

Ada Haaranen

FULL STACK -KEHITTÄJÄN PÄIVÄKIRJA

FULL STACK -KEHITTÄJÄN PÄIVÄKIRJA

Ada Haaranen
Opinnäytetyö
Kevät 2018
Tietojenkäsittely
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittely, Web-sovelluskehitys

Tekijä: Ada Haaranen

Opinnäytetyön nimi: Full stack -kehittäjän päiväkirja

Työn ohjaaja: Ritva Virkkala

Työn valmistumislukukausi ja -vuosi: Syksy 2018

Sivumäärä: 42

Opinnäytetyö on päiväkirjamuotoinen tutkielma, jonka tarkoituksena on kuvata työtä full stack -kehittäjän näkökulmasta sekä analysoida päivittäistä ja viikoittaista työprosessia. Opinnäytetyössä pohditaan työssä vastaan tulevia ongelmia ja niiden ratkaisua.

Toimeksiantajana toimii organisaatio, jonka Oulun toimipisteellä työskentely pääasiassa tapahtuu. Opinnäytetyössä kuvataan kehittäjän työyhteisöä ja vuorovaikutusta työpaikalla. Lisäksi opinnäytetyössä pohditaan työn nykytilannetta sekä tutkitaan sidosryhmien kautta, mihin kehittäjä sijoittuu organisaatiossa.

Opinnäytetyössä kuvaillaan kymmeneltä viikolta jokaiselta päivältä kehittäjän työtehtäviä, joita analysoidaan jokaisen päivän lopuksi. Lisäksi jokaisen viikon lopuksi on viikkoanalyysi siitä, miten kehittäjä on suoriutunut kyseisestä viikosta, eli tiivistelmä viikon tapahtumista ja pohdintaa siitä, mitä olisi voinut tehdä työssä toisin.

Päiväkirjakakson jälkeen opinnäytetyössä tutkitaan sekä kehittäjän omaa, että vuorovaikutuksen kehittymistä. Lisäksi pohditaan, millä tavalla kehittäjä voi muuttaa omia työskentelytapojaan tehokkaammiksi ja miten voisi kehittyä jatkossa. Pohdinnassa kerrataan kehittäjän oppimat asiat tiivistysti ja pohditaan, kuinka nämä vaikuttavat jatkoon.

Asiasanat: full stack -kehittäjä, kehittäjä, päiväkirja, analyysi, tietojenkäsittely

ABSTRACT

Oulu University of Applied Sciences
Business Information Systems, Web-application development

Author: Ada Haaranen

Title of thesis: Diary of full stack developer

Supervisor: Ritva Virkkala

Term and year when the thesis was submitted: Fall 2018

Number of pages: 42

The thesis is a diary-like study, of which meaning is to describe work from the aspect of a full stack developer and to analyze daily and weekly working process. The thesis considers problems and solutions to problems in the work environment.

The employer is an organization that has an office in Oulu where working mainly happens. The thesis describes developer's work community and interactions in their working place. Additionally the thesis considers status quo of work and examines the placement of developer in the organization through stakeholders.

The thesis describes ten weeks of every day work of developer's assignments, which are analyzed after every day. There is also weekly analyzing after every week about how the developer has managed the week, summary of happenings of the week and wondering if some tasks could have been done differently.

After the diary part, the thesis examines developer's own and their interaction's development. There is also pondering about how developer can change her working habits to be more efficient and how she can develop in future. Last deliberation repeats developer's learned points in summary and it is thought over how they will have effect to the future.

Keywords: full stack developer, developer, diary, analysis, information technology

SISÄLLYS

1	JOHDANTO	6
2	KÄSITTEISTÖ	7
3	LÄHTÖTILANTEEN KUVAUS	10
3.1	Työn nykytilanne	10
3.2	Sidosryhmät työpaikalla.....	12
3.3	Vuorovaikutus työpaikalla.....	13
4	PÄIVÄKIRJA.....	15
4.1	Viikko 10.....	15
4.2	Viikko 11.....	18
4.3	Viikko 12.....	20
4.4	Viikko 13.....	22
4.5	Viikko 14.....	25
4.6	Viikko 15.....	27
4.7	Viikko 16.....	30
4.8	Viikko 17.....	32
4.9	Viikko 18.....	35
4.10	Viikko 19.....	37
5	POHDINTA	40
	LÄHTEET.....	42

