



# Päiväkirjaopinnäytetyö sovelluskehittäjän työstä asiakasprojekteissa

Vili Muotka

2022 Laurea



Laurea-ammattikorkeakoulu

Päiväkirjaopinnäytetyö sovelluskehittäjän työstä asiakasprojekteissa

Vili Muotka  
Tradenomi  
Opinnäytetyö  
Toukokuu, 2022

Tässä päiväkirjaopinnäytetyössä seurataan ohjelmistokehittäjän asemassa aloittaneen työntekijän työarkea asiakasprojekteissa. Opinnäytetyöhön sisältyy 10 viikon seurantajakso, jonka ajalta on kirjoitettu päivittäiset päiväkirjamerkinnot. Viikkojen alussa on kerrottu tulevan viikon tavoitteista ja viikkojen lopussa on pyritty arvioimaan kulunutta viikkoa. Seurantajakso ajoittui 7.2. - 15.4.2022.

Tavoitteena oli kehittää työntekijän tietoja ja taitoja käytännön asiantuntijatehtävässä. Kehityskohteisiin lukeutui kehitettävän sovelluksen koodikantaan perehtymistä, muutospyyntöjen mukaisten toimenpiteiden toteuttamista, testien kehittämistä ja tuotantoympäristössä toimimista. Opinnäytetyön tietoperustana käytettiin C# ohjelmointikielen ja käytettyjen ohjelmistokehityksien dokumentaatiota.

Opinnäytetyön tuloksena työntekijä kehittyi sovelluskehittäjän tehtävissä ja pohti asiantuntijatehtävän eri osa-alueita. Pohdintaa syntyi kehittämisen haasteista, testien tärkeydestä ja työssä kohdatuista stressaavista tekijöistä.

Asiasanat: sovelluskehitys, agile, scrum, asiakasprojekti

Laurea University of Applied Sciences

Abstract

Degree Programme in Business Information Technology

Bachelor's thesis

Vili Muotka

**Diary-form thesis about software developer's work in client projects**

Year

2022

Pages

38

---

This diary-form thesis followed the workdays of a beginner software developer working on customer projects. The thesis consisted of a 10-week monitoring period, which included daily diary entries. The goals for the week were listed at the beginning of each week. The employee aimed to evaluate the past week at the end of the week. The monitoring period was between 7.2. - 15.4.2022.

The goal was to develop the employee's knowledge and skills by working in the professional field. Improvement targets were to get acquainted with the code base of the application that is being developed, making code changes fit the given criteria, test development and working in the production environment. Most of the sources were *C#* or software framework documentation.

As a result of this thesis, the employee improved as a software developer and reflected on different sectors of the job, eg. the challenges of software development, the importance of writing tests and causes of stress faced while working.

Keywords: software development, agile, scrum, customer project

## Sisällys

1	Johdanto.....	6
2	Nykytilanne.....	6
2.1	Nykyinen työ ja osaaminen .....	6
2.2	Sidosryhmät .....	9
2.3	Vuorovaikutustaidot .....	10
2.4	Kehittäminen .....	10
2.5	Opinnäytetyön tavoitteet .....	11
3	Päiväkirjaraportointi.....	11
3.1	Seurantaviikko 1.....	11
3.2	Seurantaviikko 2.....	13
3.3	Seurantaviikko 3.....	15
3.4	Seurantaviikko 4.....	18
3.5	Seurantaviikko 5.....	21
3.6	Seurantaviikko 6.....	24
3.7	Seurantaviikko 7.....	25
3.8	Seurantaviikko 8.....	28
3.9	Seurantaviikko 9.....	30
3.10	Seurantaviikko 10 .....	33
4	Yhteenveto ja pohdinta .....	35
	Lähteet.....	37

## 1 Johdanto

Tämä opinnäytetyö seuraa sovelluskehittäjän toimintaa asiakasprojektissa. Päiväkirjan 10 seurantaviikkoa kohdistuvat aikavälille 7.2.2022 - 15.4.2022. Opinnäytetyö tulee kuvaamaan kehittäjän päivittäistä työskentelyä. Viikkojen lopussa tarkastellaan kulunutta ajanjaksoa ja omaa kehittymistä.

Samlink on noin 400 työntekijän pankki- ja finanssialan palveluntuottaja. Yritys oli alun perin pankkien omistuksessa, mutta siirtyi Cognizantin omistukseen vuonna 2019 (Samlink 2019). Vuonna 2021 Kyndryl ilmoitti aikeistaan ostaa Samlinkin (Kyndryl 2021). Vuoden 2022 alussa yrityksen omistajuus vaihtui Kyndrylille.

Yrityksen toimiessa pankki- ja finanssialalla, on luottamuksellisuus ja tietoturva erittäin tärkeitä jo lainkin puolesta. Palveluiden on toimittava luotettavasti ja on oltava varautuneita mahdollisiin ongelmiin.

Opinnäytetyön aikana työstän monia ohjelmistokehityksen osa-alueita. Seurantaviikkojen aikana osallistun tiimin scrum-tapaamisiin kuten päivittäispalavereihin, sprintin suunnittelupalavereihin ja retroihin.

Sovelluskehittäjänä iso osa työstäni on muutosten toteuttamista. Tämä alkaa asiakkaan muutospyyntöstä, sen työmääräarvioinnista ja suunnittelusta. Muutosten toteuttamisen jälkeen se voi myös palata vielä takaisin kehitettäväksi testaaajalta.

Tiimi vastaa kehittämämme sovelluksen tuotannon hoidosta. Tähän kuuluu esimerkiksi uuden sovellusversion asennus tuotantoon ja tuotannossa ilmentyvien virheiden korjaaminen vuorolleen.

Tehtävissäni tarvitsee ohjelmointitaitoja, ongelmanratkaisukykyä, tiimityöskentelytaitoja ja ymmärrystä pankki- ja finanssialasta. Seurantajakson tavoitteena on perehtyä tarkemmin sovelluksen koodiin ja kirjoittaa parempaa koodia. Myös tuotantoympäristössä toimimiseen totutellaan seurantajakson aikana.

## 2 Nykytilanne

### 2.1 Nykyinen työ ja osaaminen

Työnkuvaani on kuulunut tähän mennessä monia tehtäviä:

- testaamista
- kehittämistä
- tuotannon häiriöiden hoitoa
- ympäristöjen päivittämistä
- perehdyttämistä

## Testaus

Aloittaessani työt kesäkuun puolessa välissä, tulin nykyiseen tiimiini testajaksi. Testajan tehtäviin kuuluu kehittäjien muutosten vertaaminen käyttäjätarinoihin. Toteutuksen tulee vastata tarinalla kuvattua toiminnallisuutta. Sen lisäksi muutosten tulee tietysti toimia, eikä ne saa rikkoa muita toiminnallisuuksia.

Tiimissämme testajien tekemä testaus tapahtuu pääasiassa manuaalisesti. Tarinoiden pohjalta kirjoitetaan testaussuunnitelmat, joissa pyritään ottamaan huomioon kaikki muutokseen mahdollisesti liittyvät skenaariot. Tämän jälkeen muutokset testataan testaussuunnitelman mukaisesti.

Viime vuoden lopussa alkoi projekti, jonka mukana tuli tarve tuottaa asiakkaalle muutoksista testiraportit. Tätä varten kehitin kyseiseen projektiin sopivan raportointimallin, joka on edelleen käytössä uudella testajalla.

Olen testajana suoriutunut tehtävistäni hyvin ja saanut positiivista palautetta. Täytin tiimin tarpeet ja vaatimukset hyvin. Ymmärsin tehtävät ja suoriuduin niistä hyvin itsenäisesti. Olen osallistunut uuden testajan perehdytykseen ja pystyn ohjeistamaan häntä tarpeen mukaan.

## Kehittäminen

Siirryin tammikuun alkupuolella testajan asemasta kehittäjän rooliin. Jatkoin samassa tiimissä ja saman sovelluksen parissa, joten pystyn hyödyntämään isoa osaa testajana oppimistani asioista kehittäjän tehtävissä.

Kehittäjän tehtävissä ensimmäiseksi tulee mieleen koodaaminen. Perustehtäviin kuuluu uusien ominaisuuksien toteuttaminen käyttäjätarinoiden kuvausten mukaan, jotka tuotteen omistaja kirjoittaa asiakkaan vaatimusten perusteella. Ominaisuuden toteuttaminen sisältää koodiin perehtymistä, toimivan toteutustavan selvittäminen ja itse koodimuutosten kirjoittaminen.

Myös kehittäjiä osallistutetaan tarinoiden suunnittelussa. Kun asiakas pyytää uutta ominaisuutta, tulee se tiimillemme ensin työmääräarvioitavaksi. Työmääräarviointi toteutetaan yleensä yhteisessä tilaisuudessa, joko sprinttisuunnittelun aikana tai sprintin refinement -tilaisuudessa. Arvioinnissa otetaan huomioon, kuinka haastavaa ja kuinka työlästä halutun

muutoksen toteutus tulee olemaan. Koskeeko muutos olemassa olevaa toiminnallisuutta vai vaatiiko se uuden koodaamista. Työmääräarviointi lähetetään asiakkaalle hyväksyttäväksi.

Kun toteutuslupa on saatu ja muutos aiotaan toteuttaa, otetaan tarina tarkemmin suunnittelun alle sprinttisuunnittelussa. Suunnittelussa mietitään, miten muutosta lähdetään toteuttamaan. Muutospaikat etsitään valmiiksi. Suunnittelun avulla tiimin työskentely on myös yhteisempää.

Uusien ominaisuuksien toteuttamisen lisäksi kehittämiseen kuuluu bugien korjausta. Ensimmäisen kuukauden aikana kehittäjän tehtävissä en ole bugeja vielä joutunut korjaamaan. Kehittäjän tehtävistä suoriudun kohtalaisen hyvin. Tarvitsen välillä vielä apua tehtävien suorittamiseen. Välillä haluan vain varmistua, että toteutustapani on hyvä, tai tietää mitä voisin tehdä toisin.

### **Tuotannon häiriöiden hoitoa**

Tiimissämme on viisi kehittäjää, joista jokainen on vuorollaan viikon ajan tikettivastaavana. Häiriötikettien tullessa teknisen tuen kautta tiimille, lähtee tikettivastaava niitä ensimmäisenä selvittämään. Häiriötiketit ovat yleensä pieniä ja koskevat vain yksittäistä rahoitusta taikka eräajoa. Rahoitusmaailmassa viat pitää kuitenkin korjata nopeasti ja tuotanto meneekin aina muun tekemisen edelle. Häiriötikettien hoitamisessa kyselen vielä paljon apua työka-vereiltani.

Teknisen tuen kautta tulee myös välillä tilauksia, joissa pitää korjata viallisia tietoja rahoituksilta tai osapuolilta. Rahoitukset ovat lainoja tai takauksia, joita kehittämällämme soveluksella voidaan käsitellä.

### **Ympäristöjen päivittäminen**

Oman tiimin testiympäristöt päivittyvät automaattisesti joka aamu. Tämän lisäksi ne saa manuaalisesti päivitettyä, mikäli katselmoinnista tulee päivän aikana lisää testattavaa. Myös asiakkaalla on testiympäristöt. Ne eivät kuitenkaan päivity joka aamu, vaan ne päivitetään manuaalisesti asiakkaan pyynnöstä. Asiakastestiympäristöjen päivitys onnistuu yleensä ongelmitta, mutta joskus asennus ei ole mennyt läpi.

Tiimissämme yleensä testaaja on päivittänyt ympäristöt. Ensimmäisen puolen vuoden aikana niitä tuli päivitettyä usein, lähellä tuotantoon menoa noin kerran viikossa. Kehitystehtäviin siirtymisen jälkeen tulen tuskin päivittämään testiympäristöjä enää kovin usein. Tuotantoa en ole vielä päivittänyt. Siihen liittyy vielä useita eri valmistelevia askeleita, joita en osaa vielä tehdä.

### **Perehdyttäminen**

Vaihdettuani testaaajan tehtävistä kehittäjän tehtäviin, tiimiimme liittyi tilalleni uusi testaaaja. Hänet on pääasiassa perehdyttänyt esimieheni, mutta myös minä olen osallistunut perehdyttämisprosessiin. Joulukuussa aloitettiin uusi projekti, jota esimieheni ei ole itsekkään testannut, joten pääsin perehdyttämään heitä molempia kyseiseen aiheeseen. Sen lisäksi olen auttanut testaajaa eteen tulleissa haasteissa ja vastannut kysymyksiin.

