

Niko Nykänen

**OHJELMISTOTESTAAJAN OPPIMISPÄIVÄKIRJA – LAADUN-
VARMISTUS SYMBIO CONNECTED CAR -PROJEKTISSA**

**OHJELMISTOTESTAAJAN OPPIMISPÄIVÄKIRJA – LAADUN-
VARMISTUS SYMBIO CONNECTED CAR -PROJEKTISSA**

Niko Nykänen
Opinnäytetyö
Syksy 2019
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu

Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä(t): Niko Nykänen

Opinnäytetyön nimi suomeksi: Ohjelmistotestaajan oppimispäiväkirja – Laadunvarmistus Symbio Connected Car -projektissa

Opinnäytetyön nimi englanniksi: A software tester's learning diary – Quality assurance in Symbio Connected Car project

Työn ohjaaja(t): Pekka Alaluukas

Työn valmistumislukukausi ja -vuosi: Syksy 2019

Sivumäärä: 41

Oppimispäiväkirja kirjoitettiin Symbio Finland Oy:n Connected Car -asiakasprojekteissa tehdyn työn perusteella. Symbio Connected Car on kirjasto, jonka avulla auton päätelaitteeseen voidaan yhdistää iOS- tai Android-älylaite Apple CarPlaytä, Googlen Android Autoa tai Baidun CarLifea varten. Edellä mainitut sovellukset mahdollistavat esimerkiksi navigaattorin käytön, hands-free-puheluiden soittamisen ja vastaanottamisen sekä musiikin kuuntelun autoa ajaessa.

Työn tavoitteena oli laadunvarmistuksen teorian ja käytännön parempi ymmärtäminen sekä Symbion laadunvarmistusprosessien kehittäminen havaintojen pohjalta. Tätä varten opinnäytetyössä on tarkasteltu testaamisen ja laadunvarmistuksen teoriaa ja käytettyjä testiprosesseja tarkasteltu sen valossa. Tätä varten opinnäytetyöhön kerättiin havaintoja ja oppimiskokemuksia tehdystä työstä viikoittain.

Opinnäytetyötä varten tehty työ tehtiin Symbion asiakasprojekteissa. Työ koostui ensisijaisesti ohjelmistojen ja laitteiden testaamisesta yhteistyössä Symbion asiakkaiden kanssa. Työhön kuului muun muassa laitteen käytettävyyden, äänenlaadun ja vakauden testaamista sekä mahdollisten ongelmakohtien etsimistä loppukäyttäjän yleisimmistä käyttötilanteista. Koska opinnäytetyö tehtiin kaupallisessa projektissa, ei laitteiden, käytettyjen ohjelmistojen tai hyödynnetyn dokumentaation teknistä sisältöä ole työn vahvan vaitiolovelvollisuuden vuoksi esitelty. Sen sijaan tehtyä työtä on esitelty yleisellä tasolla.

Työn aikana havaittiin, että Symbion laadunvarmistusprosessi noudattelee pääpiirteissään teoriassa esitettyjä suosituksia ja työstä on havaittavissa teoriassa esitellyt testauksen vaiheet. Vaikka merkittäviä uusia oppimiskokemuksia ei tehdyn työn aikana tullut, löydettiin Symbion laadunvarmistusprosessista yksittäisiä kehityskohtia kommunikaation ja projektien organisoinnin suhteen.

Asiasanat: laadunvarmistus, ohjelmistotestaus, järjestelmättestaus, testaus

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Option of Software Development

Author(s): Niko Nykänen

Title of thesis: A software tester's learning diary – Quality assurance in Symbio Connected Car project

Supervisor(s): Pekka Alaluukas

Term and year when the thesis was submitted: Autumn 2019

Pages: 41

This thesis was conducted in Symbio Finland Ltd's Symbio Connected Car customer projects. Symbio Connected Car is a software library that provides a cross-platform compatibility layer for Google's Android Auto, Apple CarPlay and Baidu CarLife. By using these applications, a user can use their smart device for, for example, navigation, music playback or for hands-free phone calls while driving a car.

The goal of this thesis was to acquire a better understanding of the theory and practice of quality assurance. To this end, a learning diary was written on a weekly basis. The learning diary consists of descriptions of the conducted work and personal experiences and observations.

The thesis was conducted in Symbio's customer projects. Most of the work consisted of testing the customers' hardware and software. The work included, for example, testing the user experience, audio quality and stability. In addition, possible software and hardware issues were teased out by exploratory testing, by using the product as the end user would, by replicating common end-user use cases. Since the work was conducted in commercial projects that include a strong non-disclosure agreement, the products, their software or their specifications are not described or laid out. Instead, the work has been described on a general level.

During the writing of this thesis, it was noticed that Symbio's quality assurance process was comparable to that outlined in the theory. Although no new major learning experiences were had, some points of improvement in Symbio's quality assurance project were discovered, namely in project communication and in organizing the quality assurance in customer projects.

Keywords: quality assurance, software testing, system testing, testing

SISÄLLYS

1 JOHDANTO	6
2 OHJELMISTOTESTAUKSEN PÄÄPERIAATTEET	7
2.1 Laadunvarmistus ohjelmistoprojekteissa – Mistä laatu koostuu?	8
2.2 Ohjelmistotestauksen organisointi ja projektin elinkaari	11
3 KÄYTETYT VÄLINEET, KOMMUNIKOINTIKANAVAT JA PROSESSIT	16
3.1 Käytetyt välineet	16
3.2 Kommunikointikanavat	17
3.3 Laadunvarmistusprosessin keskeisimmät vaiheet	17
4 OHJELMISTOTESTAAJAN OPPIMISPÄIVÄKIRJA	19
4.1 Viikko 41	19
4.2 Viikko 42	21
4.3 Viikko 43	24
4.4 Viikko 44	26
4.5 Viikko 45	26
4.6 Viikko 46	28
4.7 Viikko 47	30
4.8 Viikko 48	32
4.9 Viikko 49	33
5 YHTEENVETO	36

1 JOHDANTO

Tämä opinnäytetyö on tehty Oulun ammattikorkeakoulun tietotekniikan koulutusohjelmassa. Työ on toteutettu yhteistyössä Symbio Finland Oy:n kanssa. Symbio Finland Oy on monikansallinen yritys, joka tuottaa digitaalisia palveluita Pohjois- ja Etelä-Amerikassa, Euroopassa ja Kiinassa (1). Esimerkkejä Symbion painopisteistä ovat älykkäät laitteet muun muassa autoteollisuuden tarpeisiin sekä pilvipalvelut (2). Opinnäytetyö on jatkumoa Symbiolle suoritetuille harjoittelulle sekä yrityslähtöisille tuotekehitysprojekteille.

Symbion kehittämä Symbio Connected Car (SCC) -kirjasto mahdollistaa älylaitteen liittämisen auton päätelaitteeseen. SCC-kirjaston avulla käyttäjä voi hyödyntää Applen CarPlaytä tai Googlen Android Autoa esimerkiksi navigointiin, hands-free-puheluiden soittamiseen ja vastaanottamiseen sekä musiikin kuunteluun. Päätelaitteessa voi lisäksi olla useita natiivitoiminnallisuuksia, kuten esimerkiksi radio, musiikkisoitin, Bluetooth-yhteys sekä valmistajan oma navigointi- tai puheentunnistusjärjestelmä. Jotta päätelaitetta voidaan myydä CarPlay- tai Android Auto -yhteensopivana, on sen täytettävä Applen ja Googlen asettamat laatuvaatimukset. Tätä varten Symbio tarjoaa testaus- ja laadunvarmistuspalveluita SCC-kirjastoa ostaville asiakkaille.

Opinnäytetyö on kirjoitettu oppimispäiväkirjamuodossa Symbion asiakasprojekteissa tehdystä työstä. Oppimispäiväkirjassa kuvaillaan työn sisältöä ja peilattu sitä alan kirjallisuudesta hankittuun testauksen ja laadunvarmistuksen teoriaan. Opinnäytetyön tarkoituksena on ymmärtää testaus- ja laadunvarmistusprosessia paremmin sekä pyrkiä etsimään kehityskohteita Symbion laadunvarmistusprosessista peilaamalla tehtyjä havaintoja ja kokemuksia alan teoriaan. Koska kyseessä on kaupallinen projekti, ei käytettyjä laitteita, ohjelmistoja tai dokumentaatiota työssä esitellä. Sen sijaan työn sisältöä kuvaillaan yleisellä tasolla. Käytännössä tämä tarkoittaa asettumista loppukäyttäjän asemaan: kuinka loppukäyttäjä voi tuotteen ostaessaan odottaa laitteen toimivan, kuinka hän laitetta käyttää sekä millaisia ongelmia hän voi tuotetta käyttäessään kohdata.

2 OHJELMISTOTESTAUKSEN PÄÄPERIAATTEET

Testaamisen merkitys sovelluskehitysprojekteissa on kasvanut huomattavasti 2000-luvun alun jälkeen. Siinä missä aiemmin testaajat nähtiin sovelluskehitysprojektin ulkopuolisina toimijoina, on heitä viime vuosikymmeninä otettu yhä tiiviimmäksi osaksi varsinaista projektityöskentelyä sen eri vaiheissa. Heidät on alettu nähdä aiempaa tärkeämpänä projektin onnistumisen kannalta. Sovelluskehityksen lisäksi testaamisen merkitys on kasvanut myös esimerkiksi sulautettujen järjestelmien ja laitteistojen kehitysprojekteissa. (3, s. xvii.)

Milin ja Fairouzin (4, s. 5, 6, 7, 9) mukaan tietotekniikan tuotteiden laatuvaatimukset ovat kiristyneet 2000-luvulla. Heidän mukaansa tämä johtuu laitteiden monimutkaistumisesta. He nostavat muun muassa esiin 1960-luvulla julkaistun IBM OS360:n, joka koostui noin miljoonasta koodirivistä, sekä vuonna 2003 julkaistun Windows Server 2003:n, jonka ohjelmistokoodi oli noin 50 miljoonaa riviä. He toteavat, että ohjelmien monimutkaistuessa niiden alttius virheille kasvaa. Tämä nostaa projektin budjettia, sillä virheiden korjaukseen on käytettävä enemmän aikaa. Mili ja Fairouz huomauttavatkin, että erilaiset joustavat sovelluskehitysmenetelmät, kuten Agile, auttavat hillitsemään ohjelmistoprojektien kasvavaa budjettia. Myös Spillner, Linz ja Schaefer (5, kpl 1) toteavat, että ohjelmistoilla hallitaan 2000-luvulla monenlaisia laitteita esimerkiksi autojen jarruista kokonaisuun tehtäisiin. Heidän mukaansa ohjelmistojen laadusta on tullut kilpailuvaltti, minkä vuoksi ohjelmistotekniikassa on alettu kiinnittää aiempaa enemmän huomiota tuotekehitysprosessiin, laadunvalvontaan ja tuotteiden testaukseen.

Ohjelmistotekniikka ja etenkin siihen liittyvä testaus poikkeaa merkittävästi monista muista tekniikan aloista. Eräs esimerkki tästä on yhtenäisen ohjelmistoarkkitehtuurin puute – erilaiset ohjelmistot toteutetaan hyvin erilaisilla tekniikoilla, kuten ohjelmointikielillä, eikä ohjelmistojen osien kierrättäminen uusiin ohjelmiin ole aina helppoa. Tässä suhteessa ohjelmistotekniikka poikkeaa esimerkiksi autoteollisuudesta, jossa autojen perusarkkitehtuuri on säilynyt vuosikymmeniä hyvin samankaltaisena myös eri valmistajien välillä (4, s. 10). Toinen huomattava poikkeavuus liittyy ohjelmistojen laadun arvioinnin objektiivisten ja kvantitatiivisten ar-

viointimenetelmien puutteeseen (3, s. 11; 4, s. 11). Siinä missä esimerkiksi silanrakennuksessa voidaan luottaa luku- ja laskelmien tuottamaan numerodataan, ei ohjelmistojen toteuttamat ominaisuudet ole helposti kvantifioitavissa. Esimerkiksi kaiuttimen tuottaman äänenlaadun arviointi riippuu jossain määrin testaja- jasta.

