



Karelia-ammattikorkeakoulu  
Tietojenkäsittelyn tradenomi (AMK)

# Ohjelmistotestaajan päiväkirja

Miia Jaakkola

Opinnäytetyö, joulukuu 2021

[www.karelia.fi](http://www.karelia.fi)



**OPINNÄYTETYÖ**  
**Joulukuu 2021**  
**Tietojenkäsittelyn koulutus**

Tikkarinne 9  
80200 JOENSUU  
+358 13 260 600

Tekijä(t)  
Miia Jaakkola

Nimeke  
Ohjelmistotestaajan päiväkirja

**Tiivistelmä**

Opinnäytetyö on kirjoitettu päiväkirjamuotoisena toteutuksena, jossa seurataan kirjoittajan työskentelyä ohjelmistotestaajana 13 viikon mittaisen ajanjakson ajan. Opinnäytetyön tavoitteena on kirjoittajan ammatillisen osaamisen kehittyminen testausasiantuntijana. Seurattavia osa-alueita ovat erityisesti testausprosessi sekä kirjoittajan teknisen osaamisen kehittyminen.

Ensimmäisenä käydään lävitse aloitustilanteen kuvaus sekä opinnäytetyön tietoperusta. Tämän jälkeen siirrytään päiväkirjaosioon, jossa kirjoittaja kuvaa työtehtäviään päiväkohtaisesti sekä viikkopäiväkirjan muodossa. Päiväkirjan kirjoittamisen rinnalla kirjoittaja vertaa omaa työskentelytapaansa siihen, mitä kirjallisuudessa kerrotaan testausprosessin kulusta.

Päiväkirjan seurantajakson aikana kävi ilmi, että testausprosessin laajuus ja perinpohjaisuus riippuu monesta eri asiasta, kuten käytössä olevasta ohjelmistokehityksen elinkaarimallista ja testausprojektin luonteesta. Lisäksi testaamisen alalla on paljon erilaisia erikoistumismahdollisuuksia ja alalla riittää opittavaa loputtomiin, joten tärkeintä on uteliaisuus ja halu kehittyä ja oppia jatkuvasti uusia asioita.

Kieli  
suomi

Sivuja 80

Asiasanat  
ohjelmistotestaus, manuaalinen testaus, laadunvarmistus



**THESIS**  
**December 2021**  
**Degree Programme in Business Information  
Technology**

Tikkarinne 9  
80200 JOENSUU  
FINLAND  
+ 358 13 260 600

Author (s)  
Miia Jaakkola

Title  
Diary of a Software Tester

**Abstract**

This thesis was written in diary format, and it follows the author's working as software tester for 13 weeks. The aim of this thesis was to follow the author's professional growth as software tester, and it specifically focuses on testing process.

At first the starting point and the theory base of this thesis are introduced. After that the actual diary section starts, where author describes her work in daily and weekly diaries. Daily work and testing process are then compared to testing process found in literature.

It was found that testing process may vary in different software projects. It was also found that learning is lifelong process in this industry, so the most important thing is curiosity and willingness to improve working processes and knowledge.

Language  
Finnish

Pages 80

Keywords  
software testing, manual testing, quality assurance

# Sisältö

1	Johdanto .....	5
2	Aloitustilanteen kuvaus .....	5
2.1	Opinnäytetyön tietoperusta ja keskeiset käsitteet .....	5
2.1.1	Ohjelmistotestauksen käsitteitä .....	6
2.1.2	Testattava järjestelmä.....	8
2.2	Keskeiset työtehtävät.....	10
2.3	Työssä tarvittavat taidot.....	15
2.4	Kehittymisen arviointi .....	16
2.5	Sidosryhmät ja vuorovaikutus työpaikalla .....	17
2.5.1	Työ- ja toimintaympäristö ja sidosryhmät.....	17
2.5.2	Työ- ja toimintaympäristö, vuorovaikutus.....	19
3	Päiväkirjatyöskentely .....	19
3.1	Viikko 1 (23.-27.8.2021).....	19
3.2	Viikko 2 (30.8.2021-3.9.2021).....	26
3.3	Viikko 3 (6.9. -10.9.2021).....	31
3.4	Viikko 4 (13.9. -17.9.2021).....	36
3.5	Viikko 5 (20.9. -24.9.2021).....	40
3.6	Viikko 6 (27.9. -30.9.2021).....	43
3.7	Viikko 7 (4.10. -7.10.2021).....	48
3.8	Viikko 8 (11.10. -15.10.2021).....	53
3.9	Viikko 9 (18.10. -21.10.2021).....	58
3.10	Viikko 10 (25.10 – 28.10.2021) .....	62
3.11	Viikko 11 (1.11. - 4.11.2021).....	64
3.12	Viikko 12 (8.11. - 11.11.2021).....	68
3.13	Viikko 13 (16.11. -19.11.2021).....	73
4	Pohdinta.....	76
	Lähteet.....	80

## 1 Johdanto

Opinnäytetyö on kirjoitettu päiväkirjamuotoisena toteutuksena. Opinnäytetyössä käsitellään ohjelmistotestaamisen teoriaa oman päivittäisen työskentelyni kuvaamisen yhteydessä. Seurantajakso on 13 viikon mittainen. Päiväkirjan aloitusajankohta on viikko 34 (23.8.2021) ja lopetusajankohta viikko 46 (19.11.2021). Opinnäytetyö alkaa aloitustilanteen tarkastelulla, jossa käydään lävitse tietoperustaa ja ohjelmistotestauksen keskeisiä käsitteitä sekä keskeisiä työtehtäviäni. Seuraavaksi siirrytään varsinaiseen päiväkirjaraportointiin ja lopuksi tulee pohdinta siitä, mitä olen oppinut tarkastelujakson aikana. Opinnäytetyön tavoitteena on oman ammatillisen osaamiseni kehittyminen testausasiantuntijana. Seurattavia osa-alueita ovat erityisesti testausprosessi sekä teknisen osaamisen kehittyminen.

Työni liittyy IAM (Identity and Access Management) -järjestelmän testaamiseen. IAM-järjestelmän testaamisen parissa työskentelee yrityksessä useita testaaajia, joista osa tekee töitä tuotekehityksen puolella ja osa asiakasprojektien parissa. Tuotekehityksen puolella testaustyöhön liittyy manuaalisten testien suorittamisen lisäksi testiautomaation tekemistä. Asiakasprojektien puolella testiautomaatiota ei toistaiseksi hyödynnetä, vaan testaustyö on manuaalista. Oma työni on manuaalista testaamistyötä asiakasprojekteissa.

## 2 Aloitustilanteen kuvaus

### 2.1 Opinnäytetyön tietoperusta ja keskeiset käsitteet

Opinnäytetyön testaukseen liittyvä tietoperusta pohjautuu suurimmaksi osaksi ISTQB:n (International Software Testing Qualifications Board) perustason sertifiointimateriaaliin ja sen pohjalta kirjoitettuun Foundations of Software Testing -kirjaan. Lisäksi opinnäytetyössä tulee esiin käsitteitä, jotka liittyvät testattuun järjestelmään ja testauksen apuna käytettyihin testaustyökaluihin. Osa

käsitteistä käydään lävitse perusteellisemmin ja osa lyhyemmin. Ensin esitellään opinnäytetyössä esiin tulevia, testaukseen yleisemmin liittyviä käsitteitä ja sen jälkeen testattavaan järjestelmään liittyviä käsitteitä. Opinnäytetyössä on päädytty ratkaisuun olla käyttämättä testauksen apuna käytettyjen ohjelmien/ohjelmistojen oikeita nimiä erityisesti siitä syystä, että lukijan on mahdollisesti helpompi hahmottaa ja muistaa mitä tarkoitetaan, kun puhutaan esimerkiksi ”testauksen hallinnan järjestelmästä” kuin että puhuttaisiin järjestelmästä sen oikealla nimellä. Erilaisia järjestelmiä, niihin liittyviä liitännäissovelluksia ja käsitteitä tulee työssä esiin melko paljon myös viikkoseurannan yhteydessä. Joskus käsitteiden jälkeen on lisätty sen englanninkielinen vastine, koska näitä termejä käytetään yleisesti tai ne tunnetaan paremmin englanniksi.

### **2.1.1 Ohjelmistotestauksen käsitteitä**

Ohjelmistotestaus määritellään ISTQB:n sertifikaattisääntöissä FISTB:n (Finnish Software Testing Board) sivuilla (2018, 12) seuraavasti: ”Ohjelmistotestaus on keino arvioida ohjelmiston laatua ja pienentää tuotantokäytössä tapahtuvien häiriöiden riskiä.” Testaus ja laadunvarmistus käsitteinä eivät kuitenkaan ole sama asia, vaikkakin ne liittyvät toisiinsa. Laadunhallinta on suurempi kokonaisuus, joka sitoo sisäänsä laadunvarmistuksen ja testauksen. Laadunvarmistus sisältää kaikki dokumentit, jotka kertovat minkälaisia toimintamalleja yrityksessä seurataan ja keskittyy näin ollen prosesseihin. Testaus puolestaan sisältää ne toiminnot, jotka täytyy suorittaa, jotta organisaatio saavuttaa asetetut laatuvaatimukset ja keskittyy siis enemmän konkreettisiin tuotteisiin ja toimintoihin. Laadunvarmistus kuitenkin tukee testausta, koska testauksen aikana tulisi seurata määritettyjä prosesseja. (Graham, Black & van Veenendaal 2020, 7.)

Ohjelmistotestausta voidaan suorittaa eri tasoilla, joita ovat yksikkötestaus, integraatiotestaus, järjestelmätestaus sekä hyväksyntätestaus. Yksikkötestausta, joka tunnetaan myös komponenttitestauksena, suorittavat tavallisesti ohjelmistokehittäjät. Integraatiotestausta on kahdentyyppistä: komponentti-integraatiotestaus sekä järjestelmäintegraatiotestaus. Näistä komponentti-integraatiotestaus on tavallisesti kehittäjän vastuualuetta. Järjestelmäintegraatiotestaus

puolestaan kuuluu testaajalle. Järjestelmäintegraatiotestaus keskittyy järjestelmien, pakettien ja mikropalveluiden välisiin vuorovaikutuksiin ja rajapintoihin. Tällä testaustasolla ei siis testata yksittäistä järjestelmää, vaan järjestelmien välistä vuorovaikutusta. Järjestelmätestaus puolestaan keskittyy koko järjestelmän tai tuotteen käyttäytymiseen ja ominaisuuksiin. Myös hyväksymistestaus keskittyy koko järjestelmän tai tuotteen käyttäytymiseen ja ominaisuuksiin, mutta ero järjestelmätestauksen ja hyväksymistestauksen välillä on esimerkiksi siinä, että hyväksyntätestauksen tavoitteena ei ole enää varsinaisesti vikojen löytäminen, vaan tiedon tuottaminen sen arvioimiseksi, onko järjestelmä valmis julkaistavaksi ja asiakkaan käytettäväksi. Järjestelmätestauksen tavoitteena puolestaan on muun muassa vikojen löytäminen. (FISTB/ISTQB 2018, 27–32.)

Myös testaustyyppjä on erilaisia: toiminnallinen testaus, ei-toiminnallinen testaus, lasilaatikkotestaus ja muutokseen pohjautuva testaus. Toiminnallinen testaus tutkii järjestelmän käyttäytymistä ja sitä tehdään silloin kun järjestelmää on jo mahdollista oikeasti käyttää. Ei-toiminnallinen testaus keskittyy esimerkiksi käytettävyyteen, suorituskykyyn tai tietoturvaan. Lasilaatikkotestaus keskittyy järjestelmän sisäiseen rakenteeseen, esimerkiksi koodiin. Muutokseen pohjautuvassa testauksessa järjestelmää testataan sen jälkeen, kun siihen on tehty jokin muutos. Varmistustestauksella varmistetaan jonkin vian korjaus. Regressio-testaus puolestaan tarkoittaa muutokseen pohjautuvaa testaustyyppiä, jossa varmistetaan, että järjestelmään tehdyt muutokset eivät ole aiheuttaneet ennakkoimattomia haitallisia seurauksia järjestelmään. Ylläpitotestauksesta voidaan puhua, kun järjestelmä on jo julkaistu tuotantokäyttöön ja siihen tehdään muutoksia. Testausta tehdään tällöin sen arvioimiseksi, kuinka onnistuneesti muutokset on tehty sekä mahdollisten sivuvaikutusten löytämiseksi niistä järjestelmän osista, joita ei ole muutettu. (FISTB/ISTQB 2018, 35–38.) Testaustyypeistä lasilaatikkotestaus on testaustyyppi, jota en omassa työssäni hyödynnä. Enimmäkseen työni keskittyy toiminnalliseen testaukseen sekä muutokseen pohjautuvaan testaukseen ja on ylläpitotestausta, sillä järjestelmä on tuotantokäytössä ja siihen tehdään ajoittain muutoksia sekä suunnitellusti (Service Release, SR), että toisinaan myös pikakorjauksia (Hot Fix, HF).

Testaus ei ole pelkästään testitapausten suorittamista, vaan koostuu prosessista ja monista erilaisista tehtävistä ennen ja jälkeen varsinaisen testaustyön. Testausprosessi koostuu testauksen suunnittelusta, testauksen seurannasta ja hallinnasta, testianalyysistä, testien suunnittelusta, testien valmistelusta, testien suorituksesta ja testauksen päättämisestä (FISTB/ISTQB 2018, 16–17). Näihin kaikkiin eri vaiheisiin liittyy omat tehtävänsä, joihin viitataan opinnäytetyön edessä päivä- ja viikkoraporteissa.

### 2.1.2 Testattava järjestelmä

IAM (Identity and Access Management) -termiä käytetään yleisesti kuvaamaan identiteetin- ja pääsynhallinnan järjestelmiä. IAM-järjestelmän avulla hallinnoidaan käyttäjäidentiteettejä ja heidän pääsyjään eri resursseihin. Tämä tarkoittaa, että oikeilla käyttäjillä, laitteilla ja järjestelmillä on roolinsa mukaan tarvittavat pääsy oikeisiin resursseihin oikeaan aikaan. Identiteetin ja pääsynhallinnan järjestelmä auttaa yrityksiä hallinnoimaan esimerkiksi työntekijöidensä käyttövaltuuksia tietoturvalisella tavalla. Henkilö saa oikeudet vain niihin resursseihin ja järjestelmiin, joihin hänellä on roolinsa mukaan oikeus päästä. Kun työntekijän työsuhde päättyy, myös hänen käyttöoikeutensa yrityksen eri resursseihin päätetään. Näin työntekijälle ei voi työsuhteen päättyessä jäädä voimaan sellaisia käyttöoikeuksia, joita hänellä ei enää kuuluisi olla. (Varinder, 2021.)

Testattavan järjestelmän perusprosesseja ja toiminnallisuutta kuvataan seuraavaksi lyhyesti, jotta lukijan on myöhemmin helpompi ymmärtää, minkälaisia toiminteita ollaan testaamassa. Oletetaan, että Yritys AB:n käytössä on identiteetin- ja pääsynhallinnan järjestelmä. Yritykseen palkataan uusi työntekijä. Uuden työntekijän tiedot syötetään IAM-järjestelmään tai työntekijän tiedot voivat siirtyä järjestelmään esimerkiksi HR-järjestelmästä. Työntekijälle (käyttäjäidentiteetille) voidaan tämän jälkeen pyytää erilaisia käyttövaltuuksia yrityksen eri järjestelmiin ja resursseihin IAM-järjestelmän kautta. Käyttäjä saa esimerkiksi kirjautumisoikeudet yrityksen toimistoverkkoon ja erilaisiin työssään tarvitsemiinsa järjestelmiin.



Käyttöoikeuksia voidaan hakea käyttäjälle manuaalisesti esimerkiksi esimiehen toimesta. Käyttöoikeuksia voidaan hakea myös automatisoidusti, jolloin käyttäjä saa tietyt käyttöoikeudet perustuen esimerkiksi työrooliinsa.

Mikäli käyttäjän työsuhde päättyy, päätetään kaikki käyttöoikeudet käyttäjältä kaikista yrityksen resursseista kootusti.

Perus IAM-järjestelmään liitettynä on mahdollista käyttää myös erillistä työsuhteiden hallinnan järjestelmää tai sidosryhmien hallinnan järjestelmää. Sidosryhmien hallinnan järjestelmän avulla voidaan hallinnoida yrityksen ulkoisten työntekijöiden tai kumppaneiden käyttöoikeuksia. Käyttäjillä voi myös olla käytössä itsepalvelu, jonka kautta on mahdollista tehdä joitakin rajattuja tietojen päivityksiä omiin tietoihin sekä päivityspyyntöjä liittyen erityisesti EU:n tietosuojasetuksen, GDPR:n mukaisiin tietojen korjaus- tai poistopyyntöihin.

GDPR tulee sanoista General Data Protection Regulation (yleinen tietosuojasetus), joka on henkilötietojen käsittelyä sääntelevä laki, jota alettiin soveltaa kaikissa EU-maissa keväällä 2018. Tähän lakiin liittyen yksityishenkilöllä on oikeus muun muassa pyytää henkilötietojensa poistamista yrityksen järjestelmistä ja tietää mitä tietoja organisaatiolla on henkilöstä. (Tietosuojavaltuutetun toimisto 2018.)

IAM-järjestelmään kehitetään säännöllisesti uusia toiminnallisuuksia ja samalla korjataan vanhoja vikahavaintoja. Näitä päivityksiä kutsutaan nimellä Service Release (SR). Lisäksi järjestelmään tehdään HF (Hot Fix) -päivityksiä, joissa tehdään korjaus sellaisiin havaittuihin vikoihin, jotka eivät voi odottaa seuraavaa suurempaa SR-päivitystä.

Uuden tuoteversion kehitysten ja korjausten testaamisesta sekä hyväksyntätestauksesta vastaavat tuotekehityspuolen testaajat. Tuotekehityksen puolella osa regressiotesteistä on automatisoitu, eli tietyt regressiotestit ajetaan automaattisesti läpi jokaisen päivityksen yhteydessä. Osa testeistä tehdään manuaalisesti.

## 2.2 Keskeiset työtehtävät

Kun uuden tuoteversion hyväksyntätestaus on mennyt onnistuneesti läpi, aletaan uutta SR-päivitystä jollakin aikataululla päivittää eri asiakkaille. Tiimien esimiehet tekevät projektien aikataulutuksen ja resurssien etsimisen eri asiakas/päivitysprojekteille, mutta varsinaisia projektipäälliköitä pienillä päivitysprojekteilla ei yleensä ole.

Kun asiakasprojekteissa mukana oleva testaaja on saanut tietää uudesta jollekin asiakkaalle tulevasta päivitysprojektista, alkaa yleensä testianalyysivaihe. Testianalyysivaiheessa tarkennetaan vastaus kysymykseen ”mitä pitää testata?” (FISTB/ISTQB 2018, 17). Tässä vaiheessa käydään lävitse päivityksessä tehtävät kehitykset ja korjaukset ja arvioidaan mitkä niistä testataan asiakasversiossa. Lisäksi tarkistetaan asiakaskohtaiset toteutukset ja se, onko asiakkaalle tulossa asiakaskohtaisia korjauksia tai kehityksiä versionviennin yhteydessä sekä minkälaisia regressiotestejä versiopäivityksen yhteydessä tehdään.

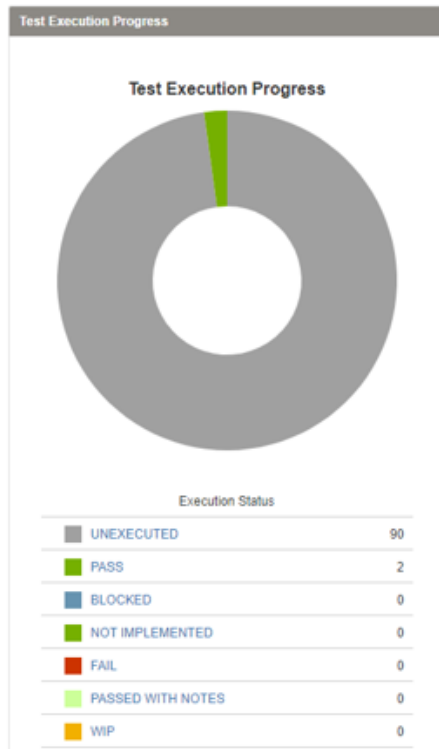
Testianalyysin jälkeen tai samaan aikaan tulee testien suunnitteluvaihe. Testien suunnitteluvaihe vastaa kysymykseen ”kuinka testata?” (FISTB/ISTQB 2018, 18). Testien suunnitteluvaiheessa suunnitellaan testitapaukset testianalyysin pohjalta. Testitapaukset jaetaan tavallisesti regressiotesteihin, asiakaskohtaisten toteutusten regressiotesteihin sekä kehityksiin ja korjauksiin. Mikäli asiakaskohtaisia kehityksiä on menossa versiopäivityksen yhteydessä, luetaan kehityksen tarkemmat määrittelyt ja tehdään vaatimuksia vastaavat testitapaukset testienhallintajärjestelmään. Tämän jälkeen testi linkitetään määrittelyyn/kehitystehtävään, jotta myöhempi seurattavuus kehityksen ja siihen liittyvän testauksen suhteen on helppo jäljittää.

ISTQB:n perustason testaajan sertifiointisäädösten mukaan jäljitettävyyden varmentaminen liittyy testien suunnitteluvaiheeseen, mutta sitä ylläpidetään koko testausprosessin lävitse. Jäljitettävyyden on hyvin tärkeää testauksen seurannan ja hallinnan kannalta. Jäljitettävyydellä tarkoitetaan sitä, että esimerkiksi vaatimusten, niihin liittyvien testitapausten ja vikojen (bugien) väliltä löytyy linkitys, jonka avulla on helppoa seurata, mikä testi on tehty mihinkin vaatimukseen

liittyen ja minkälaisia havaintoja testeistä on mahdollisesti tehty. (FISTB/ISTQB 2018, 22.) Ihan kaikissa regressiotestien tapauksissa tällaista jäljitettävyyttä testitapausten ja määritysten välillä ei tällä hetkellä ole toteutettu, mutta aina uusien kehitysten ja korjausten kohdalla kyllä. Mikäli versiopäivityksen yhteydessä viedään jokin asiakaskohtainen korjaus, tästä on olemassa vianhallinnanjärjestelmässä aiempi vikakirjaus, joka linkitetään versioon tulevaan testaustehtävään samoin kuin kehityksenkin osalta.

Tässä vaiheessa normaalisti myös luodaan valmiiksi asiakasversion päivitykselle oma Dashboard, jonka avulla voidaan (testauksen alettua) helposti seurata testauksen etenemistä. Tämä Dashboard liittyy testauksen seurannan ja hallinnan vaiheeseen. Tässä vaiheessa voidaan käyttää monenlaisia eri mittareita testauksen etenemisen arviointiin, kuten testauksen eteneminen verrattuna suunniteltuun aikatauluun, vikatiedot tai testitapausten suorittaminen.

(FISTB/ISTQB 2018, 63.) Dashboardilla testauksen seurantaan on tällä hetkellä käytetty lähinnä testitapausten suorittamiseen liittyvää, Pie Chart -tyyppistä piirakkakuviota testauksen etenemisestä. Tältä samalta Dashboardilta esimerkiksi toisten projektin jäsenten olisi helppoa seurata projektin etenemistä, mutta tähän saakka Dashboardia ei ole tähän tarkoitukseen juurikaan käytetty, vaan se on ollut testaajan oma työkalu. Dashboardilta löytyvää kuvaa (kuva 1) hyödynnetään myöhemmin myös testauksen loppuraportissa.



Kuva 1. Pie Chart testauksen etenemisestä.

Tavallisesti samaan aikaan testien analyysi- ja suunnitteluvaiheen kanssa on käynnissä myös testauksen valmisteluvaihe, jossa laaditaan ja viimeistellään testien suorituksessa tarvittava testimateriaali. Testien toteutus vastaa kysymykseen ”onko meillä kaikki valmiina testien suoritusta varten?” (FISTB/ISTQB 2018, 19). Suuri osa tämän vaiheen tehtävistä kuuluu yrityksen prosessissa järjestelmäasiantuntijalle, joka päivittää uuden ohjelmistopakettien asiakkaan testiympäristöön ja huolehtii testiympäristön valmiudesta testaukselle.

Kun testiympäristö on päivitetty järjestelmäasiantuntijan toimesta, alkaa varsinainen testien suoritusvaihe, jossa testataan ennakkoon suunnitellut testitapakset asiakkaan versiossa. Varsinaiseen testausvaiheeseen liittyy paitsi itse manuaalinen testaus, myös tehtyjen huomioiden kirjaus. Testit ja niiden stepit merkitään tehdyiksi testauksen seurantalijärjestelmään. Testi step tarkoittaa, että yksittäisellä testillä on monta suoritusvaihetta, ja yksi step on aina yksi suoritusvaihe. Esimerkiksi ensimmäinen step voisi olla ”kirjautu järjestelmään admin-tunnuksilla ja siirry paikkaan x”, joka kertoo millä roolilla testi suoritetaan ja mihin osioon järjestelmästä testi liittyy. Toinen step kertoisi mitä seuraavaksi tehdään ja mikä on toiminnon odotettu lopputulos, ja niin edelleen. Jokainen step

merkitään erikseen suoritetuksi ja lopuksi koko testin lopputulos merkitään riippuen siitä, miten eri steppien tulokset on merkitty. Jos esimerkiksi kaikkien steppien tulos on Pass, on koko testikin Pass, eli hyväksytty. Jos taas jollekin testin stepille on kirjattu tulokseksi Fail, on koko testi Fail, eli testi ei ole mennyt läpi hyväksytysti. Jos koko testin odotettu tulos ja varsinainen tulos vastaavat toisistaan, testi merkitään hyväksytyksi. Jos taas testi menee läpi, mutta siitä on kuitenkin kirjattu jokin huomioitava asia jollekin testin stepeistä tai yleisesti, testi merkitään Passed with notes. Joskus testi voi olla myös Blocked tilainen, joka tarkoittaa sitä, että jostakin syystä testiä ei voida suorittaa. Jos testin jonkin stepin tai koko testin odotettu tulos ja varsinainen tulos eroavat toisistaan, testi ei ole mennyt läpi, jolloin vikojenhallintajärjestelmään kirjataan testistä tehty huomio, ja tämä huomio linkitetään testitapauksen kanssa.