### **Kehittyminen**

En ole vielä kauaa ollut alalla töissä ja vielä vähemmän nykyisissä tehtävissä. Paljon on opittavaa. Eteen tulee usein uutta, jonka tekemiseen tarvitsee aikaa myös perehtymiseen. Ohjelmoinnissa koodikatselmointien yhteydessä kokeneemilta kehittäjiltä saa usein parannusehdotuksia niin senhetkiseen toteutukseen, kuin yleisiin koodikäytänteisiin. Koitan näitä keräillä ja sisäistää parhaani mukaan, jotta tulevaisuudessa kirjoittamani koodi olisi mahdollisimman luettavaa ja tehokasta. Töissä on käytössä minulle uusi ohjelmointikieli. Lisäksi kehittämämme sovellusta on työstetty jo kymmenisen vuotta, joten koodiperusta on hyvin laaja. Perehtymistä riittää.

## **2.2 Sidosryhmät**

### **Sisäiset sidosryhmät**

- oman tiimin jäsenet
  - kehittäjät
  - testaaajat
  - tuotteenomistaja
- asentajat

Tiimin jäsenien mielipiteet vaikuttavat työskentely- ja toteutustapoihin. Tiimin kanssa sovitaan yhteisistä käytänteistä ja kehitetään toimintatapoja. Tuotteenomistaja määrittelee vaatimukset tarinoille ja testaus tarkistaa tehdyt muutokset annettujen vaatimusten perusteella.

### **Ulkoiset sidosryhmät**

- asiakas
- loppukäyttäjät

Asiakkaalta tulee jatkokehityspyyntöjä sovelluksen toiminnallisuudelle. Heiltä saattaa tulla isompia projekteja tai pienmuutospyyntöjä. He suorittavat myös omaa hyväksymistestausta tekemiimme muutoksiin.

Loppukäyttäjät ovat asiakkaan sovellusta käyttäviä työntekijöitä. Heiltä saattaa tulla joskus bugi-ilmoituksia. Muutospyynnöt tulevat kuitenkin yleensä asiakkaan sovelluksesta vastuussa olevan tiimin kautta.

### 2.3 Vuorovaikutustaidot

Eniten töissä ollaan tekemisissä oman tiimin jäsenten kanssa. Vaikka työskentely on pääasiassa itsenäistä, tapaa tiimi joka aamu päivittäispalaverissa, eli dailyssa, jossa jaetaan töiden etenemisen tilanne. Yleinen tilanne on myös, että pyydetään työkaverilta apua tai neuvoa. Etänä työskennellessä viestiminen tapahtuu Teamsin välityksellä joko chatin tai puheluiden kautta. Pääsen myös olemaan avuksi välillä, kun tiimiin liittyneet uudet jäsenet tarvitsevat apua jossakin. Myös vanhemmat jäsenet saattavat kysyä, miten jokin tilanne saadaan toistettua sovelluksessa, sillä oma osaamiseni sovelluksen käytöstä on aika laaja testaajan tehtävieni vuoksi.

Tuotteenomistajan kanssa en ole kovin paljoa ollut kanssakäymisissä dailyjen lisäksi. Pari kertaa on testattu jotain yhdessä ja välillä on tarvinnut kysyä tarkennusta työn alla olevaan tariinaan. Asiakkaan kanssa olen joutunut olemaan tekemisissä vasta vähän. Jokainen tiimin kehittäjä on vuorollaan viikon tikettivastaavana. Olen itse ollut vasta kerran. Viikon aikana ei tullut kovin montaa tikettiä, joten kommunikoitavaa heidän suuntaansa oli aika vähän.

Tiimin kanssa kommunikoidessa ei tule paljoa haasteita vastaan. Meillä on hyvä porukka, jonka kanssa on mukavaa olla tekemisissä. Apua pyytäessä täytyy toki ymmärtää, mikä on ongelmana, jotta tilanteen osaa kuvata toiselle. Olen kuitenkin toistaiseksi selviytynyt tästä kohtuullisen hyvin, yleensä minulla on saattanut kysyessä olla jopa oma ratkaisutapa mielessä, jonka toteuttamiseen olen vain kysynyt suuntaviivoja tai mielipidettä.

Asiakkaan kanssa kommunikointi on haastavampaa. Heidän esittämänsä kysymykset saattavat olla usein vaikeita. Pitää myös miettiä tarkemmin, mitä vastaa, sillä edustan vastatessani yritystäni. Kiperissä tilanteissa kysyn tuotteenomistajan, tai jonkin muun kokeneemman tiimin jäsenen apua.

### 2.4 Kehittäminen

Aloitimme viime vuoden loppupuolella uuden projektin. Asiakkaalla on tarve lokittaa, mitä tietoja käyttäjät katselevat sovellusta käyttäessään. Tähän projektiin todettiin myös tarve toteuttaa jokaisesta tarinasta testiraportti, jotka toimitetaan asiakkaalle. Tehtävieni myötä kehitin projektille sopivan raportointimallin, jolla näytetään lokien toimivuus ilman liiallista lisätyötä. Sen lisäksi olen tehnyt Excel-pohjan helpottamaan eräiden sovelluksesta ulosviettävien tietojen tarkastelua.

Olemme myös lisänneet tarinoille vaatimuksia siitä, mitä tietoja sen tulisi sisältää, jotta tiimin uudet jäsenetkin selviävät tehtävistä helpommin ja ymmärtävät, mitä tarinalla pyydetään. Esimerkiksi mahdollinen muutokseen liittyvä käyttöliittymän osa tulisi ilmaantua tarinasta.

## 2.5 Opinnäytetyön tavoitteet

Haluan yleisesti kehittyä uusissa tehtävissäni sovelluskehittäjänä. Haluan olla itsevarmempi osaamiseeni suhteen. Haluan myös panostaa koodini luettavuuteen, sillä se on yksi ominaisuus, jota arvostan paljon. Kehittämämme sovellus sisältää paljon koodia, johon tutustuminen ja ymmärtäminen on tärkeää. Koodin tuottamisen lisäksi myös sen testaaminen yksikkötesteillä on tärkeä osata. Vaikka tuotantoympäristöjen ylläpito ei ole kovin mieluisaa, se on tärkein osa työtämme. Tuotantoympäristössä toimiminen on yksi asia, jota opinnäytetyön aikana tulen opettelemaan.

Toimeksiantajaa kiinnostaa katsaus tiimin työskentelyyn ja mahdollisesti sen myötä syntyviä kehitysideoita tiimin prosesseihin.

## 3 Päiväkirjaraportointi

### 3.1 Seurantaviikko 1

#### **Maanantai, 7.2**

Tein viime viikolla selvitystyötä erään bugin parissa. Sain bugin toistettua, joten tänään on tarkoitus käydä tuotteenomistajan kanssa bugin toisto läpi ja katsoa askelmerkit sen korjaamiseen. Täytyy myös tarkistaa bugin toistuvuus toisessa mahdollisessa tilanteessa.

Työpäivä alkoi päivittäispalaverilla eli dailyllä, jossa tiimin jäsenet päivittivät etenemistään ja tämän päivän suunnitelmia. Tästä jatkettiin suoraan sprintin suunnitteluun. Viime sprintin aikana oli aika paljon tuotannon hoitoa, joten aikaisemmalta sprintiltä siirtyi kehittämistä tälle sprintille. Katsoimme yhdessä kehittäjien kesken uusien tikettien mahdollisia toteutustapoja.

Dailyn jälkeen valmistelin omaan kehitysympäristööni tilanteen, jolla aikaisemmin mainitun bugin saisi toistettua. Iltapäivällä katsoimme tuotteenomistajan kanssa yhdessä bugin toistoaskeleet ja sain hyväksynnän bugin korjaamiselle. Päivän päätteeksi tutustuin aikaisempiin muutoksiin kyseiseen toiminnallisuuteen ja aloitin bugin toistuvuuden testaamista toisessa mahdollisessa tilanteessa.

#### **Tiistai, 8.2**

Tänään olisi tarkoitus jatkaa eilisen bugin parissa. Kyseinen toiminnallisuus palauttaa SQL-kutsusta rivejä, joita ei olisi tarkoitus kannasta hakea. Korjauksessa pitää olla tarkkana, että halettuja rivejä ei suodateta pois. Tämän lisäksi pitäisi sopia asiakkaan kanssa tuotannossa erään tiedon korjauksen ajankohdasta.

Tänään dailyn jälkeen katsottiin yhdessä parin kehittäjän kanssa erästä bugia, jonka parissa toinen heistä on työskennellyt. Katsottiin yhdessä läpi, mitä ongelmasta oli selvinnyt ja pohdittiin, miten sitä voisi lähteä korjaamaan.

Sain tänään tehtyä korjauksen eilen selvittelemääni bugiin. Muutos koodiin ei ollut suuri, vain kyseisen SQL-lauseen muokkaus niin, ettei se palauta kuin halutut rivit. Korjaus odottaa vielä katselmontia toisen kehittäjän toimesta ja testaaajan verifiointia.

Laitoin myös aamupäivällä viestiä asiakkaalle tietojen korjausajankohdasta. Ehdotin heille ajankohdaksi huomista kello 13. Iltapäivällä asiakkaalta tuli hyväksyntä ajankohdalle, joten tein muutoshallinnalle muutoslupapyynnön. Pyysin myös asentajaa huomista varten. Päivän lopuksi aloitin tutustumaan seuraavaan bugiin.

## **Keskiviikko, 9.2**

Tänään tärkeimpänä on kello 13 ajettava tietojen muutos tuotantoon. Muutos täytyy ajaa asiakkaan toisen palvelun kanssa samaan aikaan. Muun ajan työskentelen eilisen bugin parissa, joka on jälleen SQL-hommia. Tänään on myös viikon kohokohta, eli tiimin kahvitaukio. Siellä tiimi voi rentoutua ja puhua työhön liittymättömistä asioista.

Tietojen korjauksen yhteydessä tuli pieniä ongelmia vastaan. Virhe onneksi selvisi nopeasti tiimikaverin avulla ja korjaus saatiin tehtyä, eikä tarvinnut sopia uutta asennusajankohtaa. Tämän päivän bugi vaatii pientä muutosta erääseen datankäsittelytyökaluumme. Lisättyäni muutaman rivin SQL-lauseeseen testasin vielä muutosta omissa ympäristöissäni, ennen kuin laitoin muutoksen katselmontiin.

## **Torstai, 10.2**

Tänään pitäisi alkaa työstämään seuraavaa muutosta. Tarpeena on toteuttaa viimeisimmän projektimme automaattinen dokumentointi. Dailyssä testajaamme pyysi minulta apua, joten otimme jälkepäin puhelua, jonka aikana autoin häntä eteenpäin.

Automaattisen dokumentoinnin tekeminen on haastavaa, mutta etenin siitä kuitenkin tänään hyvin. Toteutan automaattista dokumentointia komentorivityökalun muodossa. Päivä sisälsi paljon Microsoftin dokumentaation sekä StackOverflow:n lukemista sekä eri ratkaisuiden koittamista. Päivän lopuksi kuitenkin sain kerättyä tarvittavat tiedot dokumentointia varten.

## **Perjantai, 11.2**

Tänään jatkan eilen aloittamaani automaattisen dokumentoinnin konsolisovelluksen toteuttamista. Sain eilen kerättyä tarvittavat tiedot dokumentointia varten ja tänään pitäisi toteuttaa tietojen tulostus CSV-tiedostoon.

Sain tulostettua tiedot CSV-tiedostoon. Huomasin kuitenkin, että tarinalla oli tarpeena myös pystyä näkemään lokeihin tehdyt muutokset viimeisimpään dokumentointiversioon verrattuna. Lisäsin siis vielä konsolisovellukseen mahdollisuuden syöttää aiemman version, johon sovellus vertaa nykytilannetta. Maanantaille jäi vielä toteutettavaksi tulevaisuudessa mahdollisesti poistettavien lokien lisääminen muodostettavaan dokumenttiin.

### **3.2 Seurantaviikko 2**

Tällä viikolla aloitan lokien muodostumisen automaattisen testauksen toteutuksen. Tarkoituksena on lisätä nykyisiin testeihin askel, joka tarkistaa lokimerkintöjä muodostavien tapahtumien lähetyksen sovelluksesta.

## **Maanantai, 14.2**

Tänään viimeistelen automaattisen dokumentoinnin. Sen jälkeen pitäisi aloittaa lokien automaattisen testauksen toteuttaminen.

Dailyn jälkeen lisäsin poistettujen lokien erittelyn dokumentointiin. Tämän jälkeen siistin koodiani, jotta se olisi luettavampaa ja siihen olisi helpompi tehdä tarvittaessa muutoksia myöhemmin.

Iltapäivällä suuntasin toimistolle. Meillä oli backlogin uudelleenmäärittelytapaaminen, jonka aluksi kävimme läpi sprintin tilanteen. Loppuajan kävimme läpi backlogilla odottavia tarinoita. Tämän tapaamisen tarkoituksena ei ole suunnitella toteutustapoja, joten emme käyneet tarinoita kovinkaan tarkkaan läpi.