Kaksi keskeisintä tekijää testauksessa ja laadunvarmistuksessa ovat tuotteen vaatimukset ja ohjelmistovirheet. Vaatimuksilla tarkoitetaan lyhyesti sitä, kuinka tuote suoriutuu sille tarkoitetuista tehtävistä. Vaikka tuote olisi oikein määritelty, voi se sisältää ohjelmistovirheitä. Ohjelmistovirheiden vakavuus voi vaihdella käyttöä häiritsevästä käytön estymiseen. Jotkin viat (*problem*) voivat aiheutua esimerkiksi tuotteen sisältämien fyysisten osien kulumisesta, kun taas jotkin viat (*fault*) aiheutuvat virheestä tuotteen laitteiston tai ohjelmiston toteutuksessa. Eri- tyisen ongelmallisia ovat virheet, jotka ilmenevät vain pitkäaikaisessa käytössä. Esimerkki tällaisesta virheestä on tuotteen säilömän datan turmeltuminen. Tällai- sia virheitä on hankala huomata tuotteen kehityksen ja testauksen aikana. Pa- himmillaan ne ilmenevät vasta loppukäyttäjän käytettyä tuotetta kauan aikaa. Li- säksi on huomioitava ero sovelluskehittäjien suorittaman virheenkorjauksen (*de- bugging*) ja varsinaisen testaamisen välillä. Siinä missä sovelluskehittäjät suorit- tavat ohjelmistokehityksen aikana virheenkorjausta löytääkseen virheitä ohjel- mistokoodista, on varsinaisen testaamisen tarkoitus varmistua siitä, että laite toi- mii oikein suhteessa tuotteen vaatimuksiin ja määrittäisiin. (5, kpl 2.1.1, kpl 2.1.2; 6, luku What Testing Is And What Testing Does.)

2.1 Laadunvarmistus ohjelmistoprojekteissa – Mistä laatu koostuu?

Mili ja Fairouz (4, s. 15–17) erittelevät viisi eri ominaisuutta, joiden perusteella ohjelmistotekniikan tuotteen laatua voidaan arvioida. Nämä ovat tilastolliset (*sta- tistical*), toiminnalliset (*functional*), käytännölliset (*usability*), liiketoiminnalliset (*business*) sekä rakenteelliset (*structural*) ominaisuudet. Näistä tilastolliset toi- minnalliset ja käytännölliset ominaisuudet ovat loppukäyttäjän kannalta tärkeim- mät. Liiketoiminnalliset ominaisuudet ovat oleellisia tuotteen omistajalle ja mark- kinoijalle. Rakenteelliset ominaisuudet puolestaan ovat keskeisiä sovelluskehit- täjien kannalta.

Tuotteen laadun tilastollisia ominaisuuksia ovat muun muassa luotettavuus (*reliability*), turvallisuus (*safety, security*), luottamuksellisuus (*confidentiality*), eheys (*integrity*) ja saatavuus (*availability*). Nämä ominaisuudet liittyvät tuotteen käytön turvallisuuteen ja tietoturvaan. Esimerkiksi tuotteen luotettavuudella tarkoitetaan sitä, kuinka järjestelmä toimii suhteessa määriksiin – luotettavuus mittaa siis tuotteen sisäistä turvallisuutta. Ulkoista turvallisuutta, eli tuotteen kestävyttä tahallista väärinkäyttöä vastaan, mitataan turvallisuudella (*security*). Järjestelmän on toimittava ilman kriittistä virhettä (*safety*) ja kyettävä jatkamaan toimintaa ja palautumaan virheestä (*availability, confidentiality*). (4, s. 15–17.)

Toiminnalliset ominaisuudet mittaavat tuotteen käyttöön liittyviä teknisiä piirteitä. Näitä ovat Milin ja Fairouzin mukaan viive (*latency*), teho (*throughput*), tehokkuus (*efficiency*), kapasiteetti (*capacity*) sekä skaalattavuus (*scalability*). Viiveellä tarkoitetaan tuotteen käytössä tehtävän syötteen ja tuotteen antaman ulostulon välistä kestoa. Teholla tarkoitetaan järjestelmän toimintakykyä suhteessa käytettyyn aikaan eli esimerkiksi sitä, kuinka kauan tietyn toiminnon suorittaminen kestää. Tehokkuus puolestaan tarkoittaa sitä, mikä on järjestelmän toimintakyky suhteessa sen resursseihin, kuten esimerkiksi keskusyksikköön tai muistiin. Toisin sanoen kyse on siitä, kuinka hyvin järjestelmä hyödyntää käytössä olevat resurssit. Kapasiteetilla tarkoitetaan samanaikaisten toimintojen toimivuutta ilman, että suorituskyky heikkenee. Skaalattavuus kuvaa sitä, kuinka hyvin järjestelmä kykenee toimimaan, jos sen työkuorma kasvaa oletettua suuremmaksi. (4, s. 17–18.)

Loppukäyttäjän kannalta kolmas oleellinen laadun ominaisuus on käytettävyys. Käytettävyys ei toiminnallisuuden tavoin viittaa laitteen suorituskykyyn vaan sen käyttäjäystävällisyyteen tavanomaisissa käyttötilanteissa. Käytettävyyden tunnusmerkkejä ovat käytön helppous (*ease of use*), oppimishelppous (*ease of learning*), mukautettavuus (*customizability*), kalibroituavuus (*calibrability*) sekä yhteentoimivuus (*interoperability*). Käytön helppous viittaa siihen, kuinka erilaiset käyttäjät on huomioitu esimerkiksi käyttöliittymän ja käytetyn kielen suhteen, kun taas oppimisen helppoudella tarkoitetaan muun muassa järjestelmän loogista ja yhdenmukaista toimintaa. Kustomoitava järjestelmä on mahdollista räätälöidä käyt-

täjän tarpeiden ja maun mukaan. Kalibroituavuus puolestaan tarkoittaa järjestelmän muokattavuutta käyttäjän toiminnallisten tarpeiden mukaan, esimerkiksi tiettyä käyttötarkoitusta varten. Yhteentoimivuudella tarkoitetaan sitä, kuinka järjestelmä toimii yhdessä muiden järjestelmien, kuten sovellusten, kanssa. (4, s. 18–19.)

Tuotteen laadun liiketoiminnalliset ominaisuudet ovat erityisen mielenkiintoisia tuotteen omistajille ja markkinoijille. Liiketoiminnallisia ominaisuuksia ovat tuotteen kehityksen kustannukset (*development cost*), ylläpidettävyys (*maintainability*), siirrettävyys (*portability*) sekä kierrätettävyys (*reusability*). Kustannuksilla tarkoitetaan luonnollisesti tuotteen kehityksen hintaa ja sitä voidaan mitata esimerkiksi kehitykseen käytettyinä työkuukausina. Ylläpidettävyys tarkoittaa sitä, kuinka suuri työmäärä vaaditaan tuotteen ylläpitoon, kuten tekniseen tukeen, tuotteen toimittamisen jälkeen. Siirrettävyys viittaa siihen, kuinka kallista järjestelmä on siirtää alustalta toiselle. Tähän tulee varautua jo kehitysvaiheessa hyödyntämällä mahdollisimman alustariippumattomia toimintoja ja esimerkiksi ohjelmistokirjastoja. Siinä missä edellä mainitut piirteet liittyvät suoraan tuotteen hintaan, liittyy kierrätettävyys tuotteen tuottavuuteen. Sillä tarkoitetaan sitä, kuinka helppoa järjestelmä tai joku sen osa on hyödynnettävissä jossain toisessa tuotteessa. (4, s. 19–20.)

Ohjelmistokehittäjien kannalta oleellisin laadun ominaisuus on rakenteellinen. Se koostuu järjestelmän teknisestä eheydestä (*design integrity*), modulaarisuudesta (*modularity*), testattavuudesta (*testability*) sekä joustavuus (*adaptability*). Järjestelmän tekninen eheys tarkoittaa sitä, että järjestelmän tekninen toteutus, esimerkiksi ohjelmiston koodi, on yhdenmukaista ja yksinkertaista. Sen tulee noudattaa annettuja määräyksiä ja on oltava kaikkien sen parissa työskentelevien henkilöiden ymmärrettävissä. Modulaarisuus viittaa siihen, kuinka järjestelmä hallinnoi ja piilottaa dataa. Ohjelman eri osat, moduulit, tekevät vain niille annettuja tehtäviä ja keskustelevat keskenään rajapintojen kautta. Järjestelmän testattavuus ilmaisee sen, kuinka kattavasti järjestelmää tai jotakin sen osaa voidaan testata. Testattavuuden osia ovat ohjattavuus (*controllability*) sekä havainnoitavuus (*observability*). Järjestelmää on kyettävä testaamaan erilaisilla syötteillä ja järjestelmän

tuottamaa dataa on kyettävä havainnoimaan mahdollisimman kattavasti. Joustava järjestelmää voidaan muokata muuttuvien vaatimusten ja määritysten mukaan. Esimerkiksi projektin aikana tuotteeseen voidaan haluta lisätä jokin uusi toiminnallisuus, jota ei alkuvaiheen määrittelyissä vielä huomioitu. Joustavan järjestelmän on kyettävä toteuttamaan uusi ominaisuus ilman regressiota eli ilman jo toteutettujen toimintojen heikkenemistä tai rikkoutumista. (4, s. 20–21.)

2.2 Ohjelmistotestauksen organisointi ja projektin elinkaari

Koska riski ohjelmistovirheille on läsnä kaikissa ohjelmistoprojektin vaiheissa, on testausta suoritettava kaikissa projektin vaiheissa. Testauksen on kuitenkin oltava ajan suhteen tehokasta, sillä vaikka teoriassa kaikki virheet olisivat löydettävissä tarpeeksi laajalla testauksella, eivät useimpien projektin ajalliset vaatimukset salli tuotteen täydellistä testaamista. Erityisesti laajojen ja monimutkaisten ohjelmistojen täysi testaaminen on mahdotonta. Tämän vuoksi testaaminen ja laadunvalvonta on projekteissa suunniteltava huolellisesti ja kohdistettava sellaisiin ohjelmiston osiin, joissa ohjelmistovirheiden mahdolliset seuraukset ovat vakavimmat. Esimerkiksi lentokoneiden ohjelmistoissa on pyrittävä löytämään ja korjaamaan kaikki virheet, jotka saattavat vaarantaa lentokoneen toiminnan. Toisaalta tietokonepelien ohjelmistovirheet eivät vaaranna ihmishenkiä, joten virheet voidaan havaita ja korjata jo tuotteen julkaisun jälkeen. Tuotteen testaaminen ja tuotteen laadun varmistaminen on siis suunniteltava hyvin jo ohjelmistoprojektin alkuvaiheessa. (6, luku Keeping Software Under Control.)

Baker ja Fisher esittävät, että sovelluskehityksen aikana on noudatettava laatuohjelmaa (*Quality Program*). Tämä prosessi kattaa kaikki ohjelmistoprojektin elinkaaren vaiheet suunnittelusta varsinaiseen testaukseen. Tämä prosessi koostuu heidän mukaansa kolmesta päävaiheesta, vaatimusten ja määritysten asettamisesta, laadunvarmistusmetodien määrittelystä sekä tuotekehitysprosessin ja lopullisen tuotteen laadun arvioinnista. (3, s. 8).

Laatuohjelman ensimmäisen osan tehtävä on asettaa tuotteelle halutut vaatimukset. Näitä ovat muun muassa erilaiset toiminnallisuudet sekä laatuvaatimukset. Tässä vaiheessa erityisen tärkeää on asettua loppukäyttäjän asemaan ja huomioidava se, kuinka loppukäyttäjä tuotetta käyttää. Laatuvaatimusten asettamisen

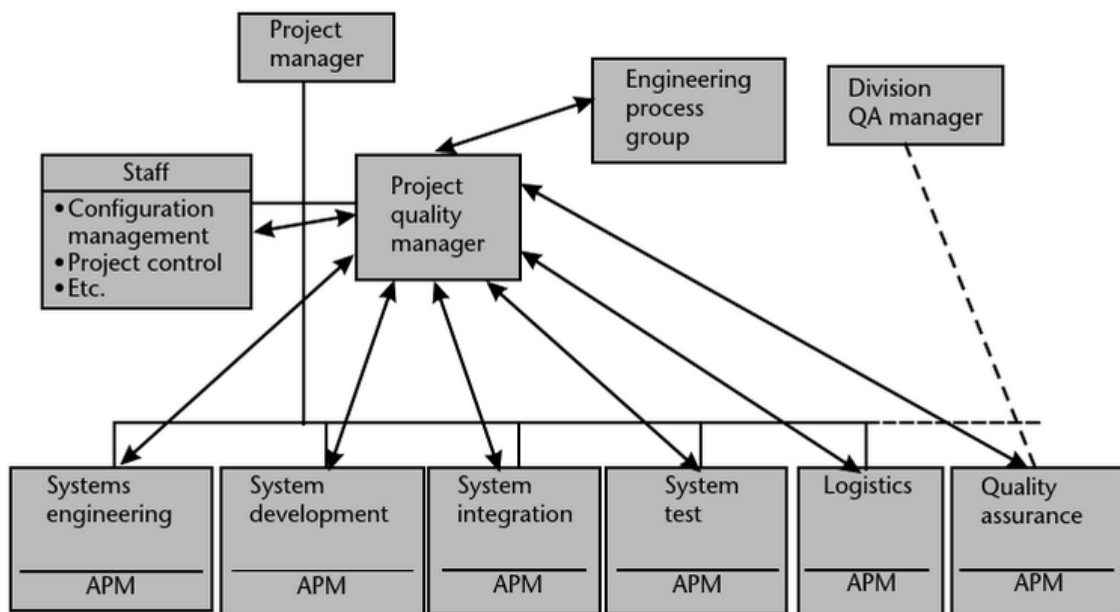
suurin haaste on objektiivisten, kvantifioitavien mittarien puute. Tämän vuoksi laatuvaatimusten dokumentointi ja kommunikointi on hankalaa. Lisäksi projektin alkuvaiheessa tulisi asettaa tuotteen laatuvaatimuksille perustaso (*baseline*). Mikäli tuotteeseen tarvitsee kehityksen aikana tehdä sellaisia muutoksia, joita ei alkuvaiheessa ole määritelty, tulee muutoksien jälkeinen ero perustasaan pitää mahdollisimman pienenä. (3, s. 11–12.)