Mikäli asiakasversion testauksessa löytyy jokin vika, selvitetään tässä yhteydessä, onko vastaava vika myös tuoteversiossa vai onko tämä asiakaskohtaiseen versioon liittyvä vika. Toimenpiteet ovat erilaiset riippuen siitä, onko kyseessä tuotteeseen liittyvä vika vai asiakaskohtaiseen versioon liittyvä vika. Mikäli vika on ollut asiakaskohtainen ja se saadaan korjattua ohjelmistokehittäjän tai järjestelmäasiantuntijan toimesta, testaaja tekee korjauksen jälkeen vielä varmistustestauksen. Jos korjaus on auttanut, testitapaus voidaan merkitä läpäistyksi. Varmistustestaus tarkoittaa, että kun jokin testitapaus on tilassa FAIL, eli on löytynyt jokin vika, jonka vuoksi testitapaus ei ole mennyt läpi, tälle vialle täytyy tehdä jotakin. Kun järjestelmäasiantuntija tai ohjelmistokehittäjä on korjannut vian, testaajan tehtävänä on vielä suorittaa korjaukselle varmistustestaus ja suorittaa uudelleen kaikki ne testitapaukset, jotka liittyvät tähän vikaan eivätkä viimeksi suoritettaessa menneet lävitse. Tarkoituksena on varmistaa, että alkuperäinen vika on korjattu. (FISTB/ISTQB 2018, 36.) Voi myös käydä niin, että havainto on sellainen, jota ei korjata heti, vaan esimerkiksi seuraavan versiونيennin yhteydessä. Silloin testi jää Fail tilaiseksi ja kirjattu vika avoimeksi odottamaan korjausta. Tällaisessa tilanteessa tehtävä siirretään odottamaan korjausta esimerkiksi seuraavan versiopäivityksen yhteydessä.

Kun testaus on tehty loppuun, eli kaikki ennakkoon suunnitellut testitapaukset on suoritettu, siirrytään testauksen päätösvaiheeseen. Tähän vaiheeseen liittyy

sen tarkistaminen, onko kaikki havaintoraportit suljettu ja mitä tehdään niille havainnoille, joita ei ole vielä ratkaistu testauksen suorituksen loppuessa. Tähän vaiheeseen liittyy myös testausraporttien ja lokien valmistelu ja lähettäminen asiakkaalle. (FISTB/ISTQB 2018, 20.)

Testiraportille kirjoitetaan tieto siitä, mitä testaus on kattanut, tarkemmat testitapaukset ja niiden tilat ja selitteet. Sen lisäksi raportille liitetään kuvallinen Pie Chart testien suorituksesta. Raportille tulee myös kooste testimetriikoista, eli tässä tapauksessa taulukkomuotoinen info siitä, minkä verran testejä on missäkin tilassa ja minkä verran prosentteina se on kokonaistestien tehdystä määrästä. Sen lisäksi testiraportille kirjoitetaan lista avoimista havainnoista ja niiden luokittelu/vakavuusasteet. Viimeisenä testiraportilla on yhteenveto testien suorituksesta, jossa käydään lävitse, milloin ja missä testit on suoritettu, minkälaisia havaintoja testauksen aikana on tehty sekä loppupäätelmä testauksen lopputuloksesta, voidaanko siirtyä seuraavaan vaiheeseen, eli asiakkaan omaan hyväksyntätestaukseen.

Asiakkaille on myös toimitettu testauksen tarkempi testiloki. Testiloki saadaan Excel-muotoisena ulos testienhallintajärjestelmästä ja se sisältää testauksen tarkemmat suoritettut stepit ja kommentit. Testilokia muokataan niin, että sille jää näkyviin vain asiakkaalle olennaiset tiedot selkeästi luettavassa muodossa. Testiraportin ja testilokin kirjoittamisen jälkeen raportit toimitetaan asiakkaalle.

Useissa tapauksissa testaajan osuus versiopäivityksessä päättyy tähän. Joissakin tapauksissa asiakkaan omista hyväksyntätestauksista voi vielä tulla huomioita, jotka vaativat aina jonkinlaisen käsittelyn, läpikäynnin mitä huomioille tehdään ja koska. Ovatko huomiot sellaisia, että ne pitää korjata ennen tuotantovientiä vai voidaanko ne korjata esimerkiksi seuraavan versiopäivityksen yhteydessä. Mikäli korjauksia tehdään vielä samaan versiopäivitykseen, tulevat nämä korjaukset vielä testaajan testattavaksi.

## 2.3 Työssä tarvittavat taidot

Jotta testaaja voi tehdä työnsä hyvin, pitää olla tiedossa, miten testatun ohjelmiston tulisi käyttäytyä, mikä on sen suunniteltu toiminnallisuus. Hyvät vaatimusmäärittelyt ovat vankka pohja testaustyölle. Testaaminen on vaikeaa, jos ei tiedä mikä haluttu toiminnallisuus on, mutta aina tarkkoja vaatimusmäärittelyjä ei ole saatavilla.

Englannin taitaminen sujuvasti on monelle asialle edellytys: yleisesti ottaen alalla tietoa löytyy hyvin paljon enemmän englanniksi kuin suomeksi, ja joidenkin asiakkaiden kohdalla esimerkiksi testiraportteja kirjoitetaan englanniksi, joten etenkin hyvä kirjallinen englannin kielen taito on työssä hyödyksi. Testaajan työ sisältää monesti paljon selvittelytyötä, muun muassa erinäisten vaatimusmäärittelyjen ja käyttöoppaiden läpikäyntiä. Työssä pitää olla tarkka yksityiskohdista, mutta toisaalta hahmottaa myös kokonaisuuksia, mikä liittyy mihinkin ja millä tavoin. On myös hyvä olla kiinnostusta uuden oppimiseen, sillä uusia asioita tulee jatkuvasti vastaan. Pitää olla myös rohkeutta kysyä, jos jokin asia on itselle epäselvä.

Kirjallisuudesta hyvän testaajan ominaisuuksista löytyi seuraavia kuvauksia:

- Uteliaisuus, hyvät testaajat ovat uteliaita sen suhteen miksi järjestelmät käyttäytyvät kuten käyttäytyvät ja miten järjestelmät on rakennettu.
- Ammattimainen pessimismi, hyvät testaajat odottavat löytävänsä vikoja.
- Kriittinen silmä, hyvillä testaajilla on henkilökohtainen slogan ”jos epäilen, se on bugi”.
- Kyky huomioida yksityiskohtia, hyvät testaajat huomaavat pienimmätkin yksityiskohdat, esimerkiksi kosmeettiset ongelmat, mutta myös hienovaraiset vihjeet, jotka voivat vihjata vakavampaan ongelmaan.
- Kokemus, hyvät testaajat eivät vain tunnista vikaa, kun törmäävät sellaiseen, vaan myös tietävät mistä etsiä.
- Hyvät kommunikaatiotaidot, testaajan pitää pystyä kommunikoimaan löydökset niin, että toiset ymmärtävät mistä on kyse, ja myös niin, ettei viasta esimerkiksi syyllistetä ketään, eli neutraalilla, asiallisella tavalla. Testaajan ajatusmaailma on ajatella mikä voi mennä vikaan ja mitä puuttuu. (Graham ym. 2020, 31.)

## 2.4 Kehittymisen arviointi

Näkisin oman osaamiseni annetulla asteikolla olevan tällä hetkellä ”taitava suoriutuja”, sillä käsitykseni työtehtävästä on jo muutaman vuoden työkokemuksella syventynyt paljon ja suoriudun työtehtävistä itsenäisesti ja projektitiimissä oman tehtäväni hoitaen. Osaamisalue on kuitenkin itselläni toistaiseksi vielä melko suppea, kun alan ajatella sitä kaikkea mitä testaajana olisi hyvä osata ja mitä pitäisi vielä oppia.

Tällä hetkellä uusia asioita tulee esiin vielä kohtuullisen usein, eli saan oppia jotakin uutta lähes joka viikko, joinakin viikkoina joka päivä. Nämä asiat liittyvät joko itse testattuun järjestelmään, sen konfiguraatioihin ja toiminnallisuuksiin, tai sitten ihan testaajan työhön. Teknisen osaamisen kasvattaminen yleisesti on itselleni kehityskohde, eli syvempi ymmärrys siitä miten esimerkiksi data järjestelmissä ja niiden välillä liikkuu ja miten hyödyntää tietokantaa testauksen apuna. Myös Active Directory osaamisen kasvattaminen olisi hyvästä. Tuntuu lisäksi siltä, että testaajan työssä on mahdollisuuksia kehittyä lähes loputtomasti: ehkei koskaan voi olla täysin valmis. Testaajana on esimerkiksi mahdollista erikoistua johonkin tiettyyn osa-alueeseen, kuten käytettävyydestestaukseen tai tietoturva-testaukseen. Opittavaa testaajana on siis kaikkeen tähän verraten itselläni vielä hyvin paljon.

Tällä hetkellä syvennän itse vielä perustason tietoperustaa ja pohjaa opiskelemalla opinnäytetyön ohessa perustason ISTQB-sertifikaattisisältöä. Lisäksi haluan vielä tulevaisuudessa oppia (esimerkiksi) paremmaksi tietokantaosaajaksi, ja jos opinnäytetyöprosessin aikana tulee mahdollisuus syventää erilaisia teknisen osaamisen alueita varsinaisen testausosaamisen lisäksi, sitä parempi. Minua kiinnostaa myös testausautomaatio, jota en toistaiseksi ole päässyt omassa työssäni hyödyntämään, mutta jota ehkä tulevaisuudessa olisi myös kiinnostavaa oppia.

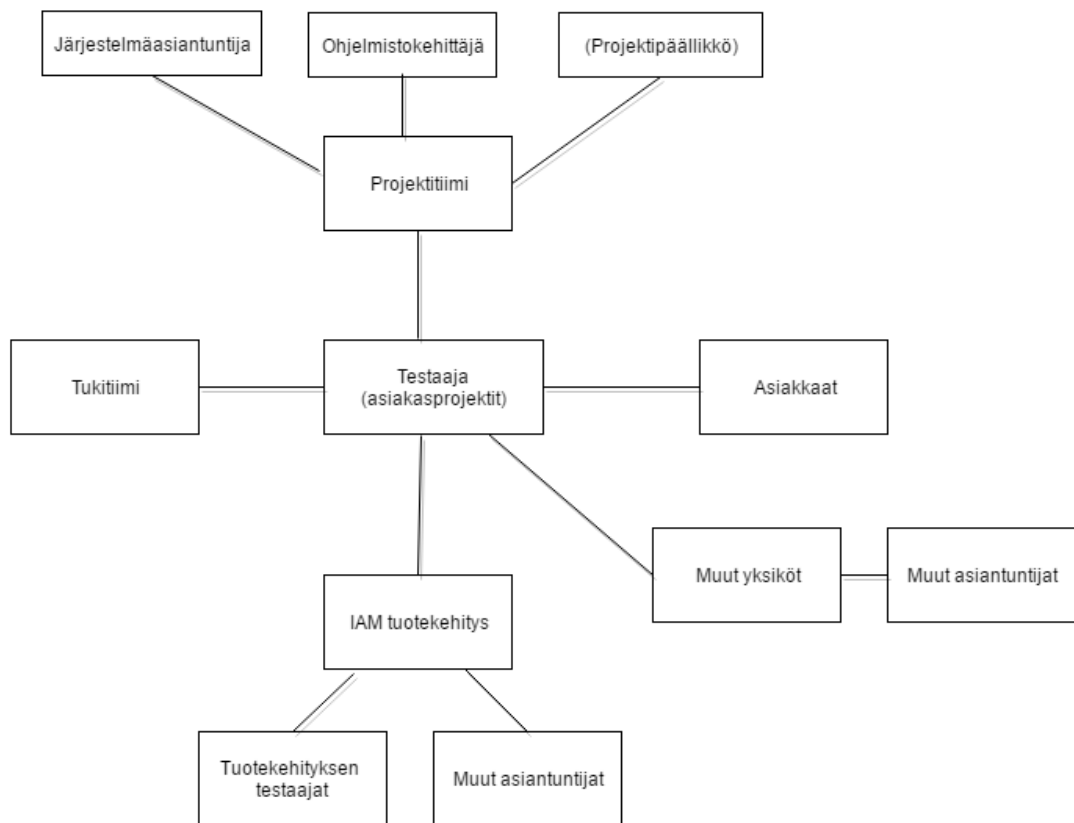


## 2.5 Sidosryhmät ja vuorovaikutus työpaikalla

### 2.5.1 Työ- ja toimintaympäristö ja sidosryhmät

Asiakasprojektien testaaja kuuluu aina tiettyyn projektitiimiin, joka muodostetaan eri asiakkaiden versiopäivitysten ajaksi. Projektitiimiin kuuluu yleensä testaajan lisäksi ainakin järjestelmäasiantuntija sekä ohjelmistokehittäjä. Pienemmissä päivitysprojekteissa ei ole erillistä projektipäällikköä, vaan tiimi hoitaa versiopäivityksen itsenäisesti alusta loppuun. Ohjelmistokehittäjä on vastuussa ohjelmistopakettien tekemisestä sekä (asiakaskohtaisten vikojen) koodiin liittyvien vikojen selvittämisestä ja korjauksesta. Järjestelmäasiantuntija on vastuussa muun muassa asiakkaan testiympäristön päivittämisestä uuteen versioon, konfiguraatioihin liittyvien vikojen selvittämisestä ja korjauksesta sekä version päivittämisestä tuotantoympäristöön. Testaaja on vastuussa kaikesta testaukseen liittyvästä, testien suunnittelusta, suorittamisesta ja tulosten raportoinnista. Tiimien esimiehet ovat niitä, jotka koordinoivat jollakin tasolla projektien aikataulutusta ja sitä, ketkä mihinkin projektiin osallistuvat. Asiakasprojektien testaaja on tekemisissä myös IAM-tuotekehityksen kanssa, asiakkaiden kanssa sekä toisinaan muiden asiantuntijoiden, kuten IAM- ja järjestelmä konsulttien kanssa. Tukitiimin kanssa kanssakäymistä on loppujen lopuksi aika vähän. Tukitiimi vastaa asiakkaiden kysymyksiin päivitysprojektien ulkopuolella, eli esimerkiksi tuotantoympäristössä ilmeneviin vikoihin ja ongelmiin liittyvistä selvityksistä.

Asiakasprojektien testaaja kuuluu projektitiimiin, ja muita sisäisiä sidosryhmiä ovat tukitiimi, tuotekehitys sekä muut asiantuntijat. Asiakkaat kuuluvat ulkoisiin sidosryhmiin (kuva 2).



Kuva 2. Sidosryhmät.

Asiakas on tietysti kaiken toiminnan keskiössä. Asiakastyytyväisyys on hyvin tärkeää ja se, että järjestelmä täyttää asiakkaiden sille asettamat odotukset. Siksi yhteistyö asiakkaiden kanssa on hyvin oleellista. Projektitiimi tekee yhteistyötä sen mahdollistamiseksi, että asiakkaan kokonaiskokemus versiopäivityksestä ja IAM tuotteesta ylipäätään olisi positiivinen. Myös tuotekehitys on oleellisessa asemassa tässä, sillä joskus asiakkaan versiosta voi löytyä jokin vika, joka ei ehkä toiselle asiakkaalle ole niin suuri, mutta kyseiselle asiakkaalle hyvinkin suuri. Jos vika on tuotteessa, tuotekehitykseltä saatetaan tarvita nopeastikin apua asian korjaamiseen. Tällöin korjaus saatetaan viedä etukäteen tietylle asiakkaalle, vaikka se olisi tulossa vasta myöhempään versioon tuotteessa.

## 2.5.2 Työ- ja toimintaympäristö, vuorovaikutus

Ohjelmistotestaajan työssäni asiakasprojekteissa olen eniten vuorovaikutuksessa järjestelmäasiantuntijan kanssa. Vuorovaikutus tapahtuu Teamsin kautta, joko kirjoittamalla tai joskus ottamalla puhelun, jos tarvitsee selvittää jotakin asiaa tarkemmin.

Olen myös tekemisissä asiakkaiden kanssa. Toisinaan saattaa olla asiakaspalavereita, joissa on paikalla koko projektiryhmä. Joskus olen palaverissa asiakkaan kanssa myös yksin, esimerkiksi liittyen pelkästään testauksen asioihin. Suurimmaksi osaksi kanssakäyminen myös asiakkaiden kanssa on kuitenkin ollut myös kirjallista, joko sähköpostitse tai Teamsin kautta.

Muita vuorovaikutustilanteita tulee esimerkiksi yhteisissä tiimipalavereissa, joissa käsitellään kaikille yhteisiä asioita ja esimerkiksi kokonaistilannetta projektien osalta. Suurin osa ihmisistä, joiden kanssa olen päivittäin tekemisissä, työskentelee eri paikkakunnilla kuin itse työskentelen, ja muutoinkin työskentelen tällä hetkellä kokonaan etänä, joten kaikki vuorovaikutus tapahtuu tällä hetkellä joko sähköpostin tai Teamsin välityksellä.

## 3 Päiväkirjatyöskentely

### 3.1 Viikko 1 (23.-27.8.2021)

#### **Maanantai 23.8.2021**

Päivän tavoitteena on aloittaa Asiakas A:n versiopäivityksen testaus uusimpaan SR-versioon. Testaukselle on normaalisti hyvä varata aikaa ainakin viikko myös pienempien versiopäivitysten osalta, koska aina on hyvä olla hieman varoaikaa. Erinäisistä syistä johtuen sain kuitenkin viime viikolla tietää, että testaukselle jääkin aikaa vain noin kaksi ja puoli päivää. Joudun mukauttamaan aiemmin tehtyä testaussuunnitelmaa käytettävissä olevaan aikatauluun nähden uusiksi. Tämän aikatauluhaasteen lisäksi etukäteen tiedossa olevia haasteita projektin

testauksen osalta on kaksi: Asiakas testaa uutta versiota samaan aikaan kuin minä, joka ei ole paras mahdollinen tilanne. Toiseksi hallinnointijärjestelmä, jossa tehdyt testisuunnitelmat säilytetään ja suoritetaan ja jossa on myös vianhallinta, on tämän päivän päivityksen alla. Tarvitsen tätä testauksen hallinnointijärjestelmää työssäni ihan koko ajan, joten on oleellista työni kannalta, että testit ovat jollakin tapaa luettavissa. Ensimmäinen päivän tehtävä onkin näin ollen varmistaa päivän työt ottamalla päivälle etukäteen suunnitellut testitapaukset ulos testienhallintajärjestelmästä, ennen päivityksen alkua.

Näistä ennakkoon tiedossa olevista haasteista huolimatta pyrin päivän aikana saamaan suoritetuksi versioon tulevien kehitysten ja korjausten testauksen ja asiakaskohtaisten toteutusten regressiotestauksen, jotka olin priorisoinut tehtäväksi ensin. Asiakkaalla on käytössä myös työsuhteiden hallinnan järjestelmä, joka myös tavallisissa olosuhteissa regressio testataan, mutta koska tähän järjestelmään ei nyt suoraan kohdistu muutoksia ja testausaikataulu on kiireinen, priorisoin testit näin.

Pääosan päivää testien hallinnointijärjestelmä ei ollut edes lukutilassa, joten onneksi aamulla tuli otettua päivän aikana tarvittavat testit ylös. Päivän töiden tekeminen oli tavallista hankalampaa, koska en päässyt etsimään testitapauksiin liittyviä asioita juuri lainkaan, joten päivälle ennakkoon asetetut testauksen tavoitteet eivät ihan täyttyneet. En pystynyt tarkistamaan kaikkia tarvitsemiani tietoja testien hallintajärjestelmästä.

Päivän aikana keskustelin järjestelmäasiantuntijan kanssa parista testaustehtävästä ja selvittelin asiakkaalle tehtyihin korjauksiin liittyviä asioita. Testaushavaintoja tein pari kappaletta, mutta niitäkään ei päässyt tänään kirjaamaan mihinkään eikä tarkistamaan onko vastaavanlaisia havaintoja aiemmin kirjattu. Näiden huomioiden tarkistus jää huomiseksi, jolloin toivottavasti testien hallinnointijärjestelmä on taas käytössä.

**Tiistai 24.8.2021**

Aion aloittaa aamun viemällä eilisen testauksen tulokset hallinnointijärjestelmään, joka oli eilen päivityksen alla. Tämän jälkeen jatkan Asiakas A:n versio-päivityksen testausta. Aamupäivällä on myös tiedossa yksi sisäinen palaveri asiakas B:n asiakaskohtaiseen toteutukseen liittyvästä projektista. Tämän päivän tavoite on saada asiakas A:n testaukset valmiiksi, jotta voin huomenna kirjoittaa ja lähettää asiakkaalle testauksen raportit. Versio on tarkoitus viedä tuotantoon jo torstaina.

Tänään selvitettäviä tapauksia oli useampia, joista toisessa oli hukassa ominaisuuden haluttu toiminnallisuus perusjärjestelmän ja työsopimusten hallinnan järjestelmän välillä. Etsin tietoa ominaisuuden määrittelyistä, käyttöoppaista, järjestelmäasiantuntijalta ja testauskollegalta. Oli helppoa selvittää miten tämän toiminnallisuuden kuuluisi toimia perusjärjestelmässä, mutta sen yhteys työsopimusjärjestelmään ja mikä (konfiguraatio, eräajo, vai koodi, määrittää milloin tiedot päivittyvät järjestelmästä toiseen), olikin jo ongelmallisempaa.

Lisäksi asiakkaalta tuli eräs huomio, joka aiheutti jatkotutkimuksia järjestelmäasiantuntijalle, minkä jälkeen havainto siirtyi tutkittavaksi kehittäjälle. Tämän muutaman päivän aikana on tullut selväksi, miksi ei ole hyvä ajatus, että asiakas testaa samaan aikaan kun omat testit ovat vielä menossa: Tietyt testauksessa tarvittavat toiminnallisuudet, kuten tarkempien lokien tarkistaminen, eivät aina ole mahdollisia rajoitettujen käyttäjämäärien vuoksi. Tämä hidastaa tekemistä, kun testaajia on useampia samaan aikaan. Toiseksi, kun asiakas testaa samaan aikaan, asiakas voi löytää itse sellaisia virheitä, jotka olisi ehkä löydetty ja keretty korjata ennen asiakkaan hyväksyntätestauksen alkua. Molemmat skenaariot kävivät tässä tapauksessa toteen.

Asiakas A:n versiosta löytyneen havainnon vuoksi tuotantoon päivitystä joudutaan siirtämään ja todennäköisesti tekemään kokonaan uusi paketti. Iltapäivästä tein erään havainnon käyttöoikeuksien hakuun liittyen. Testasin saman toiminnallisuuden standardiversiossa ja huomasin, että toiminnallisuus on

poikkeava asiakasversiossa, eikä sen kuuluisi toimia niin. Kirjasin tästä vian (bugin) ja laitoin vielä erikseen viestiä kehittäjälle, joka alkaa tutkia asiaa huomenna.

### **Keskiviikko 25.8.2021**

Koska eilen suunnitelmat Asiakas A:n version tuotantoon viennistä lykkääntyivät, tarkoituksena on tänään jatkaa eilisten havaintojen selvitystä. Huomasin eilen, että kun selvitettäviä asioita kasaantui päivän aikana useita, meni itsellä jossain vaiheessa asiat solmuun yrittäessäni selvittää kaikkea kerralla. Tämän aamun aloitan nyt niin että teen itselleni listan, selkeän etenemisjärjestyksen, jossa alan asioita selvittää yksi kerrallaan loppuun asti. Päivän tavoitteena on saada selvitettyksi keskeneräiset asiat ja selvittää tarkemmin tilanteet, joissa virheellinen toiminta ilmenee ja jatkaa testauksia.

Aamulla ihan ensimmäiseksi olin yhteydessä asiakkaan kanssa, koska tiettyjen testikäyttäjien tunnukset olivat päässeet vanhenemaan ja ne piti uusia, jotta pääsisin tekemään joitakin rooliperusteisia testejä. Sen jälkeen keskityin selvittämään käyttöoikeuksien voimaantumoon liittyvää vikatilannetta, joka toimi tuoteversiossa eri tavalla kuin asiakkaalla. Tähän liittyen toistin testitilanteen, jossa virhe oli eilen ilmennyt ja tarkastelin tapahtumalokia ja tarkempia lokitietoja tapahtumiin liittyen. Tämän jälkeen selvitin aiempaa virhetilannetta, joka liittyi eräiden tietojen välittymiseen järjestelmien välillä. Tietojen päivitys ei näyttänyt kulkevan oikein perusjärjestelmän ja työsuhteenhallinnan järjestelmän välillä. Tähän liittyen olin tehnyt selvitystä, kuuluuko tietojen välittymisessä järjestelmien välissä olla viivettä vai ei. Sain viimein tuotekehityksen kanssa selvitettyä, että viivettä ei kuulu olla ja erilaisten testitapaus skenaarioiden kautta selvisi, missä tilanteessa tilanne ilmenee ja missä ei – tässä testailin siis erilaisia testitapauksia tietojen välittymisessä, miten tilanne järjestelmässä näyttäytyy ja tilanne ratkesi. Näiden jälkeen pääsin jatkamaan testauksia.