1 JOHDANTO

Opinnäytetyön pääasiallinen tavoite on seurata full stack -kehittäjän työtä päiväkirjamuodossa. Opinnäytetyössä pohditaan ja analysoidaan työssä eteen tulevia tehtäviä ja haasteita sekä tutkitaan niiden ratkaisuja. Päiväkirjaseuranta kestää kymmenen viikkoa eli päiväkirja sijoittuu ajalle 5.3.–18.5.2018. Jokaiselle päivälle on oma osionsa, jonka lopuksi analysoidaan päivän työtehtäviä, sekä jokaisen viikon lopussa on analyysi kyseisen viikon tapahtumista.

Opinnäytetyön toimeksiantajana toimii yritys, joka kehittää rahoitus- ja finanssialan palveluita. Organisaation päätoimipiste sijaitsee Helsingissä, mutta tämä työ suoritetaan Oulun toimipisteellä.

Työyhteisö on moninainen ja koostuu useiden osa-alueiden ammattilaisista. Erilaisia osa-alueita ovat esimerkiksi kehitys, design, testaus, mobiili ja tietoturva. Kehitystyöryhmät ovat osana osastoja, joita johtaa osastopäällikkö. Työryhmät tekevät usein keskenään yhteistyötä, ja yhteyshenkilöitä voi olla niin Helsingissä kuin ulkomaillakin.

Työtehtäväni on full stack -kehitys, joka tarkoittaa pääpiirteittäin sitä, että työssäni käytän ja kehitän sekä sovelluksen front- että back end -puolta sekä autan niitä kommunikoimaan keskenään. Ongelmanratkaisutaidot ovat tärkeä osa työtä. Käytän paljon erilaisia ohjelmia, ohjelmointikieliä ja muita työkaluja ja apuvälineitä, kuten Javaa, JavaScriptiä, RESTiä, APIa, Eclipseä ja Reactia. Lisäksi työssä pitää olla tietoinen sellaisista asioista kuten julkaisuprosessit, tietoturva, ketterä kehitys, erilaiset organisaation sisäiset linjaukset ja Scrum.

2 KÄSITTEISTÖ

Tähän on koottu käsitteistöä helpottamaan opinnäytetyön lukemista.

Agile = ketterä kehitys. Kehitys on jaettu lyhyisiin jaksoihin, joista jokainen pyrkii julkaisukelpoiseen materiaaliin jakson lopussa.

Ajax = Asynchronous JavaScript and XML. Tekniikoita, joilla websovelluksista saadaan vuorovai-
kutteisempia.

Back end = hallinnoi muun muassa sitä, miten nettisivun tai sovelluksen logiikka ja datan käsittely
toimivat.

Bug = lähdekoodissa oleva ohjelmointivirhe.

CSS = Cascading Style Sheets. Tiedostomuoto, joka määrittelee nettisivun tyyliä.

Eclipse = ohjelmointiympäristö, jota käytetään muun muassa Java-ohjelmoinnin työkaluna.

Front end = koodi, joka on niin sanottua näkyvää sisältöä nettisivulla ja sovelluksessa. Front end
-kehitykseen kuuluvat muun muassa HTML, CSS ja JavaScript.

Full stack –kehittäjä = ohjelmistokehittäjä, joka osaa kehittää sekä front- että back end -puolta ja
ymmärtää niiden välisen yhteyden.

GIT = versionhallintaohjelmisto.

HTML = Hypertext Markup Language. Tunnetaan kielenä, jolla nettisivut on kirjoitettu.

IntelliJ IDEA = Java-ohjelmointiin tarkoitettu kehitysympäristö.

Java = oliopohjainen ohjelmointikieli.

JavaScript = ohjelmointikieli, joka mahdollistaa toiminnallisuuden lisäämisen sovelluksiin ja sivustoille.

Jira = tehtävän- ja projektihallintaohjelmisto.

Mockito = avoimen lähdekoodin testaustyökalu Javalle.

Mokittaa = kirjoittaa yksikkötestaustestejä Mockiton avulla.

NVDA = Non Visual Desktop Access on ruudunlukuohjelma, joka mahdollistaa tietokoneen käytön sokeille ja näkövammaisille henkilöille.