Laatuohjelman toinen osa on Bakerin ja Fisherin mukaan asettaa tuotteelle selkeät laadunvarmistusmenetelmät. Nämä kattavat kaikki erilaiset menetelmät ja prosessit, joita tuotteen kehityksessä käytetään. Nämä tulisi aina kirjata ylös, jotta ne ovat kaikkien työryhmän jäsenten tiedossa ja jotta niistä on helppo keskustella esimerkiksi tiimipalaverissa. Laatuohjelman toinen osa on Bakerin ja Fisherin mukaan erittäin tärkeä tuotteen onnistumisen kannalta. He korostavat, että vain hyvin valittujen ja suunniteltujen toimintatapojen käyttö varmistaa sen, että tuotteen kehityskaari ja mahdolliset ongelmat ovat ennakoitavissa. Lisäksi hyväksi havaittuja ja hyvin dokumentoituja toimintatapoja on helppo hyödyntää uusissa projekteissa. (3, s. 12–13.)

Kolmas laatuohjelman osa on tuotekehitysprosessin ja tuotteen laadun arviointi. Eriyishuomiota tulee kiinnittää siihen, että laatuohjelman toisessa vaiheessa sovitut menetelmät on käytetty. Tässä vaiheessa myös suoritetaan tuotteen lopullinen testaaminen. Varsinainen arviointi tehdään laatuohjelman ensimmäisen ja toisen vaiheen määritykset huomioiden ja voi siten olla luonteeltaan joko subjektiivista tai objektiivista. Tuotteesta riippuen sen arviointi voidaan tehdä muun muassa erilaisin matemaattisin tai tilastollisin analyysein, testikäyttäjille kohdistetuin kyselyin tai erilaisilla testeillä. (3, s. 14.)

Bakerin ja Fisherin mukaan laatuohjelman kantava ajatus on se, että tuotteen laadunvarmistuksesta huolehditaan koko tuotekehityksen ajan. He korostavat, ettei tuotteen laatua voida varmistaa vain testaamalla – laadunvarmistusta ei voi jättää ainoastaan testaajien varaan. Tuotekehityksessä täytyy alusta alkaen huomioida tuotteen laatu. Tämä on mahdollista vain tarkkojen määritysten ja ennalta sovittujen tuotekehitysmenetelmien avulla. (3, s. 33.)

Määrittysten ja menetelmien lisäksi tärkeä osa laadunvarmistusta on oikeanlainen projektin organisointi. Baker ja Fisher tähdentävät, että projektissa tulee aina olla mukana tuotteen laadusta vastaava henkilö, laatupäällikkö (*quality manager*). Laatupäällikön tulee huolehtia siitä, että laatuohjelmaa noudatetaan. Sen vuoksi hänen tulee olla aktiivisesti mukana tuotteen määrittely- ja kehitysvaiheissa. Pienissä projekteissa projektipäällikkö voi täyttää myös laatupäällikön tehtävät. Tärkeintä on Bakerin ja Fisherin mukaan se, että laadusta vastaava henkilö valvoo yhteistyössä muiden työntekijöiden kanssa laatuohjelman noudattamista. Kuvassa 1 on esitetty Bakerin ja Fisherin esimerkki suuren projektin organisaatiokenteesta, jossa laatupäällikkö toimii yhteistyössä muun muassa ohjelmistokehittäjien, testaajien ja muun henkilökunnan kanssa. (3, s. 23–30.)



KUVA 1. Projektin rakenne (3, s. 28)

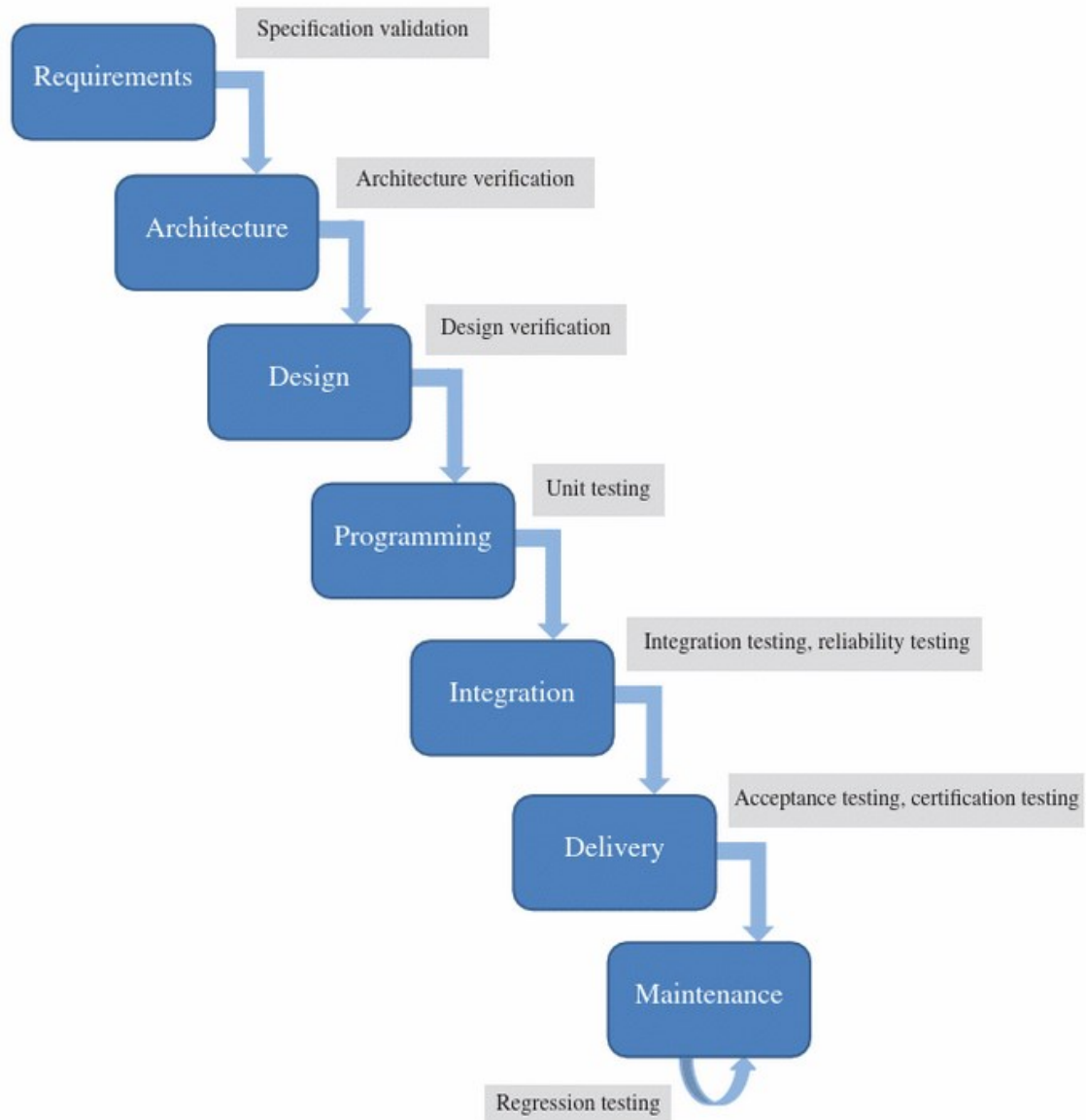
Mili ja Fairouz erottavat tuotekehitysprojektista seitsemän eri vaihetta tai osaa, joiden testaaminen koostuu eri tehtävistä. Kuitenkin hekin korostavat, että tuotteen laadunvarmistus kattaa koko projektin elinkaaren. Nämä seitsemän osaa ovat vaatimukset (*requirements*), arkkitehtuuri (*architecture*), suunnittelu (*design*), ohjelmointi (*programming*), integraatio (*integration*), toimitus (*delivery*) ja ylläpito (*maintenance*). (4, s. 23–24.)

Vaatimusten määrittelyssä varsinainen testaus on vähäistä, mutta laadunvarmistuksen kannalta oleellista on selvittää vaatimusten mielekkyys ja kannattavuus (*feasibility*). Tähän laadunvarmistuksen vaiheeseen kuuluu erilaisia analyysejä muun muassa alan markkinatilanteesta, tuotteen hinnasta, organisaatorakenteesta (mm. lisätyövoiman tarve) sekä tuotteen vaatimusten arviointi suhteessa loppukäyttäjään. Arkkitehtuurin osalta laadunvarmistus koostuu ensisijaisesti järjestelmän suunniteltujen osien yhteensopivuuden arvioinnista. Siihen kuuluu lisäksi muun muassa tuotteen suunnitellun toimituksen, jakelun ja mahdollisen tiedonkeruun toteuttamisesta. Suunnitteluvaiheessa laadunvarmistus toteutetaan tarkastelemalla järjestelmän eri osien toiminnan suhdetta koko järjestelmän toimintaan. Tavoitteena on, että mikäli jokainen järjestelmän osa toimii, myös lopullinen tuote toimii hyvin – kyseessä on siis järjestelmän toiminnan johdonmukaisuuden varmistaminen. (4, s. 23–24.)

Varsinainen testaaminen alkaa Milin ja Fairouzin mallissa ohjelmointivaiheessa. Ohjelmointivaiheessa suoritetaan yksikkötestaus (*unit testing*), jossa jokainen järjestelmän osa eli yksikkö testataan suhteessa sen määrityksiin. Tärkeää on, että jokainen osa toteuttaa sille asetetun tehtävän virheettömästi. Integraatiovaihe koostuu testaajan kannalta sekä integraatio- että luotettavuustestauksesta (*integration and reliability testing*). Integraatiotestauksessa varmistetaan, että jokainen järjestelmän osa kommunikoi oikein muiden osien kanssa. Integraatiotestauksen tavoitteena on löytää järjestelmästä virheitä, jotka estävät järjestelmän toimintaa. Luotettavuustestauksessa puolestaan arvioidaan järjestelmän luotettavuutta eli sitä, kuinka järjestelmä suoriutuu toiminnasta pitkällä aikavälillä ja esimerkiksi palautuu virheistä. (4, s. 24.)

Testejä suoritetaan myös tuotteen toimitusvaiheessa. Siinä missä integraatiotestauksen tarkoituksena on löytää virheitä korjattaviksi, osoitetaan toimitusvaiheen testeillä tuotteen virheetön toiminta. Tähän vaiheeseen voi liittyä myös sertifiointi, eli sen varmistaminen, että tuote täyttää halutut laatuvaatimukset. Tuotteen toimituksen jälkeen sen ylläpitovaiheessa suoritetaan vielä regressiotestausta (*regression testing*). Regressiotestauksessa varmistetaan, että ylläpitovaiheessa

tuotteeseen tehdyt korjaukset tai lisäykset eivät heikennä tuotteen toimintaa. Mi-
lin ja Fairouzin malli tuotekehityksen elinkaaresta ja siihen liittyvistä laadunvar-
mistuksen tehtävistä on esitetty kuvassa 2. (4, s. 24–25.)



KUVA 2. Ohjelmistoprojektin päävaiheet ja niiden testaus (4, s. 26)

3 KÄYTETYT VÄLINEET, KOMMUNIKOINTIKANAVAT JA PROSESSIT

3.1 Käytetyt välineet

Työssäni käytän useita erilaisia välineitä ongelmien etsimiseen. Useimmin käyttämäni työkalut ovat Applen ja Googlen toimittamat automaattiset testijärjestelmät, kaappauslaitteet ja -ohjelmistot sekä erilaiset testaukseen käytetyt iOS- ja Android-sovellukset.

Applen ja Googlen automaattisilla testijärjestelmillä laitteiden testaus Applen ja Googlen laatuvaatimuksia vasten on nopeaa. Lisäksi näiden järjestelmien tuottamat raportit muodostavat yhtenäisen kokonaisuuden, josta saa kattavan kuvan laitteen toiminnasta. Testijärjestelmien suorituksen kesto riippuu käytetyn päätelaitteen ominaisuuksista. Mikäli laitteessa ei ole paljoa toiminnallisuksia, voi testijärjestelmät suorittaa muutamassa tunnissa. Runsaasti ominaisuuksia sisältävien päätelaitteiden kohdalla suoritus voi kestää kaksi työpäivää.