Päivän päättyessä olen saanut etukäteen päivälle suunnitellut tehtävät tehtyä. Järjestelmällinen tapa keskittyä yhteen ongelmaan kerrallaan näytti toimivan hyvin. Opin tänään, miten erään prosessin tiedot välittyvät perusjärjestelmän ja

työsopimustenhallinnanjärjestelmän välillä. Selvitin asiaa toisen testaajan kanssa sekä tuotekehityksen ihmisen kanssa ja lopulta selvisi, mistä tilanne johtui. Kyseessä ei testitilanteessa ollut vika; toiminnallisuus oli testatussa testitilanteessa tämänhetkisen tarkoituksen mukainen, mutta se voisi toimia paremmin/automatisoidusti, joten tästä päätettiin kirjata kehitystehtävä tuotekehitykselle.

### **Torstai 26.8.2021**

Tänään olisi tarkoitus saada Asiakas A:n testaukset loppuun ja tarkistaa kehittäjältä, onko asiakkaan huomio jo selvitetty ja jos niin mitä sille ollaan tekemässä. Tarvitaanko mahdollisesti uutta pakettia ja milloin korjauksen pääsee testaamaan, jos tästä on vielä saatavissa lisätietoa.

Päivän päätteeksi päivälle asetetut tavoitteet on saavutettu ja Asiakas A:n versiopäivityksen testaus on toistaiseksi saatu päätökseen. Muiden testausten lisäksi selvittelin tänään vielä oikeuksien hakuun liittyvää ongelmaa, jonka selvityksessä on tällä viikolla ollut paljon työtä, liittyen lähinnä siihen, missä kaikissa tilanteissa vikatilanteita ilmenee. Tänään sain varmennettua yhden tilanteen, jossa vika toistuu ja päivitin kirjatulle vikakirjaukselle nämä tiedot sekä tiedotin vielä kehittäjää Teamsin kautta asiasta. En päässyt tarkastelemaan tarkempia lokitietoja vikaan liittyen, joten ne täytyy katsoa huomenna aamusta. Nyt odotetaan kahden huomion korjausta, kun ne on saatu korjattua ja vietyä testiympäristöön, testaan vielä korjaukset ennen testiraporttien kirjoittamista ja toimittamista asiakkaalle. Opin tänään myös lisää konfiguraatioista erään selvityksen myötä.

### **Perjantai 27.8.2021**

Aamun aluksi suunnittelin etsiväni eiliset lokitiedostot liittyen kirjattuun vikaan. Aion liittää lokitiedot aiemmin kirjatulle vikakirjaukselle helpottamaan kehittäjän selvitystyötä. Tänään olisi tarkoitus päästä myös aloittamaan Asiakas C:n versiopäivityksen testausta.

Päivän aluksi tein vielä Asiakas A:lla käyttöoikeuksien voimaantuloon liittyviä testejä, koska oli jäänyt vielä mietityttämään, ovatko kaikki tilanteet, joissa vika ilmenee, nyt huomioitu. Pitkin testausta olin havainnut tähän liittyen erilaisia virheitä, jotka näkyivät myös lokilla, joten varmistin vielä muutaman asian näihin liittyen.

Aamupäivästä tuli asiakkaalta myös uusi selvitettävä asia liittyen työsopimushallinnan näkyvyyksiin. Sitä varten tein testitapauksen testien hallinnointijärjestelmään ja aloitin testin tekemisen. Testin saa tehtyä loppuun vasta maanantaina, jotta testitapaus tulisi testatuksi realistisesti.

En päässyt tänään aloittamaan Asiakas C:n versiopäivityksen testausta, koska tunnukseni olivat ehtineet mennä vanhaksi. Laitoin tänään asiakkaalle viestiä tunnusten uusimisesta, joten maanantaiksi tunnusten pitäisi toivottavasti olla jälleen voimassa.

### **Viikkopäiväkirja seurantaviikko 1: 23.8.2021 – 27.8.2021**

Viikon aikana olen oppinut joitakin uusia asioita liittyen testattuun IAM-järjestelmään sekä siihen liittyvään erilliseen työsopimustenhallinnan järjestelmään. Tein melko paljon selvitystyötä tämän osalta ensinnäkin siitä, miten kyseisen toiminnallisuuden kuuluisi toimia. Ominaisuudesta löytyi kyllä määrittelyt ja käyttöoppaasta kuvaus, mutta nämä eivät kuvanneet tarkemmin kyseisen ominaisuuden yhteyttä työsopimusjärjestelmään ja miten se tällä hetkellä toimii testatussa tapauksessa. Niinpä kyselin ja selvittelin asiaa paljon kollegoiden kanssa. Tässä olisi ehkä päässyt nopeammin eteenpäin, kun olisi vain heti kysynyt tuotekehityksen ihmisiltä, koska heillä on tuotteeseen liittyen monesti tieto helpoiten/nopeimmin saatavilla. Tämän selvityksessä tein myös paljon erilaisia testitapauksia selvittääkseni minkälaisessa testitilanteessa ongelma tulee esiin. Sain sen lopulta selville ja tuotekehityksen kanssa selvitettyä, että tuotteen toiminnallisuus on tällä hetkellä tämä, jossa tietyssä käyttötapauksessa tietojen välittyminen toimii kyseisellä tavalla. Totesimme, että toteutus voisi olla erilainen, joten kirjasin tästä lopulta kehitystehtävän tuotekehitykselle.



Toinen tällä viikolla paljon selvitysaikaa vienyt asia oli asiakkaan versiosta löytynyt vika käyttöoikeuksien hakuun liittyen. Tässä meni paljon aikaa myös sen selvittämiseen, missä tilanteissa kyseinen ongelma ilmenee ja missä kohtaa toiminnallisuus toimii oikein. Tämän yhteydessä tein vertailevaa testausta tuoteversiossa. Lopulta havainnosta on kuitenkin tehty vikakirjaus, jossa on tarkemmat tiedot siitä missä yhteydessä ongelma ilmenee ja asia jää kehittäjän selvittäväksi, samoin kuin asiakkaan havainnoima ongelma. Kolmanneksi selvitteilin järjestelmäasiantuntijan kanssa erästä aiempaa (aiemman versiopäivityksen yhteydessä tehtyä) asiakkaan huomiota. Tämän selvityksen yhteydessä opin lisää konfiguraatioista ja mistä löytyy tiettyyn toiminnallisuuteen vaikuttava konfiguraatio. Neljänneksi selvitykseksi jäi työsopimushallinnan näkyvyyksiin liittyvä asia, joka tuli selvitettäväksi perjantaina. Tämän selvitys jäi vielä kesken, koska testitilannetta ei ollut mahdollista suorittaa loppuun samana päivänä, joten tämän selvitys jatkuu vielä maanantaina.

Huomioiden selvittäminen vaatii siis monesti paljon tiedonhakua eri paikoista, ja toisinaan myös kollegoiden konsultoimista ja toiseksi järjestelmän testausta erilaisilla testitilanteilla, jotta pääsee kiinni siihen missä kaikissa tilanteissa vika tarkemmin ilmenee. Lisäksi opin joitakin muutettavia/muistettavia asioita omaan työskentelyyni liittyen:

1. On tärkeää pitää järjestelmällisyys työssä, eikä lähteä selvittämään montaa asiaa yhtä aikaa, koska silloin asioiden selvitys vain viivästyy. On parempi pitää huomio (mahdollisuuksien mukaan) yhdessä asiassa kerrallaan, silloin fokus säilyy ja saa enemmän asioita aikaan samassa ajassa. Aina tämä fokuksen pitäminen ei erinäisistä syistä johtuen ole mahdollista, mutta jos siihen voi itse vaikuttaa, niin kannattaa toimia.
2. Ennakointi on tärkeää, se että ennen testausten alkua kaikki tarvittavat testitunnukset ovat voimassa ja myös, että jos on tiedossa etukäteen tiedossa olevia haasteita, pohtia jo etukäteen, miten ne ratkaistaan.

Tämän viikon omat huomiot testauksen parantamiseen liittyen liittyivät nyt siis lähinnä huomioon siitä, että pitää tehdä ennakoinnit tunnuksiin liittyen paremmin. Lisäksi jatkossa olisi hyvä, että testit suoritettaisiin aina ja joka tilanteessa niin, että ensin omat testaukset valmiiksi ja vasta sitten asiakkaan hyväksyntätestaukset. Näin toimimalla ainakin vähennetään riskiä siihen, että asiakas löytää vikoja oman hyväksyntätestauksensa aikana. Toki siinäkin tapauksessa, että omat testaukset on suoritettu ennen asiakkaan hyväksyntätestausta, on hyvin mahdollista, että asiakkaan hyväksyntätestauksen aikana löytyy vikoja, joita

itse ei ole huomannut. Tämän voi jo nähdä testauksen seitsemästä perusperiaatteesta periaatteista 1 ja 2.

Testauksen perusperiaate 1: ”Testaus osoittaa virheiden läsnäolon, ei niiden poissaoloa. Testauksella voidaan osoittaa, että vikoja on olemassa, mutta sillä ei voida osoittaa, että vikoja ei ole. Testaus pienentää todennäköisyyttä, että ohjelmistossa on vielä jäljellä löytämättömiä vikoja, mutta vaikka yhtään vikaa ei löydetä, se ei ole todiste virheettömyydestä.” Testauksen perusperiaate 2: ”Täydellinen testaus on mahdotonta. Kaiken testaaminen ei ole mahdollista lukuun ottamatta triviaaleja tapauksia”. (ISTQB 2018, 15.)

Näihin periaatteisiin liittyen, on mahdotonta testata kaikkea ja joskus voi olla tilanteita, joissa tietty virhe ilmenee esimerkiksi vain jollakin tietyllä selaimen versiolla, mitä ei olisi testauksen yhteydessä ollut kovin helppo saada kiinni. Toki aina kun asiakas löytää jonkun vian, mitä itse ei ole löytänyt, on hyvä testaajana miettiä sitä, miksi näin kävi. Olisinko voinut itse löytää tämän vian ennen asiakasta ja voinko jatkossa tehdä jotain toisin, jotta voisin löytää tämänkaltaisen vian ennen asiakasta.

### **3.2 Viikko 2 (30.8.2021-3.9.2021)**

#### **Maanantai 30.8.2021**

Tänään tarkoituksena olisi ensin testata Asiakas A:n viimeviikkoinen testitapaus loppuun ja sen jälkeen aloittaa Asiakas C:n versiopäivityksen testaus. Päivän aikana on myös tiedossa tunnin kestävä yksikkömme yhteinen palaveri.

Aamulla testasin Asiakas A:lla viimeviikkoisen testitapauksen loppuun. Sen jälkeen tein vielä uuden testitapauksen erilaisella lähtökohdalla ja tarkastan tämän testin tulokset huomenna, kun sopimukset ovat jälleen päättyneet. Asiakkaan havaitsemaan ongelmaan ei ole vielä tullut korjausta, joten en ole päässyt testaamaan tätä uudelleen.

Sen jälkeen aloitin Asiakas C:n versiopäivityksen testausta. Olen aiemmin yleensä tehnyt ensimmäisenä regressiotestit tietyssä järjestyksessä, mutta nyt totesin, että varmasti oikeastaan kaikissa tapauksissa on hyvä tehdä ensimmäisenä sellaiset testit, jotka ovat toiminnaltaan erityisen kriittisiä tai muutoin asiakkaalle tärkeitä, kuten asiakaskohtaiset toiminnallisuudet. Niinpä aloitin testauksen tällaisista erityisen kriittisistä regressiotesteistä sekä kehitysten ja korjausten testeistä. Tänään tällä asiakkaalla ei tullut mitään testaushuomioita, kaikki tehdyt testit menivät läpi. Tänään ei oikeastaan tullut mitään uutta.

### **Tiistai 31.8.2021**

Tänään jatkan Asiakas C:n versiopäivityksen testausta. Tänään on myös muutama palaveri, Asiakas B:n asiakaskohtaisen toteutuksen korjausten tilanteesta, ja palaveri esimiehen kanssa. Tänään on tavoitteena saada valmiiksi Asiakas C:n versiopäivityksen testaus.

Päivän aikana testaukset sujuivat jouhevasti asiakas C:n versiossa. Mitään havaintoja ei tullut. Asiakas B:n sisäisessä palaverissa sain tietää, että testaushavaintojen korjaukset ovat edenneet, mutta vielä ei ole testattavaa, koska kaikkia havaintoja ei vielä olla saatu korjattua tai korjauksia vietyä testiin.

Päivän aikana tuli myös yksi ennakoimaton korjaus testattavaksi Asiakas D:n osalta, jonka isomman versiopäivityksen olen testannut aiemmin ja josta oli vielä havaintoja korjaamatta. Tänään vietiin yksi korjaus testiin ja sain testattua korjauksen toimivaksi. Tämä oli korjaus, joka on korjattu tuotteeseen uusimpaan versioon, mutta asiakkaalle tämä korjaus vietiin erillisenä aiempaan versioon. Päivän päätteeksi olen saanut valmiiksi Asiakas C:n versiopäivityksen testauksen suunnitellusti.

### **Keskiviikko 1.9.2021**

Tänään tarkoituksena on kirjoittaa valmiiksi Asiakas C:n versiopäivityksen testauksen raportit sekä toimittaa ne asiakkaalle. Sen jälkeen käyn tarkistamassa Asiakas A:lle maanantaina tehdyn testin tulokset.

Aamupäivästä sain kirjoitettua Asiakas C:n testiraportit ja lähetettyä ne asiakkaalle. Sen jälkeen tarkistin Asiakas A:n testin tuloksen ja toiminnallisuutta selvittäessäni totesin, että se toimii kuten kuuluukin toimia. Tämän jälkeen oli pieni tauko, kun asiakasprojekteissa ei ollut meneillään mitään. Eilisessä keskustelussa tuli esimiehen kanssa ilmi, että olemme siirtymässä projektinhallinnan järjestelmässä uuteen ratkaisuun ja ennakkoon pitäisi testata miten nykyisenkaltaiset toiminnallisuudet on mahdollista toteuttaa uudessa ratkaisussa. Kerkesin tustua asiaan muutaman tunnin, ennen kuin sain tiedon, että Asiakas A:n korjaukset oli tehty. Tämän korjauksen ohessa vietiin myös yksi kehitys. Kun tämä uusi paketti oli päivitetty testiympäristöön, menin asiakasympäristöön suorittamaan varmistustestauksia. Kävi ilmi, että käyttöoikeuksiin liittyvä ongelma ei ollut korjaantunut ja tätä täytyy vielä jatko selvittää.

### **Torstai 2.9.2021**

Tänään tarkoituksena on testata Asiakas A:lle korjauspaketin yhteydessä viety asiakaskohtainen kehitys sekä asiakkaan tekemän huomion korjaus. Jos tänään tulee vielä korjaus käyttöoikeuksien hakuun liittyvään huomioon, saisin myös sen testattua, jolloin voisin aloittaa jo testiraporttien kirjoittamisen asiakkaalle. Tämä ei kuitenkaan ole varmaa, joten varasuunnitelmana on jatkaa omaa sisäistä testausta liittyen tulevaan projektinhallintajärjestelmän muutokseen.

Tänään aamupäivällä sain testattua asiakkaan tekemän huomion korjauksen toimivaksi. Testasin myös asiakkaalle viedyn kehityksen, joka liittyi työsuhtejärjestelmään lisättyyn uuteen kenttään ja tuon kentän tiedon välittymiseen export tiedostoon. Tämän testauksen yhteydessä huomasin, että osalta organisatioista puuttui vielä tuon uuden kentän sisältö käyttöliittymästä kokonaan ja sen

myötä opin uutta, kun kävin järjestelmäasiantuntijan kanssa lävitse, mistä nämä tiedot lisätään. Tavallisesti näitä tietoja hallinnoi asiakas itse, mutta tällä kertaa asiakkaan kanssa sovittiin, että me teemme tämän, jotta pääsemme testaamaan tuon toiminnallisuuden. Sisältöjen lisäyksen jälkeen pääsin testaamaan itse export tiedoston ja totesin tuon kehityksen toimivaksi. Linkitin testitapauksen kehityksen määrittämiseen, jotta voi nähdä, että kehitykselle on tehty testitapaus ja se on testattu.

Käyttöoikeuksien haun testauksen yhteydessä puolestaan korjaus alkuperäiseen ongelmaan tuli testattua toimivaksi. Kuitenkin tämän yhteydessä tuli uusi selvitys miksi yksi käyttäjä ei saanut käyttöoikeuksia. (Aiempi ongelma liittyi käyttöoikeuksien poistumiseen). Katsoin tilannetta lokilta, mutta koska mitään selvää virheilmoitusta ei sinne tilanteesta jäänyt, jäi asia kehittäjän selvitettäväksi. Arvelin, että ongelma saattaisi liittyä jotenkin siihen, että uusintatestissä käytettiin samoja testihenkilöitä ja juuri se testihenkilö, jonka käyttöoikeuksien poistuminen oli epäonnistunut aiemmin, ei saanut käyttöoikeuksia nyt. En kuitenkaan tiennyt tarkemmin, miten tämä skenaario olisi mahdollinen tai mikä tämän aiheuttaisi. Uusilla testikäyttäjillä testi kuitenkin onnistui, kun aiempi ongelma liittyi kaikkiin käyttäjiin.

### **Perjantai 3.9.2021**

Tänään toivoakseni kehittäjältä tulisi jokin vastaus siihen mikä aiheutti eilisen viikatilanteen käyttöoikeuksien lisäämisessä, jolloin saisin nämä korjausten testaukset tänään pakettiin ja laitettua asiakkaalle testauksen raportit ja lokin.

Aamusta kehittäjä laittoi tietoa, että eilisen käyttäjän käyttöoikeuksien voimaantulo oli epäonnistunut, koska aiemman vian vuoksi edellinen toimenpide oli jättänyt käyttäjän tietoihin eräänlaisen tilan, joka esti uusien käyttöoikeuksien voimaantulon. Kun tämä korjattiin testikäyttäjä sai oikeudet oikein.

Tänään sain Asiakas A:n testaukset päätökseen ja kirjoitin testiraportin sekä testilokin ja lähetin ne asiakkaalle. Nyt jäädään Asiakas A:n osalta vielä odottamaan, että he saavat omat hyväksyntätestaukset suoritettua loppuun ennen tuotantoon päivitystä.

## **Viikkopäiväkirja seurantaviikko 2: 30.8.2021 – 3.9.2021**

Tällä viikolla sain tehtyä Asiakas C:n versiopäivityksen testauksen valmiiksi sekä myös lähetettyä tästä asiakkaalle testiraportit. Asiakas C:n versiopäivityksessä ei tullut eteen mitään selvitettäviä asioita. Asiakkaalle oli juuri kesällä päivitetty aiempi versio ja nyt tämän uuden päivityksen yhteydessä uusia huomioita ei tullut. Asiakas kuittasi, että oli tehnyt myös oman hyväksyntätestauksen, eikä myöskään huomioita, joten tuotantoon päivitys päästään tämän asiakkaan osalta tekemään suunnitellusti.

Asiakas A:n versiopäivityksen yhteydessä on ollut paljon enemmän selvitettäviä asioita ja muutamassa selvityksessä eniten aikaa on vienyt sen selvittäminen, miten toiminnallisuuden ylipäättään kuuluisi toimia. Se ei ole aina ihan itsestään selvää, vaikka niin voisi olettaa. Tällä viikolla oli yksi sellainen selvitys kuten viime viikollakin. Tämän viikon selvitys liittyi erään käyttäjäroolin oikeuteen tarkastella tietoja. Huomio tuli asiakkaalta, joten tein tälle erillisen testitapauksen versioon, ja testasin eri käyttötapaukset. Toiminnallisuus toimii tällä hetkellä tarkoituksenmukaisesti, eli kyseessä ei ollut vika.

Lisäksi tällä viikolla testasin Asiakas A:lle versiopäivityksen yhteydessä viedyn kehityksen. Kun asiakkaalle viedään uutta kehitystä, projektinhallintajärjestelmään luodaan kehitystehtävä, jossa on määriteltynä mitä kehityksessä viedään. Testaajana teen tälle kehitykselle testien hallintajärjestelmään testin/testejä, ja linkitän tämän testitapauksen kehityksen kanssa. Tätä kutsutaan jäljitettävyydeksi (traceability). Tällöin testauksen pohjamateriaalin, testien ja mahdollisten testaushavaintojen väliltä löytyy linkki toisiinsa. ISTQB:n sertifikaattisisältö kertoo jäljitettävyyden tärkeydestä seuraavaa: "Tehokkaan testauksen seurannan ja hallinnan kannalta on tärkeää suunnitella jäljitettävyyden ja ylläpitää sitä koko testausprosessin läpi eri testauksen pohjamateriaalin osien ja erilaisten kyseisiin osiin liittyvien testauksen tuotosten välillä." (FISTB\ISTQB 2018, 22.)

Kaikissa testitapauksissa tällaista suoraa linkkiä vaatimuksiin ei kuitenkaan tällä hetkellä ole. Tällainen tilanne on erityisesti regressiotestien osalta, joissa

testataan jotakin järjestelmän perustoiminnallisuutta. Kuitenkin aina kun tulee uutta kehitystä, johon on kirjoitettu vaatimusmäärittelyt ja/tai käyttäjätarinat, tällainen linkitys tehdään. Samoin kuin linkitetään toisiinsa testitapaus ja siihen liittyvä mahdollinen testaushavainto.

Tällä viikolla testasin lisäksi Asiakas D:lle viedyn erillisen korjauksen, joka oli hyvin nopea testattava. Tälle ei ollut luotu asiakaskohtaista vikaa, mutta loin sellaisen nyt, koska tämä korjaus vietiin asiakkaalle asiakaskohtaisena tulevasta versiosta, eli tuotteeseen uusimpaan versioon tehty korjaus tehtiin asiakkaalle erillisenä aiempaan versioon, ei siis viety koko versiopäivityspakettia vaan yksittäinen korjaus siitä erillisenä. Tällöin jää jälki siitä asiakkaan julkaisulle, että tämä on erikseen viety ja testattu.

Kokoavana huomiona tästä viikosta, että tällä viikolla ei ollut niin paljon selvitettäviä asioita kuin viimeviikolla. Opin kuitenkin taas uutta työsopimustenhallinta-järjestelmän erilaisista toiminnallisuuksista.

### **3.3 Viikko 3 (6.9. -10.9.2021)**

#### **Maanantai 6.9.2021**

Tästä viikosta eteenpäin viikkopäiväkirjaa kirjoitetaan neljältä päivältä viikossa. Omista henkilökohtaisista syistäni olen sopinut työnantajan kanssa lyhennetystä työajasta, joten kirjoitan päiväkirjaa työskentelypäiviltä. Tänään ei ole ennakoon tiedossa asiakasprojektitöitä, joten päivän ennakkosuunnitelma on keskittyä testaamaan uuden projektienhallintatyökalun testausliitännäistä (pluginia). Tarkoituksena on testata, pystymmekö tekemään uudessa ratkaisussa samoja asioita kuin ennen ja jos ei, voiko näitä ennen tehtyjä asioita tehdä toisella tavalla, eli kiertää jotenkin. Tämän testauksen tarkoituksena on ennakoida tulevaa siirtymää ja kerryttää tietoa siitä, millä tavoin toteutus eroaa aiemmasta ratkaisusta, jotta kun siirtymä tapahtuu, kaikkien ei tarvitse enää selvittää samoja asioita uudelleen.

Aloitin päivän olemalla yhteydessä projektinhallintatyökalun hallinnoijaan, eli henkilöön, joka hoitaa kyseisen ohjelman hallinnointia, asennuksia, ylläpitoa ja konfiguraatioita ja hoitaa datamigraation (datan siirron vanhasta järjestelmästä uuteen). Tarvitsin testaukselle oman projektin, jota pystyn itse hallinnoimaan. Kun projekti saatiin luotua, aloitin testauksen. Tällä kertaa lähin tekniikka testaukselle voisi vastata kokempohjaisiin testitekniikoihin pohjautuvaa testausta, sillä mitään varsinaisia yksityiskohtaisia testitapauksia tälle testaukselle en ole erikseen tehnyt, vaan testit perustuvat sille millainen kokemus minulla on aiemmasta järjestelmästä ja millaisia toiminnallisuuksia sillä pystyi suorittamaan. Kirjoitin ylös erinäisiä tehtäviä ja kokonaisuuksia, joita pitäisi testata, mutta mitään yksityiskohtaisia testitapauksia en tehnyt.