Loppupäivä meni suurelta osin sosialisoituessa esimieheni kanssa, joka oli tullut Jyväskylästä käymään Espoon konttorillamme. Laitoin muutokseni katselmointiin ja aloin tutustumaan lokien automaattisen testauksen toteuttamista. Päivän loppuksi minulle tuli vielä toiselta kehittäjältä katselmoitavaa koodia, mutta se jäi huomiselle.

## **Tiistai, 15.2**

Tänään päivän aluksi katselmoin eilen tulleet koodimuutokset. Tämän jälkeen jatkan lokien automaattista testausta. Muutoksesta on kirjoitettu käyttäjätarina. Käyttäjätarinalla kuvataan ongelma, johon tarvitaan ratkaisu. Käyttäjätarinalle on kirjoitettu muutoksen perustelu

ja vaatimukset tarinan valmistumiselle. Käyttäjätarinoille annetaan myös työmääräarvio, jolla pyritään arvioimaan, kuinka suuren vaivan tarinan valmistuminen vaatii. (Turley & Rad 2018, 47-52.) Kyseiselle tarinalle on arvioitu kymmenen henkilötyöpäivää, eli minulla on seuraavat kaksi viikkoa aikaa kehittää ratkaisutapa ja toteuttaa vaadittavien osien testaus. Työmääräarviot on hyväksytetty asiakkaalla ja niiden ylittämistä pitäisi välttää.

Katselmoitava muutos koski erästä prosessia, jossa pieni määrä dataa sekoittui. Prosessia oli vuosi sitten optimoitu, jonka myötä kyseinen bugi oli syntynyt. Ennen optimointia sen ajamiseen suurella määrällä dataa meni yli kahdeksan tuntia. Optimoinnin jälkeen suurella määrällä dataa ajoaika tippui tuntiin. Tuotteenomistajan mukaan ei haittaa, vaikka ajoaika kasvaisi, kunhan se ei kestä koko päivää. Uuden muutoksen myötä ajoaika nousi, mutta pysyi kuitenkin alle kahdessa tunnissa. Muutos itsessään ei ollut kovin suuri, joten kulutin hetken luokiessani C#:n lock-lauseen dokumentaatiota (Microsoft 2022a), jonka jälkeen hyväksyin katselmoinnin.

Loppupäivän vietin perehtyen eri vaihtoehtoihin automaattisen testauksen toteuttamiseksi. Windowsiin muodostuvien lokien lukeminen olisi mahdollista, mutta testaus halutaan pitää sovelluksen sisällä. Testeihin tarvitsisi lisätä kuuntelija, jolla saadaan kiinni lokeja tuottavat komennot.

### **Keskiviikko, 16.2**

Tänään pitäisi saada tehtyä kuuntelijat testeihin sovelluksessa lähetettäville komennoille. Sovelluksen tehdessä eri toimintoja, se lähettää background workerille lokitettavia viestejä kommentoina. Pyydän työkaverilta apua näiden toteuttamisessa, sillä hän oli toteuttanut samankaltaisia kuuntelijoita itse sovelluksen puolelle.

Tutkittiin yhdessä, kuinka kuuntelijat saataisiin testien puolelle toteutettua. Toteutusta varten testiprojektiin tarvitsi tehdä konfiguraatiomuutoksia, joiden selvittämiseen yksin minulla olisi luultavasti mennyt hyvin pitkään.

Seuraavaksi toteutettiin kuuntelijat. Testien alustuksen yhteydessä kuuntelijat tilataan, eli kun niiden käsittelemiä viestejä lähetetään sovelluksesta, kuuntelijat ottavat sen vastaan. Tässä vaiheessa todettiin, että on järkevintä toteuttaa yleinen kuuntelija sen sijaan, että jokaisella tapahtumalla, josta lokimerkintä syntyy, on oma kuuntelijansa. Kulutin loppupäivän tehden yleisen kuuntelijan vaatimat muutokset sovellukseen.

### **Torstai, 17.2**

Tänään tulisi ratkaista, miten kuuntelijoilta välitetään testiskenaarioille tieto vastaanotetuista komennoista. Käyttämämme testauskirjasto (SpecFlow) tarjoaa testeille kontekstin,

jossa voi säilöä testiä varten tarvittavia tietoja. Jos tähän kontekstiin pääsisi käsiksi myös kuuntelijoissa, voisi sen kautta välittää tiedot vastaanotetuista tapahtumista.

Ongelma, joka toteutuksessa tuli tänään vastaan on se, että tapa, jolla kontekstia käsiteltiin SpecFlow:ssa, on käytöstä poistuva, eikä sen käyttöä suositella. Nykyään kontekstiin pääsee käsiksi injektioita kautta (SpecFlow 2022). Suoraan konteksti tarjotaan kuitenkin vain luokille, jotka toteuttavat testiaskeleita.

Päivän aikana onnistuin ratkaisemaan tämän ongelman. SpecFlow tarjoaa metodin, joka ajetaan ennen jokaista testiä. Tämän metodin kautta syötän kuuntelijat toteuttavalle luokalle käytettävän kontekstin, jonka myötä kuuntelijat pystyvät lisäämään vastaanottamiaan tapahtumia kontekstiin, ja testejä suorittavat luokat pääsevät niihin käsiksi.

## **Perjantai, 18.2**

Tänään aloitin lisäämään testejä, jotka testaavat, onko lokimerkintöjä muodostuvia tapahtumia muodostunut. Moni tapahtuma löytyy jo olemassa olevista testeistä, joten selvitin mitkä testit tapahtumia luo. Sen jälkeen lisäsin kyseisten tapahtumien muodostumisen varmistuksen testiskenaarioiden loppuun.

Tänään oli myös esimiehen kanssa tapaaminen, jossa keskusteltiin työn kulusta ja viihtyvyydestä. Tilattiin minulle uusi tehokkaampi työkannettava, sillä alun perin testaamiseen annettu kannettavani tehot eivät oikein riitä kehittämiseen.

Ilmapäivällä oli myös retro, jossa tiimin kanssa käytiin läpi kulunutta sprinttiä. Retrossa tuodaan esille onnistumisia, epäonnistumisia ja mahdollisia kehitysideoita tuleville sprinteille. (Turley ym. 2018, 40.) Tällä kertaa käsiteltäviä asioita ei ollut paljoa ja retro oli tavallista lyhyempi.

## **Viikkoyhteenveto 2**

Onnistuin tämän viikon tavoitteissani hyvin. Testausautomaation kehittämiseen oli varattu 10 henkilötyöpäivää. Vaikka sen työstäminen on vielä kesken, olen siinä hyvällä mallilla. Opin lisää käyttämästämme testikirjastosta ja vähän sovelluksen konfiguraatioista. Opin, kuinka testikirjastossamme injektoidaan testikonteksti, joka pitää sisällään testien käsittelyyn vaadittavaa dataa. Opettelin myös konfiguroimaan uuden projektin olemassa olevaan testiprojektiin. Tapasin myös esimieheni ensimmäistä kertaa paikan päällä.

### **3.3 Seurantaviikko 3**

Tällä viikolla jatkan testiautomaation toteuttamista. Jatkokehitän vielä nykytoteutusta niin, että varmistan tiettyjen rahoitusten tietojen olevan lokimerkinnöissä. Iso osa mahdollisista

lokimerkinnöistä syntyy, kun käyttäjät katselevat tietoja käyttöliittymällä, joten myös käyttöliittymällä muodostuvia lokimerkintöjä olisi hyvä testata. Käyttöliittymää ei vielä tällä hetkellä testata integraatiotesteillä.

### **Maanantai, 21.2**

Tänään tiimillä on jälleen varattu puoli päivää sprintin suunnittelulle. Tämän lisäksi jatkan testiautomaation kehittämistä. Tarpeena olisi lisätä lokimerkintöjä tuottavia prosesseja nykyisiin testeihin, jotta testikattavuutta saataisiin kasvatettua.

Suunnittelutilaisuudessa toin esille sen ongelman, että nykyisissä integraatiotesteissä ei testata käyttöliittymää ollenkaan. Käyttöliittymästä muodostuu tällä hetkellä suurin osa lokimerkinnöistä, sillä niiden tarkoitus on pääasiassa tallentaa tiedot siitä, mikä käyttäjä on tarkastellut mitään tietoa sovelluksessa. Testauksen toteutukseen jäljellä olevaa aikaa jaettiin jo työstämieni testien loppuun viennille ja käyttöliittymän testauksen toteutukselle. Käyttöliittymää varten aikaa on kuitenkin varattuna vain pari päivää, joten sen toteutuksen on vaarana jäädä vähäiseksi tai menevän työmääräarvioista ylitse.

Loppupäivän kulutin jatkokehittämällä testitoteutusta. Tähän mennessä testiaskel, joka varmistaa lokimerkinnän muodostavan tapahtuman lähtemisen sovelluksesta, tarkisti vain sen, että kyseinen tapahtuma otettiin vastaan. Lisäsin testiaskeleeseen myös tarkistuksen merkinnän sisällöstä. Kaikilla rahoituksilla ja osapuolilla on uniikit tunnisteet, jotka vievät lokimerkinnälle. Lisäsin tarkistuksen, joka varmistaa merkinnällä kyseisten tietojen vastaavuuden testiskenaarioon liittyviin rahoituksiin ja osapuoliin.

### **Tiistai, 22.2**

Tänään jatkan vielä lokien integraatiotestiautomaation kehittämistä. Eilen sain valmiiksi tarkistukset merkintöjen sisällölle. Olen myös jo lisäillyt merkintöjen muodostuksien testauksia paikkoihin, joissa olemassa olevat testit niitä jo muodostaa. Tarpeen olisi siis lisätä askelia olemassa oleviin testeihin, joilla muodostuisi uusia lokimerkintöjä.

Tänään edistyminen oli hieman hidasta. Olen viimeksi kesällä tehnyt uusia testejä, joten jouduin palauttelemaan mieleeni, kuinka niitä lisätään. Lisäksi piti etsiä sopivia paikkoja, joihin uusia testiaskeleita voisi lisätä. En halunnut tehdä täysin uusia testitapauksia, koska rahoitusten luonti ja muu käsittely lisäisi turhaan testien ajoaikaa. Sen sijaan lisäsin uusia testiaskeleita jo olemassa oleviin tapauksiin.

### **Keskiviikko, 23.2**

Tänään laitan tekemäni muutokset katselmointiin. Koodia tuli tuotettua aika paljon, joten siihen tulee vielä luultavasti korjattavaa. Jatkan vielä tänään kuitenkin testiautomaation

parissa. Tarkoituksena olisi käyttää Seleniumia automatisoimaan selaimella tapahtuvaa testausta. Näin testataan myös käyttöliittymän kautta muodostuvia lokimerkintöjä.

Vaikka meillä ei vielä olekaan käyttöliittymän testausta toteutettuna, sille löytyy valmiiksi projekti, joka sisältää ainakin joitakin vaadittuja valmisteluja testaamiselle. Projektista löytyy myös yksi testitapaus, joka on kuitenkin jätetty kesken. Ilmeisesti käyttöliittymän integraatiotestausta on koitettu toteuttaa aiemmin, mutta siinä on tullut joitakin ongelmia vastaan, joiden vuoksi siitä on ainakin toistaiseksi luovuttu. Valitettavasti tiimin jäsenet on vaihtuneet, niin tuolloin vastaan tulleet ongelmat ovat jääneet epäselviksi.

Käyttöliittymätestaus toteutetaan Seleniumilla. Selenium on työkalu, jolla voi automatisoida selaimia esimerkiksi testitarkoitukseen (Selenium 2022). Päivitin Seleniumin version ja hain ajettavan selaimen ajurit. Projektiin oli aiemmin asennettu Internet Explorerin ajurit, mutta kyseisen selaimen tuesta on luovuttu. Testaus toteutetaan Edgellä, joka on myös asiakkaan käyttämä selain.

Päivän aikana sain selaimen avaamaan automaattisesti syöttämäni sivun ja painamaan valikosta nappeja, jotka lataavat eri luetteloita. Luetteloiden latauksesta muodostuu lokimerkintä, joka pitää vielä ottaa vastaan testien puolella, jotta sen muodostuminen voidaan tarkistaa. Katselmoinnissa olevista muutoksista tuli myös joitakin pieniä huomioita, jotka korjasin ennen työpäivän loppua.

## **Torstai 24.2**

Tänään jatkan käyttöliittymän integraatiotestauksen toteutusta. Testausta pitäisi lisätä muille sovelluksen sivuille. Lokimerkintöjä luovat viestit pitäisi myös vastaanottaa testien puolella, jotta niiden muodostuminen voidaan varmistaa. Tämän toteutin jo toiseen testiprojektiin, mutta käyttöliittymätestiprojektissa on tällä hetkellä tehty hyvin vähän konfigurointia, joten tämä voi tuottaa vaikeuksia.