Keskeisin osa työssäni on päätelaitteen ja älylaitteen välisen dataliikenteen kaappaaminen. Kaappaustyökalut tuottavat lokeja, joista käy ilmi laitteiden suorittamien komentojen lisäksi siirretyt datapaketit ja niiden sisältö. Dataliikennettä voidaan kaapata sekä USB-yhteydestä että langattomasta yhteydestä. Eräs käyttämäni kaappari on Wireshark, jolla voidaan helposti kaapata datapaketteja langattomasta yhteydestä. Saatujen lokien perusteella on helppo selvittää, koostuuko siirretty data juuri niistä asioista, joista sen vaatimuksien mukaan kuuluukin, vai onko datassa virheitä tai puutteita.

Käyttämäni iOS- ja Android-sovellukset sisältävät erilaisia työkaluja päätelaitteiden testaamiseen. Erityisen hyödyllisiä ne ovat silloin, kun niitä käytetään automaattisten testijärjestelmien kanssa. Sovelluksilla voidaan esimerkiksi simuloida erilaisia loppukäyttäjän kannalta oleellisia käyttökkenaarioita sekä välittää lisätietoa automaattisten testijärjestelmien käyttöön.

3.2 Kommunikointikanavat

Projekteissa kommunikointi Symbion ja asiakkaan välillä hoidetaan pääsääntöisesti sähköpostin ja JIRAn välityksellä. JIRA on Atlassian Corporation Plc:n kehittämä alusta projektinhallintaan (7). Asiakasprojekteissa käytetään JIRAan luotuja projektisivuja. Näihin projektisivuihin voidaan luoda tikettejä (*ticket*), jotka voivat olla esimerkiksi virheraportteja, kysymyksiä tai tuotteeseen toivottuja ominaisuuksia. Tikettiin voidaan liittää tiedostoja, kuten lokeja tai kuvia (8). Lisäksi tiketti voidaan aina osoittaa (*assign*) halutulle projektissa työskentelevälle henkilölle. Tiketin tilaa voidaan muuttaa työvaiheeseen sopivaksi, esimerkiksi avoimeksi (*open*), arvioitavaksi (*in review*) työn alla olevaksi (*in progress*), ratkaistuksi (*resolved*) tai suljetuksi (*closed*) (9).

3.3 Laadunvarmistusprosessin keskeisimmät vaiheet

Symbion asiakasprojektien vaiheet vaihtelevat hieman projektikohtaisesti, mutta työ sisältää pääsääntöisesti samat päävaiheet. Integraatiovaiheessa Symbion SCC-kirjasto liitetään osaksi asiakkaan päätelaitetta. Tässä vaiheessa asiakas liittää SCC-kirjaston osaksi omaa ohjelmistoaan. Samanaikaisesti Symbio räätälöi kirjaston asiakkaan tarpeisiin sopivaksi. Tämän jälkeen päätelaitteeseen lisätään puuttuvia tai haluttuja ominaisuuksia ja varmistetaan niiden yhteensopivuus SCC-kirjaston kanssa. Kun tuote sisältää halutut ominaisuudet, aloittaa Symbio tuotteen laadunvarmistuksen. Annetun palautteen perusteella asiakas korjaa havaitut ongelmat. Tämän vaiheen lopussa päätelaite on valmis ulkoiseen testaukseen. Ulkoisessa testauksessa asiakkaan ja Symbion ulkopuolinen taho suorittaa päätelaitteelle kattavan testauksen, josta saadun palautteen perusteella päätelaitteen viimeiset ongelmat korjataan ja SCC-kirjastoon tehdään mahdollisia muutoksia.

Tuotteen valmistumisen jälkeen asiakas voi käyttää tuotetta pohjana, jolle kehitetään ominaisuuksiltaan kattavampi tai karsitumpi laite, variantti. Varianttien kehittämisessä prosessi on sama, mutta koska variantit perustuvat jo toimiviin laitteisiin, on niihin liittyvä työmäärä yleensä vähäisempi kuin kokonaan uusissa tuotteissa.

Laadunvarmistusprosessissa keskeisessä osassa on projektin JIRA-sivu. Laitetta testataan sekä automaattisilla testijärjestelmillä että tutkivan testauksen (*exploratory testing*) kautta. Havaituista ongelmista kirjataan uudet tiketit projektin JIRA-sivulle ja tikettiin liitetään mahdollisimman paljon ongelman korjausta edesauttavaa tietoa, kuten lokeja ja kuvia. Ongelman mukaisesti osoitetaan tiketti joko projektia johtavalle henkilölle tai joko asiakkaan tai Symbion edustajalle. Ongelman korjaututtua ja korjauksen sisältävän ohjelmistopäivityksen saavuttua verifioidaan eli varmistetaan korjauksen toimivuuden. Mikäli ongelma on korjattu, voidaan tiketti ratkaista (*resolve*). Jos korjaus ei ole onnistunut, palautetaan tiketti uusine havaintoineen korjauksen tehneelle henkilölle.

4 OHJELMISTOTESTAAJAN OPPIMISPÄIVÄKIRJA

Oppimispäiväkirja on koottu Symbio Finland Oy:n asiakasprojekteissa. Oppimispäiväkirja on jaksotettu viikon mittaisiin kokonaisuuksiin, joissa esitellään viikon aikana tehtyjä työtehtäviä yleisluontoisesti. Työ koostuu pääsääntöisesti testaukseen liittyvistä työtehtävistä mutta myös jossain määrin SCC-kirjastoon liittyvistä ohjelmointitehtävistä sekä asiakkaiden ohjauksesta SCC-kirjaston käytössä.

Koska olen työskennellyt kyseisissä tehtävissä jo noin vuoden ajan, ei oppimispäiväkirja kuvaa asiakasprojektien kulkua niiden alusta loppuun. Sen sijaan oppimispäiväkirja alkaa nykytilanteesta ja kattaa noin kahdeksan viikon kuvauksen tuona aikana tehdyistä työtehtävistä.

Työskentelen pääsääntöisesti neljän eri projektin parissa, joita kuvaan kirjaimilla A, B, C ja D. Projektissa A olen työskennellyt pisimpään. Projekti keskittyi aluksi CarPlay-toteutukseen, kunnes keväällä 2019 painopiste siirtyi Android Autoon. Syksyllä 2019 painopiste siirtyi jälleen CarPlayhin. Projekti A on siirtymässä loppusuoralle ja se on valmis keväällä tai kesällä 2020. Työtehtäväni ovat koostuneet ensisijaisesti tuotteen testaamisesta suhteessa Applen ja Googlen laatuvaatimukseen. Projektin B parissa olen työskennellyt kevästä 2019 asti. Projekti on vasta alussa, joten tuotteesta puuttuu vielä paljon keskeisiä ominaisuuksia. Sen aikataulu on kuitenkin melko tiukka, minkä vuoksi se on vienyt valtaosan työajastani. Työni projektissa on pääasiassa puuttuvien ominaisuuksien ja ongelmien raportointia sovelluskehittäjille. Kun puuttuvat ominaisuudet on lisätty tuotteeseen, siirtyy painopisteeni projektin A tavoin tuotteen laadunvarmistukseen suhteessa Applen ja Googlen laatuvaatimukseen. Projektit C ja D ovat lähes valmiita. Niiden molempien on määrä valmistua talvella 2019–2020. Tämän vuoksi työni projekteissa on ollut pääsääntöisesti Applen ja Googlen toimittamia testijärjestelmiä, joilla tuotteita valmistetaan Applen ja Googlen tarkistettavaksi.

4.1 Viikko 41

Viikko 41 oli osaltani erittäin kiireinen, työskentelin kaikkien neljän projektin parissa. Työni projektin A parissa oli ollut melko vähäistä, sillä asiakas keskittyi ke-

sän 2019 aikana Android Autoon ulkoisen laadunvarmistusyrityksen kanssa. Viikon 41 aikana asiakas siirsi minulle useita CarPlay-korjauksia verifioitavaksi. Asiakas oli myös omissa testeissään huomannut ongelmia laitteen langattomassa yhteydessä. Vahvistin asiakkaan havainnot ja kirjasin uuden virheraportin projektin JIRA-sivulle. Ongelmat langattoman yhteyden kanssa liittyivät todennäköisesti tuotteen Android Auto -muutoksiin – tehdessään muutoksia Android Auto -osioon asiakas ei ollut varautunut regressioihin tuotteen CarPlay-tuessa.

Projektin B laitteeseen oli julkaistu uusi ohjelmistoversio, jonka toimivuus oli määrä testata nopeasti. Uuden version arvioitiin korjanneen useita ongelmia erityisesti laitteen Bluetooth-tuessa. Lisäksi laitteeseen oli nyt lisätty toiminnallisuuksia sen fyysisiin painikkeisiin. Myös mikrofonin oli nyt määrä toimia äänikomentojen antamista varten. Päätin testata laitetta automaattisella testijärjestelmällä. Pian kuitenkin huomasin, ettei Bluetooth-yhteys toiminut lainkaan. Myöskään fyysiset painikkeet tai mikrofoni eivät toimineet CarPlayn tai Android Auton kanssa. Kirjasin näistä uudet raportit JIRAan ja projektissa työskentelevä insinööri varmisti Bluetooth-yhteyteen liittyvät ongelmat.

Alkuviikosta työskentelin myös projektin C parissa Symbion vanhemman soveluskehittäjän kanssa. Tuotteen äänentoistossa oli vain joitakin viikkoja aiemmin havaittu merkittäviä puutteita tutkivan testauksen aikana. Äänentoisto-ongelmat ilmenivät tyypillisessä loppukäyttäjän käyttötilanteessa: äänenlaatu heikkeni, mikäli kahta tai useampaa äänilähdettä käytettiin yhtä aikaa tai nopeasti peräjäkeen. Esimerkki tällaisesta tilanteesta on tekstiviestin vastaanottaminen puhelun aikana. Vanhempi insinööri pyrki korjaamaan ongelmaa SCC-kirjaston osalta ja testasin hänen tekemänsä korjaukset ja kerroin havainnoistani Teams-keskustelussa. Ongelma saatiin korjattua, jonka jälkeen varmistin laitteen toimivuuden automaattisella testijärjestelmällä ja toimitin tulokset asiakkaalle.

Koska projekti D oli edennyt lähes valmiiksi, halusi asiakas kiireesti korjata viimeiset tunnetut ongelmat. Tämän vuoksi siirryin auttamaan toista nuorempaa insinööriä laitteen testaamisessa. Projekti, laite ja ohjelmisto olivat minulle entuudestaan tuntemattomia, joten jouduin käyttämään pari päivää laitteen käytön

opetteluun ja tunnettujen ongelmien löytämiseen. Esimieheni antoi minulle tehtäväksi etsiä uusia ongelmia tutkivan testaamisen kautta. Löysin kuusi uutta ongelmaa, jotka raportoin toiselle testaajalle, joka puolestaan liitti havaitut ongelmat omaan raporttiinsa. Esimerkiksi eräs ongelma liittyi todennäköiseen loppukäyttäjän käyttötapaan: Mikäli päätelaitteeseen oli muodostettu langaton CarPlay-yhteys samalla, kun toinen puhelin oli kytketty laitteeseen akun lataamista varten, ei langaton yhteys palautunut, mikäli CarPlay-laite vietiin kantaman ulkopuolelle ja tuotiin takaisin. Tällainen ongelma voi loppukäyttäjällä ilmetä siten, että toinen puhelin on kytketty päätelaitteeseen lataukseen ja käyttäjä poistuu autosta musiikkia toistavan tai navigaattoria käyttävän puhelimen kanssa. Autoon palatessa käyttäjän musiikki ei jatku eikä navigaattori jatka reittiohjeiden antamista.

Viikko oli kiireinen, koska työskentelin viikon aikana kaikkien neljän projektin parissa ja koska projektien B ja D asiakkaat toivoivat korjausten nopeaa testausta. Kiirettä helpotti kuitenkin se, että projektin D parissa työskentelevä toinen nuorempi insinööri työskenteli täysipäiväisesti projektin parissa. Samalla kuitenkin huomasin, että työajasta huomattava osuus kuluu kommunikointiin ja työn kordinointiin. Etenkin kommunikointi Teamsin ja JIRA:n kautta on hidasta, koska esitettyihin kysymyksiin ei saa vastausta välittömästi.

4.2 Viikko 42

Viikko oli huomattavasti edellistä viikkoa hiljaisempi. Merkittävin osa viikon työajasta meni jo aiemmin havaittujen ongelmien tutkimiseen. Työtä hankaloitti viikolla sattunut muutaman tunnin mittainen katkos toimiston internet-yhteydessä. Tuona aikana uusia ohjelmistoja ei voinut ladata asennettavaksi, sillä vaikka internet-yhteys olisi mahdollista muodostaa puhelimesta muodostetun tukiaseman kautta, vaaditaan useimpien projektien sivuille pääsyyn yhteys Symbion sisäisen verkon kautta.