OpenGrok = ohjelma, jonka avulla ohjelmoija voi etsiä, viitata ja navigoida lähdekoodin puita.

Opset = yritys, joka tuottaa asiantuntija- ja tukipalveluita.

PO = Project Owner on Scrum-tuotteen omistaja, jonka tehtävänä on välittää näkemys tuotteen rakentamisesta kehittäjille ja ylläpitää backlogia.

Portletti = eng. Portlet. Ohjelmistokomponentti, joka tuottaa näkyvän sisältöruudun sovellukselle.

PuTTY = avoimen lähdekoodin ohjelma, jolla käytetään SSH-palvelimia/hallitaan virtuaalikoneita.

React = komponenttipohjainen JavaScript-kirjasto front end -kehitykseen.

REST = Representational State Transfer on arkkitehtuurimalli ohjelmistorajapintojen toteuttamiseen.

Retrospektiivi = osa ketterää prosessia, jonka tarkoituksena on kerrata tiettyä ajanjaksoa ja suunnitella muutoksia prosesseihin.

Scrum = ketterän kehityksen työskentelytapa, jossa kehitysjaksoja kutsutaan sprinteiksi.

Scrum Master = osa kehitystiimiä. Scrum Masterin vastuulla on, että kehitystiimi käyttää ketteriä menetelmiä.

SOA = Service Oriented Architecture eli palvelukeskeinen arkkitehtuuri on suunnittelutapa, jolla eri osiot on suunniteltu toimimaan itsenäisesti.

SoapUI = avoimen lähdekoodin web-testaussovellus palvelupohjaisille arkkitehtuureille.

Splunk = datanhallintaohjelma, jonka avulla kerätään järjestelmästä tietoa reaaliaikaisesti.

Spring Framework = avoimen lähdekoodin Java-alusta, jolla luodaan tehokasta, testattavaa ja uudelleen käytettävää koodia.

Sprintti = yksi Scrumin kehitysjakso, joka suunnitellaan etukäteen ja lopuksi esitellään valmista tuotetta asianomaisille.

SVN = subversion eli versionhallintajärjestelmä, jonka avulla voidaan muokata lähdekoodia tietoverkon yli niin, että kaikki kopiot pysyvät ajan tasalla.

3 LÄHTÖTILANTEEN KUVAUS

3.1 Työn nykytilanne

Työtehtäviini kuuluu niin front- kuin back end -kehittäminen. Lisäksi työhön kuuluu osallistuminen Scrum-kokouksiin sekä muihin kehittäjää koskeviin kokouksiin ja koulutuksiin. Työryhmäämme kuuluu minun lisäksi toinen full stack -kehittäjä, front end -kehittäjä, kaksi back end -kehittäjää, Scrum Master, PO sekä testaaja. Työryhmämme kehittää tällä hetkellä vakuutuspuolen palveluja, ja olemme aloittaneet työryhmän kanssa tämän vuoden alussa.

Tällä hetkellä front end -kehittäminen on hitaampaa, koska tehtävät vaativat osia muilta tahoilta, kuten designereiden suunnitelmia. Tämän vuoksi myös front end -tekemistä on vähemmän, mutta autan tarvittaessa sillä puolella sitten, kun tehtävät lisääntyvät. Back end -puolella olen juuri saanut otettua käyttöön uusimmat koodit ja pystytettyä Eclipse-projektin. Lisäksi sain asetettua REST-ra-japinnan toimintavalmiiksi.

Työn ohella opiskelen muun muassa front end -puolen React-ohjelman käyttöä, joka tuli minulle uutena asiana tämän projektin myötä. Back end -puolella minun täytyy tutustua samalla sen toimintamalleihin. Työ on enimmäkseen ongelmanratkaisua, joten asioiden opiskelu kuuluu päivittäiseen työprosessiin joka tapauksessa.

Front- ja back end -kehittämisen lisäksi minun täytyy työssäni päivittäin kommunikoida molempien puolien muiden kehittäjien kanssa sekä usein myös designiimiin, Scrum Masterin ja projektin omistajan kanssa. Kokouksiimme osallistuu ajoittain tekijöitä myös Helsingistä. Autan edellisen projektin työryhmää sovelluksen kehityksessä. Joskus yksiköstä myös täytyy etsiä jonkin tietyn erikoisosa-alueen osaavia ihmisiä, jotta ongelmat saadaan ratkaistua.