Lisäsin toisen testitapauksen, jolla avataan eri sivu. Kuten aikaisemmalla sivulla, myös tällä on valikko, josta aukeaa eri luetteloita. Valikon napit saa Seleniumilla etsittyä selaimesta esimerkiksi HTML:n id-tunnisteen avulla, jonka jälkeen selaimen voi automaattisesti laittaa painamaan kyseistä nappia.

Toteutin käyttöliittymätestiprojektiin kuuntelijat, joilla voin vastaanottaa lokimerkintöjä muodostavat viestit. Valitettavasti ne eivät vielä tee mitään, sillä projektista puuttuu konfiguraatioita, jotka mahdollistavat viestien kulkemisen sovelluksesta myös käyttöliittymätestejä ajavaan projektiin.

## **Perjantai 25.2**

Tänään jatkan käyttöliittymätestiprojektin konfigurointia. Mahdollisia tarvittavia ulkoisia kirjastoja ja referenssejä sovelluksen eri projekteihin, sekä vaadittavia asetuksia on niin paljon, että pyydän apua tiimikaveriltani.

Aamupäivä kului Teams-puhelussa tiimikaverin kanssa. Selviteltiin yhdessä, mitä alustuksia testien ajamiseen tarvitsee tehdä, jotta viestit voitaisiin vastaanottaa sovelluksesta. Toisessa integraatiotestiprojektissa alustetaan paljonkin asioita, mutta siinä myös testataan kattavasti useita sovelluksen prosesseja. Pitääksemme riippuvuudet eri sovelluksen osiin mahdollisimman vähäisenä, koitimme pitää alustettavia prosesseja mahdollisimman vähäisinä.

Valitettavasti tämä ei toiminut ja jouduimme tuomaan lisää referenssejä eri projekteihin. Loppuen lopuksi saimme kuitenkin alustukset kuntoon. Vielä kuitenkin viestit eivät kulkeet sovelluksesta testiprojektiin. Tiimikaverini lopetteli tässä kohtaa työt, joten sovimme jatkavamme tästä maanantaina.

Sen sijaan, että olisin yksin yrittänyt jatkaa konfigurointia, laitoin toiselle tiimikaverilleni viestiä. Hän oli jonkin aikaa sitten toteuttanut erään lokituksen, joka on istunut siitä asti katselmoinnissa. Tiimikaverillani on nimittäin ollut kiireitä toisen asiakasprojektin parissa, eikä ole ehtinyt tehdä katselmoinnin myötä haluttuja muutoksia toteutukseensa. Joten katsoimme yhdessä, mitä hän oli tehnyt ja mitä pitäisi vielä tehdä, jonka jälkeen lähdin itse työstämään katselmointimuutoksia.

Kyseisen lokitustarinan yhteydessä oli ilmennyt tarve pilkkoa muodostuvaa lokia useammalle lokimerkinnälle. Windowsin tukemilla lokiviesteillä on nimittäin maksimipituus, jonka tämä kyseinen lokimerkintä saattaa ylittää. Halutaan siis toteuttaa yleiskäyttöinen metodi, jolla voidaan pilkkoa sille syötettyjä listoja, jotta tarvittaessa voidaan muodostaa useampi kuin yksi lokimerkintä. Tiimikaverini oli metodin jo pääasiassa toteuttanut, minulle jäi tehtäväksi vain tehdä siitä vähän selkeämpi ja helppokäyttöisempi. Loppupäivän käytin tutustumaan tähän mennessä toteutettuun menetelmään.

#### 3.4 Seurantaviikko 4

Tällä viikolla pitäisi viimeistellä testiautomaation toteuttaminen. Täytyy selvittää vielä, mitä konfiguraatioita tarvitsee, jotta testiympäristö saa vastaan sovelluksen lähettämät lokiviestit. Tiimimme tuotteenomistaja ja toinen testaja ovat talvilomalla tällä viikolla. Tuotteenomistaja tulee torstaina takaisin.

#### **Maanantai, 28.2**

Jatkan vielä tänään testiautomaation kehittämistä. Selvitellään työkaverin kanssa viime viikolla ilmenneet ongelmat. Sen lisäksi tarvitsee viimeistellä testit, joita viime viikolla aloitin. Valitettavasti kaikkia viestejä ei ehditä lisätä testattavaksi. Iltapäivällä on myös sprintin

uudelleenmäärittelytapaaminen, jossa meillä on työmääräarvioitavana yksi asiakkaalta tullut ominaisuus.

Sprintin uudelleenmäärittelytapaamisessa otettiin asiakkaan uusi muutospyyntö työmääräarvioitavaksi. Yleensä tuotteenomistaja kirjoittaa asiakkaan lähettämästä muutosdokumentista tarinan, jonka perusteella teemme työmääräarvion, mutta hänen ollessa lomalla jouduimme käyttämään asiakkaan toimittamaa dokumenttia. Joskus asiakkaan vaatimusten ja finanssialan prosessien ymmärtäminen voi olla hankalaa, jolloin tuotteenomistajan osaamista tarvitaan. Selvisimme kuitenkin kyseisestä työmääräarvioinnista. Erään rahoitustyyppin voimaansaattamisprosessiin halutaan muutoksia, jotka on jo toteutettu muille rahoitustyypeille.

Muu päivä meni testiautomaation toteutukseen. Saatiin vastaanotettua sovelluksen lähettämiä viestejä testien puolella, joten iltapäivän kulutin viimeistelemällä viime viikolla aloittamani testitapaukset. Seleniumilla kyseisten testien toteutus onnistui helposti. Tarvitsi vain antaa osoite ladattavaksi ja tämän lisäksi ohjeistaa Seleniumia painamaan tiettyjä nappeja. Seleniumi osaa odottaa, että kyseinen nappi latautuu, jonka jälkeen se painaa haluttua nappia. Päivän lopuksi laitoin koodimuutokseni katselmointiin. Muutoksia oli jälleen tavallista enemmän, joten mitä todennäköisimmin katselmoinnista tulee vielä jotain korjattavaa.

### **Tiistai, 1.3**

Tänään jatkan perjantaina työkaveriltani perimän muutoksen kehittämistä. Tämänhetkinen toteutus pilkkoo yksittäisen syötetyn listan useampaan. Yhteen lokimerkintään saatetaan kuitenkin tarvita yhdestä neljään listaa, jotka pitää kaikki pilkkoa. Joten sen sijaan, että pilkkojametodia kutsuttaisiin jokaista listaa varten, metodin tulisi voida ottaa niin monta listaa vastaan, kuin tarvitaan.

Tämän toteutuksessa oli haasteena listojen syöttäminen metodille ja missä muodossa listat tulisi palauttaa kutsujalle. C# mahdollistaa ennalta määrittelemättömän määrän parametreja syöttämisen metodille params-avainsanan avulla (Microsoft 2022b). Tällöin syntyy yksi taulukko, joka sisältää kaikki metodille syötetyt listat. Tämän myötä saadaan otettua myös huomioon tarkemmin lokimerkintöjen todellinen koko ja riski siitä, että merkinnöistä tulee liian pitkiä, pienenee.

Pyörin myös päässäni useita eri vaihtoehtoja listojen palauttamiselle kutsujalle. Yritin toteuttaa mahdollisimman helppokäyttöistä ja selkeää ratkaisua. Jokaista syötettyä listaa kohden pitää palauttaa lista, joka sisältää listoja. Sisemmät listat pitävät sisällään alkuperäisen listan tiedot, jotka on jaettu ulkoisiin listoihin. Yksi ulkoinen lista vastaa yhtä lokimerkintää. Palautettu tieto menee vain helposti erittäin monimutkaiseksi, kun näitä ulkoisia listojakin voi olla useita. Päädyin lopulta palauttamaan ne yhdessä taulukossa, jossa tiedot ovat samassa järjestyksessä kuin ne metodille syötettiin.

### **Keskiviikko, 2.3**

Tänään koitan vielä siistiä eilen kehittämäni metodia. Siinä on vielä eräs vaikeasti luettava apumetodi, jota koitan siistiä. Eräs työkaveri oli kyseiseen apumetodiin aiemmin selvittänyt selkeämpää toteutustapaa internetistä ja oli mahdollisen ratkaisun löytänyt, joten koitan käyttää hänen löytämäänsä toteutustapaa. Sen lisäksi kirjoitan vielä yksikkötestejä. Nämä olisi voinut myös kirjoittaa aiemmin, jolloin kyseisen metodin toimivuuden testaaminen olisi ollut helpompaa. Mutta tällä varmistetaan, että metodi toimii eri tilanteissa. Mikäli tulevaisuudessa tehdään metodiin muutoksia, niin varmistetaan tällä myös, ettei muutokset riko toteutusta.

Päivän suunnitelman lisäksi pyysin yhtä kokeneempaa tiimikaveriani käymään muutokseni yhdessä läpi, jotta voisin saada häneltä mahdollista palautetta toteutustapaani tai koodin siisteyteen. Mietittiin yhdessä vähän parempaa nimeä aiemmin mainitulle apumetodille, mutta muuten hän oli tyytyväinen työhöni.

Saatuani koodimuutokset valmiiksi, laitoin ne katselmointiin. Korjasin myös joitakin testiautomaattiototeutukseen tulleita huomioita ja kommitoin korjaukset. Tämän jälkeen päivä oli melkein lopussa, joten kulutin lopun aikaa tutustuessa seuraavaan muutospyyntöön, jota alan huomenna työstää.

### **Torstai, 4.3**

Tänään pääsen toteuttamaan uutta muutospyyntöä. Tarpeena on tehdä muutoksia perinnässä olevien rahoitusten maksusuoritusten käsittelyyn. Automaattinen käsittely halutaan keskeyttää ja ohjata manuaaliseen käsittelyyn, jos rahoitus ei ole tietyssä tilassa.

Aloitin muutosten teon yksikkötesteistä. Lisäsin muutosta varten muutaman testin ja muutin olemassa olevat testit muutoksen mukaiseksi. Kun tein yksikkötestit ensimmäisenä, on muutoksen testaaminen helppoa ja nopeaa. Sovellusta ei tarvitse erikseen käynnistää eikä valmistella tilannetta, jolla testata muutosta. Itse muutos vaati vain pienen muutoksen. Laitoin päivän päätteeksi muutokset katselmointiin.

### **Perjantai, 5.3**

Testit, jotka ajetaan joka aamu, tuottivat virheitä. Testikoneella ajamamme testit eivät ilmeisesti ole jostain syystä saaneet sovelluksen lähettämiä lokiviestejä vastaan, joten kyseiset testit eivät ole menneet läpi. Tutkin tänään asiaa. Tämän lisäksi tänään on retro sekä town hall. Town hall on koko Samlinkin laajuinen tilannekatsaus, missä yrityksen johto kertoo ajankohtaisista asioista työntekijöille. Samalla työntekijöillä on tilaisuus tuoda itselleen tärkeitä asioita esille yrityksen johdolle. (Company Webcast 2022.)

Koitin tutkia eroavaisuuksia oman ympäristöni ja testikoneen välillä. Tarkistin eri tietokanta-  
tauluja ja koitin tutkia, puuttuuko testikoneen konfiguraatiosta vielä jotakin. Huomasin vii-  
mein, että testien alustamisesta puuttuu metodikutsu, joka luo kuuntelijat lokiviestien muo-  
dostumiselle. Lisäsin metodikutsun alustukseen. Loppupäivän käytin tutustuessa seuraavaan uu-  
teen muutospyyntöön ja aloittelin sen toteuttamista.

#### **Viikkoyhteenveto 4**

Pääsin kirjoittamaan tällä viikolla yksikkötestejä. Yksikkötesteillä saadaan testattua koodia pie-  
nissä osissa luokkatasolla. Yksikkötestit ovat hyvä tapa sovelluksen regressiotestaamiselle.  
Niillä varmistetaan, että testattava luokka tai metodi tuottaa halutun lopputuloksen. Mikäli  
yksikkötestattua koodia refaktoroidaan, voidaan yksikkötestit ajamalla varmistaa, että sen toi-  
minnallisuus ei muutu. Jos yksikkötestit kirjoittaa ensiksi, voi niitä käyttää myös kehityksen  
apuvälineenä. Niillä saa testattua kehitettävää toiminnallisuutta ilman, että testattava tilanne  
tarvitsee rakentaa manuaalisesti uudestaan ja uudestaan. (Kenett & Ruggeri & Faltin 2018. 415-  
416.)