Projektissa A viikko oli hiljainen. Asiakas välitti minulle verifioitavaksi korjauksia edellisen viikon ohjelmistoversiosta, jotka olivat minulta jääneet huomaamatta. Kuitenkin edellisellä viikolla havaittujen yhteysongelmien vuoksi jouduin merkitsemään lähes kaikki korjaukset epäonnistuneiksi, sillä en voinut testata korjauksia. Kyseessä on testausta estävä ongelma (*blocker*). Tämän ongelman vuoksi

päätin verifioida muut korjaukset uudelleen, kun yhteysongelmat on ratkaistu. Ainoa hyväksymäni korjaus liittyi ongelmaan, jonka seurauksena päätelaite näytti virheellisesti Android Auton logon, kun CarPlay-yhteys muodostettiin.

Suurimman osan viikon työstä tein projektin B parissa. Asensin päätelaitteeseen uuden ohjelmistopäivityksen ja verifioin laitteeseen tehdyt korjaukset. Minun oli myös määrä suorittaa automaattinen testijärjestelmä ja kirjata uudet ongelmat JIRAan. Huomasin kuitenkin, että jonkin muutoksen seurauksena käyttämäni kaappaustyökalu ei enää toiminut oikein. Kaapatussa liikenteessä oli jokin virhe, jonka vuoksi lähes kaikki kaapattu data oli turmeltunutta. Tämä ongelma toistui lähes jokaisella yrityksellä, joten kirjasin siitä välittömästi uuden tiketin JIRAan. Merkitsin sen korkeimman prioriteetin ongelmaksi, sillä se esti minua testaamasta laitetta. Yritimme muiden projektissa työskentelevien henkilöiden kanssa korjata ongelmaa. Vaihdoin kaapparin, johdot sekä liittimet ja kokeilin käyttää eri puhelimia sekä eri versioita niin päätelaitteen ohjelmistosta, puhelimen käyttöjärjestelmästä kuin käyttämästäni kaappariohjelmistostakin. Totesin myös käyttämieni laitteiden toimivan ongelmitta muiden asiakkaiden päätelaitteissa. Ongelma ei ratkennut, joten päätimme seuraavalla viikolla kaapata kaiken mahdollisen liikenteen päätelaitteen ja puhelimen välillä löytääksemme mahdollisen ongelmakohdan.

Projektien C ja D osalta viikko oli myös melko hiljainen. Suoritin asiakkaan pyynnöstä projektin C päätelaitteelle automaattisen testijärjestelmän ja toimitin tulokset JIRAan. Koska projektin parissa työskennellyt vanhempi insinööri oli jäänyt edellisellä viikolla lomalle, oli hänen tehtävänsä siirretty toiselle vanhemmalle insinöörille. Koska hän ei ole viime aikoina työskennellyt laitteen parissa, opastin häntä laitteen käytössä ja autoin häntä toistamaan ja selvittämään aiemmin havaittuja, vaikkakin uusimmissa ohjelmistoverioissa jo korjattuja äänentoisto-ongelmia.

Projektin D parissa työskentelevä nuorempi insinööri oli matkustanut edellisellä viikolla asiakkaan toimistolle suorittamaan testausta yhdessä asiakkaan kanssa. Teimme jonkin verran yhteistyötä projektin Teams-kanavan kautta. Suoritin lait-

teelle kosketusviiveeseen liittyviä testejä. Kosketusviiveellä tarkoitetaan sitä aikaa, joka kuluu päätelaitteen näytön koskettamisesta päätelaitteen reagointiin. Käytännössä testi suoritetaan kuvaamalla suurinopeuksisella videokameralla laitteen käyttöä ja laskemalla viive videon kuvien (*frames*) määrän ja kameran nopeuden avulla (kaava 1). Totesin kosketusviiveen olevan huomattavasti suurempi kuin korkein sallittu viive ja toimitin mittauksista Excel-taulukon työtoverilleni, joka välitti tulokset asiakkaalle.

$$((F_e - F_s) / FPS) \cdot 1000ms$$

KAAVA 1

F_e = videon kuva, jossa päätelaite alkaa reagoida kosketukseen

F_s = kuva, jossa laitetta kosketetaan

FPS = videokameran kuvaamien kuvien määrä sekunnissa

Koska viikko oli edellistä hiljaisempi, ehdin keskittyä työtehtäviini hyvin. Esimerkiksi projektille C suoritettuna automaattisen testijärjestelmän ehdin suorittaa hyvin ja kiinnittää huomiota testijärjestelmän tuottamiin lokeihin. Tämän vuoksi ongelmien raportointi asiakkaalle oli helppoa. Myös projektiin siirretyn vanhemman insinöörin opastukseen oli runsaasti aikaa, joten hän saattoi aloittaa ongelmien korjaamisen nopeasti. Kommunikointi vanhemman insinöörin kanssa oli helppoa, sillä hänen työpisteensä sijaitsee samassa toimistossa. Sen sijaan kommunikointi projektin D parissa työskentelevän nuoremman insinöörin kanssa Teamsin kautta oli edelleen hieman hidasta. Koska työtehtäviä ei ollut paljoa, ei tämä kuitenkaan haitannut niiden suorittamista.

4.3 Viikko 43

Suurimman osan viikon työajasta vei projektissa B edellisellä viikolla havaitun ja raportoidun kaappariongelman selvittely. Moni työntekijä vietti syyslomaa, minkä vuoksi korjauksia asiakasprojektien SCC-osioihin ei juurikaan tullut. Viikko oli erittäin kiireinen, sillä minun oli määrä suorittaa automaattinen testijärjestelmä projektin D laitteelle viikon loppuun mennessä samalla kun selvittelen projektin B kaapparissa havaittua ongelmaa. Lisäksi projektin A ohjelmistoon julkaistiin loppuviikolla uusi ohjelmisto ja minulle osoitettiin useita tikettejä. Koska olin sopinut syysloman viikolle 44, jäisi osa tehtävistä ylitöistä huolimatta viikolle 45.

Koska projektin A parissa työskentelevä vanhempi insinööri oli lomalla ja koska asiakas ei ollut toimittanut uutta ohjelmistoa, oli projektin parissa alkuvuikolla varsin vähän tekemistä. Esimieheni osoitti minulle muutaman tiketin, jotka asiakas oli pyytänyt verifioimaan, mutta joihin vanhempi insinööri ei ollut reagoinut ja joista en ollut saanut ilmoitusta. Tavanomaisesti JIRA-kommenteista ja -päivityksistä saavat tiketin raportoinut henkilö, sen parissa työskentelevä henkilö (*assignee*) sekä tiketin seuraajat (*watchers*). Koska en ollut mikään näistä, en saanut kysymyksistä ilmoitusta. Totesin, etteivät edellisen päivityksen korjaukset olleet toimineet ja päivitin havaintoni tiketteihin. Tämän jälkeen osoitin tiketit uudelleen niiden parissa työskennelleille henkilöille. Loppuviikolla asiakas julkaisi uuden ohjelmistoversion ja osoitti minulle yli 10 tikettiä, jotka ehdin verifioida. Suurin osa korjauksista oli toimivia, joten suljin tiketit. Sen sijaan langattomaan yhteyteen liittyviä tikettejä en edelleenkään voinut testata, sillä langattoman yhteyden ongelmia ei päivityksessä ollut korjattu.

Kaapparien toiminnassa havaittu ongelma projektissa B nousi viikon aikana projektin korkeimman prioriteetin ongelmaksi. Koska kaappariin kaappaamassa dataliikenteessä esiintyi vieläkin virheellistä dataa, ei laitteen testaaminen ollut mahdollista. Olin edellisellä viikolla varmistanut, ettei vika ole käyttämässäni laitteistossa. Lisäksi toinen projektin parissa työskentelevä henkilö sai toistettua saman ongelman toisella laitteistolla. Tämä varmisti, ettei kyseessä ole esimerkiksi laitevika kaapparissa. Ongelmaa ryhdyttiin etsimään kahdella tavalla. Ensinnäkin käyttämällä kahta kaapparia yhtä aikaa, saatiin laitteen toiminnasta lokitiedostot,

jossa ongelman toistuminen näkyy kahdessa eri dataväylässä. Vertaamalla näitä toimivan laitteen lokitiedostoihin, ongelman mahdollista juurisyitä on mahdollista rajata. Toiseksi ryhdyimme asentamaan päätelaitteeseen vanhoja ohjelmistoversioita siinä toivossa, että toimiva versio löytyy. Tämän jälkeen olisi mahdollista asentaa uudempia ohjelmistoversioita, kunnes vian lisännyt päivitys löytyy. Valittavasti vanha ohjelmistoversio, joka vain kuukautta aiemmin toimi ongelmitta, ei enää toiminut. Syyksi epäiltiin sitä, etteivät ohjelmistopäivitykset ole todella päivittäneet järjestelmän kaikkia osia. Joka tapauksessa ongelma ei vielääkään ratkennut.

Projektin C päätelaitteessa oli havaittu vika, jonka seurauksena koko päätelaite jumiutuu, mikäli kytketty puhelin irrotetaan päätelaitteen ollessa valmiustilassa. Lisäksi laitteessa havaittiin merkittävä ongelma CarPlayn käynnistyksessä. Autoin projektissa työskentelevää vanhempaa insinööriä testaamaan ongelmaa. Vertasimme päätelaitteen toimintaa muiden projektien päätelaitteisiin ja totesimme, ettei ongelma toistunut. Toimitin havainnot ongelmia koskeviin tiketteihin projektin JIRA-sivulle.

Keskiviikkona projektin D asiakas ilmoitti haluavansa molempien automaattisten testijärjestelmien tulokset viikon loppuun mennessä. Tavallisesti molempien suorittaminen kestäisi kolme-neljä päivää, mutta päätimme jakaa työn toisen nuoremman insinöörin kanssa. Koska toimistollamme on kaksi projektin D päätelaitetta, pystyimme suorittamaan testejä yhtä aikaa. Projektin A kiireiden vuoksi järjestimme perjantain aikataulut niin, että aloitin testauksen kello 6.00. Työtoverini tullessa töihin kello 10 jälkeen hän jatkoi testijärjestelmän loppuun, joten ehdin myös verifioida lähes kaikki projektin A korjaukset.

Viikko oli kaiken kaikkiaan erittäin kiireinen. Erityisen haasteen aiheuttaa projektien päällekkäisyys, minkä vuoksi yhteen projektiin tai yhteen ongelmaan ei ehdi keskittyä kovin kauan. Tämän vuoksi työ on katkonaista. Esimerkiksi viikon aikana työni projektin D parissa keskeytyi toistuvasti projektin B ongelmien vuoksi. Lisäksi koska yhteen työtehtävään ei ehdi keskittyä täysipainoisesti, voi testeissä tapahtua huolimattomuusvirheitä. Tämä voi aiheuttaa ongelmia esimerkiksi siten, että ongelmia ei havaita ajoissa, jolloin niiden korjaaminen viivästyy.

4.4 Viikko 44

Viikolla 44 pidin syysloman.

4.5 Viikko 45

Viikolla 45 työskentelin projektien A ja B parissa. Lisäksi minun oli määrä aloittaa laadunvarmistus syksyn aikana käynnistettyä projektia E. Projektin E tuotteeseen on syksyn aikana integroitu SCC-kirjasto pääpiirteissään valmiiksi ja tuotteen ominaisuudet ovat pääsääntöisesti valmiita laadunvarmistusta varten.

Olin ehtinyt viikolla 43 ennen syyslomalle jääntiä verifioimaan kaikki projektin A korjaukset. Ennen lomaani osoitin muutamia tikettejä projektissa työskentelevälle vanhemmalle insinöörille. Hän ei kuitenkaan ollut ehtinyt perehtyä ongelmiin, minkä vuoksi asiakas osoitti tiketit uudelleen minulle. Palattuani töihin osoitin tiketit muiden kiireiden vuoksi jälleen vanhemmalle insinöörille ja hän vastasi asiakkaan kysymyksiin nopeasti. Tämän lisäksi asiakas julkaisi maanantaina uuden ohjelmistoversion ja osoitti minulle verifioitavaksi noin kymmenen korjausta. Valitettavasti edellisillä viikoilla ongelmia aiheuttanutta ohjelmistovikaa langattomassa yhteydessä ei ollut korjattu. En voinut verifioida muita langattomaan yhteyteen liittyviä korjauksia, vaan merkitsin JIRAssa alkuperäisen langattoman yhteyden ongelman estäjäksi (*blocker*) ja linkitin uudet ongelmat tähän alkuperäiseen ongelmaan. Tämä helpottaa asiakasta huomaamaan pahimmat ongelmistoviat ja korjaamaan testaamisen estävät ongelmat nopeasti. Muut ongelmat erityisesti kiinteässä yhteydessä oli korjattu. Kirjasin näihin tiketteihin havainnot ja merkitsin tiketit ratkaistuksi (*resolved*).