Pääasiallisesti työni on ottaa tehtävä Jirasta, sopia tehtävistä muiden kehittäjien kanssa, siirtää koodia Gitin tai SVN:n kautta, tarvittaessa kysyä apua ja suorittaa annettu tehtävä usein joko Eclipsen tai Sublimen avulla. Lisäksi etsin tietoa ja käytän ongelmanratkaisua lopputuloksen saavuttamiseksi. Kehitystyö tehdään joko käyttäen Java-työkaluja back end -päässä ja esimerkiksi Reactia, HTML:ää, CSS:ää, ES6:tta front end -päässä. Kun kehitystyö on tehty, tehtävä usein katsel-

moidaan jonkun toisen kehittäjän toimesta ja siirretään testausympäristöön, jossa testaja voi testata sen. Muita tarvittavia ohjelmia prosessissa ovat esimerkiksi SoapUI, jolla voidaan hallinnoida sovelluksen kutsuja, ja PuTTY, jonka avulla voi esimerkiksi muokata virtuaalikoneen asetuksia, hallinnoida palvelinta ja porttettia ja tarkastella lokeja. Lisäksi käytössä ovat eri selaimet, Skype ja Outlook. Front- ja back end -koodin toimintaa käsitellään komentoriviltä komentotyökalulla.

Työssäni tarvitaan todella monipuolista web-kehityksen osaamista. Lisäksi pitää osata kommunikoida ja soveltaa erilaisia työtapoja muiden työntekijöiden kanssa. Pitää hallita Agile ja Scrum-prosessi ja osata soveltaa niitä sujuvasti. Tärkeää on, että uskaltaa ja osaa pyytää apua, koska sovelluksen kehitystä ei pysty kukaan hoitamaan itsenäisesti ilman kommunikointia alusta loppuun. Lisäksi on oltava paljon tietoa organisaation sisäisistä toimintatavoista ja organisaatorakenteesta. Työssä pitää olla hyvä tietämys tietoturvasta.

Työssä olen jo tähän mennessä oppinut Scrum-prosessin ja sen mitä siihen kuuluu ja miksi. Tiedän paljon tietoturvasta ja siitä, mitä pitää ottaa huomioon kehityksessä, jotta tietoturva olisi taattu yrityksen kannalta. Olen oppinut paljon myös tuotantoprosessin etenemisestä, etenemisen esteistä ja julkaisusta, ja minulla on kokonaiskuva julkaisujunasta. Tiedän nyt aiempaa enemmän testauksesta, sen eri vaiheista, raportoinnista ja siitä, miksi se on välttämätöntä ja tärkeä osa tuotantoa. Tärkeimpänä opin päivittäin uutta kehityksestä ja siitä, miten voin tehdä oman työni paremmin, tehokkaammin ja nopeammin.