#### **3.5 Seurantaviikko 5**

Tämän viikon perjantaina on seuraavan toimituksemme code complete, eli kaikki seuraavaan  
sovellusversioon menevät koodimuutokset tulee olla toteutettuna perjantaihin mennessä. Tä-  
män jälkeen viedään asiakkaan testiympäristöön tuorein versio sovelluksesta. Tämän lisäksi  
olen tämän viikon ajan tikettivuorossa, eli vastaan tuotannon häiriöistä.

#### **Maanantai, 7.3**

Tänään päivä alkoi sprintin suunnittelutapaamisella. Sprintille nostettiin useita tarinoita,  
jotka tulisi saada perjantain code completeen valmiiksi. Yksi näistä tarinoista on se, jota  
aloittelin jo perjantaina. Meidän tiimissämme kehittäjät yleensä ottavat seuraavan vapaan  
tarinan sprintiltä, kun ovat saaneet edellisen toteutettua, mutta code completen vuoksi so-  
vimme tämän viikon tarinoiden tekijät etukäteen. Näin varmistimme, että tarvittavat tarinat  
saadaan tehtyä perjantaihin mennessä.

Iltapäivällä jatkoin viime viikolla aloittamaani tarinaa. En edennyt sen kanssa vielä kovin pit-  
källe. Tarinaan sisältyy osapuolten, eli henkilöiden tai yritysten liitokset toisiinsa, joiden ym-  
märtäminen tuottaa minulle hankaluuksia. Aion kysyä huomenna dailyssä asiasta tuotteen-  
omistajaltamme, jotta voisimme esimerkiksi yhdessä katsoa miten osapuolten väliset liitokset  
toimivat.

#### **Tiistai, 8.3**

Tänään dailyn jälkeen kävimme tuotteenomistajan kanssa yhdessä läpi tarinaan liittyvät osapuoliliitokset. Tarkoituksena on etsiä annettuun osapuoleen tiettyjen roolien mukaan liittyvät toiset osapuolet. Yhteisselvityksen yhteydessä tuotteenomistaja kuitenkin alkoi pohtimaan, riittääkö tarinalla määritellyt rooliliitokset vai tarvitseeko ominaisuuden huomioida muutkin mahdolliset roolit. Hän lupasi selvittää asian iltapäivällä.

Asiakkaalta oli tullut eilen tietopyyntö. Tietopyynnöllä asiakas haluaa listauksen esimerkiksi tietyistä aktiivisista rahoituksista. Koska asiakkaalla ei ole suoraa pääsyä sovelluksen tietokantaan, jää meidän tehtäväksi kerätä tarvittavat tiedot. Koska olen tämän viikon tikettivuorossa, myös tietopyyntöihin vastaaminen on minun vastuullani. Onneksi iso osa tietopyyntöistä ei ole ainutlaatuisia, eli vaadittavat toimenpiteet tiedon keruuseen löytyy usein aiemmalta samankaltaiselta tietopyyntötiketiltä. Suurempana haasteena oli datan toimitus asiakkaalle. Koska data saattaa sisältää luottamuksellista tietoa, on se toimitettava asiakkaalle turvallisesti ja salasanalla suojattuna. Selvitin nämä askeleet työkaverilta ja sain tiedot asiakkaalle.

Tämän jälkeen tuotteenomistaja otti minuun yhteyttä. Hän välitti minulle päivitetyn roolilistan. Jouduimme myös laittamaan kyseisen tarinan sivuun toistaiseksi, koska meiltä oli jäänyt prioriteetiltaan korkeammalla oleva bugi backlogille, jonka pitäisi olla korjattuna seuraavaan versioon. Vietin siis loppupäivän tutkimalla kyseistä bugia.

### **Keskiviikko, 9.3**

Tänään asiakkaalta tuli testihavainto bugiin, jota korjasin ensimmäisellä seurantaviikolla. Heillä tilanne oli toistunut, vaikka korjauksen yhteydessä en ollut saanut enää tilannetta toistumaan. Heidän havaintonsa vaikuttaa myös johtuvan eri syystä, kuin aiempi ongelma, sillä heidän testiraportissaan ei ole mainintaa rahoituksen sulkemisesta, mikä johti viimeksi virhetilanteeseen.

Tämän testihavainnon myötä luovutin eilen tutkimani bugin toiselle tiimijäsenelle ja aloin tutkimaan, mistä asiakkaan havainto voisi johtua. Teimme näin sen takia, että minä olen paremmin perillä aiemmasta korjauksestani ja perjantaina hämmöittävä code complete ei tarjoa paljoa ylimääräistä aikaa virheen selvittämiseksi.

En saanut tilannetta itse toistettua vielä aamupäivän aikana, joten pyysin toiselta testaajaltamme apua tilanteen tutkimiseen. Kävimme läpi asiakkaan testiraportin toistoaskeleet ja koitimme selvittää, mitä he tekivät eri tavalla. Löysimme yhden mahdollisen syyn, mutta itse epäilen, ettei se johdu siitä.

Tänään tuli myös tuotantotiketti liittyen erääseen parametrimuutokseen, mikä vietiin tuotantoon viime viikolla. Asiakkaan mukaan parametrimuutos ei ole vaikuttanut toimivan; ne

laskelmat, joiden olisi pitänyt muuttua, eivät ole muuttuneet. Tämä johtuu todennäköisimmin siitä, että sovelluksen ydinprosessit lukevat parametrit välimuistiin silloin, kun ne käynnistetään. Parametrejä muuttaessa olisi siis pitänyt myös käynnistää ydinprosessit uudelleen, mutta tämä oli jäänyt tekemättä. Lähetin tiketin kautta viestin, jossa kerroin, että ydinprosessit käynnistetään uudelleen huomenna, jonka jälkeen parametrimuutoksen pitäisi alkaa vaikuttamaan.

### **Torstai, 10.3**

Tänään pitäisi tuotannossa uudelleen käynnistää ydinprosessit, jotta aiemmin tehty parametrimuutos tulisi voimaan. Tämä vaatii muutospyyntöä tekemistä Samlinkin muutostenhallinnalle ja asentajan kanssa ajankohdan sopimista, jotta prosessit saadaan uudelleenkäynnistettyä. Tiimin jäsenillä on rajoitetut oikeudet tuotannossa toimimiseen ja monet toimenpiteet vaativat tiimin ulkopuolisen asentajan. Tämän lisäksi jatkan eilisen bugin parissa. Huomenna ilta-päivällä on code complete ja sen hetkinen versio viedään asiakkaan testiympäristöihin. Tämän jälkeen ei tehdä muutoksia tuotantoon vietävään versioon, ellei asiakkaalta tule ensi viikon aikana testihavaintoja.

Muutospyyntöä teon yhteydessä syötetään toteutusaskeleet, joita asentajan tulee seurata. Näin asentaja tietää, mitä hänen tulee tehdä ja samalla dokumentoidaan, mitä tuotannossa on tehty. Selvitin askeleet sovelluksen dokumentaatiosta ja varmistin ne vielä tiimikaverilta, ennen kuin aloin tekemään muutostickettiä. Kysyin vapaata asentajaa tälle päivälle tiimin ja asentajien muodostamasta ryhmästä ja laitoin muutospyyntöä muutoshallinnalle käsittelyyn.

Muutoshallinnan hyväksytyä muutospyyntöni saimme asentajan kanssa aloitettua prosessien uudelleenkäynnistämisen. Uudelleenkäynnistäminen on onneksi yksinkertaista, eikä ongelmia ilmennyt. Merkittyäni muutospyyntöä toteutetuksi ja vapautettua asentajan viestin vielä asiakkaalle, että ydinprosessit on uudelleenkäynnistetty ja parametrimuutoksen pitäisi olla voimassa.

Ilta-päivän aikana jatkoin vielä bugin parissa. Kävin tarkistamassa asiakkaan testiympäristön tietokannasta, miltä bugiin liittyvät tiedot näyttävät. Huomasin, että heidän ympäristössään bugiin liittyvä tietokantakysely palauttaa taas ylimääräisiä rivejä. Iso osa sovelluksen prosesseista toteutetaan eräajoina, jotka toteuttavat prosessin tarvitsemat toimenpiteet. Asiakkaan testin yhteydessä kyseinen eräajo oli ajettu useampaan kertaan, koska aiemmat ajot olivat tuottaneet virheitä, eikä ajo ollut tehnyt niiden vuoksi haluttuja toimenpiteitä. Bugiin liittyvä tietokantakysely ottaa tällä hetkellä myös virheitä tuottaneet eräajot mukaan tuloksiinsa, vaikka tätä ei haluta.

Aiempi korjaus johti tähän uuteen bugiin, joten peruin aiemman korjauksen. Alkuperäinen bugi johtui siitä, että tietokantakysely palautti rivejä, jotka liittyivät väärin eräajoihin, joten

lisäsin kyselyyn tarkastelun, joka varmistaa eräajojen olevan oikeat. Testasin, ettei alkuperäinen bugi toistu ja laitoin korjauksen katselmointiin.

### **Perjantai, 11.3**

Tänään on ohjelmassa muiden töiden lisäksi town hall ja osastopalaveri. Nämä ovat olleet viikoittaisia siitä asti, kun Samlink siirtyi Kyndryllin omistukseen. Osastopalaverissa käydään läpi osaston ajankohtaisia asioita. Näiden lisäksi tänään on code complete ja iltapäivällä viedään versio asiakkaan testiympäristöön, jotta he voivat ensi viikolla testata tuotantoon menevät ominaisuudet.

Koska sain eilen bugini katselmointiin, jatkan tänään tiistaina sivuun laitettun tarinan työstämistä. Kyseistä ominaisuutta ei olla enää tähän versioon viemässä, joten minulla ei ole kiirettä saada sitä tänään valmiiksi. En kuitenkaan ehtinyt kovin paljoa tänään kyseistä tarinaa työstämään, sillä työkaveri pyysi apua bugin kanssa, joka tänään pitäisi saada korjattua. Hän oli korjauksen jo tehnyt, mutta koska olen testaajan hommissa kerryttänyt jonkin verran kokemusta sovelluksesta, hän halusi varmistaa minulta erään teoreettisen virhetilanteen muodostumisen.

### **Viikkoyhteenveto 5**

Olin tällä viikolla toista kertaa tikettivuorossa. Vuorossa ollessa aina toivoo, ettei tikettejä tule, mutta tälläkin kertaa niitä tuli muutama. Selviydyin niistä hyvin ja opin sekä tietopyyntötikettien käsittelyn ja tiedon viemisen turvallisesti asiakkaalle, että sovelluksen prosessien uudelleenkäynnistämisen tarvittaessa. Tikettivuorossa oleminen ja tuotannossa toimiminen on kuitenkin itselleni vielä aika pelottavaa ja stressaavaa. Tiimimme piti myös kiinni code completesta, eli jätimme koodimuutosten tekemisen ja tuotantoon viennin välille aikaa. Näin ehdimme korjata asiakkaan testihavainnot, emmekä puske viimeisiä muutoksia edellisenä päivänä ennen tuotantoon asennusta.

### **3.6 Seurantaviikko 6**

Tällä viikolla asiakkaalla on testattavana ensi viikolla tuotantoon menevä versio. Mikäli heiltä tulee testihavainnot kyseisestä versiosta, tulee ne korjata pikimmiten tämän viikon aikana. Tiimiimme on myös mahdollisesti liittymässä Kyndryllin työntekijöitä Unkarista ja tiistaina yksi heistä tulee dailyymme käymään. Kaikki tarinamme ja dokumentaatiomme on suomeksi, joten uusien jäsenien liittyminen voi olla haastavaa. Lisäksi palaverit, joissa he ovat mukana, käydään englanniksi. Ilmeisesti tarkoituksena on kuitenkin, että he työstävät joitakin teknisen backlogimme asioita ja nykyinen tiimimme työstää uusia ominaisuuksia.

### **Maanantai, 14.3**

Sairaslomalla.

### **Tiistai, 15.3**

Sairaslomalla.

### **Keskiviikko, 16.3**

Sairaslomalla.

### **Torstai, 17.3**

Tänään työstän meidän BDD-testejämme. Olin pari viikkoa sitten tehnyt muutoksia perinnän suoritusten kohdistamiseen ja samassa yhteydessä tehnyt tarvittavat muutokset yksikkötesteihin. Olin kuitenkin unohtanut tehdä tarvittavat muutokset BDD-testeihin, joten toteutan tarvittavat muutokset nyt. Tämän lisäksi lokiviestien BDD-testit eivät vielääkään toimi Azuressa. Tiimikaverini, jolta tämän toteuttamiseen alun perin sain apua, on vielä tänään lomilla, joten joudun tätä yksin koittaa selvittämään. Luulen, että ongelma on Azure-ympäristön konfiguraatiossa, mutta tämän selvittäminen on aika hankalaa.