Kun projekti oli luotu, opin miten projektien asetuksista pystyy lisäämään komponentteja tehtäville, koska näitä ei ollut valmiina. Komponentteja käytetään testauksen tehtävillä luokittelemaan mihin järjestelmän osa-alueeseen testi liittyy. Ensimmäiseen ongelmaan törmäsin aika pian. Yksittäisen uuden testin sai luotua, mutta testin liittäminen testausyykliin ei enää onnistunut. Testausyykli on testauksen kokonaisuus, joka liittyy esimerkiksi johonkin yksittäiseen asiakasprojektin versiopäivitykseen, ja joka koostuu kaikista versiopäivityksen testeistä. Kun yritin liittää yksittäistä testiä tällaiseen sykliin, tuli virheilmoitus "Not a test". Selvitin tätä järjestelmän pääkäyttäjän kanssa ja syykin selvisi pian. Vanhoista tiedoista oli tuotuna testauksen tyyppi, ja tätä testauksen liitännäistä asentaessa ei olisi saanut valmiina olla mitään toista tehtävää nimellä "testi". Tämä saatiin korjattua, kun ongelma selvisi. Tämän jälkeen löytyikin vielä joitakin eroavaisuuksia nykyiseen toteutukseen nähden, mutta näyttäisi siltä, että kaikki ovat konfiguroitavissa olevia asioita.

## **Tiistai 7.9.2021**

Tänään on tarkoituksena jatkaa projektinhallintatyökalun testausliitännäisen testausta.



Tänään sain itselleni admin oikeudet testiympäristöön ja pääsin itse selvittämään mikä konfiguraatio vaikuttaa minkäkin kentän näkyvyyteen, jotta pääsisin itse lisäämään puuttuvia tietoja. Sain myös hyvää tietoa siitä, minkälaisia lisäominaisuuksia nykyiseen verrattuna löytyy uudesta systeemistä ja näyttäisi, että ainakin jäljitettävyyteen liittyen olisi helpompi linkittää vaatimuksiin liittyvä tieto testeihin jatkossa. Tämä päivä meni siis pitkälti selvittäessä mistä konfiguraatiosta pääsee muokkaamaan asiat samalla tavalla kuin ne ovat olleet aiemmassa järjestelmässä. En kuitenkaan päässyt tekemään itse konfiguraatiomuutoksia, koska admin oikeudet eivät jostain syystä toimineet oikein ja tämä jäi selvitykseen. Tänään testailin esimerkiksi dashboardeja ja löysin muutamia mukavia uusia gadgeteja joita voi lisätä esimerkiksi tiettyyn projektiin liittyvien testaushavaintojen seuraamiseksi. Gadgetit ovat eräänlaisia dashboardille lisättäviä erillisiä kokoavia näyttöjä, jotka esittävät esimerkiksi kuvallisena tietona jonkin kootun kokonaisuuden, kuten minkälaisia vikoja tietystä versiosta on löytynyt tai minkä verran testitapauksista tietyssä versiopäivityksessä on suoritettuna ja missä tilassa ne ovat.

### **Keskiviikko 8.9.2021**

Tänään on tarkoitus edelleen jatkaa projektinhallintatyökalun testausta. Tälle päivälle tuli myös muita kuin testaustöitä, sain tehtäväkseni erään käännöstyön, ja siihen meni osa päivästä. Sain myös admin oikeudet vihdoinkin tänään kuntoon testausliitännäiseen liittyen ja pääsin loppupäivästä testaamaan hieinan konfiguraatioiden muutoksia, jotta voi kokeilla saadaanko kaikki ominaisuudet samanlaisiksi kuin aiemmin konfiguraatioiden avulla.

### **Perjantai 10.9.2021**

Päivälle ei ole ennakoon tiedossa asiakastöitä, joten tarkoituksena on jatkaa samaa työtä kuin aiemmin tällä viikolla.

Kun päivä alkoi, olin saanut sähköpostia, että Asiakas A:n asiakkaan hyväksyntätestauksesta on tullut huomio liittyen tiettyyn erityiseen tilanteeseen, jossa

työntekijän käyttöoikeudet eivät päättyneet oikein. Minulle oli tullut pyyntö testata tätä huomiota tuoteversiossa. Tuoteversiossa kyseinen tilanne päätti käyttöoikeudet oikein, joten menin asiakasversioon testaamaan samaa skenaariota. Huomio saattaisi hyvin liittyä aiempaan tilanteeseen, jossa oikeudet eivät päättyneet käyttäjiltä oikein, vaikka toiminnallisuus oli korjattu ja testattu toimivaksi. Juuri kyseistä tapahtumasarjaa ei kuitenkaan ollut testattu. Tämän huomion selvittelyyn ja testaamiseen kuluikin melkein koko päivä, koska testattua virhettä en saanut toistumaan, vaan oikeudet päättyivät kaikissa testatuissa tilanteissa oikein. Selvittelin vielä lokilta asiakkaan testikäyttäjän avulla tapahtumien kulkua, mutta täyttä selvyyttä ei tullut, sillä testikäyttäjän käyttöoikeudet näyttivät nyt olevan päätetyt ja lokin perusteella näyttäisi, että ne ovat päättyneet kuten piti. Voi kuitenkin olla, että samaa testikäyttäjää on käytetty toisissakin testeissä, joten en voinut mitenkään olla varma mihin testiin mikäkin tapahtumasarja liittyy. Pelkkien lokitietojen avulla on tämän vuoksi tosi hankalaa selvittää mitä tarkalleen on tapahtunut, joten asiakkaalle lähetettiin tästä vielä lisäkyselyä selvitystä varten. Vaikka käyttöoikeudet päättyisivät kaikissa omissa testitilanteissa oikein, on silti tärkeää selvittää syy sille, miksi jokin yksittäinen tilanne on päättynyt virheeseen, koska syy voi olla sellainen, että se voi toistua tietyssä tilanteessa uudelleen. Lisäksi kysymys on asiakkaan luottamuksesta järjestelmään.

### **Viikkopäiväkirja seurantaviikko 3: 6.9.2021 – 10.9.2021**

Tällä viikolla tein pääosin omia sisäisiä töitä, eli testasin sisäistä projektienhallintaa työkalua, jota käytetään projektien- ja testauksenhallintaan. Testauksen tarkoituksena oli tehdä ennakoivaa selvitystyötä siitä, miten uusi ratkaisu eroaa nykyisestä ja onko asioita, joita ei enää voi tehdä samoin kuin ennen ja jos niin miten ne voi jatkossa tehdä.

Tällä viikolla opin uutta käytössä olevasta projektienhallinnan työkalusta, ymmärsin esimerkiksi paremmin, miten voin jatkossa hyödyntää vielä paremmin dashboardeja, eli projektille luotavia tilastotyyppisiä näkymiä. Hiljaisempina aikoina osaamisen kehittymiseen on mahdollista käyttää aikaa, ja meillä on mahdollisuus hyödyntää erilaisia koulutusmahdollisuuksia. Kävinkin viikon aikana

muutamien kurssien, jotka liittyivät projektienhallintatyökaluun ja opin sieltä myös paljon uutta, mitä en ole ennen tiennyt. Opin miten kyseisen projektienhallintajärjestelmän kyselykieli toimii, ja miten tietynlaisia kyselyfilttereitä on mahdollista luoda, jolloin haettu tieto löytyy tehokkaammin kuin miten olen aiemmin sitä etsinyt. Kävin myös admin kurssia, koska en ole aiemmin ollut missään tekemisissä kyseisen järjestelmän konfiguraatioiden kanssa. Toki tietoa konfiguraatioiden hallintaan löytyy pääosin hyvin myös kyseisen järjestelmän dokumentaatiosta. En tule varmasti jatkossakaan hallinnoimaan kyseisen projektienhallintajärjestelmän tuoteversion konfiguraatioita, mutta nyt testausta varten sain itse mahdollisuuden päästä kokeilemaan mitä kaikkea testauksen liitännäiseen liittyvää konfiguraatiota järjestelmästä löytyy ja koen tämän itse hyvin hyödyllisenä ja mielenkiintoisena.

Selvityksiä viikon aikana tuli joitakin. Osaan sain ratkottua dokumentaation kautta. Osaan piti kysyä apua, esimerkiksi tilanteessa, kun minulle annetut admin oikeudet eivät toimineetkaan. Tähänkin löytyi lopulta syy ja sain admin oikeudet toimimaan. Yksi asia jäi vielä myös jatkoon selvitettäväksi, sillä en onnistunut löytämään erääseen toiminnallisuuteen liittyvää konfiguraatiota ja jäinkin etsimään tietoa onko kyseessä jonkinlainen vika toiminnallisuudessa, vai eikö tuota toiminnetta enää saada toimimaan kuin aiemmin, vai onko kyseessä konfiguroitava asia.

Perjantaina tuli selvitykseen Asiakas A:n hyväksyntätestauksessa ilmennyt virhe. Tämä meni mutkikkaaksi, koska virhettä ei saanutkaan toistettua. Tässä joutui tekemään monenlaisia eri testiskenaarioita ja lopulta lähteä jäljittämään tapahtumia asiakkaan testihenkilön yksittäisessä testitapauksessa. Tätä varten asiakkaalta kysyttiin tarkempia tietoja, ja asiakkaalta löytyikin hyvin dokumentoituja testihenkilö ja joitakin kuvia testitilanteesta. Lisäksi on lokitiedot. Tällä viikolla tämä mysteeri ei vielä kuitenkaan täysin selvinnyt, vaan vielä tarvitaan asiakkaalta lisätietoja sille, oliko kyseisen testihenkilön käyttöoikeudet päätetty muuta kautta, koska enää oikeudet eivät olleet voimassa. (Alkuperäinen ongelma oli, että käyttöoikeudet eivät olleet päättyneet).

Testauksen tyypillisiin tavoitteisiin kuuluu ISTQB:n sertifikaattisisällön mukaan seuraavia toimintoja:

1. arvioida eri tuotoksia, kuten vaatimuksia, käyttäjätarinoita, suunnittelukuvauksia ja koodia.
2. todentaa, että kaikki kuvatut vaatimukset on täytetty
3. kelpuuttaa testauksen kohteen valmius ja toimivuus käyttäjien ja muiden sidosryhmien odotusten mukaisesti
4. kasvattaa luottamusta testauksen kohteen laadun tasoon
5. ennaltaehkäistä vikoja
6. löytää häiriöitä ja vikoja
7. tuottaa sidosryhmille riittävästi tietoa, jotta nämä voivat tehdä valistuneita päätöksiä erityisesti testauksen kohteen laadun tason suhteen
8. vähentää riittämättömään ohjelmiston laatuun liittyvien riskien tasoa (esim. aikaisemmin havaitsemattomien häiriöiden esiintyminen tuotantokäytössä)
9. täyttää sopimukseen, lakeihin tai muuhun sääntelyyn perustuvat vaatimukset ja/tai todentaa, että testauksen kohde on tällaisten vaatimusten tai standardien mukainen. (ISTQB 2018, 12.)

Perjantaisen asiakkaan huomion selvittäminen liittyy mielestäni olennaisesti kohtaan neljä, eli testauksen tehtävänä on kasvattaa luottamusta testauksen kohteen laadun tasoon. Jos asiakkaalle jäisi epäselväksi, miksi jokin virhetilanne on esiintynyt, tämä voisi horjuttaa luottamusta. Vaikka virhettä ei saatu omassa testauksessa sellaisenaan enää toistumaan, on silti tärkeää selvittää syy alkuperäiselle virheelle. Voiko tilanne toistua joissakin olosuhteissa?

### **3.4 Viikko 4 (13.9. -17.9.2021)**

#### **Maanantai 13.9.2021**

Tänään on ennakkoon tiedossa jälleen sisäisiä töitä, hieman projektinhallinnan työkalun testausta ja koulutuksia.

Tänään katselin jäljitettävyyden raportin (traceability report) toimintaa testauksen hallintaan liittyen, jota en ole ennen hyödyntänyt, mutta jota voisi jatkossa mahdollisesti hyödyntää. Tämän tarkoituksena on seurata miten jäljitettävyyden vaatimusten, testitapausten ja testaushavaintojen välillä toteutuu. Lisäksi kävin koulutusta liittyen tutkivaan testaukseen.

**Tiistai 14.9.2021**

Tänään ei ole ennakkoon tiedossa olevia asiakastöitä, joten päivän tavoitteena on jatkaa omia sisäisiä testauksia ja tämän lisäksi etsiä jotakin sopivaa koulutusta.

Tänään tein vielä muutaman tunnin projektinhallinnan työkaluun liittyvää selvitystä, koska uudessa ratkaisussa ei ole täysin samanlaisia testauksen dashboardeja kuin aiemmin. Etsin kiertotietä tälle nykyiselle versiolle ja löysinkin sellaisen, vaikka järjestelmän dokumentaatioissa ei ollut varsinaisesti tätä vaihtoehtoa esitetty suoraan. Kokeilemalla eri variaatioita onnistuin saamaan samanlaisen kuvallisen tiedon testitapauksista. Lisäksi jatkoin tutkivan testauksen koulutusta ja katsoin muutaman webinaarin, jossa puhuttiin IAM-liiketoiminnasta. Tänään myös saatiin selvitettyä loppuun Asiakas A:n havainto, jota testasin viime viikolla.

**Keskiviikko 15.9.2021**

Ennakkotietojen mukaan tänään pitäisi tulla Asiakas B:lle korjauspaketti, jolloin pääsisin tekemään neljään testauksen aikana havaittuun testaushavaintoon uudelleentestausta. Tähän todennäköisesti menee ainakin puoli päivää.

Tänään tuli taas yksi uusi yllättävä projekti, testauksen, - tai ainakin testisuunnittelun -, tarve tulisi nopeasti. Viime kerrasta oppineena menin heti ensimmäisenä tarkistamaan tunnusteni voimassaolon asiakasympäristöön. Ne olivatkin vanhentuneet. Tänään oli myös asiakaspalaveri asiakkaan kanssa, johon menin mukaan tällä kertaa lähinnä kuuntelemaan, koska en ollut kerennyt perehtymään juuri lainkaan tähän projektiin ja mistä asiassa on kyse. Kyse ei ole tavallisesta versiopäivityksestä, vaan asiakkaan kahdesta erillisestä kehitysprojektista, johon liittyy suurempi kokonaisuus kuin vain IAM.

Tänään uudelleen testasin Asiakas B:lle tehtyjä korjauksia. Yhden korjauksen sain kuitattua suoraan lävitse korjattuna, muista heräsi lisäkysymyksiä, jotka täytyy selvittää tiimin kanssa.

## **Torstai 16.9.2021**

Tänään on tarkoitus jatkaa jonkin aikaa Asiakas B:n asiakaskohtaisen toteutuksen korjausten testausta ja selvitystä järjestelmäasiantuntijan ja kehittäjän kanssa. Lisäksi pitäisi selvittää eilen tulleen Asiakas F:n tulevien testauksen testien pohja, eli ensinnä mihin kaikkeen tehtyt muutokset vaikuttavat, toisin sanoen: mitä pitää testata.

Tänään aamulla katsoin ensimmäisenä Asiakas B:n eilen hieman kesken jääneitä testauksia loppuun. Kirjoitin aiemmin kirjattuun vikahavainto tehtävään havainnot ja kysymykset, joita sitten selvittelin kehittäjän ja järjestelmäasiantuntijan kanssa. Kaikkia asioita ei saatu tänään päätökseen. Sen lisäksi pyysin Asiakas F:n projektinvetäjää palaveriin, jotta saisin hieman lisäselvennystä siihen mitä näissä asiakkaan projekteissa ollaan tarkemmin tekemässä. Ensiviikolla minulla on asiakkaan kanssa palaveri testaukseen liittyen ja siihen mennessä pitäisi saada selvyyttä testitapauksiin. Kävin palaverin lisäksi lävitse projektista kirjoitettuja materiaaleja. Haasteena tässä projektissa on, että asiat muuttuvat usein, eikä mitään tarkkaa olla vielä lyöty lukkoon, joten testitapausten tarkempi suunnittelu tässä kohtaa ei oikeastaan ole edes mahdollista. Voin miettiä tässä kohtaa ainoastaan yleispätevästi mitä kaikkea pitää testata ja toisaalta tarvitsen asiakkaalta myös vielä lisätietoja tähän. Näiden lisäksi testasin Asiakas D:lle tehdyn korjauksen asiakkaan testissä. Korjaus oli toimiva.

## **Viikkopäiväkirja seurantaviikko 4: 13.9.2021 – 16.9.2021**

Alkuviikko oli tällä viikolla asiakasprojektien osalta hiljaisempaa, joten pyörittelin vielä hieman meidän omaa projektienhallinnan työkalua. Nyt oli myös aikaa tutustua raportteihin, joita ei ole aiemmin tullut käytettyä. Kävin myös muutamia koulutuksia.

Keskiviikkona pääsinkin sitten palaamaan asiakastestausten pariin, kun Asiakas B:lle tehtyt korjaukset saatiin vietyä testiin. Testit eivät aivan suoraan menneet läpi. Yhdessä oli vielä korjattavaa ja kolmen muun avoimen tehtävän osalta jäi kysymyksiä, joiden oikeasta toiminnallisuudesta en ollut varma, mutta joiden

toiminta mielestäni vaatimusmäärittelyjen pohjalta vaati vielä selvitystä. Näitä käytiin lävitse vielä kehittäjän ja järjestelmäasiantuntijan kanssa. Näiden testien osalta tein erään huomion kirjaamastani testaushavainnosta: Olin kirjannut ikään kuin kaksi samaan asiaan liittyvää vikaa yhdelle tehtävälle. Tätä ei pitäisi tehdä, vaan kaikki huomiot pitäisi kirjata omina tehtävinään. Tässä kyseisessä viassa oli kyse järjestelmän käyttäjille lähettämistä sähköposteista ja niiden linkeistä, jotka eivät toimineet. Huomioin kuitenkin, että toisen näistä luokittelu kuuluisi olla "Minor" ja toisen "Major". Vaikka viat ovat samantyyppisiä: käyttäjä ei saa sähköpostin linkkiä auki. Toisessa tapauksessa linkin avaamiselle on kuitenkin olemassa helppo kiertotie (workaround): käyttäjä voi mennä järjestelmään ja tehdä saman toiminnon helposti sitä kautta, linkki vain nopeuttaa siirtymistä oikeaan paikkaan. Toisessa tapauksessa taas helppoa workaroundia ei ole, vaan käyttäjä ei saa tarvitsemiaan tietoja auki. Siten, vaikka nämä kaksi testaushuomiota liittyvät tavallaan samantyyppiseen ongelmaan, ne ovat kuitenkin kaksi eri asiaan liittyvää sähköpostia ja vakavuusasteiltaan erilaiset, vaikka ne liittyvätkin samantyyppiseen asiaan korjauksineen. Jatkossa ehkä kirjaisin nämä silti erikseen.

Asiakas F:lle tuleva testaus vaatii jonkin verran perehtymistä koska kyseessä on testaus, johon liittyy paljon integraatioita asiakkaan muihin järjestelmiin, ja jossa tiedon siirtyminen pitää varmentaa asiakkaan järjestelmien, IAM:n sekä AD:n välillä. Todennäköisesti tätä testausta tehdään jollakin tavalla asiakkaan kanssa yhteistyössä, koska minulla ei ole pääsyä asiakkaan muihin järjestelmiin. Testauksen pohjamateriaalina minulla on tällä hetkellä asiakkaan toimittamat tietovirtakaaviot (data flow diagram), käyttötapaukset sekä alkumäärittelyt.

Tässä testauksessa on siis kyse integraatiotestauksesta. Integraatiotestaus on testaustasoista se taso, joka keskittyy "komponenttien tai järjestelmien väliseen vuorovaikutukseen." Testaus keskittyy siis ensisijaisesti järjestelmien väliseen viestintään, ei mihinkään järjestelmistä yksittäisesti. (ISTQB 2018, 28.)

### **3.5 Viikko 5 (20.9. -24.9.2021)**

#### **Maanantai 20.9.2021**

Tänään on tarkoitus tutustua Asiakas F:n toimittamiin materiaaleihin testitapausten suunnittelua varten. Kirjoitan yleisluontoiset testitapaukset testienhallintajärjestelmään.

Tämä päivä meni aika pitkälti testitapausten kirjoittamiseen ja sen hahmottamiseen, miten data liikkuu eri järjestelmien välillä ja mikä data siirtyy mihinkin. Tutustuin asiakkaan toimittamiin materiaaleihin ja määrittäisiin ja niiden pohjalta aloin kirjoittaa alustavia testitapauksia testienhallintajärjestelmään. Testitapaukset tarkentuvat datan suhteen myöhemmin.

#### **Tiistai 21.9.2021**

Tänään olisi tarkoitus tehdä uusintatarkastuksia Asiakas B:lle, sillä eilen tuli muutamia korjauksia tälle asiakkaalle testaukseen.

Korjatut testitapaukset menivät tällä kertaa lävitse hyväksytysti. Tällä hetkellä avoimia selvitettäviä asioita ennen mahdollista asiakkaan omaa hyväksyntätarkastusta ja tuotantoon vientiä on kuitenkin vielä kolme.

#### **Keskiviikko 22.9.2021**

Tänään ei ole ennakoon tiedossa olevia testauksia, joten päivän alussa suunnitelmassa on käyttää aikaa osaamisen kehittämiseen koulutuksilla.

Tästä päivästä tuli lopulta melko silppuinen, joitakin erillisiä palavereja, Asiakas B:n testauksiin liittyviä selvityksiä, jotka liittyivät käyttäjien AD-tunnusten saamiseen ja sen prosessiin. Sen lisäksi kävin koulutuksia, jotka liittyivät IAM-käyttövaltuushallintaan ja AD:hen. Iltapäivällä sain vielä uusia käännöstöitä, joitakin asiakkaan ohjeita piti kääntää englanniksi.



## **Torstai 23.9.2021**

Tänään on tarkoituksena jatkaa eilisiä käännöstöitä ja sen lisäksi katsoa ennen iltapäivän asiakaspalaveria Asiakas F:n kanssa mitä kysymyksiä on vielä asiakkaalta selvennettävänä ennen kuin testitapauksia voi tehdä valmiimmaksi.

Aamupäivästä käänsin Asiakas G:n tiettyjä ohjemateriaaleja englanniksi. Iltapäivällä oli asiakaspalaveri Asiakas F:n kanssa liittyen käynnissä olevaan projektiin. Palaverissa keskusteltiin esimerkiksi kuinka tuleva integraatiotestaus toteutetaan, koska se täytyy toteuttaa asiakkaan ja meidän yhteistyönä. Lisäksi keskusteltiin minkälaisia raportteja ja todennuksia asiakkaalla tarvitaan testauksesta ja sovittiin aikataulusta, jolloin toimitan asiakkaalle ennakkoon suunnitellut testitapaukset.

## **Viikkopäiväkirja seurantaviikko 5: 20.9.2021 – 23.9.2021**

Tämän viikon testaukset liittyivät Asiakas B:n asiakaskohtaisen toteutuksen korjausten testaamiseen. Lisäksi tällä viikolla oli jonkin verran testaussuunnitteluun liittyvää työtä.

Tällä viikolla olen oppinut paljon uutta. En ole ollut aiemmin mukana tekemässä projektia, joka pohjautuu pääosin integraatioiden testaukseen. Tällaisen testisuunnitelman tekeminen on siis ollut itselleni uuden oppimista, sillä tavallisesti teen testausta, joka tehdään puhtaasti IAM-järjestelmässä. Tällainen integraatiotestaus, jossa erityisesti keskitytään datan liikkumiseen eri järjestelmien välillä, on ollut uutta ja mielenkiintoista, ja vaatinut paljon eri dokumentteihin tutustumista ja myös termien selvitystä.

AD noin yleisesti ottaen on kyllä tuttu, koska olen sitä jonkin verran käyttänyt testaustyössä. Tavallisesti olen IAM perustestausta tehdessäni tarkistanut AD:lta sen, että käyttäjän AD tunnus näkyy AD:lla voimassa olevana ja että käyttöoikeuden päättyessä käyttäjän tunnus poistetaan oikein, eli kovin syvälistä AD osaamista itselläni ei vielä ole. Näiden projektien kautta uskon osaamiseni syvenevän. Selvitin termejä, jotka ovat aiemmin olleet vielä vähän

epäselviä. Esimerkiksi termit AD, DC, LDAP, ja mikä on ero sAMAccountin ja UPN:n välillä, kun molemmat liittyvät jotenkin kirjautumistunnukseen. Seuraavassa hieman määrittelyistä, joita opin hakiessani tietoa.

**Active Directory (AD)** on hakemistopalvelu (directory service) joka on saatavilla Windows Server alustalle. Se tallentaa tietoja keskitettyyn tietokantaan ja antaa käyttäjille yksilöllisen käyttäjätilin tietoverkkoon. Tätä käyttäjätiliä kutsutaan domain käyttäjä tiliksi (domain user account). (Jobskillshare Community. 2019.)

**Domain Controller (DC)** on palvelin, joka vastaa autentikointi pyynnöistä Windows Server domainin sisällä. Domain controller säilyttää käyttäjien kirjautumistietoja ja autentikoi käyttäjät. Eli kun käyttäjä kirjautuu tiettyyn domainiin, DC autentikoi ja validoi käyttäjän tunnukset ja sallii tai kieltää pääsyn. (Techopedia. 2020.)

**Lightweight Directory Access Protocol (LDAP)** on protokolla, jota käytetään hakemistopalvelun palvelimelle menemiseen ja tiedonhakuun. (Microsoft. 2018.)