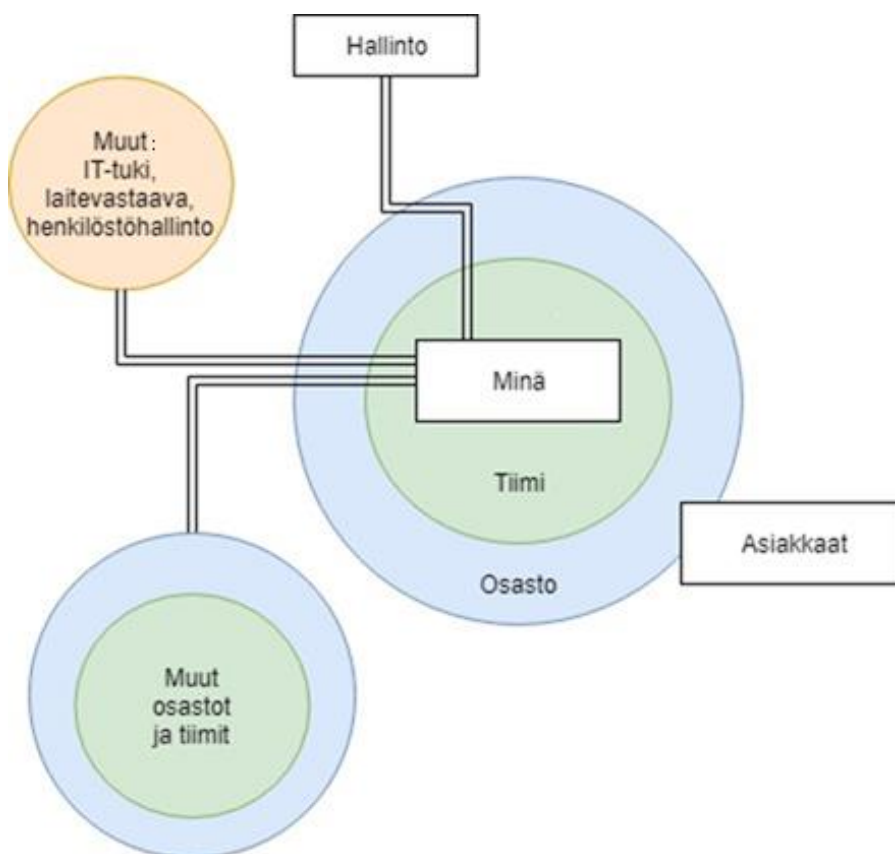
Mielestäni juuri uusien asioiden jatkuva oppiminen tekee työstäni erityisen mielekästä. Ennen tätä työtä olen saanut sille hyvän pohjan koulutuksestani Oamkissa, jossa olen saanut monipuolisesti tietoa useista eri osa-alueista. Ennen koulutusta olen opiskellut itsenäisesti ainoastaan eri ohjelmointikieliä, mutta en olisi voinut silloin uskoa, kuinka paljon muutakin osaamista tulen tarvitsemaan suoriutuakseni alan työtehtävistä.

Joskus huomaan olevani aika uusi alalla ja tarvitsen ohjeistusta työtehtäviin ja joskus osaan ohjeistaa muita. Varsinkin tämänhetkinen tilanne tuntuu siltä, että minulla on vielä paljon opittavaa, sillä kehityksessä käytetään työkaluja, joita en ole aiemmin käyttänyt. Haluan kuitenkin yrittää ensin ratkaista ongelmat itsenäisesti kuitenkin käyttämättä niihin liikaa aikaa. Työtehtävät voivat olla hyvinkin vaihtelevia, joten osaaminen ja ymmärrys voi vaihdella paljonkin, vaikka päivittäin. Uskon kuitenkin, etten ole aivan aloittelija työssäni vaan jo aika lähellä suoriutujaa, joka voi hyvinkin hoitaa annetut työtehtävät mahdollisimman itsenäisesti.

Kehittäjän oppiminen ja kehittyminen ei lopu koskaan, sillä jatkuvasti tulee uusia kieliä, tapoja ja tekniikoita tehdä asioita. Minulle tärkeää työssä on se, että saan oppia koko ajan uutta. Otan mielelläni vastaan tehtäviä, joiden aihealueet eivät ole minulle ennestään tuttuja. Jatkossakin aion ottaa vastaan tehtäviä monipuolisesti ja kokeilla rohkeasti erilaisia toteutuksia.

3.2 Sidosryhmät työpaikalla

Organisaation tärkein ulkoinen sidosryhmä on asiakkaat (Kuvio 1.). Minun työni ei kuitenkaan suoranaisesti vaadi asiakaskontakteja, vaan tällaisiin tehtäviin on olemassa omat toimihenkilöt. Asiakkaille viestintä on kuitenkin hyvin tärkeää, ja he ovat myös tuotteen loppukäyttäjiä. Asiakkaat voivat myös vaikuttaa toimintaan ja toimintatapoihin esimerkiksi palautteen kautta. Organisaatiossa myös jatkuvasti kartoitetaan asiakastyytyväisyyttä erilaisilla tutkimuksilla.



KUVIO 1. Yksinkertaistettu sidosryhmäkaavio

Muita sidosryhmiä ovat alihankinta eli konsultit, joihin minäkin lukeudun. Organisaatiolla on alihankkijoita monista eri yrityksistä. Alihankkijat eli konsultit vastaavat muun muassa kehittämisestä ja suunnittelusta samalla tavalla kuin organisaation sisäisetkin toimijat.