### **Perjantai, 18.3**

Tänään kalenterissa on jälleen town hall -tapahtuma sekä tiimin retro. Lisäksi minulla on esimieheni kanssa one to one, missä keskustelemme työn sujumisesta, mahdollisista kysymyksistä tai muista työntekijän tarpeista. Lisäksi työkaverini palasi tänään lomilta, joten toivottavasti BDD-testiongelmien kanssa päästään tänään eteenpäin. Haluan myös käydä tuotteenomistajan kanssa erästä tarinaa läpi, jossa tarvitsen vähän selvennystä.

Iltapäivän aikana sain mahdollisen läpimurron BDD-testien suhteen. BDD-projektissa on App.config-tiedosto, mistä testiajot keräävät konfiguraationsa. Tässä tiedostossa on määriteltynä myös eri viestiväylät, kuten lokiviestejä lähettävät väylät. Azure-ympäristö käyttää kuitenkin eri konfiguraatitiedostoa, App.Azure.config-tiedostoa. Muutokset, jotka tehtiin App.config-tiedostoon, pitäisi tehdä myös tähän tiedostoon. Visual Studiossa tämä tiedosto oli ikävästi piilotettuna App.config-tiedoston alle, minkä vuoksi ongelman löytämiseen meni näin paljon aikaa. Päivän päätteeksi käytiin jälleen tuotteenomistajan kanssa läpi viimeviikkoista tarinaa. Saatiin yhdessä selkeäksi, mitä tarvitsee tehdä, joten jatkoin sen toteuttamista loppupäivän. En saanut sitä vielä valmiiksi, joten jatkan sen toteuttamista maanantaina.

## **3.7 Seurantaviikko 7**

### **Maanantai, 21.3**

Tänään päivä aloitettiin uuden sprintin suunnittelulla. Tämän sprintin aikana on tarkoitus tehdä muutama uusi ominaisuus ja korjata pari bugia. Yksi tarinoista sisältää useita eri asioita, joita en ole ennen tehnyt, joten pyysin saada tehdä kyseisen tarinan. Tarinan myötä pääsen toteuttamaan migraatioita ja asennusskriptejä, joilla tehdään tarvittavia konfiguraatiomuutoksia eri ympäristöihin.

Iltapäivän jatkoin perjantaina kesken jäänyttä tarinaa. Kirjoitin tarvittavat tietokantakyselyt LINQ-lauseina (Microsoft 2022c). Testattuani kyselyitä, en kuitenkaan saanut haluttuja tuloksia tietokannasta. Yritin debuggerilla selvittää tämän syytä, mutta siinä onnistumatta. Lopuksi kirjoitin haut uudelleen SQL-lauseen muotoisena käyttämällä C#:n query-avainsanoja (Microsoft 2022d). Laitoin koodit katselmointiin ja lopettelini työt.

### Tiistai, 22.3

Tänään aloitin työstämään uutta tarinaa. Varsinaisia koodimuutoksia ei tähän tarinaan tarvita. Tarpeena on lisätä sovellukseen uusi rahoituslaitos, jonka kautta rahoituksille tulee lyhennysilmoituksia, eli rahoitusten takaisinmaksuja. Rahoituslaitoksen lisääminen ei vaadi varsinaisia koodimuutoksia, vaan tulee tietokantaan lisätä tietuerivi. Tähän käytetään Entity Frameworkin migraatioita (Microsoft 2022d).

Migraatioilla saadaan ajettua tietokantaan SQL-lauseita asennusskriptin ajon yhteydessä. Mikä tärkeämpää, migraatioita ajava toiminto tarkistaa, ettei kyseistä migraatiota ole vielä ajettu, ennen kuin se suorittaa muutokset. Näin migraatiot ajetaan ympäristöön vain kerran. Yleisimmin migraatioita käytetään silloin, kun tietokantataulujen malliin tarvitsee tehdä muutoksia. Entity Framework osaa jopa tehdä tarvittavia muutoksia automaattisesti.

### Keskiviikko, 23.3

Migraatioiden lisäksi tulee uuteen rahoituslaitokseen liittyviä tietoja lisätä myös eräajoihin. Jokaisella rahoituslaitoksella on oma eräajonsa, joka käsittelee kyseisen rahoituslaitoksen tuottaman lyhennysilmoitusmateriaalin. Nämä eräajot näkyvät käyttäjille jokaista rahoituslaitosta vastaan omina eräajoinaan, vaikka pellin alla ne ovatkin sama luokka. Erona näiden välillä on niiden konfiguraatiot. Rahoituslaitoksilla on omat tunnuksensa, joiden kautta lyhennysilmoitusmateriaali toimitetaan.

Eräajojen konfiguraatiot luodaan myös tietokantaan. Uudet rivit lisätään asennusskripteihin, jolloin ne ajetaan eri ympäristöihin uusien versioiden asennuksen yhteydessä. Tuotantoympäristön lisäksi jokaista asiakkaan testiympäristöä varten on olemassa omat asennusskriptit, joten muutos piti tehdä aika moneen paikkaan. Näiden lisäksi Samlinkillä on omat testiympäristönsä, joihin on kuitenkin yhdet yhteiset skriptit. Näiden lisäksi on vielä kuitenkin kehittäjien lokaalin ympäristön asennusskriptit. Valitettavasti näiden lisäksi on vielä uuden

testiympäristön ja uuden tuotantoympäristön asennuskriptit, joita tarvitsee ylläpitää. Konfiguraatiot pitää siis tehdä ikävän moneen paikkaan. Pitää myös olla tarkkana, että jokaisessa ympäristössä on varmasti oikeat parametrit.

### **Torstai, 24.3**

Maanantaina suunnittelupalaverin yhteydessä tarkastelimme myös eräajon nykytoteutusta. Rahoituslaitosten lähettämien lyhennysilmoitusaineistoja jäsenitelevän koodin lukeminen koettiin haastavaksi. Mikäli tulevaisuudessa metodeja refaktoitaisiin, eli muokattaisiin helpommin luettavaksi tai paremmin toimivaksi, olisi metodeja hyvä yksikkötestata. Näin varmistettaisiin, ettei refaktoroinnin yhteydessä rikota toiminnallisuuksia.

Yksikkötestien kirjoittamisen yhteydessä tuli ongelma vastaan. Metodit ovat suojattuja (protected), jolloin yksikkötestit eivät pääse käsiksi niihin. Koitin selvittää ongelmaan ratkaisuja internetistä. StackOverFlow:ssa kehoitettiin periyttämään yksikkötestit toteuttava luokka testattavasta luokasta. Tällä tavalla testit pääsivät käsiksi haluttuihin metodeihin. Valitettavasti testien ajaminen ei kuitenkaan vielä onnistunut, sillä eräajoluokan alustaminen vaati monen muun sovelluksen prosessin toiminnassa olemista, vaikka testattavat metodit eivät näitä tarvitsekaan.

### **Perjantai, 25.3**

Tänään jatkoin yksikkötestien toteuttamista. Jonkin aikaa etsittyäni löysin ehdotuksen, jossa kehoitettiin käyttämään PrivateObject-luokkaa (Microsoft 2022f) pitämään testattavaa luokkaa sisällään. PrivateObjectin avulla pystyin kutsumaan testattavan luokan metodeja ja vihdoinkin ajamaan niitä testaavia yksikkötestejä.

Tänään pidimme myös valmistautumispalaverin käyttöliittymän uudistamisprojektin aloitusta varten. Samlinkin siirtyessä Kyndrylin omistukseen meille on tarjoutunut mahdollisuus käyttää Kyndrylin työntekijöiden osaamista omissa projekteissamme. Tiimimme esimiehet ovat halunneet olla Samlinkissä johtamassa tätä muutosta. Käyttöliittymän uudistamisprojekti voisi sopia tähän tarkoitukseen hyvin.

Nykyistä sovellusta aloitettiin kehittämään jo vuonna 2012 korvaamaan asiakkaalla aiemmin käytössä ollut vanhempi sovellus. Silloin sovelluksen käyttöliittymän toteutukseen valittiin Knockout.js niminen JavaScript-kirjasto. Kyseinen kirjasto on kuitenkin tänä päivänä aika vanhentunut. Siihen ei itseasiassa ole pariin vuoteen edes julkaistu uutta versiota. Kirjasto tulisi vaihtaa sellaiseen, jolla on jatkuvuutta mahdollisimman pitkäksi aikaa tulevaisuuteen. Yksi todennäköisimmistä vaihtoehtoista on React (Meta 2022). React on Metan kehittämä ja ylläpitämä avoimen lähdekoodin JavaScript-kirjasto. Tiistaina on sovittu palaveri unkarilaisten työntekijöiden kanssa, jossa käsittelemme aihetta tarkemmin.

### 3.8 Seurantaviikko 8

Tällä viikolla asennetaan tuotantoon uusi hotfix-versio. Eräässä eräajossa havaittiin virhe, joka täytyy viedä tuotantoon ennen kuin asiakas ajaa kyseisen eräajon kuun alussa. Olen ollut kerran seuraamassa tuotantoasennusta, mutta en ole toteuttanut siihen vaadittavia valmisteluita, joten on minun vuoroni hoitaa tuotantoasennus tällä kertaa.

#### **Maantantai 28.3**

Tänään dailyn jälkeen tarkistin vielä muutoksiani viime viikon tikettiin. Tiimikaverini kommentoi parit huomiot katselmoinnissa. Olin joissakin kohdissa käyttänyt väärä arvoja. Varsinkin tämänkaltaisessa muutoksessa on helppo tehdä virheitä, joten on hyvä, että toinen henkilö tarkistaa muutokset.

Iltapäivällä oli sprintin uudelleensuunnittelutapaaminen, jonka aikana työmääräarvioitiin joi-takin asiakkaan lähettämiä muutospyyntöjä. Meille muodostui kuitenkin useita tarkentavia kysymyksiä muutosten määrittelyistä, joten emme antaneet monestakaan tarinasta vielä arvi-  
oita, vaan lähetimme tarkentavat kysymykset takaisin asiakkaalle.

Iltapäivällä tiimimme testaajalla tuli ongelmia, kun hän tuoreutti asiakkaan testiympäristöjä. Testiympäristön tuoreutuksella tarkoitetaan ympäristön tietokannan päivitystä tuoreempaan dataan ja version päivitystä tuoreempaan. Uuden version asennuksen yhteydessä kuitenkin tuli virheilmoitus, josta hän joutui kysymään minulta.

Virheilmoitus viittasi juuri valmistuneeseen muutokseeni. Virhe on SQL-virhe, joka syntyy sil-loin, kun yritetään luoda uutta tietuetta, joka löytyy jo tietokannasta. Migraatioita ajavan oh-jelman pitäisi kuitenkin tietää, jos kyseinen migraatio eli kyseinen SQL-lause on jo ajettu ym-päristöön.

Pienen tutkimisen jälkeen minulle selvisi, mistä kyseinen ongelma johtuu. Tuoreutuksen yh-  
teydessä testiympäristöihin asennettiin ensin uusi versio, jonka myötä myös kyseinen migraa-  
tio ajettiin.

Sen jälkeen ajettiin tuoreuttava PowerShell-skripti, joka lataa sille syötetyn tietokantatiedos-ton, joka yliajaa suurimman osan tietokantatauluista, mukaan lukien ajettut migraatiot säilyt-tävän tietokantataulun. Kyseinen skripti ei kuitenkaan yliaja systeemiparametritaulua, johon kyseinen migraatio lisäsi uuden tietueen. Tämän vuoksi, kun tuoreutuksen jälkeen yritettiin asentaa uutta versiota, ajettiin myös uudelleen kyseinen migraatio ja kyseinen virhe syntyi. Lisäsin migraation alkuun tarkistuksen, että lisättävää tietuetta ei vielä ole olemassa. Tämän jälkeen asiakkaan testiympäristöt saatiin onnistuneesti päivitettyä.

### Tiistai, 29.3

Tänään meillä oli tapaaminen Kyndrylin unkarilaisten työntekijöiden kanssa liittyen käyttöliittymän käyttämän JavaScript-kirjaston vaihtamiseen. Nykyinen kirjasto on vanhentunut, eikä sitä ole päivitetty pariin vuoteen. Unkarilaisten työntekijöiden esimiehellä on kokemusta samankaltaisista projekteista ja hän osasi antaa tiimillemme useita huomioon otettavia seikkoja.

Koko käyttöliittymää ei kannata päivittää kerralla. Kerralla päivityksen mukana tulee liian suuri riski, että jotakin hajoaisi. Sen sijaan vaihto kannattaa toteuttaa osissa. Nykyinen toteutus käyttää .NET:n Model View Controlleria (MVC), joka tullaan todennäköisesti myös vaihtamaan. Yhtenä vaihtoehtona harkitsimme, että käyttöliittymän koodi ja rajapinnan koodi olisivat erillään. Nykyiset Controllerit vaihdettaisiin mahdollisesti NSwagiin (Microsoft 2022g), jonka kautta sovelluksen rajapintaan tehtäisiin jatkossa API-kutsuja.