**userPrincipalName (UPN)** on käyttäjän kirjautumistunnus sähköpostimuodossa. UPN:n pitäisi täsmätä käyttäjän sähköpostiin, mutta näin ei välttämättä aina ole. UPN muodostuu UPN prefixistä (etuliitteestä) ja suffixista (DNS domainin nimi). Prefix ja suffix liitetään yhteen @ symbolilla. Esim. [some-one@example.com](#). UPN:n pitää olla yksilöllinen. (Microsoft. 2018.)

**Security Account Manager (SAM)** Windows säilyttää ja hallinnoi paikallisia käyttäjiä (local user) ja ryhmätilejä tietokannassa, jota kutsutaan SecurityAccountManageriksi (SAM). SAM autentikoi eli tunnistaa paikalliset käyttäjäkirjautumiset. (Microsoft. 2018.)

Ero paikallisen käyttäjän (local user) ja domain käyttäjän (domain user) välillä on se, että domain userin tiedot säilytetään AD:lla ja paikallisen käyttäjän tiedot tietyllä tietokoneella. Esimerkiksi Local user tilillä voi kirjautua ilman internet

yhteyttä, kun taas Domain User Account vaatii internet yhteyden client koneen ja domain serverin välillä, jotta kirjautuminen voidaan tehdä. (Jobskillshare Community. 2021.)

**sAMAccountName (SAM-Account)** on myöskin kirjautumisnimi, lyhyemmässä muodossa, jota käytetään kirjautumiseen vanhemmissa kuin Windows 2000 kirjautumisissa. (Microsoft.2018)

Projektiin liittyen tuli siis jonkin verran termien opiskelua ensin, että ymmärsi kunnolla mitä ollaan tekemässä (jotta voi vastata testianalysivaiheen kysymyseen: mitä pitää testata).

Kun tehdään integraatiotestausta asiakkaan ympäristössä, en voi tehdä kaikkea testausta itse, koska minulla ei ole pääsyä asiakkaan muihin järjestelmiin. Testaus täytyy siis toteuttaa yhteistyönä asiakkaan kanssa, jossa asiakas tekee testauksista sen osuuden, joka liittyy niihin järjestelmiin, joista data varsinaisesti siirtyy IAM-järjestelmään. Oma osuuteni testauksessa liittyy siihen, miten data sitten siirtyy IAM-järjestelmään ja AD:lle. Tähän liittyen pitää selvittää, mitä tietoja järjestelmistä siirtyy ja missä tapauksessa. Asiakkaalla oli jo valmiiksi mietittynä omia käyttötapauksia, jotka tässä tapauksessa helpottivat omaa testisuunnittelua. Asiakkaan prosessien ymmärtäminen on tärkeää testisuunnitelman teossa, ja nämä osittain valmiit käyttötapaukset auttoivat alustavan testisuunnitelman teossa. Jatkan kuitenkin testisuunnitelman tekoa vielä ensiviikolla.

### **3.6 Viikko 6 (27.9. -30.9.2021)**

#### **Maanantai 27.9.2021**

Tänään on etukäteen tiedossa muutama sisäinen palaveri, ja loppupäivä todennäköisesti menee testitapausten suunnitteluun ja kirjoittamiseen Asiakas F:n integraatiotestaukseen liittyen.

Tänään suunnittelin ja kirjoitin testitapauksia Asiakas F:n tulevaa testausta varten. Tämä on jonkin verran haasteellisempaa kuin tavallisen versiopäivityksen testauksen suunnittelu, etenkin kun samassa yhteydessä tehdään kahta eri projektia ikään kuin yhtä aikaa, ja koska en ole ollut mukana alusta asti, on kestänyt jonkin aikaa hahmottaa eri materiaalien perusteella mitä kaikkea pitää ottaa testauksessa huomioon. Lisäksi toki vaikuttaa myös se, että tämänkaltaiseen projektiin en ole aiemmin osallistunut, eli on tullut yleisestikin paljon uutta. Olen tehnyt testianalyysi vaihetta ja testisuunnitteluvaihetta päällekkäin ja täydentänyt testitapauksia sitä mukaa kun lisätietoa on kertynyt.

### **Tiistai 28.9.2021**

Tänään on tarkoitus tarkistaa Asiakas B:n korjausten tilanne ja jos saan kirjautumistunnukset erilaisille testikäyttäjille voimaan, tarkoitus on myös suorittaa rooliperustaisia testejä, joita en ole pystynyt aiemmin tekemään. Eri testikäyttäjiä tarvitaan sellaisten testien todentamiseen, jolla tarkistetaan rooliperusteisesti käyttäjien oikeuksia nähdä tietoja/suorittaa toimintoja. Tarkoitus on myös jatkaa testisuunnitelmaa Asiakas F:n kahdelle eri projektille, jotta saisin huomiseen mennessä asiakkaalle laitettua alustavaa testisuunnitelmaa.

Tänään sain testattua Asiakas B:llä aiotut testit, ja testeistä heräsi jatkoselvityksiä jälleen. Määrityksissä oli mainittu, että kaikki tietyn organisaation tietoja tarkastelemaan pääsevät henkilöt saavat nähdä tietyn tiedon, mutta vain admin käyttäjä saa tehdä muutoksia tähän tietoon. Toiminne toimi periaatteessa oikein, sillä peruskäyttäjä ei päässyt tekemään muutosta, mutta peruskäyttäjälle oli kuitenkin näkyvissä painike, jolla muutosta voi yrittää tehdä. Painikkeen painamisesta ei vain tapahtunut käyttäjälle näkyvästi mitään, eli muutosta ei suoritettu. Tämän painikkeen ei kuitenkaan tulisi olla näkyvissä käyttäjälle, joka ei pysty muutosta suorittamaan, joten tämä joudutaan vielä korjaamaan.

Asiakas F:n testisuunnittelu eteni tänään hyvin, sain muutaman asian varmennettua järjestelmäasiantuntijalta, joten huomenna pitäisi saada testitapaukset viimeistelyä. Testianalyysi/suunnitteluvaiheeseen on mennyt melko paljon aikaa näiden kahden projektin osalta.

## **Keskiviikko 29.9.2021**

Tänään on tarkoitus saada Asiakas F:n alustava testisuunnitelma valmiiksi, eli testianalyysi + testisuunnittelun tuloksena aikaansaadut testitapaukset lähetetään asiakkaalle. Tämä tarkoittaa, että otan testitapaukset ulos testienhallintajärjestelmästä Exceliin, ja muokkaan Excelin sellaiseen muotoon, jossa se voidaan lähettää asiakkaalle mahdollisimman selkeässä muodossa.

Tänään selvittelin järjestelmäasiantuntijan ja IAM konsultin kanssa muutamia vielä epäselviä asioita toteutettavasta muutoksesta ja sen jälkeen täydensin vielä testitapauksia ja muokkasin testitapaus Excelin asiakasta varten ja lähetin nämä eteenpäin asiakkaalle katsottavaksi. Tavalliseen versiopäivitykseen liittyen testitapauksia ei olla normaalisti lähetetty asiakkaalle etukäteen katsottavaksi, mutta on tiettyjä asiakkaita ja projekteja, joissa näin toimitaan.

## **Torstai 30.9.2021**

Tänään on ennakkoon tiedossa vain yksi asiakastestauksiin liittyvä palaveri, jossa on tarkoitus käydä lävitse Asiakas B:n avoimia havaintoja ja mitä niille tehdään. Tämä palaveri on erittäin tarpeellinen tässä kohtaa, jotta saadaan yhdessä selvitettyä, miten avoimien havaintojen kanssa edetään.

Tällä viikolla tuli ohjelmistokehittäjän kanssa puheeksi, että testiautomaatiota voisi ehkä jatkossa hyödyntää asiakkaille asiakaskohtaisten toteutusten regressiotestauksessa ja tänään tarkoituksenani onkin tutustua alustavasti käytettyyn testiautomaatio-ohjelmaan. Testiautomaatio kiinnostaa itseäni, mutta en ole vielä toistaiseksi tehnyt sitä yhtään, ja niinpä tietämispohja on tässä kohtaa aiheesta melko pientä, jollei olematonta.

Tänään asiakaspalaverissa käytiin lävitse avoimia asioita Asiakas B:n asiakaskohtaiseen toteutukseen liittyen. Kävimme yhdessä lävitse mm. koodia ja konfiguraatioita liittyen havaintoihin ja näiden perusteella päätettiin, miten korjauksia

lähdetään toteuttamaan. Yhden havainnon osalta tarvittiin vielä lisätietoa ongelman juurisyystä, ja tätä varten lokituksen lisääminen on tarpeellista. Pääsen testaamaan näitä korjauksia mahdollisesti maanantaina.

Muuten tämä päivä meni Robot Frameworkin opiskeluun, aloitin erään kurssin tähän liittyen ajatuksena, että jos jossakin vaiheessa pääsisin tekemään testiautomaatiota asiakaskohtaisuuksien regressiotesteille, olisi jotakin pohjaosaaamista/tietämystä ennen kuin lähtisi toteuttamaan testejä oikeasti.

### **Viikkopäiväkirja seurantaviikko 6: 27.9.2021 – 1.10.2021**

Alkuviikko meni melko tiiviisti Asiakas F:n projektien testianalyysin/suunnittelun puitteissa. Torstaina pääsin tutustumaan hyvin mielenkiintoiseen, itselleni uuteen aiheeseen, nimittäin testiautomaatioon. Perustuotteen osalta testiautomaatiota on jo hyödynnetty regressiotestaukseen, mutta asiakkailla testiautomaatiota ei olla vielä tehty. Tällä viikolla tulikin puheeksi olisiko asiakaskohtaisten toteutusten regressiotestausta mahdollista tehdä testiautomaatiolla. Tässä vaiheessa itselläni ei ole mitään käsitystä, miten tämä teknisesti toteutettaisiin, eikä itselläni ole muutenkaan juurikaan kokemusta testiautomaation toteuttamisesta, saati siihen käytettävistä ohjelmista. Niinpä kun torstaina ei ollut mitään asiakastestauksiin liittyvää projektia juuri menossa, päätin hyödyntää tämän ajan tutustumalla aiheeseen. Ensin etsin tietoa siitä, miten testiautomaatiota on hyödynnetty tuotekehityksen puolella ja millaisia ohjelmia tähän liittyy. Tiesin jo ennestään, että Jenkins liittyy tähän jotenkin, sillä olin aiemminkin käynyt katsomassa automatisoituja testituloksia Jenkinsissä, mutta en oikeastaan tiennyt yhtään enempää siitä, mikä Jenkins tässä yhteydessä tarkemmin on, eli mitä muuta sillä voidaan tehdä, kun tarkastella tehtyjen automaattitestien tuloksia.

Tämän yhteydessä tulikin taas paljon uusia selvitettäviä termejä liittyen ohjelmiin/frameworkeihin, kirjastoihin ja varsinaiseen hallinnointiin käytettäviin ohjelmiin. Tämä onkin monesti uusiin ohjelmiin ja asioihin liittyvissä selvityksissä haaste: Kun lähtee tekemään jotakin, asioihin liittyy monia ehkä tuntemattomienkin termejä, ja ennen kuin pääsee syvälle siihen mitä on tekemässä, pitää selvittää tiensä erinäisten termien viidakon lävitse. Mikä on Robot Framework,

mikä on Selenium, miten Jenkins liittyy näihin, miten CI/CD putki liittyy Jenkin-siin, mitä CI/CD putkella itseasiassa tarkoitetaan tässä yhteydessä? Millä ohjelmalla konkreettisesti tehdään automatisoituja testejä?

Robot Framework on avoimen lähdekoodin automaatiokehys (automation framework), jota voidaan käyttää testiautomaatioon ja RPA:han. Selenium puolestaan on Robot frameworkin päällä toimiva kirjasto, jota käytetään Web testaamiseen. (Robot Framework Foundation. 2021.) Itse testiautomaatiota voidaan tehdä monella IDE:llä (Integrated Development Environment), joka tarjoaa ympäristön ohjelmistokehitykseen. Tässä tapauksessa käytetty IDE on se ohjelma, jolla testiautomaatiota konkreettisesti tehdään, eli millä automatisoitavat testitapaukset kirjoitetaan. Jenkins puolestaan on ohjelma, jossa tehtyjä testejä voidaan ajoitetusti ajaa ja tuloksia tarkastella.

Tällä viikolla en kuitenkaan ehtinyt tutustua Jenkinsiin vielä tämän syvemmin, kuin mitä jo aiemmin siitä tiesin. Aion ensin käydä loppuun kurssin, jossa käydään läpi Robot Frameworkia ja miten automatisoituja testejä tehdään sen avulla. Tällä viikolla kerkesin oppia, miten saadaan asennettua Robot Framework ja sen päälle Selenium kirjasto ja kerkesin myös harjoitella hieman automaattitestien kirjoittamista. Opin, miten tietyn projektin alle tehdään erilliset kansiot resursseille, tuloksille ja testeille. Testikansioon kirjoitetaan varsinaiset automaattitestit, ja tulos kansioon tulee testauksen tulokset. Opin, että ennen kuin esimerkiksi Selenium kirjastoa voi käyttää omissa automaattitesteissä, pitää kirjasto tuoda testille avainsanalla "Library Selenium Library". Kun tämä kirjasto on tuotu projektiin, sen jälkeen sen sisältämiä avainsanoja voi hyödyntää testissä. Esimerkiksi selaimen saa auki hyvin helposti komennolla Open Browser sen jälkeen kirjoitetaan avattavan sivuston url ja millä selaimella automaattisesti haluaa sivuston aukeavan, esim. IE, Chrome, Firefox. Selaimen avaamiseen automaation kautta liittyi vielä välivaihe, jolla kaikille selaimille, joita haluaa automaattitesteissä käyttää, pitää asentaa Web Driver. Selenium kirjaston (tai muiden testiautomaatioissa käytettyjen kirjastojen) käyttöön löytyy omaa dokumentaationsa, josta avainsanat ja miten niitä käytetään -löytyvät.

Innostuin oikeastaan aika paljon tästä testiautomaation mahdollisesta hyödyntämisestä omassa työssä, joten toivon, että pääsen jossakin vaiheessa konkreettisesti kokeilemaan automaattitestien tekemistä oikeilla asiakaskohtaisuuksien testeillä, mutta tässä kohtaa aion vielä käydä loppuun ehkä ensiviikolla, jos aikaa on, tuon nykyisen kurssin ja mahdollisesti myös jonkun Jenkins kurssin. Sen jälkeen on helpompi lähteä tutustumaan meidän omiin prosesseihin testiautomaation suhteen, kun on peruskäsitys asiasta jo olemassa. Näillä muutamilla viimeviikolla olen oppinut ainakin sen, miten paljon opittavaa itsellä vielä onkaan, varsinkin liittyen teknisempään osaamiseen.

### **3.7 Viikko 7 (4.10. -7.10.2021)**

#### **Maanantai 4.10.2021**

Päivälle on ennakkoon tiedossa vain yksi, Asiakas E:n tulevaan versiopäivitykseen liittyvä palaveri. Palaverissa on tarkoitus käydä yhteisesti läpi tulevaa testausta lähinnä siitä näkökulmasta, mitkä ovat tälle asiakkaalle erityisen tärkeitä kehityksiä/korjauksia tässä päivityksessä. Tähän testaukseen olen osallistumassa lähinnä auttamaan toista testaajaa, joka normaalisti hoitaa tämän asiakkaan testauksen. Samalle asiakkaalle on niin paljon myös muita projekteja menossa, että versiopäivityksen testaukseen kaivattiin apuja ja koska itselläni ei juuri nyt ole muita varmoja projekteja tiedossa, osallistun tähän testaukseen apuna. Loppupäivästä, jos asiakas B:n korjaukset tulevat, testaan korjauksia. Muutoin jatkan Robot Frameworkin opettelyn parissa.

Tänään käytiin lävitse asiakas E:n tulevaa testausta palaverissa. Toinen testaaja on jo tehnyt testitapaukset valmiiksi, joten omaksi osakseni tässä testauksessa jää todennäköisesti vain itse testausvaiheessa avustaminen. Huomasin kuitenkin, että on testaajalle huomattavan mukava asia, kun päivityksen kehitykset ja korjaukset käydään projektiryhmän kanssa yhdessä lävitse, silloin saa myös toisten asiantuntemuksen käyttöön asiakkaasta ja siitä mitkä päivityksen kehitykset ja korjaukset ovat erityisen olennaisia juuri tälle asiakkaalle, eikä ole vain omassa varassaan. Asiakas B:lle ei ollut vielä tänään tullut



korjauspakettia, joten päiväni kului testiautomaatio kouluttautumisen parissa. Lisäksi tänään oli myös palaveri, jossa käytiin lävitse kaikkien asiakkaiden tilannetta.

### **Tiistai 5.10.2021**

Tänään aamusta pääsen testaamaan Asiakas B:lle tehtyjä korjauksia, joten oleuksena on, että tähän varmistustestaukseen menee ainakin muutama tunti päivstä. Loppuosan päivää saattaa tulla testianalyysia/suunnittelua Asiakas H:n versiopäivitystä varten, sillä eilisessä palaverissa tuli ilmi, että tälle asiakkaalle päivitetään uutta versiota muutaman viikon sisällä.

Aamusta aloitin Asiakas B:lle tehtyjen korjausten testauksen, mutta melko nopeasti törmäsin Blocker tyyppiseen virheeseen, joka estää testauksen jatkumisen ennen kuin virhe on korjattu. Toiseen vikaan tehty korjaus on vaikuttanut odottamattomalla tavalla ja tämä täytyy korjata, ennen kuin testausta pääsee jatkamaan. Informoin tästä sekä järjestelmäasiantuntijaa, että kehittäjää Teamsin kautta, jotta virhe saadaan mahdollisimman pian käsittelyyn ja korjatuksi uuden paketin myötä.

Tänään pääosa aikaa meni Asiakas H:n versiopäivityksen testianalyysin/testisuunnittelun parissa. Uusi versio on juuri ilmestynyt. Päivityksessä tulee mukana muutama välissä ollut HF-päivitys, sekä SR-päivitys. Uusimmassa päivityksessä tuleviin kehityksiin ja korjauksiin en ole ehtinyt vielä tutustua, joten testianalyysin ohessa pitää käydä lävitse mitä kehityksiä ja korjauksia on ylipäättään versiossa tullut.

### **Keskiviikko 6.10.2021**

Tänään on tiedossa Asiakas B:n asioista sisäinen palaveri, jossa käydään lävitse korjausten tilanne. Sen lisäksi tänään teen loppuun asiakas H:n versiopäivityksen testianalyysin ja testisuunnitelman.

Kävin tänään lävitse loput uudessa versiossa tulleet kehitykset ja korjaukset. Yleensä kaikkia versiossa tulevia kehityksiä ja korjauksia ei testata asiakasversiossa, vaan niitä täytyy valikoida, usein siksikin, että monesti päivityksessä saattaa tulla useamman version nosto kerralla, joten kaikkea ei ole mahdollista rajallisen ajan puitteissa testata. Toinen syy testauksen pois jättämiselle on, että kaikki tuoteversioon tehdyt kehitykset tai korjaukset eivät välttämättä koske kaikkia asiakkaita, sillä tietyt ominaisuudet eivät ole välttämättä kaikilla käytössä. Versiossa tulleet kehitykset ja korjaukset täytyy siten käydä lävitse ja priorisoida, mitkä niistä täytyy ainakin asiakkaan versiossa testata.

### **Torstai 7.10.2021**

Tänään pääsisin aloittamaan Asiakas E:n versiopäivityksen testausta toisen testaajan kanssa, jos vain olisin saanut toimivat tunnukset. Jos en saa tänään tunnuksia, en pääse tänään aloittamaan testauksia tällä asiakkaalla. Mitään muutakaan asiakastestauksissa edistettävää ei ole tänään etukäteen tiedossa, sillä Asiakas B:n korjaukset ovat vielä kesken, ja Asiakas F:stä ei myöskään ole kuulunut mitään sen jälkeen, kun viimeviikolla toimitin heille alustavat testitapaukset. Joten jos en tänään pääse edistämään asiakastestauksia, jatkan testi-automaatio kurssin käymistä. Olen tästä edelleen innoissani, olisi hienoa päästä tekemään asiakaskohtaisille toteutuksille automaatiotestejä. Lisäksi koen, että tämä olisi myös aidosti hyödyllistä sen kannalta, että asiakaskohtaisuuksien regressiotestit suoritetaan jokaisessa testauksessa manuaalisesti, automaatisointi voisi siis vähentää regressiotestaukseen käytettävää aikaa.

Tänään tulikin hieman yllättäen tieto, että Asiakas H:n ja Asiakas E:n versiopäivitysten lisäksi tässä kuussa tulisi testaukseen myös Asiakas I:n ja asiakas J:n versiopäivitykset. Näistä kummallakaan asiakkaalla en ole testannut aiemmin, joten se että tuleekin näin monta versiopäivitystä yhdessä kuussa, joista kaksi on itselleni uusia asiakkaita, mietityttää kyllä etukäteen aikataulullisesti, että kuinka tämän kaiken ennättää tehdä. Aina uuden asiakkaan kohdalla menee kuitenkin jonkin verran aikaa tutustua asiakaskohtaisiin toteutuksiin ja asiakkaan ympäristöön yleensä. Asiakas I:n versiopäivitys on tulossa ensimmäisenä, melko pian jo, joten tänään ohjelmassa olikin testianalyysin ja suunnitelman

teko tämän asiakkaan versiopäivitykselle. Päivitykseen on myöskin tulossa uusin SR-versio. Aiemmin tällä viikolla ennätin käydä päivityksen sisältöä jo lävitse, joten se nopeutti hieman testisuunnitelman tekoa, mutta silti testianalyysi pitää aina tehdä asiakaskohtaisesti ja käydä lävitse mitkä kehitykset ja korjaukset ovat juuri tälle asiakkaalle olennaisia. Tällä asiakkaalla olikin paljon sellaisia, jotka tässä versiossa ovat asiakkaalle olennaisia.

### **Viikkopäiväkirja seurantaviikko 7: 4.10.2021 – 7.10.2021**

Tästäkin viikosta suurin osa on kulunut testaustöiden osalta lähinnä testianalyysi- ja suunnitteluvaiheeseen. Sen lisäksi olen opiskellut testiautomaatiota sitä ajatellen, jos hiljaisempina aikoina pääsisin tekemään mahdollisesti jollekin asiakkaalle asiakaskohtaisten toteutusten regressiotestejä testiautomaation avulla. Tällä viikolla viikkopäiväkirjan pohdinnassa keskityinkin testianalyysivaiheeseen, koska se on ollut nyt muutaman viime viikon ajan suurin osa työtä, mitä olen tehnyt.

Testianalyysiä tehdään niin, että testauksen pohjamateriaalia (test basis) analysoidaan testaustasoon pohjautuen. Tarkoituksena on löytää vastaus kysymykseen, mitä testata? (FISTB\ISTQB 2018, 17–18). Testauksen pohjamateriaalia on kaikki se, mihin testit pohjustetaan. Pohjamateriaalia voivat olla esimerkiksi vaatimusmäärittelyt, käyttäjätarinat, riskianalyytiraportit, järjestelmän arkkitehtuurikuvaukset ja rajapinta määrittelyt. (Graham ym. 2020, 20.)

Tässä vaiheessa kehitykset ja korjaukset on jo tuoteversiossa testattu ja tuotteen hyväksyntätestaus on tehty, joten omalla kohdallani SR-päivitysten testianalyysi keskittyy nimenomaan siihen, mitkä asiat testattavassa versiopäivityksessä ovat juuri tälle asiakkaalle olennaisia. Testisuunnittelun pohjamateriaalina ovat tässä vaiheessa yleensä vaatimusmäärittelyt, jotka on tehty uusille kehityksille. Testianalyysivaiheessa olisi myös hyvä pystyä tunnistamaan, mihin kaikkiin asioihin tehdyillä muutoksilla on vaikutuksia, mutta tämä on toisinaan todella haastava tehtävä. Joskus määrittelyissä voi olla erillinen maininta siitä, että tehty muutos vaikuttaa erityisesti myös johonkin toiseen prosessiin ja silloin on helppo löytää regressiotestaukselle olennaisia asioita liittyen muutokseen,

mutta aina tällaisia erillisiä mainintoja ei ole, eikä välttämättä ole aina niin helppoa tulkita pelkästään määrittämisestä, mihin kaikkeen muuhun tämä kyseinen kehitys/korjaus voi vaikuttaa. Tässä voisi helpottaa, jos olisi enemmän yhteistyötä tuotekehityksen ja asiakastestauksen välillä, esimerkiksi niin, että olisi jokin yhteinen palaveri, jossa tuotekehityksen testaajien kanssa käytäisiin lävitse versio-päivityksen testauksia, koska heillä on tässä vaiheessa jo hyvä asiantuntemus siitä, mihin kaikkiin asioihin nämä asiat vaikuttavat. Tästä heräsiikin kehityseh-dotus, jonka voisi ottaa esille tuotekehityksen testaajan ja toisen asiakastestaajan kanssa, voisimmeko jatkossa ehkä toimiakin näin.