Minua koskeva organisaatio koostuu erilaisista osastoista, joilla jokaisella on omat vastualueet. Minä kuulun osastoon, jonka vastuualueena on vakuutuspuolen kehittäminen. Osastot koostuvat eri määrästä tiimejä, joille vastuualueen tehtävät on pilkottu pienemmiksi osa-alueiksi. Meidän tiimimme kuuluu tällä hetkellä seitsemän kehittäjää ja testaajaa sekä yksi Scrum Master/PO.

Huomioitavia sidosryhmiä on muun muassa henkilöstöhallinto ja IT-tuki. Henkilöstöhallinto vastaa työtuntien oikeellisuudesta, tuntien laskemisesta, palkanmaksusta ja laskutuksesta. IT-tukeen olen yhteydessä enimmäkseen laitekohtaisista ongelmista, kuten salasanoista ja ympäristöjen asennuksesta. Meillä on talon sisällä myös laitevastaava, joka vastaa laitteiden kunnossapidosta.

Tässä on käyty läpi vain osa erilaisista sidosryhmistä, jotka koskettavat minua kehittäjänä jollakin tavalla, sillä näin ison organisaation suunniteltu tämänhetkinen organisaatiokaaviokin on monta kymmentä sivua pitkä. Organisaatorakenne muuttuu koko ajan ja on muuttunut huomattavasti tämän opinnäytetyöprosessin loppuun mennessä.

3.3 Vuorovaikutus työpaikalla

Kehitämme työryhmämme kanssa saman sovelluksen tai palvelun eri osia. Keskustelemme paljon toimistolla kasvatusten sekä Skypen kautta myös toimiston ulkopuolella. Meillä on käytössämme myös muunlaisia yhteydenpitovälineitä puhelimille ja tietokoneille. Teemme todella tiivistä yhteistyötä, ja osa tehtävistä tehdäänkin yhdessä työryhmäläisten kanssa. Työryhmässä on myös helppo pyytää apua, kommentteja ja mielipiteitä tehtävistä.

Teemme yhteistyötä myös muiden osaston työryhmien kanssa esimerkiksi tapauksissa, joissa pitää saada tietoa jostakin toteutettavasta ominaisuudesta. Työryhmät keskustelevat sujuvasti keskenään esimerkiksi sähköpostin tai Skypen välityksellä. Lisäksi pidämme kokouksia, joissa muut voivat tulla kuuntelemaan ja kommentoimaan kehittämisen nykytilannetta. Myös muiden osastojen työryhmien yksittäisiin henkilöihin voi olla yhteydessä esimerkiksi erilaisissa toteutuksen teknisissä ongelmissa.

Työpaikallamme järjestetään myös erilaisia tapaamisia, tilaisuuksia ja koulutuksia, joissa on mahdollista tutustua paremmin muiden osastojen ja tiimien työskentelyyn ja verkostoitua esimerkiksi toisten kehittäjien kanssa. Meillä on myös yhteisiä kahvihuoneita, joissa voi helposti mennä juttelemaan uusille ihmisille. Myös esimieheni on usein tavoitettavissa toimistolla, puhelimella, sähköpostilla tai Skypellä. Haasteena näin isossa organisaatiossa on löytää oikea asiantuntija johonkin tiettyyn aihealueeseen. Täytyy kysellä paljon muilta, kuka tietäisi asiasta, ja selvittää, mistä kyseinen henkilö löytyy.

4 PÄIVÄKIRJA

4.1 Viikko 10

5.3.2018 maanantai

Päivän aikana tarkoitukseni on viimeistellä animaatio. Se on ollut minulla kesken jonkin aikaa, koska olen odottanut designereilta UI-speksejä. Animaation pitäisi olla nopea ja helppo tehdä, sillä se vaatii ainoastaan CSS-muutoksia luokkiin. Tämän jälkeen aion keskittyä toiseen tehtävääni, jossa otetaan käyttöön back end -puolella uusi palvelu, jolla voidaan hakea sovellukseen tiettyjä lisäominaisuuksia. Näiden tehtävien lisäksi meillä on sovelluksen demo, esikatselu ja työryhmän retrospektiivi.

Huomasin front end -kehittäjän kanssa, että animaatio tulee olemaan paljon monimutkaisempi kuin ajattelimme, koska se vaatii laskuja ja muiden lisäominaisuuksien kehittämistä, jotta se toimisi. Designin myöhästymisen ja sprintin loppumisen vuoksi tehtävä jäi seuraavaan sprinttiin. Minulla jäi myös toisen tehtävän edistäminen tämän takia, mutta tehtävä oli vasta seuraavasta sprintistä.