Suurimmaksi osaksi työskentelin kuitenkin projektin B parissa. Maanantaina myös projektin B tuotteeseen julkaistiin uusi ohjelmistoversio ja minulle osoitettiin lista TestRail-palvelussa, johon oli merkitty lähes 80 suoritettavaa CarPlay-testiä. TestRail muistuttaa JIRAA ja siihen merkityt testit voidaan merkitä niiden suorittamisen jälkeen esimerkiksi onnistuneiksi (*pass*), epäonnistuneiksi (*fail*) tai estyneiksi (*blocked*). Testaaminen oli kuitenkin erittäin hankalaa, sillä edellisillä viikoilla ilmennyt ongelma kaappareiden liikenteessä esti minua saamasta testeistä

lokeja joka yrityksellä. Tästä huolimatta, onnistuin saamaan muutamia lokitiedostoja, joita analysoimalla sain verifioitua suurimman osan korjauksista. Kun huomasin kaappauksen onnistuvan, pyrin suorittamaan saman CarPlay-session aikana mahdollisimman monta testiä. Lopulta kaapparin keräämä data alkoi turmeltua joka yrityksellä, minkä vuoksi merkitsin loput, noin 20 testiä, estyneiksi (*blocked*) ja huomautin tiketeissä kaappariongelmasta. Tämän jälkeen minun oli määrä suorittaa laitteelle USB-yhteyden suorituskykyä mittaava testi. Testi on automaattinen, mutta vaatii testisovelluksen käyttämistä päätelaitteella esimerkiksi sarjayhteyden kautta. Koska päätelaitteen testisovellus ei ollut yhteensopiva puhelimen testisovelluksen kanssa, jouduin kääntämään (*build*) testisovelluksen oikean version lähdekoodista. Tämä ei kuitenkaan onnistunut, koska työkoneeni arkkitehtuuri poikkesi päätelaitteen arkkitehtuurista. Sain käyttööni päätelaitteen kehitystyökalut (*software development kit*), jonka avulla kääntäminen onnistui ja sain suoritettua USB-yhteyden suorituskykytestit onnistuneesti.

Minun oli viikon aikana määrä aloittaa sekä CarPlayn että Android Auton automaattiset testijärjestelmät projektissa E. Suoritin alustavan testauksen elokuussa, minkä seurauksena asiakas lisäsi vaadittuja puuttuvia ominaisuuksia tuotteeseen. Automaattiset testijärjestelmät oli määrä suorittaa nyt uudelleen, jotta asiakas voi varmistua siitä, että puuttuvia ominaisuuksia ei enää ole. Lisäksi testiajon jälkeen oli määrä kirjata JIRAan kaikki havaitut ongelmat, jotta asiakas voi ryhtyä korjaamaan niitä. Ongelmana oli kuitenkin se, että asiakas toimitti uuden ohjelmistoversion eri muodossa kuin ennen. Projektin parissa työskentelevät vanhemmat insinöörit eivät tienneet, kuinka uusi ohjelmistoversio asennetaan, joten kysyin sitä asiakkaalta. Tämä ei kuitenkaan vastannut projektin CarPlay-sivulla esittämäni kysymykseen, joten kirjasin uuden erillisen kysymystiketin, jonka osoitin suoraan asiakkaan edustajalle. Koska asiakas ei viikon aikana vastannut, en voinut aloittaa testausta.

Viikko poikkesi merkittävästi tyypillisestä työviikostani. Yleensä työni koostuu vain testaamisesta, mutta projektin B käännöstehtävien myötä pääsin työskentelemään Linux-järjestelmän, kääntäjien ja sovelluskehitystyökalujen parissa. Tämä oli mielestäni mukavaa vaihtelua. Ennen siirtymistäni laadunvarmistusteh-

täviin työskentelin Symbion omassa ohjelmistoprojektissa, johon kuului ohjelmoinnin lisäksi ohjelman käännöstyökalujen käyttöä ja optimointia Linux-ympäristössä. Projektin B käännöstehtävät toimivat muistinvirkistykseenä ja koin myös oppineeni uutta erityisesti sovelluskehitystyökalujen käytöstä eri arkkitehtuurilla toteutettujen alustojen välillä. Myös TestRail-palvelu oli minulle ennestään tuntematon, mutta mielestäni se oli hyvin helppokäyttöinen eikä palvelun käyttö tuottanut vaikeuksia. Laadunvarmistustyössä ongelmat kommunikaation kanssa nousivat viikon aikana jälleen pintaan. Koska projektin A asiakas ei ollut tietoinen syyslomastani ja koska projektissa työskentelevä vanhempi insinööri ei ollut ehtinyt vastata asiakkaan kysymyksiin, oli asiakas ymmärrettävästi hämillään. Tämä ongelma ilmenee toisinaan myös siten, että ilmoitukset JIRA-tiketteihin lisätyistä kommentteista tai kysymyksistä saapuvat sähköpostitse. Mikäli työpäivän aikana saa useita kymmeniä sähköposti-ilmoituksia, jäävät yksittäiset kysymykset helposti huomaamatta. Ongelman voisi ratkaista osoittamalla tiketti välittömästi sille henkilölle, jolle kysymys tai kommentti on esitetty. Tällöin kyseinen henkilö näkee etusivullaan heti hänelle osoitetut tiketit eivätkä kysymykset jää huomaamatta.

4.6 Viikko 46

Viikko 46 poikkesi hieman tavallisesta, sillä projektin B asiakkaan edustajia oli vierailulla Symbion toimistolla. Tämän vuoksi työskentelin viikon aikana suurimmaksi osaksi projektin B parissa.

Projektiin A ei tullut viikon aikana lainkaan korjauksia. Sen sijaan olin maanantaina projektin B kaappariongelman suhteen toiveikas, sillä testijärjestelmä oli päivittynyt viikonlopun aikana. Päivitin ohjelmiston maanantaiaamuna ja ryhdyin ottamaan kaapparilla lokitiedostoja. 15 onnistuneen lokitiedoston jälkeen olin ryhtymässä kirjaamaan kaappauksen estävän ongelman tikettiin havaintoa, että ongelma oli johtunut päätelaitteen sijaan käytetystä testiohjelmistosta. Päätin kuitenkin jatkaa testausta ja ottaa vielä muutamia kymmeniä lokeja siltä varalta, että ongelma ilmenisi uudestaan. Jo 16. kaappaus epäonnistui, minkä jälkeen lähes jokainen yritys tuotti käyttökelvottomia lokeja. Ohjelmistopäivitys ei ollut korjannut ongelmaa. Tämän jälkeen ryhdyin tutkimaan ongelmaa tarkemmin

käyttäen toista kaappausohjelmistoa, jonka havaitsemia virheitä selvitin toimistoltamme löytyneen USB-tekniikkaa käsittelevän kirjan avulla. Ongelmaan ei kuitenkaan löytynyt mitään yksiselitteistä syytä. Neuvottelin ongelmasta myös asiakkaan edustajien kanssa. Neuvottelussa tultiin siihen tulokseen, että ongelmasta kerätään mahdollisimman kattavat lokitiedostot ja niiden perusteella kirjaan kysymys Applelle. Totesimme myös, että kysymykseen ei ole odotettavissa nopeaa vastausta, minkä vuoksi laitteen testaus viivästyy mahdollisesti useilla viikoilla.

Projektin C asiakas oli viikonlopun aikana toivonut lisätestejä laitteessa havaittuun ongelmaan, jossa CarPlay ei käynnisty säädetyn aikarajan puitteissa. 20 yrityksestä jopa 14 ylitti tavoiteajan. Testatessamme ongelmaa vanhemman insinöörin kanssa huomasimme, että kaapparin ilmoittama CarPlayn käynnistysaika riippuu siitä, yhdistetäänkö puhelin päätelaitteeseen iPhoneen Lightning-liitännän vai päätelaitteen USB-liitännän kautta. Yhteyden muodostus liittämällä Lightning-kaapeli puhelimeen oli poikkeuksetta nopeampi. Ero oli 20 yrityksellä keskimäärin puoli sekuntia. Suoritin 20 yhteyden muodostusta sekä Lightning- että USB-liitännän kautta ja latasin lokitiedostot ongelman JIRA-tikettiin.

Koska projektin E asiakas ei maanantaihin mennessä ollut vastannut kysymykseeni uuden ohjelmistoversion asennuksesta, muistutti esimieheni asiakasta kysymyksestä sähköpostitse. Tiistaina asiakas vastasi ja toimitti asennusohjeet. Huomasin kuitenkin, että käynnistettäessä päätelaitteen käyttöliittymä ei ollut käytettävissä, vaan päätelaite näytti aloitusruudun jälkeen mustaa ruutua. Muodostin sarjaporttiyhteyden laitteeseen ja totesin laitteen toimivan. Kysyttyäni ongelmasta toiselta projektin parissa työskentelevältä insinööriltä, hän toimitti minulle projektin sovelluskehitystyökalut, joiden avulla käänsin päätelaitteen ohjelmiston lähdekoodista. Tämän jälkeen laite käynnistyi ja sain asennettua uusimman ohjelmistoversion. Suoritin laitteelle CarPlayn ja Android Auton automaattiset testijärjestelmät. Erilaisia kriittisiä ongelmia löytyi yli 80, jotka kirjasin projektin JIRA-sivustolle.

Viikon mielenkiintoisin havainto oli projektin C CarPlay-yhteyden käynnistysajan riippuvuus siitä, kuinka puhelin yhdistetään päätelaitteeseen. Käynnistysaika on

aiemmin testattu lähes yksinomaan USB-porttia käyttämällä, koska testatessa on haluttu välttää puhelimen Lightning-portin kulumista. Vastedes testit on kuitenkin tehtävä Lightning-porttia käyttäen, sillä sen todettiin yhdessä projektin C vanhemman insinöörin ja esimiehen kanssa vastaavan paremmin loppukäyttäjän tilannetta.

Mieluisin tehtävä viikon aikana oli projektin E ohjelmiston kääntäminen lähdekoodista. Vaikka olin edellisellä viikolla kääntänyt projektin B ohjelmiston, oli tehtävä melko haastava projektien laitteistojen ja ohjelmistojen poikkeavien arkkitehtuurien vuoksi. Projektin E testauksessa huomasin, että havaittujen ongelmien kirjaaminen JIRAan on melko työlästä, erityisesti mikäli tikettejä täytyy tehdä useita – tässä tapauksessa 85 kappaletta. Pelkästään ongelmien kirjaukseen käytin lähes kaksi työpäivää. Valitettavasti käytetyt automaattiset testijärjestelmät eivät tue minkäänlaista JIRA-integraatiota, jonka avulla ongelmat saisi automaattisesti kirjattua ylös, vaan tiketit on luotava manuaalisesti.

4.7 Viikko 47

Olin viikon 47 alussa sairauslomalla kaksi päivää. Loppuviikolla työskentelin pääasiassa projektin B parissa. Lisäksi verifioin projektin A päätelaitteeseen julkaistun uuden ohjelmistoversion korjaukset.

Projektin A päätelaitteeseen oli julkaistu uusi ohjelmistoversio tiistaina. Päivityksessä oli korjattu noin 10 ongelmaa käyttöliittymään ja langattomaan yhteyteen liittyen. Pahinta ongelmaa langattoman yhteyden kanssa ei kuitenkaan vielä ole ollut korjattu. Koska langattoman yhteyden muodostus tietyillä vanhemmilla puhelimilla onnistui, verifioin korjaukset. Suljin kahta lukuun ottamatta kaikki tiketit ja kirjasin havainnot tiketteihin, joiden korjaukset eivät olleet onnistuneet. Osoitin toisen tiketin takaisin asiakkaan edustajalle, toisen vanhemmalle Symbion insinöörille. Projektin A päätelaitteen ongelmia on muutaman viikon aikana korjattu paljon ja laite toimii jo melko vakaasti. Odotettavissa on lähiviikkoina uusi täysi ajo automaattisella testijärjestelmällä.