Tällä hetkellä oma prosessini testianalyysiin on seuraavanlainen: Tehdessäni testianalyysia liittyen versio-päivityksiin, aloitan yleensä siitä, onko korjauksissa jotakin, joka on erityisen tärkeää tälle asiakkaalle. Esimerkiksi onko asiakkaalta tullut huomioita liittyen johonkin asiaan, mikä on nyt korjauksissa mukana. Seuraavaksi poissuljen sellaiset korjaukset/kehitykset, jotka eivät ole asiakkaalle olennaisia. Esimerkiksi, jos asiakkaalla ei ole lainkaan käytössä jokin tietty ominaisuus, silloin sitä ei voida tietenkään myöskään testata. Sitten on joitakin korjauksia, jotka on hankalaa todentaa asiakkaan testi-ympäristössä. – Tarvittaisiin esimerkiksi paljon enemmän testidataa mitä ympäristössä nyt on, jotta voisi todentaa jotakin ei-toiminnalliseen testauksen alueeseen liittyvää korjausta, kuten suorituskyky parannusta. Myös jotkin trivial- luokitellut korjaukset voivat olla pois jätettäviä testattavia asiakasversiossa. Nämä ovat yleensä sellaisia, jotka eivät vaikuta mitenkään järjestelmän toiminnallisuuteen, vaan ovat ns. kosmeettisia korjauksia. Kun tein Asiakas H:n testianalyysiä näin, näiden poissulkemisten jälkeen jäljelle jäi vielä kahdeksantoista testattavaa kehitystä ja korjausta. Asiakas I:llä puolestaan testattavaa tuli enemmän, koska tälle asiakkaalle olennaisia kehityksiä ja korjauksia oli versio-päivityksessä enemmän, kuin Asiakas H:lle.

Joskus on tehty niin, että kaikki versio-päivityksessä tulevat korjaukset esimerkiksi listataan yhdelle testitehtävälle ikään kuin testistepeille. Se on ollut hyvä käytäntö siinä mielessä, että tällöin jokainen kehitys ja korjaus tulee raportille listattuna ja jää näkyviin onko sitä testattu tai ei ja jos niin miksi ei. Itse olen päättänyt nyt tekemään niin, että jokaiselle korjaukselle tulee oma tehtävänsä, kuten muillekin testeille. Tämä sen vuoksi, että esimerkiksi kuvallinen

dashboard raportti ottaa tällaisen ”yhdistetyn” korjaus testitapauksen ikään kuin yhtenä testinä, vaikka sen alla voi olla kaksikymmentä eri korjaukseen liittyvää testitapausta. Se ei anna realistista kuvaa testattujen testitapausten määrästä ja tällöin muutkaan automaattisesti saatavat testimetriikat ja raportit eivät täsmää.

### **3.8 Viikko 8 (11.10. -15.10.2021)**

#### **Maanantai 11.10.2021**

Tänään on ennakkoon tiedossa palaveri siitä, miten tulevan kuukauden testaus-työt aikataulutetaan, nyt kun töitä onkin yllättäen todella paljon. Jos en pääse vielä tänään aloittamaan testauksia Asiakas I:llä, menee tämä päivä hyvin todennäköisesti Asiakas E:n versiopäivityksen testauksessa, sillä tunnukseni olivat tulleet perjantaina.

Tämän päiväisessä palaverissa sovittiin, että Asiakas E:n testaukset jäävät minun osaltani todennäköisesti kokonaan tekemättä, koska tulevia muita projekteja on nyt niin paljon. Tänään en vielä päässyt aloittamaan Asiakas I:n testauksia, mutta sen sijaan tein jo valmiiksi seuraavan testauksen, Asiakas J:n testisuunnitelman. Tuli myöskin melko selväksi, että näillä tulevilla projekteilla loppuvuosi alkaa olla täyteen varattu, joten ainakaan tämän vuoden puolella näitä asiakaskohtaisuuksien testiautomaatio asioita ei taida enää päästä edistämään.

#### **Tiistai 12.10.2021**

En päässyt tänäänkään vielä aloittamaan Asiakas I:n testauksia, vaan kerkesin sittenkin täksi päiväksi avustamaan Asiakas E:n versiopäivityksen testauksessa. Tänään testasin sidosryhmien hallintaan tarkoitettua järjestelmää ja tein yhden trivial luokkaisen havainnon, joka liittyi erään käyttäjälle näkyvän ohjek tekstin näkyvyyteen järjestelmässä. Muita testaushavaintoja ei tämän päivän aikana tullut.

### **Torstai 14.10.2021**

Tänään aamusta on tavoitteena päästä aloittamaan Asiakas I:n testauksia, ja tämän päivän tavoitteena on saada ainakin suurin osa perusjärjestelmän regressiotestauksista tehtyä.

Tämän päivän aikana pääsin melko hyvin päivän alussa asetettuun tavoitteeseen, regressiotesteistä jäi vielä suoritettavaksi kolme. Yhden näistä testeistä saan loppuun huomenna ja yksi jäi selvitettäväksi: onko asiakkaalla tarkoituksellisesti pois päältä eräs oleellinen sähköpostin lähetys konfiguraatioista. Tämä on itselleni uusi asiakas, joten tulee aina enemmän selviteltävää, kun ei ole ennestään tuttu asiakas.

### **Perjantai 15.10.2021**

Tänään on tavoitteena jatkaa testauksia kehitysten ja korjausten osalta ja saada regressiotestit loppuun.

Tänään opin taas uusia asioita, jotka liittyivät tällä kertaa erinäisiin AD:lla tehtäviin muutoksiin ja AD ryhmien luomisiin. Tämä liittyi erääseen kehitykseen, joka tuotteeseen on version myötä tullut ja pääsin näin oppimaan itselleni uusia asioita.

### **Viikkopäiväkirja seurantaviikko 8: 11.10.2021 – 14.10.2021**

Tällä viikolla tein yhden uuden testisuunnitelman ja pääsin vihdoinkin myös tekemään varsinaista testausta pelkän suunnittelun lisäksi. Viime viikot ovat olleet melko suunnittelupainotteisia, joten tällä viikolla oli mukavaa päästä välillä myös tekemään konkreettisesti testaustyötä. Koska mitään erityistä uutta verrattuna aiempiin viikkoihin itse testausprosessissa ei tapahtunut, kirjoitan tällä viikolla siitä, miten testiprosessi Asiakas I:n tapauksessa on mennyt verrattuna siihen, mitä kirjallisuudessa testiprosessin kulusta kerrotaan eri vaiheissa, missä projektin osalta ollaan tässä kohtaa menossa.

Ensimmäinen vaihe testauksessa on testauksen suunnittelu. Testisuunnittelu vaiheessa määritetään testauksen tavoitteet ja lähestymistavat, joilla nämä tavoitteet saavutetaan projektin kontekstissa. Valitaan sopivat testaustekniikat, ja päätetään mitä tehtäviä pitää tehdä sekä muodostetaan testauksen aikataulu. Esimerkiksi pienissä ja tutuissa projekteissa voidaan mennä pitkälti tutulla kaavalla, kun taas isommissa projekteissa joudutaan tekemään testisuunnitteluvaiheessa enemmän työtä. (Graham ym. 2020, 18.) Asiakas I on itselleni uusi, mutta projekti sinänsä on tuttu, koska kysymys on tavallisesta versiopäivitysprojektistä, jotka etenevät yleensä hyvin samalla kaavalla. Niinpä tällaista varsinaista testisuunnitteluvaihetta ei oikeastaan ole, tai ainakin se on hyvin huomaamaton, sillä prosessit ja testitekniikat versiopäivitykselle ovat hyvin pitkälti samanlaiset projektista riippumatta. Aikataulun päättäminen kuuluu tähän vaiheeseen, mutta siihenkään en juuri voi itse vaikuttaa, sillä aikataulu tulee yleensä annettuna.

Testianalyysivaiheeseen olen viimeviikkoina keskittynyt aika paljon. Testianalyysivaiheessa analysoidaan testauksen pohjamateriaalia, jotta voidaan tunnistaa testattavat ominaisuudet ja niihin liittyvät testattavat tilanteet. Tämän ohella voidaan myös testauksen pohjamateriaalista, esimerkiksi määrittelyistä, löytää samalla puutteita. Testianalyysi vastaa kysymykseen ”mitä testata”, sisältäen mitattavat kattavuuskriteerit. Testauksen pohjamateriaalia on kaikki se, mihin testit perustuvat, esimerkiksi vaatimusmäärittelyt ja käyttäjätarinat. Testattavat ehdot voidaan tunnistaa testauksen pohjamateriaalista eri testaustekniikoiden avulla. Esimerkiksi mustalaatikkotestauksen tekniikoiden avulla tunnistetaan toiminnallisia ja ei-toiminnallisia ehtoja. (Graham ym. 2020, 18.) Kun itse alan testata asiakasversiota, silloin perusteellisempi testianalyysivaihe on jo ohitse, sillä se on tehty tuotekehityksen testaajien toimesta. Itse en ole mukana näiden alkuperäisten toiminnallisuuksien suunnitteluvaiheessa, enkä siis katselmoimassa esimerkiksi uusien ominaisuuksien käyttäjätarinoita tai vaatimusmäärittelyitä. Testattavat tilanteet on käyty lävitse tuotekehityksen puolella ja niistä on monesti myös valmiita testejä valmiina. Asiakastestauksen testianalyysivaihe liittyy yleensä enemmänkin sen tunnistamiseen, mitkä kehityksistä ja korjauksista ovat kullekin asiakkaalle oleellisia ja ollaanko asiakkaalle viemässä joitakin asiakaskohtaisia kehityksiä ja korjauksia versiopäivityksen yhteydessä. Kaikkia

kehityksiä ja korjauksia ei testata – eikä edes välttämättä pystytä testaamaan – asiakasversiossa siksi, että näitä ominaisuuksia ei välttämättä asiakkaalla lainkaan käytetä.

Testauksen seurantavaiheessa verrataan todellista edistymistä testisuunnitelmaa vastaan, tarkistetaan testausaktiviteettien edistymistä ja raportoidaan testauksen statuksesta ja mahdollisista poikkeamista suunnitelmaan. Testauksen seurantavaiheessa voidaan käyttää erilaisia testauksen monitoroinnin metriikoita, jotka on määritelty testisuunnitteluvaiheessa. Testauksen edistymisestä raportoidaan oleellisille tahoille testauksen edistymisraporteilla tai tapaamisissa. Yksi tapa arvioida testausprosessia on käyttää exit-kriteeriä. Esimerkiksi: testaus ei ole päätöksessä ennen kuin kaikki suunnitellut testit on tehty. (Graham ym. 2020, 18.) Testauksen seuranta ja hallinta -vaihetta tässä projektissa ei myöskään kovin paljon ole, kyseessä on hyvin pieni ja lyhytkestoinen projekti, ja jos mitään poikkeamia ennalta suunnitellusta aikataulusta ei ole, ei mitään erillisiä tapaamisia projektiryhmän tai asiakkaan kanssa ainakaan testaajan osalta järjestetä. Seuraan itse testauksen edistymistä Dashboardille lisäämäni Pie Chartin avulla, jotta pystyn itse arvioimaan tulenko pysymään aikataulussa. Jos kyseessä olisi isompi/pidempikestoisempi projekti, silloin tämä vaihe tulisi varmaankin toteutetuksi ainakin erinäisissä päivä/viikkopalaverissa, joihin osallistuisi pelkästään projektiryhmä, tai myös asiakas. Varsinaisia testauksen edistymisraportteja ei ainakaan vielä toistaiseksi ole käytetty.

Testien suunnitteluvaihe vastaa kysymykseen ”kuinka testata”. Tässä vaiheessa tehdään korkean tason testitapaukset tai testijoukot. Tässä vaiheessa suunnitellaan ja priorisoidaan testitapaukset ja testijoukot. Tässä vaiheessa myös tunnistetaan tarvittava testidata, joka tukee määritettyjä testauksen ehtoja. Suunnitellaan testiympäristö ja tunnistetaan tarvittava infrastruktuuri ja työkalut. Sekä toteutetaan kaksisuuntainen jäljitettävyyys testauksen pohjamateriaalin, testauksen ehtojen, testitapausten ja proseduurien välillä. Myös tässä vaiheessa voi tulla esiin uusia vikoja, joita ei löytynyt testianalyysivaiheessa, koska testien suunnitteluvaiheessa mietitään jo syvemmillä tasolla testissä käytettävää testidataa. (Graham ym. 2020, 21.) Omassa työssäni testaajana en juurikaan puutu testiympäristöjen suunnitteluun vaan nämä asiat ovat



järjestelmäasiantuntijan vastuulla. Tässä vaiheessa itse normaalisti teen konkreettiset testitapaukset testienhallintajärjestelmään. Jos on asiakaskohtaisia kehityksiä tai korjauksia, linkitän joko havaitut viat tai kehitystehtävät testitapaukseen jäljitettävyyden varmistamiseksi.

Testien valmisteluvaiheessa suoritetaan testiympäristön pystytys ja muu valmistelu. Tässä vaiheessa kysytään ”onko meillä kaikki valmiina testien suoritusta varten?” Tähän vaiheeseen liittyy erityisesti automaattitestauksen osalta testien ja testijoukkojen järjestäminen ja priorisointi sekä testidatan valmistelu. (Graham ym. 2020, 21.) Myös tämä testien valmisteluvaihe on omalta osaltani hyvin pienessä roolissa, johtuen siitä, että paljon tähän vaiheeseen liittyvistä tehtävistä kuuluu prosessissamme järjestelmäasiantuntijan vastuulle. Koska automaattitestausta ei myöskään tällä hetkellä tehdä, näitä testijoukkojen järjestämiä tai priorisointeja ei tule tehtäväksi. Manuaalisten testien priorisointia en etukäteen oikeastaan tee, koska testauksen hallintajärjestelmään ei ole mahdollista erikseen merkitä testien prioriteettia. Niinpä tämä priorisointi on lähinnä omassa mielessä toteutettu tehtävä. Tämä olisi mahdollista toki toteuttaa esim. niin, että merkitsisi testitapausten otsikkoon luvuilla testien suoritusjärjestyksen esimerkiksi testin kriittisyyden perusteella.

Testien suoritusvaiheessa testit konkreettisesti suoritetaan joko automaattisesti tai manuaalisesti, suunnitellun aikataulun mukaan. Tässä vaiheessa verrataan testien odotettuja ja varsinaisia tuloksia keskenään. Mikäli vikoja havaitaan, ne raportoidaan. Testauksen tulokset kirjataan ylös, esim. Pass, Fail. Testitapauksia voidaan joutua myös uusimaan, esimerkiksi tapauksessa, jossa testi on ollut Fail -tilainen ja sama testi testataan varmistustestauksella uudelleen korjauksen jälkeen. (Graham ym. 2020, 21.) Testien suoritusvaiheen kuvaus täsmää aika hyvin siihen, mitä omassa työssäni tähän vaiheeseen liittyy.

### **3.9 Viikko 9 (18.10. -21.10.2021)**

#### **Maanantai 18.10.2021**

Tänään jatkuu Asiakas I:n testaukset. Tämän päivän testauksien aikana testasin versiopäivityksessä tulleita kehityksiä ja korjauksia. Päivän aikana tuli kaksi huomiota, joissa tuotteeseen tehdyt korjaukset eivät toimineet asiakasversiossa. Kirjasin näistä testaushuomiot, joista molemmat menivät tällä kertaa kehittäjälle katsottavaksi.

#### **Tiistai 19.10.2021**

Tänään jatkan Asiakas I:n versiopäivityksen testausta. Huomenna olisi päämäärä saada testaukset valmiiksi, niin että torstaina viimeistään saisin lähetettyä asiakkaalle testauksen raportit.

Tänään tuli joitakin niin sanottuja vääriä positiivisia, jotka johtuivat siitä, että testien odotettu tulos poikkesi oikeasta testituloksesta, mutta niin, että kyseessä ei ollut vika. Kyseessä oli asioita, jotka ovat kyseisellä asiakkaalla eri tavalla kuin tuoteversiossa.

#### **Keskiviikko 20.10.2021**

Tänään olisi tarkoitus saada Asiakas I:n versiopäivityksen testaus loppuun. Tämä päivä meni melkein kokonaan yhden ongelman selvittämisessä. Eräaseen toiminnallisuuteen oli tullut muutosta, mikä ei suoraan liittynyt kehityksiin, joita versiopäivityksessä tuotiin, mutta kehityksen myötä tähän toiminnallisuuteen oli lisätty uusia kenttiä. Itselleni oli kuitenkin epäselvää, minkä muotoinen järjestelmään sisään luettavan tiedoston piti olla, sillä uusimpaan käyttöoppaaseen ei ollut päivitetty ajantasaisia kenttiä eikä näitä löytynyt muualtakaan. Tiedostoa järjestelmään sisään lukiessa ei tullut käyttöliittymälle virhettä, eikä myöskään lokille jäänyt virhettä, miksei tuonti onnistunut, joten selvittäminen muuttui haastavaksi. Tuoteversiossa en pystynyt suoraan tätä testaamaan koska siellä ei ollut juuri nyt samaa versiota testissä. Tämä jouduttiin ensin

asentamaan standardi testiin, jotta päästiin testaamaan tilanne. Tästä kirjattiin lopulta critical tyyppinen virhe, joka täytyy korjata ennen kuin voidaan päivitystä tehdä.

Laitoin palautetta myös käyttöohjeesta dokumentoijalle, sillä testauksen yhteydessä käyttöohjeesta löytyi vanhaa tietoa kenttien määrityksistä sekä myös kenttämäärät eivät täsmänneet nykyiseen. Korjaus itse vikaan tulee todennäköisesti tuotekehityksen puolelta melko nopeasti. Lopun päivää testasin Asiakas B:lle tehtyjä korjauksia ja tällä kertaa varmistustestaus meni läpi, eli kaikki tiedossa olevat virheet on nyt korjattu.

### **Torstai 21.10.2021**

Tänään on edessä avoinna olevien havaintojen tarkistaminen: Mitä havainnoille tehdään ja koska. Sen lisäksi tarkoituksena on kirjoittaa raportit ja loki valmiiksi Asiakas I:n versiopäivityksen testauksesta.

Tänään päivä meni näissä testauksen päättämiseen liittyvissä tehtävissä, eli selvittelin järjestelmäasiantuntijan ja asiakkaan pääkäyttäjän kanssa mitä avoinna oleville havainnoille tehdään ja milloin. Sen lisäksi kirjoitin testiraportin ja testilokin testauksen tuloksista.

### **Viikkopäiväkirja seurantaviikko 9: 18.10.2021 – 21.10.2021**

Tällä viikolla sain päätöksen Asiakas I:n versiopäivityksen testauksen. Kyseessä oli itselleni uusi asiakas, ja aina kun on uusi asiakas, testauksen kanssa menee enemmän aikaa. Se johtuu siitä, että asiakkailta on omia, asiakaskohtaisia toteutuksiaan, joihin tutustuminen vie jonkin verran aikaa. Lisäksi myös siitä, että kaikkia asioita, jotka poikkeavat tuotteen toiminnallisuudesta, ei ole mitenkään dokumentoitu mihinkään. Näin ollen uusien asiakkaiden kohdalla tulee usein tilanteita, joissa herää kysymys, onko tämä vika vai kuuluuko tämä toimia näin tällä asiakkaalla. Joissakin tilanteissa nykyisin on jo hieman helpompi tietää, mikä asia johtuu vain erilaisista konfiguraatioista. Etenkin aloittelevana testaajana, kun järjestelmä ei ollut vielä kovin tuttu, tuli todella paljon tilanteita,

joissa luulin jonkun asian olevan vika, joka ei lopulta sitä ollut. Tämä johtui tilanteesta, jossa odotettu testitulokset poikkesi siitä, mikä testin lopputulokseksi tuli, mutta kyseessä olikin vain erilaiset konfiguroinnit. Nyttemmin järjestelmän konfiguraatiot ovat jo tutumpia, mutta koska konfiguraatioita on paljon, tulee edelleen eteen tilanteita, joissa se mitä luulee viaksi, johtuukin esimerkiksi jostakin asiakkaan erilaisesta konfiguraatiosta. Nämä eivät aina ole niin yksiselitteisiä asioita myöskään.

Kaikki odottamattomat testitulokset eivät aina ole häiriöitä. Sellaisia testituloksia, jotka raportoidaan vikoina, mutta jotka eivät sellaisia kuitenkaan ole, kutsutaan vääriksi positiivisiksi. Väriä positiivisia tuloksia voi syntyä siksi, että testit on suoritettu virheellisesti, testiaineistossa, ympäristössä tai muussa testimateriaalissa on vikaa. (FISTB\ISTQB. 2018, 14.)

Tällä viikolla minulle tuli eteen tilanne, jossa en pystynyt lisäämään erästä tietyn roolin omaavaa hallinnoijaa ulkoiselle organisaatiolle, vaan lisääminen päättyi käyttöliittymällä virheeseen. Lokille jäänyt virhe oli mielestäni hämmentävä, sillä vaikutti siltä, kuin hallinnoijan lisäämiselle olisi tarvittu kommentti, vaikka järjestelmässä ei ole mitään mahdollisuutta lisätä kommenttia siinä vaiheessa. Standard versiossa kyseinen testitilanne toimi. Selvitin asiaa järjestelmäasiantuntijan ja konsultin kanssa ja ilmeni, että kyseessä olikin tietty vaatimus, joka ei varsinaisesti ollut asiakaskohtainen toteutus, mutta toimi tällä asiakkaalla eri tavalla kuin kenelläkään muulla aiemmin, joten en ollut törmännyt vikaan aiemmin. Tämä vika oli kuitenkin jo tiedossa ja tulossa korjaukseen seuraavaan versio-päivitykseen.

Toisaalta muutama huomio, joiden oletin olevan virheitä, eivät sitä olleet. Sidosryhmien hallinnoinnin järjestelmään oli versiopäivityksen myötä tullut kehitys, jossa eräs toiminnallisuus, joka aiemmin oli tullut suoraan koodista, oli muutettu konfiguraatioiden kautta hallinnoitavaksi. Kun testasin tätä ominaisuutta ja vertasin käyttöliittymässä olevia pakollisia kenttiä konfiguraatioon, huomasin, että eräs kenttä, joka oli konfiguraatioissa merkitty ei-pakolliseksi kentäksi käyttöliittymässä, oli kuitenkin pakollinen kenttä. Kirjasin tästä vian, mutta selvitettäessä asiaa enemmän, kävikin ilmi, että tämä kenttä, joka oli konfiguraatiossa, ei ollut

asiakkaalla käyttöliittymässä ollenkaan näkyvissä, vaan käyttöliittymässä näkyi toinen vastaavaa tietoa merkitsevä kenttä, jonka pakollisuus määritettiin kokonaan toisesta konfiguraatiosta. Koska nämä kentät oli kuitenkin käyttöliittymässä nimetty vastaavaksi kuin konfiguraatiossa, en tullut edes ajatelleeksi, että kyseessä voisi olla eri tietokenttä.

Keskiviikkona tuli vastaan tilanne, jossa en saanut erästä järjestelmän kannalta erästä melko oleellista toiminnetta toimimaan. Tämä liittyi toiminnallisuuteen, johon ei ollut suoraan tehty kehityksiä ja korjauksia, mutta joihin eräät tehdyt kehitykset olivat epäsuorasti vaikuttaneet. Järjestelmä antoi tietyissä tilanteissa selkeän virheilmoituksen siitä, miksi en onnistunut viemään tiettyjä tietoja järjestelmään, mutta eräessä testitapauksessa mitään virheilmoitusta ei tullut, eikä lokilekaan jäänyt mitään virhettä. Tietoja ei vain tuotu. Epäilin aluksi pitkään, että vika oli tekemässäni tiedostossa, sillä tarkkoja tietoja siitä, minkä muotoinen tiedoston pitäisi nykyään olla, ei meinannut löytyä mistään. Yleensä kehityksistä ja korjauksista löytyy hyvin dokumentaatiot, mutta tästä välillisestä muutoksesta tiedostoon en meinannut löytää ajantasaista tietoa mistään. Etsin uusimman version käyttöoppaasta ohjeen tähän importiin. Tämä oli ainoa dokumentaatio, jonka löysin tiedoston nykyisestä muodosta. Käyttöoppaaseen oli kuitenkin jäänyt vanhentunutta tietoa, sillä kenttämäärä oli yhä sama kuin aiemmassa oppaassa. Sain kuitenkin järjestelmäasiantuntijalta tiedon tiedoston loppuun lisättyjen kenttien määrästä, ja nämä piti voida jättää tyhjäksi. Tämänkin jälkeen tuonnissa ilmeni virheitä, mutta ainakin virheilmoitus muuttui. Kävi ilmi, että käyttöoppaan mukaan tehtynä import ei onnistunut, sillä muutama tiedon määrittäminen ei toiminut enää samalla tavalla kuin ennen. Sain kuitenkin muuta kautta selvitettyä nämä asiat ja sain osittain toiminnallisuuden toimimaan. Tiettyjä tietoja pystyi tuomaan järjestelmään, mutta tiettyjä edelleenkin ei.

Seuraavaksi kysyin tuotekehityksen testaajalta asiasta, ja sitä kautta löytyi tietoa mitä lisätyt kentät ovat. Perustuotteessa toiminnallisuutta ei voinut suoraan testata, koska juuri samaa versiota ei ollut sillä hetkellä missään testissä. Ensin piti siis saada tämä versio pyörimään perustuotteella, että tuotekehityksen testaaja pystyi varmistamaan testitilanteen samassa versiossa myös perustuotteen testissä. Kävi ilmi, että toiminnallisuus oli tosiaan mennyt rikki muutosten myötä

ja että tiedoston tuonti onnistuu vain tietyssä rajatussa testitilanteessa, kaikki muut tilanteet epäonnistuvat, mutta eivät jätä epäonnistuessaan mitään virhettä käyttäjälle tai lokillekaan. Tästä kirjattiin critical tason virhe, josta tarvitaan pikainen Hot Fix päivitys. Korjaus saadaankin käyttöön viimeistään maanantaina.