6.3.2018 tiistai

Aion jatkaa animaation työstämistä. Katsoin jo edellisenä päivänä valmiiksi, miten voisin alkaa ratkaista ongelmaa. Sain myös frontti-kehittäjältä linkin, jossa on esimerkki React-lisäosasta, jolla kyseinen toiminto voisi olla helpompi toteuttaa, ja aion katsoa mallia toisen palvelun toteutuksesta. Loppupäivän teen palvelutehtävääni. Aion myös katsoa nykyisen sprintin tehtävät, mitä ne sisältävät, mitä on jo otettu, ja keskustella muiden kanssa, mitä voisin mahdollisesti ottaa työn alle.

Otin aamulla ensimmäiseksi uusimmat koodit Gitistä, minkä seurauksena projektiin piti tehdä jonkin verran konfigurointia. Projektissa oli ongelmia lokalisoinnissa, jota on vasta aloitettu työstää. Korjailin ja lisäsin projektiin pari uutta riippuvuutta animaatiota varten. Tein animaation alusta valmiilla react-animate-lisäosalla, jolloin sovelluksen elementin korkeuden laskua ei välttämättä tarvita. Tein animaation toteutuksen loppuun, ja seuraavaksi sitä pitäisi katselmoida ja samalla pohtia, kannattaisiko siihen lisätä jotakin tulevia animaatioita varten. Jatkoin päivän loppuun palvelutehtävää, jossa jatkoin palvelun viemistä eteenpäin.

Huomasin kuinka, yksinkertaisen kuuloiset tehtävät voivatkin viedä todella paljon aikaa, kun pitää ensiksi löytää oikeanlainen tapa tehdä se. Tällaisissa tehtävissä voi helposti tuntea itsensä hieman osaamattomaksi, kun mikään itsensä tuntema vaihtoehto ei toimi. Näissä tilanteissa on hyvä pyytää työkaveria yhdessä miettimään ratkaisua, jolloin pelkkä ongelmasta keskusteleminen voi tuoda ratkaisun.

7.3.2018 keskiviikko

Tänään aion jatkaa palvelutehtävää ja jutella pääkehittäjän kanssa tehtävän etenemisestä. Uskoin, että tämän tehtävän parissa menee koko työpäivä. Aion tutkia aiempia integraatioita, ja sitä miten ne on toteutettu. Päivä tulee sisältämään paljon ongelmanratkaisua. Aion saada tehtävän tänään siihen pisteeseen, että voimme sen jälkeen katselmoida sen pääkehittäjän kanssa.

Sain hyvin edistettyä palvelutehtävääni, mutta vielä se ei ole valmis. Pääsin tehtävässä kuitenkin hyvin eteenpäin ja sain luotua uusia tarvittavia tiedostoja ja etsittyä tietoa uudesta palvelusta. Minulla oli ongelmia Eclipse-ohjelman kanssa, koska se ilmoittaa välillä virheistä, joita ei ole.

8.3.2018 torstai

Tänään aion taas jatkaa palvelutehtävääni ja jutella pääkehittäjän kanssa, koska en eilen ehtinyt. Minun ei tarvitsisi ottaa uusia tehtäviä enää tässä sprintissä, mutta aion saada tämän valmiiksi ajoissa ja ottaa sitten kehitysjonosta jonkin tehtävän työn alle. Lisäksi tänään on kehitysjonon työstö, jossa siistimme pisteyttämättömiä tehtäviä. Aion myös jatkaa kehittämämme palvelun eteenpäinvientiongelmien ratkaisemista.

Pääkehittäjä olikin lomalla pari päivää, joten keskustelu back end -puolen tekemisestä jäi. Jatkoinkin kuitenkin palvelutehtävän tekoa jonkin verran. Katsoimme front end -kehittäjän kanssa animaatiota ja päätimme hienosäätää sitä. Lisäksi muutin funktioiden logiikkaa, jotta painikkeet ja tekstit toimivat paremmin animaation kanssa. Tein puolikkaan työpäivän, sillä minulla oli kertyneenä ylityötunteja.

