta. Lisäksi välttämällä ulkoisen konsultin palkkaamista voidaan Kehitys-osastolla säästää merkittäviä määriä rahaa. Jos Kehitys-osaston kehittämistahti kasvaa entisestään, voisi testauksen sujuvuuden varmistamiseksi olla kannattavaa palkata lisää testauksen ammattilaisia.

Kehittämishankkeen lopputulos on käyttökelpoinen niin tekniseltä, taloudelliselta kuin inhimilliseltäkin näkökannalta.

### 7.3 Siirrettävyys

Kehittämishankkeen lopputulos on sellaisenaan suoraan kopioitavissa ja siirrettävissä maaorganisaation muihin kehitysjuniin. Testauspäällikön ja Release Train Enginöörin olisi hyvä käydä esittelemässä kehittämishankkeen aikana luotua testausmallia muille kehitysjunille ja kertoa, mitä kehittämishankkeen aikana saavutettiin sekä mihin haasteisiin törmättiin ja miten niistä päästiin yli.

Suurin yksittäinen haaste siirrettävyydelle on ihmisten asenteissa. Jotkut saattavat olla kiintyneitä vanhoihin malleihin, jolloin uuden mallin sisäänajo vaatii aikaa ja kärsivällisyyttä. Tätä varten kannattaisi koota erillinen fokusryhmä ja oma kehittämishankkeensa.

### 7.4 Johtopäätökset

Kehittämishanke kokonaisuutena oli onnistunut. Hankkeen aikana Kehitys-osaston testauksessa saavutettiin parempi läpinäkyvyys yli tiimi- ja osastorajojen. Lisäksi testauksen tukena käytettävien projektinjohto, raportointi ja dokumentointityökalujen avulla testauksesta saatiin strukturoitua ja helposti johdettavaa. Testaus saatiin pääosin siirrettyä testiympäristöön. Jatkoon kannalta on tärkeää, että kaikki testaaminen tapahtuu testiympäristöissä, jotta testaaminen on mahdollisimman sujuvaa ja riskitöntä.

Tiiviimpi yhteistyö testaajien, kehittäjien ja muiden sidosryhmien välillä auttoi laadukkaampien testitapausten suunnittelussa ja nopeutti bugi-korjauksia. Lisäksi Kehitys-osaston testauksen ammattimaistamisessa otettiin suuria askelia kun osastolle saatiin palkattua ammattitestaaja ja testauspäällikkö. Näiden edellä mainittujen seikkojen takia kehittämishankkeen aikana saavutettiin parempi testauksen laatu. Lisäksi tulee taloudellisia säästöjä, kun ulkoisesta konsultista voitiin luopua.

Tarveanalyysivaiheessa listatuista testauksen ja sen johtamisen **Taulukko 1**. Testauksen ja sen johtamisen tarpeet Kehitys-osastolla listatuista välttämättömistä



tarpeista täysin saavutettuja asioita olivat läpinäkyvä testaus tiimi- ja osastorajojen yli, strukturoidun testausmallin käyttö testauksen läpinäkyvyyden tukemiseksi ja organisoitu, dokumentoitu ja raportoitu testaus, sekä ammattitestaaja, jolla on tuntemusta tärkeimmistä sidosryhmistä ja järjestelmistä. Lisäksi välttämättömistä tarpeista osittain saavutettiin myös, se että testaus suoritetaan pääasiassa testiympäristössä ja tiivis yhteistyö testaajien, kehitystiimien ja sidosryhmien välillä.

Tarveanalyysivaiheessa listatuista testauksen ja sen johtamisen **Taulukko 1.** Testauksen ja sen johtamisen tarpeet Kehitys-osastolla listatuista hyödyllisistä tarpeista täysin saavutettuja asioita oli se, että testauspäällikkö tukee testitapausten suunnittelua, selkeä vastuunjako testitapausten suunnittelussa sekä laajamittainen projektinhallintatyökalujen hyödyntäminen testauksen tukena. Lisäksi saavutettiin laadukas raportointi. Sen lisäksi viestintä sidosryhmien suuntaan tehostui.

Kehittämishankkeen aikana rajattiin, että testauspäällikkö vastaa jatkossa testauksen johtamisesta, sen kehittämisestä, laadun valvonnasta ja oltavana tukena testitapausten suunnittelussa. Lisäksi testauspäällikön vastuulla on varmistaa, että testaus tapahtuu testausympäristöissä ja että kaikilla tarvittavilla resursseilla on tarvittavat järjestelmät testausta varten. Nämä seikat tukevat sitä, että testaus on hyvin johdettua ja sujuvaa.

Kehittämishankkeen lopputulos on käyttökelpoinen Kehitys-osaston tarpeisiin nähden ja lisäksi monistettavissa toisiinkin kehitysjuniin ja sitä myötä koko yrityksen kehitykseen.

## LÄHTEET

- Atlassian. 2023a. Confluence. Viitattu 19.5.2023. <https://www.atlassian.com/software/confluence>
- Atlassian. 2023b. Jira. Viitattu 19.5.2023. <https://www.atlassian.com/software/jira>
- Atlassian. 2023c. Xray. Viitattu 19.5.2023. <https://marketplace.atlassian.com/apps/1211769/xray-test-management-for-jira?tab=overview&hosting=cloud>
- Automation Anywhere. 2022. Intelligent Automation. Viitattu 6.5.2022. <https://www.automationanywhere.com/rpa/intelligent-automation>
- Bornet, P. 2021. What You Need to Know About the Age of Intelligent Automation. Viitattu 2.6.2023. <https://www.uipath.com/blog/automation/what-you-need-to-know-about-intelligent-automation>
- Docheva, T. 2023. The Comprehensive Guide to Xray for Jira: The Leading Test Management Tool for 2023 Viitattu 2.6.2023. <https://www.idalko.com/xray-for-jira/>
- Hamilton, T. 2023. What is System Testing? Types with Example. Viitattu 2.6.2023. <https://www.guru99.com/user-acceptance-testing.html>
- Hamilton, T. 2023. Unit Testing Tutorial – What is, Types & Test Example. Viitattu 2.6.2023. <https://www.guru99.com/unit-testing-guide.html>
- Hamilton, T. 2023. What is System Integration Testing (SIT) with Example. Viitattu 2.6.2023. <https://www.guru99.com/system-integration-testing.html>
- Hamilton, T. 2023. What is User Acceptance Testing (UAT)? with Examples. Viitattu 2.6.2023. <https://www.guru99.com/user-acceptance-testing.html>
- Harrin, E. 2018. Project manager. Swindon: BCS, The Chartered Institute for IT.
- Kasurinen, J. 2013. Ohjelmistotestauksen käsikirja. Jyväskylä: Docendo.
- Mustafa, K & Khan, R. 2007. Software Testing Concepts and Practices. Oxford: Alpha Science International.
- Piikkila, J. 2023. What is SAgile? Viitattu 5.6.2023. <https://www.atlassian.com/agile/agile-at-scale/what-is-safe>

Salonen, K., Eloranta, S., Hautala T. & Kinos S. 2017. Kehittämistoiminta ja kehittämisen menetelmiä ammatillisessa korkeakoulutuksessa. Turun ammattikorkeakoulu. Tampere: Suomen Ylipistopaino Oy.

©Scaled Agile. Agile Release Train. 2023. Viitattu 5.6.2023 <https://scaledagileframework.com/agile-release-train/>

Spillner, A., Rossner, T., Winter, M. & Linz, T. 2007. Software Testing Practice: Test Management. Santa Barbara: Rocky Nook Inc.