Tässäkin kävi siis ilmi, miten tärkeää on, että kaikista muutoksista löytyy hyvät, ajantasaiset dokumentaatiot, asioiden selvitys olisi huomattavasti nopeampaa. Muuten aikaa menee turhaan ihmettelyyn: onko testitilanne väärä tälle asiakkaalle, onko tässä joku asiakaskohtainen konfiguraatio, onko tiedosto vääränmallinen jne. Näissä tilanteissa aikaa kuluu paljon siihen, että täytyy selvittää, miten joku toiminnallisuus toimii ja monessa tapauksessa kysellä eri ihmisiltä, miten tämän kuuluisi toimia, ennen kuin voi todeta asian virheeksi. Toisaalta haasteena on myös se, miten dokumentoida kaikki asiakaskohtainen tuoteversiosta poikkeava toiminnallisuus, koska poikkeavuuksien selvittäminen alun perin voi olla perin haastavaa. Itse olen toiminut nyt niin, että aina kun asiakasversiossa tulee vastaan toiminnallisuutta, joka ei varsinaisesti ole asiakaskohtainen toteutus (kooditasolla), mutta poikkeaa kuitenkin tuoteversion vastaavasta, teen kyseiseen testitapaukseen tämän muutoksen, jolloin seuraavan kerran testatessa kyseisellä asiakkaalla, ainakin testitapaus on ajan tasalla.

Lopulta kirjattujen virheiden lisäksi laitoin palautetta käyttöoppaasta dokumentoijalle, jotta nykyversiossa olevat poikkeavat tiedot saataisiin käyttöoppaaseenkin ajan tasalle. Loppuviikosta tein myös hieman varmistustestauksia Asiakas B:lle ja kirjoitin testauksen raportit Asiakas I:n testauksesta.

### **3.10 Viikko 10 (25.10 – 28.10.2021)**

#### **Maanantai 25.10.2021**

Tällä hetkellä jonossa seuraavana on kaksi versiopäivitystä, Asiakas J:n, ja Asiakas G:n. Kummallakaan asiakkaalla en ole itse testannut aiemmin. Viime viikon perjantaina oli tuotekehitykseltä tullut korjaus tuotteessa havaitsemani vikaan. Asiakas J:lle oli jo ehditty tehdä paketti, mutta nyt se pitää tehdä uudelleen, sillä tuo tehty korjaus pitää tulla mukaan tähän versiopäivitykseen.

Asiakas J:n versiopäivitys tehdään ensin, jos saadaan kehittäjältä paketti, joka sisältää HF-korjauksen. Jos taas Asiakas G:n testaukset pääsee aloittamaan aiemmin, tulee tämä testaus ensin. Tällä hetkellä on siis vähän epäselvä tilanne, kumpi testaus tulee ensin, mutta tänään on edessä joka tapauksessa testisuunnitelman teko Asiakas G:lle.

Aloitin tänään testisuunnitelman tekoa Asiakas G:lle. Kävin läpi aiempien testauksien testitapauksia sekä dokumentaatiota liittyen asiakkaaseen, koska tämä on myös itselleni uusi testattava asiakkuus. Tämän asiakkaan kohdalla on myöskin monia erilaisia projekteja, joten testitapauksia löytyi useammasta paikasta. Asiakkaalle ei ollut aiemmin tehty tämän muotoista versiopäivitystä, joten testitapaukset eivät vielä olleet samassa mallissa, mihin ne normaalisti asetellaan, eli esimerkiksi asiakaskohtaisten toteutusten testejä ei ollut missään ennestään vielä valmiina. Asiakkuudelle on muutoinkin tullut muutoksia, joten tietoa löytyy hyvin hajanaisesti eri paikoista.

### **Tiistai 26.10.2021**

Uutta pakettia ei vielä tullut Asiakas J:lle eilen, joten tänään tarkoituksena on jatkaa testisuunnitelman tekoa Asiakas G:lle. Suunnitelman teko on normaalia työläämpi siksi, koska asiakaskohtaisuuksien testaus täytyy selvittää kokonaan alusta, ja versiopäivityksessä on muun muassa rajapintoja testattavana. Tänään keskustelin järjestelmäasiantuntijan kanssa rajapintojen testauksesta ja yritin selvittää, pystynkö testaamaan rajapintoja suoraan lähdejärjestelmästä (eli voinko saada pääsyä siihen asiakkaan järjestelmään, josta tieto normaalisti kulki IAM-järjestelmään), vai pitääkö testaus tehdä jonkun rajapintojen testaus työkalun kautta. Nämä olisi hyvä saada testattua oikeiden prosessien mukaan, mutta hyvin harvoin testaajalla on pääsyä asiakkaan muihin järjestelmiin testauksen yhteydessä kuin IAM-järjestelmään. Tänään paikalla ei ollut henkilöitä, joiden kanssa olisin voinut käydä läpi tarkemmin tätä asiaa, joten joudun selvittämään rajapintojen testauksen myöhemmin. Tänään aloitin jo regressiotestausta, jotta testauksissa päästiin eteenpäin.

## **Keskiviikko 27.10.2021**

Asiakas G:n versiopäivityksen regressiotestaus jatkuu. Asiakas J:lle on nyt tullut uusi paketti, joka sisältää korjauksen viime viikolla tuotteessa havaitsemaani vikaan. Tätä pakettia ei vielä olla asennettu asiakkaan testiin, joten en pääse aloittamaan Asiakas J:n testauksia vielä.

## **Viikkopäiväkirja seurantaviikko 10: 25.10.2021 – 28.10.2021**

Tällä viikolla tekeminen oli alkuviikosta jälleen testisuunnitelman tekoa sekä selvitystyötä. Olin yhteydessä testajaan, joka on aiemmin tehnyt testauksia tällä asiakkaalla toisessa projektissa. Lisäksi selvittelin asioita järjestelmäasiantuntijan kanssa ja kävin lävitse rajapintojen testauksen osalta asiakkaan prosesseja, jotta ymmärtäisin mistä data kulkee IAM-järjestelmään ja toisinpäin. Rajapintoihin on tullut muutoksia verraten aiempaan asiakkaan versioon, joten ne pitää testata huolellisesti tämän testauksen yhteydessä. Olisi ollut ehkä helpompaa hahmottaa asiakkaan prosessit, jos ne olisi käyty yhdessä lävitse jossakin lyhyessä palaverissa tekijöiden kanssa, joille asiakkaan prosessit ovat tuttuja, olisi ollut nopeampi hahmottaa mitä kaikkia rajapintoja pitää testata ja mikä on niihin liittyvä prosessi. Aiemmin asiakkaalla testannut testaja ei enää muistanut näitä niin tarkkaan koska edellisistä testauksista oli jo niin kauan, ja toisaalta, koska tällä viikolla oli syysloma viikko, iso osa ihmisistä oli lomalla.

## **3.11 Viikko 11 (1.11. - 4.11.2021)**

### **Maanantai 1.11.2021**

Tänään on tarkoitus jatkaa Asiakas G:n testauksia, muutama regressiotesti on vielä kesken ja sitten jatkan korjauksilla ja kehityksillä.

Tänään on vihdoin se päivä, kun projektien-, vian- ja testienhallinnan järjestelmä siirtyy uuteen ratkaisuun. Datan migroimisessa juuri testauksen liitännäisen osalta on ollut ongelmaa, jonka takia koko järjestelmän vaihto siirtyi jo



kerran kuukaudella, joten odotettavissa on, että joudumme tekemään manuaalista työtä testisykliä siirtämisessä uuteen järjestelmään jossakin kohtaa. Juuri nyt näin isoon manuaaliseen työhön ei olisi aikaa koska on niin paljon asiakasprojekteja odottamassa testaamista, mutta ilmeisesti testisykliä siirtäminen ei ole automatisoidusti mahdollista tai ainakin vaatisi jopa enemmän työtä kuin manuaalinen tiedonsiirto. Tänään testauksissa tein yhden huomion, jota ei tällä hetkellä voinut kirjata mihinkään, koska vanhaan järjestelmään ei saanut enää huomioita kirjata ja uusi järjestelmä ei vielä ollut käytössä. Niinpä otin huomion itselleni ylös, jotta kirjaan sen huomenna, kun uusi järjestelmä on käytössä. Testauksen ajot sai muuten kirjata vanhaan järjestelmään, sillä ajatuksella, että ne eivät siirry uuteen, eli jossakin kohtaa täytyy sitten siirtää nämä tiedot itse uuteen järjestelmään.

### **Tiistai 2.11.2021**

Tälle päivälle olisi tarkoitus saada Asiakas G:n kehitykset ja korjaukset testattua ja selvitettyä miten rajapintatestaus olisi mahdollista tehdä. Tällä hetkellä itselläni ei ole pääsyä asiakkaan muihin järjestelmiin.

Pääsin tänään kuitenkin testauksissa eteenpäin korjausten ja kehitysten osalta. Sidosryhmien hallinnan järjestelmän osalta ilmeni sellainen blocker tyyppinen virhe, joka estää testauksen jatkumisia tämän järjestelmän osalta.

### **Keskiviikko 3.11.2021**

Tänään testaukset Asiakas G:llä jatkuivat ja selvisi myös, että tähän saakka tehtyjä havaintoja ei ole tarpeen korjata tähän versiopäivitykseen. Havainnot oli käyty lävitse palaverissa, jossa en itse ollut läsnä. Tämä päätös, ettei havaintoa korjata tähän versiopäivitykseen hieman kummastutti itseäni, koska tehty havainto liittyy kuitenkin erittäin oleelliseen osaan järjestelmän toimintaa. Olin kuitenkin aiemmin saanut kuulla, että asiakkaan prosessit ovat muuttumassa tämän osalta, joten mahdollisesti tätä sidosryhmien hallinnan järjestelmää ei sitten enää käytetä näiltä osin. Koska kuitenkin eilen ilmennyt virhe estää tiettyjen testien ajamisen, osa etukäteen suunnitelluista testeistä jää blocked tilaiseksi,

joka tarkoittaa, että testejä ei voi suorittaa. Tästä johtuen sain tänään kaikki muut testaukset tämän asiakkaan osalta valmiiksi, pois lukien rajapintojen testaus. Näiden osalta odotan yhä vastausta siihen, saanko tunnuksia asiakkaan järjestelmään, josta testauksen voisi tehdä suoraan, vai täytyykö rajapintates-  
taus tehdä restlet clientin kautta. Olisi parempi, jos rajapinnat pystyisi testaa-  
maan koko prosessilla alkaen lähtöjärjestelmästä, mutta tämä on harvoin mah-  
dollista. Toisinaan tällaiset testaukset voidaan tehdä asiakkaan kanssa yhteis-  
työssäkin, mutta tässä tapauksessa, jos tunnuksia toiseen järjestelmään ei  
saada käyttöön, teen testauksen tuon rajapintatyökalun kautta. Se on itselleni  
uusi, joten siinä kohtaa tulee taas uutta opeteltavaa, miten saan testaukset suo-  
ritetuksi. Tämän lisäksi tänään piti siirtää seuraavan asiakastestauksen testisyk-  
lin testit uuteen testienhallintajärjestelmään. Näitä testisyklejä ei onnistuttu siir-  
tämään minkään automaation kautta, joten tästä siirrosta seuraa jonkin verran  
manuaalista työtä.

#### **Torstai 4.11.2021**

Tänään tein tietojen manuaalista siirtoa uuteen testienhallintajärjestelmään ja  
pääsin lisäksi muistelevaan, miten Dashboardille saatiinkaan samanlainen  
kuva testien suorituksesta kuin ennen. Uudessa ratkaisussa samaa gadgetia ei  
enää ole käytettävissä kuin aiemmin. Onneksi olin kirjoittanut itselleni ylös vii-  
meksi testatessa selvittämiäni asioita, sillä tässä kohtaa en olisi enää muistanut  
suoraan, miten sain viimeksi tuon samanlaisen kuvan tehtyä, sillä se vaati hie-  
man enemmän työtä, eikä tuon järjestelmän dokumentaatiosta löytynyt edes oh-  
jetta tällaiseen tilanteeseen. Huomasin kuitenkin viimeksi testatessa, että sa-  
manlaisen raportin saa aikaan, jos käyttää komponentteja oikein. Olin ottanut  
itselleni kuvakaappauksia ja niistä olikin helppo selvittää mitä olin viimeksi saa-  
nut selville ja kerroin nämä myös toiselle testaajalle. Joudumme muuttamaan  
työtapoja hieman ja tekemään enemmän pohjatyötä tuota kuvallista raporttia  
varten, mutta se kuitenkin onnistuu samoin kuin ennen, mikä on pääasia. Ensi-  
viikolle maanantaille jääkin sitten rajapintatestien tekeminen, ja testiraporttien  
kirjoittaminen Asiakas G:n osalta ja pääsen siirtymään seuraavan asiakkaan  
testaukseen.

## Viikkopäiväkirja seurantaviikko 11: 1.11.2021 – 4.11.2021

Tämä viikko koostui pääasiassa melko perus tekemisestä, versiopäivityksen testien tekeminen sujui rutiinilla. Järjestelmän siirtyminen uuteen ratkaisuun toki aiheutti omat työnsä tälle viikolle. Selvisi, että emme saa mitenkään siirrettyä vanhoja testisyklejä uuteen järjestelmään automaattisesti, vaan joudumme tekemään sen manuaalisesti. Se tarkoittaa, että testien ajohistoriaa ei olisi jatkossakaan näkyvillä. Itse testit saatiin kuitenkin siirrettyä, joten manuaalisesti täytyisi luoda testisyklit. Testisykli tarkoittaa sitä, kun tietylle asiakkuudelle, esim. Asiakas G:lle, tehdään uusi versiopäivitys, asiakkaalle luodaan tällöin uusi testisykli SR-päivitykselle, jolle luodaan kansiot regressiotestaukselle, kehitysten ja korjausten testaamiselle sekä asiakaskohtaisuuksille. Näiden kansioiden alle sitten luodaan itse testit. Jos haluaisimme säilyttää testien ajohistorian, eli sen millaisia virheitä esimerkiksi viimeisen versiopäivityksen yhteydessä on kirjattu, ja mitkä testit ovat menneet läpi ja mitä huomioita näille testeille on kirjattu, pitäisi nämä testisyklit luoda manuaalisesti uudestaan, kirjata testit ajetuksi, tehdä linkitykset testeistä kirjatuille huomioille ja kirjata tehdyt kommentit testin jokaiselle stepille uudestaan. Tässä olisi aika paljon manuaalista tekemistä ja tällöinkään testien ajopäivät eivät jäisi uuteen järjestelmään oikeina, sillä ajopäivää ei pysty muuttamaan. Mahdollisesti testien kommentteihin voisi kirjata milloin testi on oikeasti suoritettu. En tiedä vielä, miten laajasti lähdemme näitä tietoja manuaalisesti näin tuomaan, sillä oikein kummallakaan testaajalla ei olisi näin isoon työhön juuri nyt aikaa. Mutta toisaalta jos testien ajohistoriaa ei löydy helposti, se voi jatkossa aiheuttaa jonkin verran hankaluuksia. Vähintään pitäisi tuoda jokaiselle asiakkaalle edellisen testauksen testisykli, jotta seuraavan testauksen suunnittelu helpottuu, kun asiakaskohtaiset testit ovat jo valmiina.

Suoritin asiakas G:n testit vielä vanhassa järjestelmässä, ja otan sieltä ulos myös testilokin ja raportin. Siirsin joka tapauksessa torstaina tämän asiakkaan testisyklin myös uuteen järjestelmään, sillä vikakirjaukset oli jo tehty tänne. Kirjasin Fail tilaiseksi menneet testit uudelleen ja linkitin niihin kirjatut testihuomiot. Näin jää ainakin jälki niistä testeistä, joista on kirjattu vika, joka tietysti olisi muidenkin asiakkaiden kohdalla olennaisinta. Jäljitettävyyden on kuitenkin erittäin tärkeää testauksen kannalta. Tällä viikolla tuli myös ilmi, että

etukäteistestauksesta oli tosiaan hyötyä myös siltä kannalta, että sillä tavoin säästettiin aikaa nyt, kun järjestelmä todella siirtyi uuteen ratkaisuun. Ei tarvinnut enää tässä kohtaa ruveta selvittämään miten esimerkiksi korvaamme sen Dashboardin kuvan, jota ei enää saanut samoin kuin viimeksi, vaan pystyin kertomaan toiselle testaajalle sen mitä sain testatessa selville aiemmin. Mutta oli myös tärkeää, että olin tehnyt näistä dokumentaatiota, sillä en olisi enää muistanut miten viimeksi saman raportin tein, koska se ei tosiaan ollut kovin suoraviivaista, eikä järjestelmän omassa dokumentaatioissa tätä suoraan neuvottu. Vaikka sitä aina olettaa, että asiat muistaa, kun ne on kerran tehnyt niin ei se aina niinkään mene, kun on uudesta asiasta kyse. Kuvakaappauksien ottamisessa ei mene kauan, mutta ne nopeuttavat monen asian selvitystä jatkossa.

### **3.12 Viikko 12 (8.11. - 11.11.2021)**

#### **Maanantai 8.11.2021**

Tänään päivän tavoitteena on saada Asiakas G:n testaukset loppuun, mikäli on löydetty ratkaisu siihen, miten rajapinnat saa testattua.

Tänään en päässyt tekemään Asiakas G:n testauksia loppuun, sillä vieläkään ei ollut ratkaisua rajapintojen testaukseen. En ole saanut omia tunnuksia toiseen järjestelmään, enkä pysty testaamaan rajapintoja rajapintojen testaustyökalun avulla myöskään, sillä en pysty asentamaan asiakkaan palvelimelle selainlaajennuksia itse. Tämä jää siis edelleen odottamaan asiakasta tunnusten osalta.

Sillä aikaa aloitin tänään Asiakas J:n testauksia. Suuri osa päivästä meni kuitenkin pääsyjen selvitykseen. Useinkin uudella asiakkaalla pääsyjen selvittämiseen menee aikaa, sillä kaikilla asiakkailla on omat prosessinsa pääsyihin. Tässä hankaluutena oli vielä, kun en pysty suoraan asentamaan omalle työkohteelle sovelluksia, - mikä on tietysti tietoturvasyistä ihan järkevää -, mutta hidastaa hieman asioita joissakin tapauksissa, kun asennuksiin täytyy hakea erikseen lupa, jotta pystyy asentamaan uusia sovelluksia. Sinänsä prosessi on helppo ja etenee hyvinkin nopeasti, mutta asiaan liittyi myös muutama mutka,

joiden ohjeen etsimiseen meni aikaa. Kun sain nämä vihdoin kuntoon, kävin pääkäyttäjän kanssa vielä lävitse, miten pääsen tietokantaan, ad:lle ja eri palvelimille. Tämänkin jälkeen muutamia pääsyjä vielä uupui, mutta pääsin kuitenkin aloittamaan testaukset jo tänään. Asiakkaan ympäristössä ei ollut yhtään testikäyttäjää valmiina, jolla olisi päässyt kirjautumaan järjestelmään. Tämä on hyvin harvinainen tilanne, normaalisti testikäyttäjiä on valmiina vähintään yksi. Jouduin siis aloittamaan testikäyttäjien luomisella. Loin roolit admin käyttäjälle + peruskäyttäjälle ja hyväksyjälle. Tässäkin tuli yksi asiakaskohtainen erikoisuus kuvaan ja jouduin taas konsultoimaan pääkäyttäjää, miksen saanut tunnuksia toimintaan, vaikka käyttäjien kirjautumisoikeudet normaalien prosessien mukaan pitäisivät toimia. Nämä ovat asioita, jotka eivät lukeneet missään, joten nämä oli pakko kysyä henkilöltä, joka tiesi nämä ennestään. Asia selvisi kysymällä ja sain testikäyttäjät toimintaan.

### **Tiistai 9.11.2021**

Tänään on tarkoitus jatkaa testauksia asiakas J:llä, jos asiakas G:n rajapintatestauksiin ei tule uutta tietoa.

Tämä päivä meni osittain Asiakas G:llä ja osittain asiakas J:llä. Rajapintatestauksia en päässyt tänään vielä tekemään, mutta oli selvinnyt uutta tietoa aiempaan testaukseen liittyen, jossa en saanut vaihdettua testikäyttäjän salasanaa itsepalveluun. Tätä pääsin tänään uudelleentestaamaan. Nyt sain jo salasanan sähköpostiin testikäyttäjälle, mutta edelleenkaan kirjautuminen ei onnistunut vaihdetulla salasanalla, joka jäi vielä selvitykseen. Asiakas J:n osalta huomioita ei tänään tullut.

### **Keskiviikko 10.11.2021**

Tänään testasin Asiakas J:llä kehityksiä ja korjauksia. Sen lisäksi oli palaveri Asiakas G:n asioista, joissa käytiin läpi kaksi avointa havaintoa, joista toinen on blocker tyyppinen havainto. Kysyin vielä asiasta tässä isommalla porukalla järjestetyssä palaverissa varmistaakseni, eikö tällä asiakkaalla tosiaan haittaa, että näin olennainen toiminnallisuus ei toimi. Kävikin ilmi, että kyllä se haittaa.

Ilmeisesti aiemmassa palaverissa asiaa läpikäydessä ei ollut käyty havaintoa läpi kovin perusteellisesti tai sitten oma vikakirjaus oli epäselvä. Joka tapauksessa tehtyjä huomioita lähdettiin nyt selvittämään. Tässä palaverissa kävi ilmi, miksi on niin tärkeää, että kaikissa projekteissa olisi lopuksi yhteinen läpikäynti, koska näissä viimeistään löytyy yhteinen näkemys siitä mitä havainnoille tulisi tehdä.

### **Torstai 11.11.2021**

Tänään testasin Asiakas G:llä raportteihin liittyvää ongelmaa ja yritin löytää jotakin selittävää syytä raporttiongelmiaan. Eilen palaverissa ehdotettiin yhtä vaihtoehtoa mistä ongelma voisi johtua. Testasin tästä näkökulmasta, mutta ongelma ei ratkennut. Myöskään standard versiossa samaa ongelmaa ei ollut ja kyse oli samanlaisesta raportista, joka on perusversiossa myös, joten vertailin tuote- ja asiakasversion sql kyselyjä keskenään. Huomasin, että kyselyissä on eroja. Oma sql osaamiseni on kuitenkin vielä heiveröistä, osaan tehdä vain hyvin perusmuotoisia kyselyitä ja raporttikyselyt ovat siihen osaamiseen nähden vielä monimutkaisia. En pystynyt itse selvittämään mikä tarkemmin aiheuttaisi tiettyjen tietojen poisjäännin, joten asia meni vielä järjestelmäasiantuntijan tarkempaan tutkintaan. Tässä kohtaa olisi taas ollut hyötyä paremmasta sql osaamisesta, joten tietokantojen ja kyselykielen osaaminen on ehdottomasti yksi oma kehityskohde tulevaisuudessa.

### **Viikkopäiväkirja seurantaviikko 11: 8.11.2021 – 11.11.2021**

Tällä viikolla oli tarkoituksena saada Asiakas G:n testaukset loppuun myös rajapintojen osalta, mutta tämä tavoite ei nyt täytynyt. Tein tällä viikolla pääosin testauksia Asiakas J:llä odotellessani vastausta siihen, miten saan rajapinnat testattua Asiakas G:llä. Tähän löytyi osittain vastaus torstaina, mutta en kerennyt enää tällä viikolla tehdä näitä testauksia loppuun. Tällä hetkellä on menossa yhtä aikaa kolme projektia, joita edistän sitä mukaa kun se on mahdollista ja prioriteettien mukaan.

Jokaisessa testausprojektissa ei ole välttämättä tarvetta tietokantaan menemiseen tai sql kyselykielen osaamiseen, mutta tällä viikolla paremmasta tietokantaosaamisesta olisi ollut hyötyä. Koen että tietokantaosaaminen on myös hyvin tärkeä osaamisalue testaajalle, joten tämän osa-alueen kehittämiseen aion ehdottomasti tulevaisuudessa perehtyä.

Tällä viikolla myös Asiakas G:n osalta kävi ilmi, miksi yhteiset palaverit ovat tärkeitä pienemmissäkin päivitysprojekteissa. Tekemäni havainnot oli käyty läpi palaverissa, jossa en itse ollut läsnä ja sen jälkeen minulle kerrottiin, että näitä havaintoja ei korjata tässä versiopäivityksessä. Ihmettelin asiaa, sillä olihan kirjaamani vika este hyvin oleelliselle toiminnolle sidosryhmien hallinnan järjestelmässä. Olin kuitenkin ymmärtänyt, että kyseinen asiakas ei välttämättä ole käyttämässä tätä järjestelmää enää kauan, joten oletin, että järjestelmä on enää niin kevyellä käytöllä, ettei havainto sitten haittaa. Nyt kuitenkin, kun tällä viikolla järjestettiin projektiryhmän yhteinen palaveri ja otin havainnot esiin varmistaakseni, eivätkö nämä tosiaan haittaa versiopäivitystä, kävi ilmi, että haittaavat kyllä. Tässä kohtaa aloin miettiä mikä meni pieleen. Oma puutteellinen toiminta saattoi olla seuraava: Pohdin, oliko kirjattu vika tarpeeksi selkeä. Siinä oli kyllä mainittu mitä kyseinen häiriö aiheuttaa laajemmassa mittakaavassa, mutta ehkä sitä olisi pitänyt korostaa vielä paremmin. Toiseksi olisi pitänyt luottaa itseeseen enemmän havainnon vakavuuden suhteen ja varmistaa se tiimin yhteisellä Teams kanavalla. Nyt kun nämä testushavainnot käytiin lävitse jossain palaverissa, jossa en itse ollut läsnä, toisen näistä havainnoista oli ymmärretty liittyvän johonkin vanhaan havaintoon, joka ei ollut vielä korjaantunut ja siksi ajateltiin, että korjaus voidaan siirtää seuraavaan päivitykseen. Kun itse asiassa kyse oli uudesta ja paljon vakavammasta viasta. Eli kyseessä oli viestinnän ongelma, josta voin ottaa osittain omalle kontolleni vastuun ainakin sen suhteen, että selkeästi vian kirjaus ei ole ollut tarpeeksi selkeä, enkä toisaalta varmistanut asiaa koko projektitiimiltä vaan keskustelin vain järjestelmäasiantuntijan kanssa. Mutta toisaalta kyllä myös yleisenä huomiona siitä, miten tärkeää on, että nämä havainnot käydään läpi yhteisessä palaverissa koko projektitiimin kanssa väärinymmärrysten välttämiseksi.