Projektin B kaappariongelmaa ei vieläkään ole korjattu. Tästä huolimatta minulle osoitettiin yli 80 CarPlay-testiä TestRail-palvelussa, jotka oli määrä suorittaa viikon 47 aikana. Koska dataliikenteen kaappaus onnistui satunnaisesti, olisi testien suorittamiseen kulunut huomattavan kauan. Sen vuoksi muodostin CarPlay-yhteyttä uudelleen, kunnes sain kelvollisen lokitiedoston. Tämän kaappauksen aikana käytin laitetta mahdollisimman kattavasti ja lokitiedoston tietojen perusteella ratkaisin annetut testit. Kaikki sellaiset testit, joissa vaaditaan useita yhteyden katkaisuja ja uudelleenmuodostuksia, merkitsin estyneiksi. Tästä huolimatta 80 testin suorittaminen kesti noin kaksi työpäivää. Testien lisäksi tutkin kaappariongelmaa tarkemmin, vaikka ongelman syytä ei edelleenkään löytynyt. Päätimme ryhtyä kääntämään, asentamaan ja testaamaan vanhoja ohjelmistoversioita löytääksemme sellaisen version, jolla kaappariongelmaa ei ilmene. Kun tällainen versio löytyy, voidaan tarkastella seuraavassa päivityksessä tehtyjä muutoksia, mikä voi auttaa ongelman paikantamisessa. Versioita ei kuitenkaan ehditty testata viikon aikana kuin yksi, sillä ohjelmiston kääntäminen projektin laitteelle kestää useita tunteja.

Viikko oli pääasiassa melko turhauttava. Projektin B kaappariongelman vuoksi hyödyllisen lokitiedoston saaminen vaati usein jopa 20 yritystä. Koska projektissa kuitenkin vaaditaan testitulokset säännöllisesti, on testit yritettävä suorittaa. Koska kaappariongelma ei suoraan liity korjattuihin ongelmiin, ei testejä voida suoraan merkitä estyneiksi, vaan testit on pyrittävä suorittamaan mahdollisimman kattavasti. Sain myös viikon aikana tietää, ettei asiakkaan tai Symbion työryhmässä kaappariongelmaa ollut raporteistani huolimatta pidetty korkeimman prioriteetin ongelmana, minkä vuoksi vanhojen versioiden testausta ei ollut aiemmin järjestetty. Koska oma roolini projektissa on testaaja, ei minulla ole pääsyä ohjelmistoversioiden tietokantaan eikä minulla ole ohjelmistotyökaluja vanhojen versioiden kääntämiseksi. Toisaalta kaappariongelma ei hankaloita ohjelmistokehittäjien työtä, minkä vuoksi on jossain määrin ymmärrettävää, ettei ongelmaa ollut priorisoitu kovin korkealle.

4.8 Viikko 48

Viikon 47 viikonloppuna sairastuin uudestaan, minkä vuoksi työterveyslääkäri määräsi minulle sairauslomaa kolme päivää. Pääsin kuitenkin viikon 48 torstaina jatkamaan töitä ja sain heti tehtäväkseni suorittaa projektin B laitteelle automaattisen testijärjestelmän langattomien testien osalta. Lisäksi minun oli määrä aloittaa automaattisen testijärjestelmän täysi ajo projektin A laitteelle CarPlayn osalta.

Projektin A laitteelle oli jälleen julkaistu uusi ohjelmistoversio, jossa korjauksia oli noin kymmenen. Korjausten verifiointi kesti muutaman tunnin. Korjaukset liittyivät jälleen pääasiassa käyttöliittymään ja langattoman yhteyden toimintaan. Lähes kaikki korjaukset olivat onnistuneita, joten suljin kahta lukuun ottamatta kaikki minulle tarkistusta varten osoitetut tiketit. Eräs korjaus liittyi raportoimaani ongelmaan, joka ei ollut varsinainen ohjelmistovirhe. Päätelaitteen käynnistyessä käyttäjälle näytettiin kaksi ilmoitusta. Ensimmäinen ilmoitus liittyi siihen, että laite muodosti automaattisesti Bluetooth-yhteyden edellisellä käyttökerralla yhdistettyyn laitteeseen. Toinen ilmoitus sisälsi ohjeet Android Auton yhdistämistä varten. Tein näistä ilmoituksista tiketin pari viikkoa aiemmin, sillä erityisesti toinen ilmoitus Android Autoon liittyen on loppukäyttäjän kannalta turha. Loppukäyttäjällä ei välttämättä ole lainkaan Android-laitetta, jolloin ilmoituksesta ei ole hyötyä. Asiakas oli tulkinnut Googlen dokumentaatiota väärin, sillä dokumentaation mukaan päätelaitteen tulee ohjeistaa Android Auton yhdistämisessä, kun käyttäjä koskettaa Android Auto -kuvaketta silloin, kun Android-laitetta ei ole yhdistetty. Asiakas oli korjannut ongelman. Verifiointin jälkeen aloitin automaattisen testijärjestelmän suorittamisen. Koska edellisestä täydestä testauksesta on kulunut jo useita viikkoja, toimitin asiakkaan varmistettavaksi viimeisimmät testimäärytykset (*test configuration*). Määrytykset vaikuttavat siihen, mitä testejä testijärjestelmä sisältää. Vaikka testit ovat pääasiassa samat määrytyksistä riippumatta, vaatii täsmällinen testaus oikeiden määrytysten asettamisen. Ehdin perjantaina aloittamaan testijärjestelmän suorituksen, mutta suurin osa testeistä jäi seuraavalle viikolle.

Projektin B osalta paluuni sairauslomalta oli hyvin kiireinen. Esimieheni oli vierailulla asiakkaan luona ja tarvitsi testitulokset viimeistään perjantaiksi. Automaatti-

sen testijärjestelmän suorittaminen langattoman yhteyden osalta kestää noin yhden työpäivän, mikäli langaton yhteys toimii hyvin ja laite on testaajalle tuttu. Koska langattomassa yhteydessä oli vakavia ongelmia yhteyden muodostamisessa, päätin suorittaa testit ohjelmiston sijaan Excel-taulukkoon. Täten saatoin yhdellä virheettömällä lokitiedostolla merkitä useamman testin suoritetuksi. Ongelmana tässä oli se, ettei testikohtaisia lokitiedostoja saanut talteen. Ehdin suorittamaan testit torstain aikana, vaikka jouduinkin jäämään joksikin aikaa ylitöihin. Huomasin myös, että aiempina viikkoina havaittu kaappariongelma on lähes täysin korjaantunut. Havaitsin testauksen aikana turmeltunutta dataa vain muutamia kertoja. Tiedustelin ongelmaa muilta projektin työntekijöiltä, mutta heidän mukaansa korjauksia USB-liikenteeseen ei ole tehty. Tämä helpotti testausta huomattavasti, sillä yhteyden muodostamisessa oli niin paljon ongelmia, että mikäli kaapparin keräämä data olisi lisäksi ollut turmeltunutta, olisi testaus ollut mahdotonta.

Vaikka viikko oli sairauslomani takia lyhyt, olivat kaksi työpäivää mielenkiintoisia. Vaikka toisaalta oli hyvä, että projektin B kaappariongelma on korjattu, oli hieman huolestuttavaa, että vika oli korjaantunut ikään kuin vahingossa. Koska ongelman perimmäinen syy ei selvinnyt, voi ongelma ilmetä uudelleen myöhemmin projektin edetessä. Automaattisen testijärjestelmän suorittaminen projektin A laitteelle oli mukavaa. Olen työskennellyt projektin parissa noin vuoden, ja ero vuoden takaiseen tilanteeseen on huomattava. Vaikka ehdin suorittamaan vain osan testeistä ja vaikka osa testeistä ei vielääkään mennyt läpi, oli ilo huomata, että kriittisiä tai testaamista estäviä ongelmia ei juurikaan löytynyt. Tämä jouduttaa testaamista huomattavasti, minkä vuoksi voin suorittaa testit nopeammin ja luotettavammin ja antaa asiakkaalle hyödyllisempää palautetta viimeistenkin ongelmien korjausta varten. Tämän edistyksen vuoksi työskentely projektin A parissa on ollut erittäin palkitsevaa.

4.9 Viikko 49

Viikko 49 oli melko kiireinen. Suoritin viikolla 48 aloittamani automaattisen testijärjestelmän ajon loppuun projektin A päätelaitteelle. Lisäksi minun oli määrä suorittaa täysi vastaava testiajo projektin B laitteelle.

Jatkoin projektin A laitteen testiajota maanantaina. Esimieheni oli luvannut tulokset asiakkaalle maanantain loppuun mennessä. Testeillä oli siis kova kiire, sillä en ollut ehtinyt aloittaa edellisellä viikolla langattomia testejä, jotka vievät suurimman osan testiajon ajasta. Jäin joksikin aikaa ylitöihin, mutta ehdin suorittaa testit loppuun. Ilokseni huomasin, että langaton yhteys toimi melko hyvin. Päätelaitteen laitevalikon käyttöliittymä oli aiempiin versioihin verrattuna keho eikä esimerkiksi useiden laitteiden yhdistäminen ja niiden CarPlay-yhteyksien vaihtaminen onnistunut. Käytin jälleen langattomien testien apuna sekä Wiresharkia että Applen Bluetooth-kaappaukseen tarkoitettua sovellusta. Kirjasin viikon aikana kaikki uudet ongelmat JIRAan sekä avasin muutamia aiemmin suljettuja tikettejä uudelleen. Yhteensä ongelmia löytyi noin 25 kappaletta. Huolestuttavaa oli useiden aiemmin ratkaistujen ongelmien ilmaantuminen uudelleen. Tämä kertoo siitä, etteivät asiakkaan tekemät korjaukset ole täysin onnistuneet ja uusien korjausten myötä ohjelmistossa on ilmennyt lievää regressiota.

Projektin B laitteelle oli jälleen julkaistu uusi ohjelmistoversio. Työtoverini projektista D kuitenkin sanoi haluavansa testata laitetta, sillä hän ei ole vielä suorittanut projektille B täyttä testausta. Tämä oli mielestäni hyvä ajatus, sillä hän voi tulkita joitakin testejä eri tavalla ja löytää siten ongelmia, joita en itse ole aiemmin havainnut. Opastin häntä laitteen käyttöönotossa, sillä esimerkiksi langatonta yhteyttä ei aiemmissa versioissa voinut muodostaa ilman tiettyjä sarjaportin kautta syötettyjä komentoja. Työtoverini ehti suorittaa kaikki kiinteään yhteyteen liittyvät testit ja jatkoin testit loppuviikolla loppuun langattoman yhteyden osalta. Uusia ongelmia löytyi kaikkiaan noin 40, ottaen huomioon, että osaa testeistä ei voinut suorittaa puuttuvien toiminnallisuuksien takia. Testijärjestelmän suorituksen aikana minulle oli jälleen osoitettu yli 80 uutta testiä projektin B laitteelle. Koska suurin osa testeistä oli tuttuja, onnistui niiden suorittaminen melko nopeasti. Harmikseni totesin, että mukana oli sellaisia testejä, jotka olin aiemmilla viikoilla merkinnyt yhteensopimattomaksi päätelaitteen kanssa. Tällaisia olivat esimerkiksi väärin konfiguroidut testit. Suoritettuani testit, kirjasin ylös puutteelliset ja virheelliset testit ja toimitin listan niistä vastaavalle Symbion työntekijälle ja pyysin häntä korjaamaan testit.

Koska viikon 49 perjantaina vietettiin itsenäisyyspäivää, oli viikko hieman tavallista lyhyempi. Viikko ei myöskään ollut yhtä kiireinen kuin tavallisesti, koska työverini halusi suorittaa osan projektin B testeistä. Työskentely vain kahden projektin parissa oli myös mukavaa vaihtelua. Koska pystyin keskittymään projektin B päätelaitteen testeihin melko kiireettömästi, huomasin ongelmia, jotka olivat aiemmilla testikerroilla jääneet huomaamatta. Lisäksi minulla oli aikaa tehdä vapaamuotoista testausta, jonka seurauksena kirjasin muutaman tiketin laitteen käyttöliittymäongelmista. Erityisesti päätelaitteen käyttöliittymä langattoman yhteyden muodostamisen osalta toimii epäjohdonmukaisesti ja sen sisältämiä ongelmia ei välttämättä huomaa automaattisen testijärjestelmän ajon aikana. Vapaamuotoisella testauksella voidaan havaita ongelmia, joita automaattiset testijärjestelmät eivät havaitse.

5 YHTEENVETO

Työn tarkoitus oli kirjoittaa oppimispäiväkirja ohjelmistotestaajan tehtävistä Symbio Finland Oy:n asiakasprojekteissa ja peilata työssä kertyneitä kokemuksia ja havaintoja alan kirjallisuudessa esiteltyyn testaamisen ja laadunvarmistuksen teoriaan. Oppimispäiväkirja sisältää havaintoja yhdeksän viikon ajalta. Vaikka asiakasprojektit jatkuvat oppimispäiväkirjan jälkeen, on oppimispäiväkirjassa esitelty ohjelmistotestaajan työn sisältöä sekä varsinaisen ohjelmistotestauksen että muiden alan työtehtävien osalta. Yhdeksän viikon aikana työtehtävät sisälsivät ohjelmointia lukuun ottamatta kaikkia työtehtäviä, joita olen Symbiolla kahden vuoden aikana tehnyt.