### Keskiviikko, 30.3

Tämä päivä tulee menemään suurelta osin tuotantoasennukseen valmistuessa. Tuotantoasennusta varten on askeleittain toimenpideohjeet, joiden mukaan asennuksen pitäisi onnistua, eikä asennuksen yhteydestä jää puuttumaan mitään. Aloitin asennukseen valmistumisen kopioidulla aiemman asennusohjeen pohjaksi. Aiempi asennusohje sisältää joitakin askeleita, joita ei tällä kertaa tarvita, joten karsin ylimääräiset askeleet ohjeista pois.

Tuotantoon vientipäivän aikana, ennen tuotantoon asennusta asennetaan tuotantoon vietävä versio myös yhteen testiympäristöön. Kyseinen testiympäristö on eniten tuotannon kaltainen. Siellä pyörii kehittämämme sovelluksen lisäksi palvelu, jolla viedään asiakkaan toisen sovellukseen rahoitusten joitakin tietoja. Mikäli tämän ympäristön asennus onnistuu, voidaan jatkaa tuotantoasennukseen.

Tuotantoon tehtävät muutokset vaativat hyväksyntää Samlinkin muutoksenhallinnalta. Tätä varten Efecteen tehdään tiketti, jolle kuvataan muutos ja sen laajuus. Tiketillä arvioidaan myös muutoksen epäonnistumisen riskiä ja sen vakavuutta. Tämän lisäksi asennusta varten tarvitaan asentaja. Sovelluksen kehittäjillä ei ole tuotantoympäristössä kattavia oikeuksia. Tällä pyritään ehkäisemään tilannetta, missä kehittäjä kävisi vahingossa tai tahallaan tekemässä tuhojaan tuotannossa.

Asennusajankohta sovitaan yleensä päivän päätteeksi, jotta siitä aiheutuva käyttökatko aiheuttaisi mahdollisimman vähän haittaa asiakkaan työntekijöille. Ennen asennusta tulee myös lähettää varoitussähköpostit käyttökatkosta asiakkaalle. Tämän jälkeen asennus voidaan aloittaa sovittuna ajankohtana. Asentaja hoitaa asennuksen seuraten aiemmin tehtyjä asennusohjeita. Itse olin asentajalle tukena, mikäli tulisi kysymyksiä tai ongelmia. Onneksi ongelmia

asennuksen yhteydessä ei juurikaan tullut ja asennus saatiin tehtyä parissa kymmenessä minuutissa. Tämän jälkeen kuitattiin vain asiakkaalle ja Efecte-tiketille, että asennus onnistui ja vapautetaan asentaja.

### **Torstai, 31.3**

Tänään yhdessä testiaan kanssa testasimme uuden rahoituslaitoksen lähettämää testidataa. Heille on toimitettu dokumentaatiota siitä, missä muodossa heidän toimittamansa datan tulisi olla, joten varmistamme, että data näyttää oikealta. Sen lisäksi varmistamme, että datan merkistön koodaus tukee skandeja. Muuten joidenkin loppuasiakkaiden nimet saattaisivat vääristyä, kun dataa tuodaan sovellukseen.

Rahoituslaitoksen lähettämän testidatan tietuerivit vaihtelivat suuresti. Hyvin monessa rivissä oli arvo, joka ei dokumentaation mukaan ollut validi ja josta sovellus heitti virhettä. Joissakin riveissä tämä arvo oli validi. Näissä riveissä moni muu arvo oli kuitenkin pelkästään nollaa. Lähetimme huomioistamme heille sähköpostia.

### **Perjantai, 1.4**

Lähdin tänään tutkimaan sprintillä seuraavana odottavaa bugia. Bugi koskettaa loppulaskun ja sen lähetystilaa. Loppulasku on lainoilla ja takaussaamisilla oleva lasku, joka muodostetaan, kun kyseinen rahoitus halutaan maksaa loppuun. Kun kyseinen lasku muodostetaan, tehdään ensin rahoitukselle loppulaskuehdotelma, joka lähetetään asiakkaalle. Tämän muodostamisesta ilmestyy kyseisen rahoituksen asiakirjoihin loppulaskuehdotelmadokumentti.

Kun loppulaskuehdotelma vahvistetaan loppulaskuksi, ilmestyy rahoituksen laskuihin uusi loppulasku. Laskun lähetystila on tässä vaiheessa ”odottaa lähetystä”. Laskun luonnin yhteydessä ilmestyy myös rahoituksen asiakirjoihin uusi loppulaskuehdotelma-dokumentti, jonka lähetysten tila on ”odottaa lähetystä”. Tätä ”loppulaskuehdotelmaa” ei kuitenkaan koskaan lähetetä, eikä sitä myöskään kuulu lähettää. Tämän vuoksi kyseisen dokumentin tilan tulisi siis olla ”ei lähetystä”.

## **3.9 Seurantaviikko 9**

Tällä viikolla alkaa viimeinen kokonainen sprintti, eli kahden viikon työjakso, jolla työstetään seuraavan tuotantoon otton sisältöä. Sprintin jälkeen viedään jälleen versio asiakasympäristöön testattavaksi. Heillä on viikko aikaa testata muutokset. Mikäli asiakkaalta tulee testihaivantoja, ne kuitenkin korjataan ennen tuotantoon ottoa.

Tällä viikolla edistetään seuraavaan tuotantoversioon vietäviä tarinoita ja bugeja. Sprintin suunnittelutilaisuudessa määritellään lopulliset tehtävät, jotka tulee tämän sprintin aikana saada valmiiksi. Tämä tuotantoversio on pienempi kuin tähänastiset tuotantoversiot, joissa

olen ollut mukana. Tärkeimmän muutoksen toteutin viime sprintillä, mutta sitä ei ole vielä testattu kokonaan.

#### **Maanantai, 4.4**

Aamupäivä kului sprintin suunnittelutapaamisessa. Käytiin tiimin kanssa läpi tulevan sprintin sisältöä. Tuotteenomistajalla oli asiakkaalta saatu lista seuraavaan versioon halutuista tikeistä tärkeysjärjestyksessä. Listaan sisältyi kuusi käyttäjätarinaa ja kolme bugia. Kaksi tarinaa on jo toteutettuna, joten lähdimme jäljellä olevia tikettejä sovittelemaan sprinteille.

Bugeille on hankalaa antaa työmääräarvioita. Niiden selvittämiseen saattaa mennä paljonkin aikaa ja niiden korjaamisen yhteydessä pitää olla tarkka siitä, ettei riko muuta toiminnallisuutta. Kaksi vaadituista bugeista kuitenkin todennäköisesti liittyvät toisiinsa ja korjaantuvat samalla muutoksella. Toinen näistä bugeista on jo tiimillä työn alla. Kolmannen bugin selvitystä aloittelin itse perjantaina.

Neljästä jäljellä olevasta tarinasta yksi on jo tiimillä työn alla ja toinen on pieni muutos. Kahden viimeiseen muutoksen kehitykseen on kuitenkin arvioitu kuluvan useampi työpäivä. Koska tuotteenomistaja ilmoittaa tänään sprintille otettavien tarinoiden tulevan seuraavaan versioon, joutuu tiimi sitoutumaan siihen, että ne myös valmistuvat tämän sprintin aikana. Tämän vuoksi jätimme pienimmällä prioriteetilla olevan tarinan vielä toistaiseksi pois sprintiltä.

Sprintin sisällön päättämisen jälkeen tuotteenomistaja poistui paikalta. Tämän jälkeen aloimme tiimin kehittäjien kesken jakamaan tarinoita pienempiin tehtäviin. Tällä tarkoitetaan toteutustavan suunnittelua ja työn jakamista eri tehtäviin. Tarinan kehitykselle varattu aika jaetaan näille tehtäville sen mukaan, miten paljon eri tehtäviin arvioidaan kuluvan aikaa. Tämän myötä hankaliakin muutoksia on helpompi lähteä työstämään ja tiimin ymmärrys sovelluksen eri alueista laajenee.

Illtapäivän aikana edistin perjantaista bugia. Yritin monesta eri paikasta etsiä, missä dokumentin lähetystietoa asetettaisiin dokumentin luonnin yhteydessä. En kuitenkaan löytänyt kyseistä arvoa. Päivän lopuksi sain kuitenkin asetettua halutun tilan dokumentille sen luomisen jälkeen. Huomenna testaan ratkaisua ja tutkin vielä, jos saisin asetettua halutun tilan jo dokumentin luonnin yhteydessä.

#### **Tiistai, 5.4**

Huomasin tänään testatessani eilistä korjaustani ongelman. Sain onnistuneesti vaihdettua laskun luonnin yhteydessä dokumentin tilaan ”ei lähetystä”, mutta samalla itse laskun tilaksi vaihtui ”lähetys estetty manuaalisesti”. Hetken tutkittuani huomasin, että laskun lähetysten tila ottaa myös huomioon itse dokumentin lähetysten tilan. Samalla selvisi, että asiakirjoissa

näkyvä loppulaskuehdotelma on täysin sama dokumentti kuin itse lasku. Loppulaskuksi vahvistaminen ei luonut laskua ja uutta loppulaskuehdotelmaa erikseen.

Otin tuotteenomistajaan yhteyttä asian tiimoilta. Halusin selvittää, millä tavalla kyseistä bugia saa lähteä korjaamaan. Yksi tapa olisi loppulaskujen tapauksessa olla välittämättä dokumentin tilasta. Tämä vaatisi muutosta sekä rahoitusten laskut-välilehdelle, että laskuja lähettävään eräajoon. Toinen tapa olisi selvittää, miksi loppulaskun vahvistamisen yhteydessä syntyy uusi ehdotelma asiakirjoihin ja estää dokumentin sinne ajautuminen. Toinen vaihtoehto olisi saattanut vaatia isompaa remonttia loppulaskun toiminnalle, joten päädyimme tuotteenomistajan kanssa ensimmäisen vaihtoehdon toteuttamiseen.

#### **Keskiviikko, 6.4**

Tänään tutkin, saisinko asetettua loppulaskuehdotelmadokumentille halutun ”ei lähetystä” -tilan jo dokumentin luontivaiheessa. Sovellus sisältää paljon koodia ja arvojen tarkasteluja dokumenttien luonnin yhteydessä, joten jouduin sukeltamaan syvälle, jotta löysin haluamani. Lopulta sain asetettua muodostettavalle dokumentille halutun tilan jo luontivaiheessa. Tein myös eilen suunnitellut muutokset.

Viimeiset pari tuntia työpäivästä yritimme tiimikaverin kanssa refaktoroida eräitä koodista löytyviä metodeja. Kyseiset metodit ovat usein muuten samanlaisia, mutta niille syötetään eri luokka parametreina. Nämä luokat kuitenkin periytyvät samasta yläluokasta. Kun tein BDD-testejä lokiviesteille, onnistuin refaktorimaan 70 luokan kuuntelijametodit yhdeksi metodiksi toteuttamalla kuuntelijametodin geneeriselle yläluokalle, josta muut lokiviestiluokat periytyvät. Valitettavasti emme onnistuneet tavoitteissamme tänään. Kyseiset metodit eivät ole samanlaisia kuuntelijoita. Ymmärryksemme niistä jäi vielä hataralle. Emme ole varmoja, epäonnistuimmeko vai emmekö onnistuneet kutsumaan haluttua geneeristä metodia.

#### **Torstai, 7.4**

Päivä alkoi työmääräarvioinnilla. Asiakkaalta tuli useita muutospyyntöjä, joihin he haluavat työmääräarviot. Lähes kaikki muutospyyntöt liittyivät yrityslainoihin. Valitettavasti lähes kaikki muutospyyntöt vaativat asiakirjapalveluun muutoksia. Asiakirjapalvelulla viitataan toiseen sovellukseen, jota kehittämämme sovellus käyttää luomaan dokumentteja. Asiakirjapalvelumuutoksia ei voi toteuttaa kuin yksi kehittäjä samaan aikaan, sillä kun muutoksia aletaan tekemään, täytyy kehittäjällä olla usuin versio asiakirjapalvelun käyttämistä malleista. Mikäli useampi kehittäjä tekee asiakirjamuutoksia ja molempien muutokset sulautetaan päähaaraan, asiakirjapalvelu hajoaa.