Hyvän vikaraportin kirjoittamisesta löytyi ainakin seuraavia vinkkejä vastaisuu-  
den varalle: Vialla pitää olla yksilöllinen numero, jolla vikaa voidaan etsiä.  
Tämä tulee automaattisesti silloin, kun käytössä on vianhallinnanjärjestelmä, jol-  
loin aina uudelle luotavalle tehtävälle, on se sitten kehitystehtävä, testi tai vika,  
tulee oma yksilöllinen tunnisteensa. Tämä asia on siis kunnossa automaatti-  
sesti. Toiseksi vian tulee olla toistettava. Jos vikaa ei voida toistaa, silloin sitä ei  
koskaan korjata. Tämän vuoksi vikaraportille olisi hyvä kirjata askeleet, joilla  
vika toistetaan. Ei pidä olettaa tai hypätä yli yhtään askelta testin toistamisessa.  
Tällöin vika on helposti toistettavissa ja on helpompi etsiä vian syy. Tämä oli tie-  
tyllä tavalla kirjaamallani vika tehtävällä toteutettuna, sillä kirjoitin mistä toiminto-  
ketjusta virhe aiheutui, mutta toki stepit voisi kirjoittaa vielä selkeämmin nume-  
roituna allekkain. Kolmantena vihjeenä on pysyä oleellisessa, ongelmasta ei  
ole tarkoitus kirjoittaa esseetä, vaan pitää olla tarkka ja kirjoittaa vain oleellinen.  
Ei pidä yhdistää monia ongelmia, vaikka ne näyttäisivät olevan samanlaisia. Tu-  
lisi kirjoittaa eri raportit jokaiselle ongelmalle. (Software Testing Help. 2021.)

Viimeisen vinkin suhteen saattoi olla omassa kirjauksessani puutteita, sillä suu-  
rempi ongelma, joka johtui varsinaisesta otsikkokuvauksen ongelmasta, oli jää-  
nyt pienemmäksi maininnaksi kirjauksen loppuun. Eli vaikka kirjaus itsessään  
kuvasi koko tapahtumaketjun, se ei silti tullut huomatuksi niiden taholta, jotka vi-  
kakirjausta olivat tarkastelleet. Ongelma oli ehkä jo siinä, että itse otsikkokuvaus  
ei tuonut esiin suurinta ongelmaa. Lisäksi havainnolle tulisi kirjata aina priori-  
teetti, joka voi olla Blocker, Critical, Major, Minor tai Trivial (Software Testing  
Help. 2021).

Prioriteetti on myös pakollista lisätä jokaiselle kirjatulle testaushuomiolle, mutta  
tässä tapauksessa se oli väärä. Olin laittanut prioriteetiksi Major, joka olisi toi-  
minnallinen ongelma – mikä se toki myös onkin -, mutta oikea prioriteetti olisi ol-  
lut blocker, sillä ongelma esti tietyn toiminnallisuuden käytön kokonaan. Olin  
kuitenkin kirjannut prioriteetin ensin huomatuksen ongelman mukaan, enkä sen  
mukaan, mitä huomasin myöhemmin alkuperäisestä ongelmasta seuraavan.

Lisäksi kirjaukselle voidaan lisätä se mikä odotettu tulos oli ja mikä oli saatu tu-  
los. Myös kuvakaappaukset ovat tärkeitä, sillä kuvat kertovat usein enemmän



kuin tuhat sanaa. (Software Testing Help. 2021.) Kuvakaappauksia en ollut tälle bugikirjaukselle liittänyt, mutta lokille syntyneet virheet olivat liitteinä. Eli tiivistyksenä omalle kontolleni voin ottaa selvästi hieman huonosti tehdyn vikakirjauksen, ja sen, että olisi pitänyt viestiä koko projektitiimin kanssa jo aiemmin asiasta. Jatkossa täytyy kiinnittää enemmän huomiota tehtyihin vian kirjauksiin, jotta ne ovat yksilöllisiä ja selkeitä sekä varmasti oikein priorisoituja ja otsikoituja, mutta myös etenkin viestintäpuoleen koko projektiryhmän kesken.

### **3.13 Viikko 13 (16.11. -19.11.2021)**

#### **Maanantai 16.11.2021**

Tänään meni iso osa päivästä yrittäessä selvittää, miten asiakas G:llä rajapintojen testaus onnistuu. Pääsin jo eteenpäin, kun sain tunnukseni toimimaan, mutta sitten toisen järjestelmän käyttämisessä tuli eteen ongelmia, kun en tiennytkään mitä kaikkia tietoja täytetään mihinkin ja mikä kenttä siirtyy sidosryhmien hallinnan järjestelmään mistäkin kentästä. Vielä isompi ongelma tuli siitä, kun sidosryhmien hallinnan järjestelmässä pitäisi olla olemassa sama testikäyttäjä kuin tuossa toisessa järjestelmässä, jotta testin voisi tehdä, ja tällaista testikäyttäjää ei löytynyt, enkä tiedä miten saisin sellaisen luotua. Kyselin ympäriinsä asiasta eri henkilöiltä, mutta kukaan ei oikein tarkkaan tunnu tuntevan tätä prosessia, eikä asiakkaan puoleltakaan löytynyt henkilöä, joka olisi osannut asiassa auttaa. Niinpä loppupäivän testasin Asiakas J:llä, kun odotin vastauksia henkilöiltä, joille olin kysymyksiä laittanut.

#### **Tiistai 17.11.2021**

Tänään testasin aamupäivän Asiakas J:llä, mutta havaintoja testauksista ei tullut. Asiakas G:llä oli tullut asiakkaalta havainto, joka vaati pikaista korjausta ja minulta kysyttiin, pääsenkö testaamaan tuon tänään. Myös saatiin korjauspaketti erääseen havaintoon, joten loppupäivän vietin Asiakas G:llä. Jouduin myös käyttämään aikaa testisyklin siirtämiseen uuteen testienhallintajärjestelmään, koska näitä ei saatu automaattisesti siirrettyä. Tämän asiakkaan osalta viat täytyi jo kirjata uuteen järjestelmään, mutta ison osan vanhoista testeistä olin jo

kerennyt suorittamaan vanhalla puolella. Silti joitakin testejä on vielä suorittamatta, joten minun täytyy siirtää uuteen ympäristöön kaikki vanhat suoritukset ja niiden kommentit, jotta saan ne testilokille. Tämä vie aikaa, jota ei juuri nyt ylimääräisiä olisi, kun itse testaustyötä on niin paljon, mutta se on pakko tehdä, jotta saan raportit testauksen lopussa kuntoon. Uudelleentestattava toiminnallisuus toimi nyt oikein, joten pääsin myös suorittamaan testejä, jotka oli aiemmin jääneet blocked tilaan siksi, että niitä ei voinut testata toiminnallisuuden ollessa rikki. Tällä hetkellä työnteko tuntuu hyvin sekavalta pomppiessa projektista ja tekemisestä toiseen, kun yleensä tietyn asiakasprojektin saa melko hyvin keskittyä tekemään kerralla loppuun ja siirtyä vasta sitten seuraavaan. Nyt päällekkäin on kolmen projektin tekemistä.

### **Keskiviikko 18.11.2021**

Tänään testasin koko päivän Asiakas J:llä, yhden testaushavainnon tein ja kirjasin. Havainto liittyi sidosryhmien hallinnan järjestelmään ja organisaatioiden luomiseen. Liitin vika tehtävälle lokia ja ohjasin tehtävän kehittäjälle, laitoin myös erikseen viestiä hänelle, että havainto on tullut ja pitäisi selvittää.

### **Torstai 19.11.2021**

Tänään ennakolta suunnitelmissa on vain Asiakas G:lle tehdyn korjauksen testaus, sillä suurin osa päivästä kuluu tänään tiimipäivän merkeissä.

Aamusta kerkesin hieman aloitella tehdyn korjauksen testausta, mutta sitten kävi ilmi, että korjaukseen täytyy vielä tehdä muutoksia. Sitten pitikin jo lähteä tiimipäivään. Nyt on mennyt pitkä aika niin, että kaikki palaverit on pidetty etänä, enkä ole käynyt toimistolla ylipäätään pitkään aikaan. Olikin mukavaa nähdä tiimiläisiä, sillä joitakuista en ole tavannut kertaakaan kasvotusten korona-ajan vuoksi. Tämän opinnäytetyöpäiväkirjan osalta tämä tiimipäivä osuikin hauska juuri, kun päiväkirjan kirjoittamista ollaan lopettelemassa, sillä päivässä käsiteltiin mennyttä vuotta IAM-liiketoiminnan kannalta ja päivän lopussa, vapaamuotoisemmassa osuudessa jokainen tiimiläinen oli saanut valita kuvan, joka kuvastaa jollakin tavalla omaa työtä tai itseä omassa työssä. Tämä oli hauska

tapa oppia hieman lisää tiimikavereista. Jotkut olivat valinneet vitsikkäitä, itseironisia kuvia ja toiset taas ottaneet kuvan omalta kotityöpisteeltä lemmikin kanssa. Itselleni tuli ensimmäisenä mieleen kuvaa valitessa, että viimeisiä kuu-kausia on leimannut yksi iso kysymysmerkki, eli paljon on tullut uusia selvitettäviä ja opittavia asioita ja lisäksi kysymysmerkki myös mielestäni kuvaa testaajan työtä sinänsä hyvin, että testaajana sitä kysyy paljon kysymyksiä mielessään: miten tämän toiminnallisuuden kuuluisi toimia, onko tässä otettu nämä asiat huomioon, mitenkäs tämä nyt sitten toimiikaan? En kuitenkaan valinnut kysymysmerkkiä kuvakseni sillä se olisi ollut liian pelkistetty kuva. Tein eräänlaisen kuvasarjan siitä, miten työni on ollut uuden oppimista ja oivaltamista, sitten hetkeksi epätoivon alhoon putoamista sen suhteen miten paljon opittavaa ja tiedettävää vielä on, sitten toisaalta miten tärkeää testaajan työssä kommunikaatio toisten kanssa onkaan ja miten apu usein löytyy kollegaa konsultoimalla. Toisilta oppii paljon ja koen olevani tosi onnekas nykyisen työyhteisöni suhteen siitä, että kysymistä ei tarvitse pelätä vaan kaikki kollegat ovat avuliaita ja työpaikalla vallitsee toisia kunnioittava, ystävällinen ja rento ilmapiiri. Se ei suinkaan ole itsestään selvää, mutta koen, että työpaikan hyvä ilmapiiri on todella tärkeää ja olennaista jokaisen yksilön ammatillisen kehittymisen kannalta.

### **Viikkopäiväkirja seurantaviikko 13: 16.11.2021 – 19.11.2021**

Alkuviikkoani leimasi tällä viikolla melkoinen väsymys, sillä töissä on ollut kiirettä ja sen lisäksi opinnäytetyön tekeminen työn ohella on ollut syksylle melkoinen ponnistus. Kun siihen lisättiin yhtälö, että vastaan tuli tilanne, jossa minun pitäisi saada testattua ominaisuus, jonka testaamisessa eteen oli tullut paljon ongelmia ja tie tuntui nousevan pystyyn siinä, että tässä harvinaisessa tapauksessa oikein kukaan ei tuntunut tietävän miten tuo toiminnallisuus toimii, asiakkaan päässä yhteyshenkilöissä oli tapahtunut muutoksia ja sieltäkään ei löydetty henkilöä auttamaan, koin maanantaina hetkellisen väsähtämisen tunteen. Olin tässä asiassa etsinyt tietoa eri paikoista, sillä dokumentaatio on erinäisistä syistä johtuen hajallaan vähän siellä sun täällä ja yrittänyt koota kokonaiskuvaa siitä, miten voisin testauksen luotettavalla tavalla tehdä, mutta sitten oli asioita, joihin oikein mistään ei löytynytäkään dokumentaatiota.

Joskus, kun asiakkuuksissa tapahtuu joitakin muutoksia paljon ja samalla tulee vaihtuvuutta asioita aiemmin hoitaneisiin henkilöihin, voi käydä niin. Tämä asia jää edelleen selvitettäväksi, sillä tällä viikolla on tullut kiireellisempiä tehtäviä, jotka on pitänyt hoitaa ensin ja Asiakas G:n versiopäivitykselle ei ole mitään tarkkaa sovittua tuotantoon vieni päivämäärää, joten työpanosta on täytynyt kohdistaa kiireellisyyksien mukaan. Tällä hetkellä on ollut kaksi asiasta ikään kuin yhtä aikaa testauksessa ja sen lisäksi välillä tulee muita, yksittäisiä testattavia ja kun joutuu hyppimään asiasta toiseen kovin paljon, tulee helposti hallitsemattomuuden tunne omaan työhön.

Kuitenkin myös onnistumisen kokemuksia tuli tällä viikolla. Keskustelin Asiakas J:n pääkäyttäjän kanssa ja sain selvitettyä asioita niin, että pääsin asiakaskohtaisuuksien testaamisessa hyvin eteenpäin ja opin taas uusia asioita kyseisen asiakkaan asiakaskohtaisista toiminnallisuuksista. Tulee itselle testaajana paljon varmempi tunne, kun voi keskustella jonkun kanssa, jolle kyseinen asiakas on tuttu ja varmistaa että on ymmärtänyt oikein sen, miten toiminnallisuuden kuuluisi toimia ja että sen näin testaa oikein.

## 4 Pohdinta

Olen tarkastellut omaa työtäni ja työn tekemisen tapani kolmentoista viikon ajanjakson ajan. Tähän aikaan on sisältynyt paljon uuden oppimista, oivalluksia, iloa siitä, että on huomannut oman osaamisen kehittyvän, mutta toisaalta myös väsymyksen hetkiä, sillä ei ole ollut ihan kevyttä tehdä samaan aikaan töitä ja kirjoittaa opinnäytetyötä muiden elämän haasteiden lisäksi.

Olen oppinut paljon uusia asioita testatusta järjestelmästä, AD:sta ja toisaalta ihan testauksen hyvistä käytänteistä vertaamalla omaa testausprosessia siihen, mitä kirjallisuudessa testausprosessin sisällöistä kerrotaan. Kaikkia testausprosessin osa-alueita, joihin kirjallisuudessa törmäsin, ei ole itselläni käytössä. Osa niistä on sellaisia, joita ei edes voi hyödyntää kaikissa projekteissa.

Testausprosessi voi olla raskaampi tai kevyempi projektista riippuen. Jos tekisin töitä tuotekehityksen puolella, olisi prosessi varmasti hyvin toisen näköinen kuin tällä hetkellä. Tuotekehityksen puolella testaja tekee varmasti enemmän dokumenttien katselmointia, mutta asiakasprojektien puolella tällaista työtä ei juurikaan tule. Staattiseen testaukseen liittyviä tehtäviä ei juurikaan itselläni ole ja ei-toiminnallisia testauksia ei myöskään asiakastestausvaiheessa juurikaan tehdä. Oma tekemiseni pohjautuu siis paljon järjestelmän toiminnalliseen testaamiseen. Iso osa omaa työtäni ovat myös testauksen suunnitteluun ja testianalyysiin liittyvät vaiheet, jossa käydään lävitse mitä missäkin versiopäivityksessä pitää testata. Asiakasprojektien puolella tekeminen on hyvin projektiluontoista, tekemiseen on olemassa laadunhallinnan mukaiset kuvatut prosessit, ja asiakasprojektit pyörivät tämän prosessin mukaan. Omassa työssäni monessa projektissa mennään melko kevyellä testauksella. Esimerkiksi suurimmassa osassa projekteja testauksen suunnitteluvaiheeseen kuuluvia kuvattuja töitä ei juurikaan tule, eikä oikeastaan ole tarpeenkaan tulla, koska projektit ovat niin pieniä. Muunkinlaisia projekteja on tulossa, joten odotankin mielenkiinnolla sitä, miten soveltaa opittuja tietoja projektiin, jossa testausprosessi on perusteellisempi.

Ehkä suurimpana oppina ja pääällimmäisenä ajatuksena itselleni jäivät tietyt keskustelut ja kollegoiden sanat hetkissä, kun itselle heräsi tunne ”osaanko vielä mitään”. Pitkään talossa ollut työkaveri sanoi, että jopa hänelle tulee uusia asioita vastaan päivittäin. Tämä oli itselle rohkaiseva ja lohduttava keskustelu. Toisaalta myös muiden kollegoiden sanoja on jäänyt mieleen siitä, että asiakkaiden eteen tehdään töitä yhdessä ja kukaan ei tee tätä yksin, kenenkään ei tarvitse osata kaikkea yksin. Aina löytyy joku, joka osaa auttaa tai ainakin tietää jonkun, joka osaa. Tärkeintä on asenne. Tärkeintä on se, että haluaa selvittää ja tehdä asiat loppuun asti hyvin. Muut asiat ratkeavat, kaiken muun voi oppia ja kehittymistä ja ammatillista kasvamista tapahtuu koko loppuelämän ajan. Koko ajan tulee uutta, se on totuus. Ei voi koskaan tuudittautua siihen, että olisi valmis ja osaisi kaiken, sillä kaikki kehittyy koko ajan, tulee uusia asioita, jotka on pystyttävä ottamaan haltuun.

Joskus tulee epäonnistumisia, hetkiä, joina tuntuu, että vika, jonka asiakas on hyväksyntätestauksessa havainnut, olisi pitänyt itse havaita oman testauksen

aikana. Mutta nämä ovat myös kehittymisen lähtökohtia. Kun käy läpi huomioita, voi luokitella ne karkeasti kahteen ryhmään.

Toinen ryhmistä on: Minun olisi ollut hyvin vaikea havaita tätä. Tällaisia voivat olla esimerkiksi ei toiminnalliseen testaukseen liittyvät, käytettävyyteen tai suorituskykyyn liittyvät huomiot, jotka ovat esimerkiksi ongelmia juuri tälle kyseiselle asiakkaalle, mutta eivät välttämättä muille. Ongelma saattaa liittyä esimerkiksi datamääriin ja tulla esiin vain tuotantoympäristössä (testiympäristön dataa ei aina saada vastaamaan tuotantoympäristön vastaavaa). Ongelma voi olla myös jollakin tietyllä selaimella, jota asiakkaan testiympäristössä ei ole ollut käytettävissä. Yleensäkin kaikkia manuaalisia testejä ei ole mahdollista kaikilla selaimilla suorittaaakaan, siihen ei aika riitä.

Toisaalta on vikoja, jotka voin liittää kategoriaan ”tämä olisi pitänyt löytää tai tämän olisin voinut löytää ja voin löytää tällaiset jatkossa, jos toimin tavalla x”. Tämän kategorian tapaukset vaativat jatkopohdintaa omalta osalta sen suhteen, oliko suunnitelluissa testitapauksissa otettu huomioon kaikki oleellinen, tai mikä muu on saattanut vaikuttaa siihen, etten ole havainnut ongelmaa.

Toisaalta pitää myös pitää mielessä se, että vaikka kuinka perusteellisesti haluaisikaan testata, aina on olemassa aikarajat sille miten perusteellista testausta voi tehdä, eikä kaikkea voi testata. Tällöin testaus vähentää vikojen todennäköisyyttä, mutta ei koskaan voi löytää kaikkia mahdollisia järjestelmän vikoja. Tämän vuoksi olisikin itse asiassa todella hyvä, jos jossakin vaiheessa pääsimme myös asiakastestauksessa siihen, että asiakaskohtaiset regressiotestit saataisiin automatisoitua. Tällöin testaajalle jäisi versiopäivitysprojektissa enemmän aikaa muulle testaukselle.

Pidän työstäni ja haluan edelleen jatkaa kehittymistä testaajana. Se, minkä havaitsin omasta mielestäni kehityskohteeksi tai asiaksi, joka voisi tehostaa kehittämistä, on yhteistyö tuotekehityksen testaajien kanssa. Olisi hyvä olla yhteinen palaveri aina kun uusi SR-päivitys tulee julki, sillä tässä kohtaa tuotekehityksen testaajilla on jo hyvä näkemys siitä, mihin kaikkialle tehdyt kehitykset ja

korjaukset vaikuttavat. Tällöin asiakastestauksen ei tarvitsisi pähkäillä asioita erikseen, joka voisi tehostaa testausta.

Toiseksi itse asiakasprojekteissa, myös pienemmissä, olisi hyvä olla jonkinlainen lopetuspalaveri, jossa voitaisiin käydä yhdessä lävitse tehdyt havainnot ja vielä varmistaa se, että kaikilla on yhteinen ymmärrys siitä mitä havainnoille tehdään ja milloin ja mitä tapahtuu seuraavaksi. Muista jatkossa kehitettävistä osa-alueista tulee kyseeseen henkilökohtaisessa kehittämisessäni testaajana ainakin tietokantaosaamisen kehittäminen sekä mahdollisesti testausautomaatio osaamisen kehittäminen. Myös integraatiotestaus on selvästi ollut pinnalla viime kuukausina, eli rajapintatestaustyökalujen opettelu on yksi tulevaisuuden haltuun otettavista taidoista.

Haasteena tänä syksynä on ollut hieman töiden epätasainen jakaantuminen. Alkusyksystä ei ollut paljon projekteja, joten aikaa saattoi käyttää osaamisen kehittämiseen. Toisaalta loppuvuodesta tuli niin paljon projekteja, että osaa on joutunut tekemään päällekkäin. Tähän saattaa olla monia syitä, mutta ainakin tämän syksyn aikana tiimi on puhaltanut hyvin yhteen hiileen sen suhteen, että sovitut työt saadaan tehdyksi. Jatkossa tämäkin on otettu huomioon, jotta projektit jakaantuisivat tasaisemmin vuoden mittaan.

Kaiken kaikkiaan opinnäytetyön kirjoittaminen on ollut hyvin opettavainen kokemus, sillä se on asettanut minut tarkastelijan paikalle oman työni suhteen. On tullut kiinnitettyä huomiota siihen, mitä voisi jatkossa tehdä paremmin ja mikä toisaalta ehkä sujuu nyt hyvin. On tullut selväksi, mitä asioita omassa osaamisessa olisi vielä jatkossa hyvä kehittää ja toisaalta kiinnitettyä huomiota siihen miten paljon syksyn aikana on oppinut uutta. Tämän projektin päättyessä jatko-suunnitelmia oman osaamisen kehittämisen osalta on edelleen paljon.

## Lähteet

- FISTB/ISTQB. 2018. ISTQB Foundation Syllabus 2018 – Sertifioitu Testaaja Perustason sertifikaattisisältö. <https://tivia.fi/fistb-testi/wp-content/uploads/sites/30/2020/12/CTFL-2018-Sertifikaattisisalto-20181010-1-Valmis.pdf>. 17.9.2021.
- Graham, D., Black, R. & van Veenendaal, E. 2020. Foundations of software testing: ISTQP certification, 4th edition. Cengage Learning EMEA.
- Jobskillshare Community. 2021. IT Support Interview Question: What is local vs Domain user accounts. <https://www.youtube.com/watch?v=6CdIkIQdmx8>. 1.10.2021.
- Jobskillshare Community. 2019. Active Directory Training For Entry Level Help. Desk. <https://www.youtube.com/watch?v=zrCNGzwLjcU>. 1.10.2021.
- Microsoft. 2018. User Naming Attributes. <https://docs.microsoft.com/en-us/windows/win32/ad/naming-properties>. 1.10.2021.
- Microsoft. 2021. Security Account Manager. <https://www.windows-active-directory.com/windows-security-account-manager.html>. 1.10.2021.
- Robot Framework Foundation. 2021. Introduction. <https://robotframework.org/#introduction>. 1.10.2021.
- Software Testing Help. 2021. How To Write A Good Bug Report? Tips And Tricks. <https://www.softwaretestinghelp.com/how-to-write-good-bug-report/>. 20.11.2021.
- Techopedia. 2020. What does Domain Controller (DC) Mean? <https://www.techopedia.com/definition/4193/domain-controller-dc>. 1.10.2021.
- TietosuojaValtuutetun toimisto. 2021. Usein kysyttyä EU:n tietosuoja-asetuksesta. <https://tietosuoja.fi/gdpr>. 17.9.2021.
- Varinder, K. 2021. Identity and Access Management. Udemy. <https://www.udemy.com/course/identity-access-management-learn-oauth-openidsaml-ldap/learn/lecture/15430226#overview>. 1.10.2021.