Ensimmäinen havaintoni on, että laadunvarmistus vaihtelee Symbiolla huomattavasti eri projektien välillä. Opinnäytetyön aikana työskentelin viiden projektin parissa ja vaikka näissä käytetyt välineet olivatkin pääsääntöisesti samat, oli eri projektien laadunvarmistus järjestetty hieman eri tavoin. Yhteistä kaikille projekteille oli Applen ja Googlen toimittamien automaattisten testijärjestelmien ja -ohjelmistojen käyttö. Merkittävimmät erot liittyivät käytettyihin kommunikointikanaviin. Siinä missä muissa projekteissa hyödynnettiin projektinhallinnassa JIRAA, toimitettiin projektin D testien tulokset asiakkaalle Excel-taulukossa sähköpostitse. Lisäksi muissa projekteissa testaaminen keskittyi automaattisten testijärjestelmien ajoon, kun taas projektissa B toimitettiin testijärjestelmien lisäksi säännöllisesti erillisiä testejä suoritettavaksi TestRail-palvelussa. Muissa projekteissa TestRail-palvelu ei työn kirjoittamisen aikaan ollut käytössä.

Testaamisen tarkoitus on verrata laitteen ja ohjelmiston toimintaa sille asetettuihin määrittämiin ja havaita loppukäyttäjän kannalta oleellisia ongelmia. Koska projektien päätelaitteiden toiminnallisuudet on tarkkaan määritelty, voidaan Applen ja Googlen automaattisilla testijärjestelmillä varmistaa laitteen oikea toiminta. Loppukäyttäjän kannalta monia oleellisia ominaisuuksia, kuten järjestelmän käyttöliittymän helppoutta tai suorituskykyä, ei automaattisilla testijärjestelmillä juurikaan testata, minkä vuoksi niiden arviointi riippuu vapaamuotoisesta testaamisesta. Projektien päällekkäisyyden ja työn kiireellisyyden vuoksi vapaa-

muotoiselle testaamiselle ei kuitenkaan jää paljoa aikaa. Toisaalta keskeisin työtehtäväni on varmistaa, että testatut laitteet täyttävät Applen ja Googlen laatu-standardit, joten esimerkiksi käyttöliittymäsuunnittelun arviointi ei toimenkuvaani varsinaisesti kuulu.

Symbion asiakasprojektien laadunvarmistus oli työn aikana järjestetty lähes täysin samoin, kuin alan kirjallisuudessa suositellaan. Lähes kaikki teoriassa esitetyt päävaiheet vaatimusten määrittelystä integrointitestaukseen ja sertifiointitestaukseen olivat projekteissa havaittavissa. Toimenkuvaani ei kuulunut arkkitehtuurin tai ohjelmiston osien yksikkötestaus. Myös Symbion projektien organisaatiota-kenne noudatti melko tarkasti teoriassa esitettyjä suosituksia. On kuitenkin tärkeää huomioida, että Symbion asiakasprojektit olivat kirjoittamisen aikaan melko pieniä – niissä työskenteli vain muutamia työntekijöitä. Tämän vuoksi erillinen laadunvarmistuksesta vastaava henkilö, laatupäällikkö, oli vain projektissa B. Muissa projekteissa laadunvarmistuksesta vastasi viime kädessä Symbion tiimin esimies. Erillisen laatupäällikön hyödyt suuremmassa projektissa näkyivät kuitenkin työssä selvästi. Kommunikointi oli hyvin suoraviivaista ja laatupäällikön hoitaessa kommunikoinnin ohjelmistokehittäjien kanssa jäi testaajalle enemmän aikaa varsinaisen työn eli testauksen hoitamiseen.

Teoriassa esiteltyä laatuohjelmaa ei ollut missään projektissa varsinaisesti eritelty. Sen päävaiheita, kuten määrittelyä, metodien asettamista ja sovittujen menetelmien käytön arviointia, ei sellaisenaan noudatettu. Ne kuitenkin sisältyivät jossain muodossa kaikkiin projekteihin. Projektien asiakkaat toimittivat Symbiolle laitteiden määritykset esimerkiksi määritystiedostoina. Testimenetelminä kaikissa projekteissa hyödynnettiin Applen ja Googlen ohjelmistoja. Tuotetta paranneltiin korjaamalla virheitä testeissä tehtyjen havaintojen perusteella. Koska automaattiset testijärjestelmät olivat Symbion laadunvarmistustyössä keskeisessä osassa, ei muita laadunvarmistusmetodeja juurikaan käytetty. Koska kuitenkin testijärjestelmien tuottamissa raporteissa nähdään tuotteen laadun yleistilanne, oli käytettyjen menetelmien ja projektin tilanteen arviointi helppoa.

Testauksen kenties suurin ongelma, jota kirjallisuudessa korostetaan, on testauksen viime kädessä subjektiivinen luonne. Tämä pitää jossain määrin paikkansa

myös automaattisia testijärjestelmiä käytettäessä. Osa testeistä on objektiivisia: ne esimerkiksi tarkastavat, onko laite määritelty ohjelmallisesti oikein sovittuihin määrityksiin nähden. Toisaalta osa testeistä on hyvin subjektiivisia. Äänenlaatua tai tiettyjä käyttöliittymäelementtejä arvioitaessa tulokset eri testaajien välillä voivat vaihdella. Tämä ilmeni oppimispäiväkirjassa projektissa B. Siinä missä olin merkinnyt osan testeistä onnistuneiksi, merkitsi kollegani samat testit omassa testiajossaan epäonnistuneiksi. Tämän vuoksi olisi suotavaa, ettei testausta jätettäisi vain yhden henkilön varaan, vaan että projektin aikana kaksi tai useampi suorittaisi samat testit.

Opinnäytetyön ensimmäinen tarkoitus oli oppia ymmärtämään testaamisen ja laadunvarmistuksen teoriaa paremmin ja oppia soveltamaan sitä käytännössä. Merkittäviä oppimiskokemuksia ei kuitenkaan kuluneen yhdeksän viikon aikana tullut. Tämä johtuu siitä, että olin työskennellyt samoissa työtehtävissä ennen oppimispäiväkirjan aloittamista jo vuoden ajan ja työ on minulle suurimmaksi osaksi tuttua. Tästä huolimatta työssä oppi joitakin uusia asioita. Esimerkiksi projektien B ja E sovellustyökalujen käyttö ja käännöstehtävät olivat selvää vaihtelua tuttuun testaustyöhön. Myös Applen ja Googlen dokumentaatio ja ohjelmistot päivittyvät melko usein ja vaikka ne säilyvätkin pääpiirteissään samanlaisina, on jokaisessa päivityksessä uusia opeteltavia asioita. Esimerkiksi automaattisiin testijärjestelmiin lisätään toisinaan uusia testejä ja dokumentaatioon lisätään uusia laatuvaatimuksia. Lisäksi Symbion laadunvarmistusprosessi noudatti melko tarkasti alan kirjallisuuden suosituksia, joten uusia näkökulmia laadunvarmistuksen järjestämiseen ei juurikaan tullut. Keskeisin oppimiskokemus liittyi testauksen subjektiivisuuteen, ja lisäksi opin ymmärtämään useiden testaajien ja testaajien välisen kommunikoinnin merkityksen laadunvarmistustyössä.

Opinnäytetyön toinen tarkoitus oli pyrkiä etsimään kehityskohtia Symbion laadunvarmistusprosessista. Suurimmat havaitsemani epäkohdat olivat projektien hajanainen kommunikaatio sekä kiire. Kuluneen yhdeksän viikon aikana totesin useita kertoja, etten minä tai asiakkaan edustaja huomannut esitettyjä kysymyksiä ja että kysymyksiin tai kommentteihin vastaaminen kesti joissain tapauksissa jopa viikon. Useimmin tämä johtui siitä, ettei havaitun ongelman JIRA-tikettiä ollut

osoitettu kysymyksen esittämisen jälkeen oikealle henkilölle. Tällöin kysymyksestä tulee ilmoitus sähköpostiin, josta se helposti jää huomaamatta. Näin käy erityisesti silloin, kun erilaisia sähköposti-ilmoituksia on tarkastettavana päivässä useita kymmeniä tai jopa satoja. Tätä ongelmaa tulisi pyrkiä ratkaisemaan keskustelemalla asiakkaan ja Symbion työntekijöiden kesken JIRAn käytöstä ja muistuttaa tikettien osoittamisen tärkeydestä. Lisäksi olisi syytä harkita toisen nopeamman kommunikaatiokanavan, kuten Teamsin, hyödyntämistä myös asiakkaan kanssa. Tällöin kysymyksiä voisi esittää lähes reaaliajassa ilman vaaraa, että kysymys unohtuu muiden sähköpostien sekaan.

Sen sijaan toinen merkittävä ongelma, työssä usein läsnä oleva kiire, on hankalampi ongelma ratkaista. Kiire on luonteeltaan jaksottaista. Jotkin viikot ovat hyvin kiireisiä esimerkiksi silloin, kun kahteen tai useampaan projektiin on tehtävä täydelliset automaattisen testijärjestelmän suoritukset. Toisinaan useampaan projektiin saapuu samanaikaisesti uusia ohjelmistopäivityksiä, joissa korjatut ongelmat on pyrittävä verifioimaan nopeasti. Toisinaan taas viikot ovat hiljaisia, kun uusia versioita ei ole julkaistu ja testijärjestelmän suorituksia ei tarvitse tehdä. Kiireen jaksottaisuuden vuoksi lisätyövoiman palkkaaminen ei välttämättä ole tehokas ratkaisu. Sen sijaan eräs ratkaisu olisi pyrkiä järjestelmään projektit niin, että kiireisinä viikkoina työtehtävät priorisoidaan ja työvoimaa kohdistetaan kiireisimpien työtehtävien suorittamiseen. Tätä varten esimerkiksi projektin B järjestäminen siten, että laatu päällikkö vastaa työtehtävien valmistelusta ja testauksen ohjauksesta, oli mielestäni toimiva ratkaisu. Muissa projekteissa, joissa erillistä laatu päällikköä ei ole, tulisi harkita esimerkiksi lyhyiden palaverien pitämistä viikoittain. Näin työtehtävät voitaisiin priorisoida ja työntekijöitä ohjata kiireellisimpien tehtävien pariin.

Työ onnistui kokonaisuudessaan hyvin. Vaikka merkittäviä oppimiskokemuksia ei kuluneen yhdeksän viikon aikana tullutkaan, oli alan kirjallisuuteen perehtyminen opettavaista. Teorian pohjalta Symbion laadunvarmistusprosessista löytyi ainakin kaksi kehityskohtaa projektien kommunikaatiossa ja organisoinnissa. Lisäksi oppimispäiväkirjan kirjoittaminen tehdystä työstä oli hyödyllistä, sillä se auttoi jäsentämään tehtyä työtä, pohtimaan opittuja asioita ja arvioimaan omia työskentelytapoja. Näistä kaikista on hyötyä projektien jatkuessa.

LÄHTEET

1. We Think Global And Act Local. 2019. Symbio. Saatavissa: <https://www.symbio.com/contact-us/>. Hakupäivä 16.10.2019.
2. Who We Are. 2019. Symbio. Saatavissa: <https://www.symbio.com/who-we-are/>. Hakupäivä 16.10.2019.
3. Baker, Emanuel – Fisher Matthew 2008. Organizing for Quality Management. Teoksessa Schulmeyer, Gordon. (toim.), Handbook of Software Quality Assurance. Norwood, Massachusetts: Artech House, Inc. S. 1–34. Saatavissa Ebook Central Academic Complete International Edition (vaatii käyttöoikeuden).
4. Mili, Ali – Fairouz, Tchier 2010. Software Testing. Hoboken, New Jersey: John Wiley & Sons Inc. Saatavissa Ebook Central Academic Complete International Edition (vaatii käyttöoikeuden).
5. Spillner, Andreas – Linz, Tilo – Schaefer, Hans 2014. Software Testing Foundations. Rocky Nook. Saatavissa O'Reilly for Higher Education -tietokannasta (vaatii käyttöoikeuden).
6. Hambling, Brian (toim.) – Morgan, Peter – Samaroo, Angelina. Software Testing. British Computer Society, The Chartered Institute for IT. British Informatics Society Ltd. Saatavissa O'Reilly for Higher Education -tietokannasta (vaatii käyttöoikeuden).
7. Jira Software. 2019. Atlassian Corporation Plc. Saatavissa: <https://www.atlassian.com/software/jira>. Hakupäivä 18.10.2019.
8. Attaching files and screenshots to issues. 2019. Atlassian Corporation Plc. Saatavissa: <https://confluence.atlassian.com/jirasoftwareserver073/attaching-files-and-screenshots-to-issues-861256335.html>. Hakupäivä 18.10.2019.

9. Issue statuses, priorities and resolutions. 2019. Atlassian Corporation Plc. Saatavissa: <https://confluence.atlassian.com/adminjiracloud/issue-statuses-priorities-and-resolutions-973500867.html>. Hakupäivä 18.10.2019.