Muutamasta työmääräarvioitavasta muutospyyntöstä meille muodostui vielä tarkentavia kysymyksiä. Osa kysymyksistä ei vaikuttaisi työmäärään, joten niille annoimme arviot, mutta pari

muutospyyntöä vaatii tarkennuksia, ennen kuin niille voidaan antaa lopulliset arviot. Havaitimme myös aikataulullisen ongelman. Pyydetyt muutokset haluttaisiin kesäkuun alussa tuotantoon vietävään versioon. Asiakkaalta tulee toteutusluvut muutospyyntöihin vasta työmääräarvioinnin jälkeen. Nämä yleensä käsitellään asiakkaan toimesta heidän palaverissaan, joka tuotteenomistajamme mukaan tapahtuu seuraavan kerran vasta kuukauden päästä. Koska asiakirjapalvelun vaatimia muutoksia voi tehdä vain yksi kehittäjä kerrallaan, tarkoittaa tämä sitä, että meillä tulee olemaan arviolta vain puolitoista sprinttiä aikaa toteuttaa halutut muutokset. Tästä varoitettiin myös asiakasta.

En ollut kovin tyytyväinen loppulaskuehdotelmabugin korjaukseen, joten kysyin vielä erästä tänään lomilta palannutta tiimikaveriani vilkaisemaan korjausta yhdessä. Olimme sitä mieltä, että vain tietäntyyppisen laskun lähetyksen tilan päättely eri tavalla kuin muiden laskujen ei ole hyvä idea. Pohdimme, kuuluisiko loppulaskun vahvistamisen yhteydessä edes muodostua uutta ”loppulaskuehdotelmaa” rahoituksen asiakirjoihin. Dokumentin tallentaminen eri paikkaan on kuitenkin tuotteenomistajan mielestä liian suuri remontti juuri ennen tuotantoon asennusta, joten etenemme nykyisellä korjauksella.

#### **Perjantai, 8.4**

Lähin toteuttamaan uutta muutosta tänään. Emme olleet maanantain suunnittelutilaisuudessa vielä pohtineet tämän tarinan toteutusta, joten sovellusarkkitehtimme tarjoutui katsomaan mahdollista toteutustapaa yhdessä. Tarinalla halutaan muuttaa tietyn dokumentin rivien selitteitä tarkemmiksi. Koodissa tulee siis erotella eri rivit ja antaa niille uudet selitteet.

Tuotteenomistajamme on jo pitkään yrittänyt myydä kehittämäämme sovellusta muillekin asiakkaille. Nykyinen asiakkaamme on välillä hyvinkin tarkka joistakin sovelluksen ilmoituksista ja tekstimuodoista. Mikäli toinen asiakas haluaisi tulevaisuudessa ilmoitusten näyttävän eri tekstejä, olisi näiden tekstien hyvä olla tallessa muualla kuin kovakoodattuna sovelluksen koodiin. Jotta uudelleenkodeattavaa ei olisi niin paljoa myöhemmin, koitamme toteuttaa tätäkin muutosta aina, kun koskemme uuteen sovelluksen osaan.

Iltapäivällä edistimme koko tiimin voimin sovelluksemme jatkuvuussuunnitelmaa. Kyseisen suunnitelman on tarkoitus kartoittaa sovelluksen toiminnalle mahdollisia uhkia ja niiden aiheuttamia haittoja. Suunnitelma sisältää myös riskien ehkäisyä ja mahdollisen häiriötilanteen sattua määrätellään ennalta vastuualueita ja toimia.

#### **3.10 Seurantaviikko 10**

Tämä on opinnäytetyön seurantajakson viimeinen viikko. Sen lisäksi code complete on ensi viikon keskiviikkona. Tämän viikon perjantaina on pääsiäisloma ja olen ensi viikon lomalla, joten omalta osaltani minulla on neljä päivää aikaa varmistaa, että omat muutokseni ovat

valmiita tuotantoon ottoa varten. Korjaustarpeiden sattuessa tämän viikon jälkeen, joku muu joutuu ne tekemään.

#### **Maanantai, 11.4**

Työstin tänään viime viikolla aloittamaani muutosta. Uudet selitteet tulee syöttää kahdessa paikassa. Toinen on paikka syöttää selitteet dokumentille tämän muodostuksen yhteydessä ja toinen on tiedon sovelluksesta ulosvietävän datan muodostuksen yhteydessä. Toteutin tänään ulosvietävään dataan kyseisen selitteen päättelyn. Tämä onnistui helposti, sillä rivit tunnistettiin jo nykyisessä toteutuksessa.

Testaajalta tuli iltapäivällä pieni havainto viikolla viisi työstämäni muutokseen. Kyseisessä muutoksessa haettiin henkilösapuolten tekemiin tietosuojakyselyihin osapuoliin jollakin rooliliitoksella liittyviä osapuolia. Näistä liitoksista jäi kuitenkin puuttumaan vielä liittyvät rahoitukset. Lisäsin rahoitukset tietosuojakyselyyn ja laitoin korjauksen katselmointiin.

#### **Tiistai, 12.4**

Tänään otin viime viikolla työstämäni bugin oma-aloitteisesti takaisin työn alle. Oman korjaukseni myötä loppulaskun lähetystä ei saa enää manuaalisesti estettyä. Manuaalisen eston päättely tapahtuu nimittäin dokumentin lähetysten tilasta, eikä laskun lähetysten tilasta.

#### **Keskiviikko, 13.4**

Toteutin tänään loppulaskuehdotelmabugiiin uuden korjauksen. Keskustelimme asiasta päivittäin, jossa tuotteenomistaja totesi, ettei loppulaskuun aleta tekemään suurempia muutoksia tässä kohtaa. Tämän vuoksi toteutin korjauksen niin, että loppulaskudokumentin tila vaihdetaan vasta siinä kohtaa, kun loppulasku lähetetään.

päätyvä takaus

#### **Torstai, 14.4**

Minulle tilattiin joitakin viikkoja sitten uusi tehokkaampi työkone. Sain nykyisen työkoneeni, kun tulin töihin testaajan tehtäviin, mutta koneen tehot eivät riittäneet hyvin sovelluskehitykseen. Uusi kone oli saapunut toimistolle, joten kävin hakemassa sen. Käytin loppupäivän asennellessa sovelluksemme kehitykseen tarpeellista ohjelmistoa. Olen ensi viikon lomalla, joten puskin maanantaina tekemäni vientiselitemuutokset git:iin muun tiimin saataville. Kirjoitin myös tarinalle ylös tähänastisen edistymisen ja puuttuvat vaiheet, jotta tiimiläiseni tietävät, mistä jatkaa ensi viikolla.

#### **Perjantai, 15.4**

Pääsiäisloma.

#### 4 Yhteenveto ja pohdinta

Olen työskennellyt sovelluskehittäjän tehtävissä nyt noin neljä kuukautta. Jatkuvasti pääsee tekemään jotakin itselle uutta ja erilaista. Suurin osa muutoksista ei ole täysin uuden ominaisuuden lisäämistä sovellukseen, vaan olemassa olevien ominaisuuksien jatkokehittämistä tai muuttamista. Tämän myötä olen päässyt tutustumaan sovelluksen koodiin. Muutosta tehdessä tutustuu aina relevanttiin olemassa olevaan koodiin. Täytyy ymmärtää, mitä koodi tekee, jotta siihen voi tehdä muutoksia rikkomatta mitään. Sovelluksessa on silti paljon koodia, mihin en ole päässyt vielä tutustumaan.

Monesti pitää pohtia muutoksien eri toteutustapoja ja päättää, mikä niistä on järkevin. Päätöksessä pitää osata ottaa huomioon monia eri seikkoja. Jotta sovellusta voidaan myös jatkossa kehittää ja muokata, tulee koodin olla selkeälukuista. Tiimikaverini ovat katselmoinnin yhteydessä kommentoineet kirjoittamaani koodia ja olemme yhdessä tehneet siitä luettavampaa. Jotta sovellusten prosessien ajaminen suurella määrällä dataa kestäisi mahdollisimman vähän aikaa, tulee prosessien olla optimisoituja. Kehittäjien tulee myös seurata tiimin asettamia koodikäytänteitä. Tällä varmistetaan, että koodi on yhteneväistä. Esimerkiksi tietokantakyselyt tulisi toteuttaa samalla syntaksilla. Meidän tiimissämme suositaan LINQ-lauseiden käyttämistä.

Olen opinnäytetyön aikana myös päässyt kirjoittamaan jonkin verran testejä. Olen opetellut BDD-testien kirjoittamista SpecFlow:lla. Lokiviestien testauksen yhteydessä opettelin myös pystyttämään kokonaan uuden testiprojektin käyttöliittymää testaaville testeille. Joidenkin muutosten yhteydessä olen myös päässyt kirjoittamaan yksikkötestejä. Yksikkötestit ovat hyvä tapa varmistaa, että koodi tuottaa haluttuja lopputuloksia. Niillä varmistetaan myös, ettei uudet muutokset riko sovelluksen toiminnallisuutta. Lisäksi yksikkötesteillä saa helposti testattua muutosta sen kehityksen yhteydessä. Yksikkötestit jäävät valitettavasti tiimissämme helposti taka-alalle. Niitä ei muisteta ottaa huomioon työmääräarvioinnin yhteydessä, jonka vuoksi ne jäävät helposti taka-alalle. Tämä on yksi hyvä kehityskohde jatkossa.

Tuotantoympäristössä toimiminen on stressaavaa. Olen opinnäytetyöni aikana ollut tikettivuorossa, jonka vuoksi olen joutunut käsittelemään tuotantodataa. Tuotantodata saattaa sisältää luottamuksellista tietoa, joten sen käsittelyssä pitää olla tarkkana. Olen myös vastannut tuotantoasennuksesta. Asennus oli erittäin stressaavaa. Ongelmien ilmetessä ne täytyy ratkaista erittäin nopeasti. Tuotannossa toimimisen aiheuttama stressi vähentyy vain tuotannossa toimiseen totuttelun myötä.

Toinen stressiä aiheuttava tekijä on ollut annetuissa työmääräarvioissa pysyminen. Opinnäytetyön seurantajakson viimeisten viikkojen aikana oli pari tarinaa, jotka tuottivat ongelmia. Näissä tilanteissa pitää kysyä apua ajoissa, eikä jäädä ongelmiensa kanssa yksin.

Päiväkirjan kirjoittaminen seurantajakson aikana auttoi sisäistämään joitakin työssä oppimiani asioita kertauksen myötä. Seurantajakson aikana opin laajalti eri asioita, joista monia oli vaikea selittää yleisesti ja kirjallisesti, mutta tämä auttoi jonkin verran konkretisoimaan oppimaani. Päiväkirjan kirjoittaminen lisäsi kuitenkin työtaakkaa ja sen myötä minulla oli vähemmän vapaa-aikaa seurantajakson aikana.

## Lähteet

### Sähköiset

Company Webcast 2022. 6 Steps To Hosting Your Company's Town Hall Meeting Online. Viitattu 2.5.2022. <https://www.companywebcast.com/news/internal-communications-company-town-hall-meeting/>

Kenett & Ruggeri & Faltin. Analytic Methods in Systems and Software Testing. E-kirja. John Wiley & Sons, Incorporated.

Kyndryl 2021. Kyndryl plans to acquire Finnish financial IT service provider Samlink. Viitattu 14.2.2022. <https://www.kyndryl.com/us/en/news/2021/11/2021-11-18-kyndryl-plans-to-acquire-finnish-financial-it-service>

Meta 2022. React. Viitattu 12.4.2022. <https://reactjs.org/>

Microsoft 2022a. Lock statement (C# reference). Viitattu 18.2.2022. <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/statements/lock>

Microsoft 2022b. Params keyword (C# reference). Viitattu 6.3.2022. <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/params>

Microsoft 2022c. Language Integrated Query (LINQ) (C#). Viitattu 4.4.2022. <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>

Microsoft 2022d. Query keywords (C# Reference) Viitattu 4.4.2022. <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/query-keywords>

Microsoft 2022e. Migrations Overview. Viitattu 4.4.2022. <https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/?tabs=vs>

Microsoft 2022f. PrivateObject class. Viitattu 12.4.2022. <https://docs.microsoft.com/en-us/dotnet/api/microsoft.visualstudio.testtools.unittesting.privateobject?view=visualstudiosdk-2022>

Microsoft 2022g. Viitattu 23.4.2022. <https://docs.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-nswag?view=aspnetcore-6.0&tabs=visual-studio>

Samlink 2019. Samlink becomes part of Cognizant Viitattu 14.2.2022. <https://samlink.fi/en/blog/samlink-becomes-part-of-cognizant/>

Selenium. Viitattu 27.2.2022. <https://www.selenium.dev/>

SpecFlow. FeatureContext. Viitattu 20.2.2022. <https://docs.specflow.org/projects/specflow/en/latest/Bindings/FeatureContext.html>

Turley & Rad. 2018. Agile Scrum Handbook. 2. painos. E-kirja. Van Haren Publishing.