9.3.2018 perjantai

Tänään aion tehdä laskukaavoja animaatioon, jotta ei tule ongelmia, jos korkeuksia pitää muuttaa ja ettei korkeus ole kovakoodattuna. Minun pitää huomioida myös mobiili, koska siellä on elementin

korkeus aivan toinen kuin isommissa ko'issa, kuten esimerkiksi isommalla tietokoneen näytöllä. Tämän jälkeen aion tehdä yleistä refaktorointia koodiin.

Sain laskukaavat melkein valmiiksi, mutta ne vaativat vielä viimeistelyjä. Lisäksi sain siistittyä front end -puolta ja kirjoitettua, mitä mikäkin funktio sisälsi. Ehdin näiden lisäksi myös hieman jatkaa palvelutehtävää. Pidin lyhyemmän päivän, joten siihen nähden ehdin saada paljonkin aikaiseksi ja ylitinkin päivän tavoitteet.

Viikkoanalyysi

Tällä viikolla edistyin paljon sekä front- että back end -puolen osaamisessa. Aloitin tähän projektiin React-ohjelman opiskelun, joten tästä aiheesta tulee joka päivä uutta osaamista. Olen huomannut, että Reactissa on hieman erilainen jaottelu, eli missä järjestyksessä tai mihin aikaan asiat tapahtuvat funktioissa kuin Angularissa. Varsinkin esimerkiksi animaation teossa oikeanlainen ajoitus on tärkeässä roolissa. Java-työskentelystäkin sain kuluvalle viikolla paljon kertausta.

Alkuviikosta selvitin vielä speksejä siitä, mitä designerit halusivat animaatioissa tapahtuvan. Selvitin erilaisia keinoja tehdä animaatioita Reactilla, joista sitten löysin sopivan lisäosan toteutukseen. Selvitin muiden kehittäjien kanssa back end -asioita, jotta kehitys on yhtenäinen ja linjassa myös muiden palveluiden kanssa. Viikon aikana tuli muutamia ongelmia animaation toteutuksessa ja palvelun käyttöönotossa, mutta nämä sain helposti selvitettyä muiden työryhmäläisten ja tiedonhaun avulla.

Toimivin toteutus meidän animaatiollemme onnistuu React Animate Height -lisäosan avulla, koska se on kevyt ja se käyttää valmiita CSS-luokkia, joten se ei vaadi paljoa konfigurointia (Stanko 2018, viitattu 11.3.2018). Näin animaatiosta tulee tarpeeksi kevyt ja yksinkertainen. Ennen tätä suunnitelin CSS:llä animaation toteutusta, mutta siitä olisi tullut liian monimutkainen prosessi ja elementit olisivat vaatineet uusia luokkia. Toinen vaihtoehto, jonka kerrottiin olevan helppo lisäys, oli ReactTransitionGroup (React 2018, viitattu 11.3.2018), mutta jostain syystä konfiguroinnissa meni joka kerta jotakin pieleen ja loppujen lopuksi se ei toiminut meidän sovelluksemme koontiversion kanssa. Tämä lisäosa oli myös paljon monimutkaisempi, emmekä tarvinneet niin monimutkaista animaatiota ainakaan vielä.

4.2 Viikko 11

12.3.2018 maanantai

Tänään aion katselmoida toisen kehittäjän koodia. Viimeistelen animaation laskelmat, jotta elementit näkyvät oikeissa paikoissa. Iltapäivällä jatkan palvelutehtävää ja keskustelen siitä mahdollisesti pääkehittäjän kanssa.

Laskelmien ja muiden asioiden näyttäminen meni ristiin sovelluksessa, joten animoinnin laskelmat on jätettävä myöhemmälle, kunnes näyttäminen on korjattu. Mobiilissa tässä kohtaan oli eniten ongelmia. Siirsin laskelmien tekemisen loppuun toiselle kehittäjälle, jotta hän voi laskea korkeudet samalla, kun hän korjaa mobiilinäkymän. Myös katselmointi liittyi kyseiseen ongelmaan, koska siinä juuri korjattiin sovelluksen näkymää eri tilanteissa. Loppupäivän valmistelin palvelutehtävää ja sain yhdistettyä osoita, joiden kanssa aiemmin oli ongelmia.

13.3.2018 tiistai

Jatkan uuden palvelun valmistelua. Meillä on kokous GIT-käytännöistä ja siitä, kuinka haluamme sitä käyttää työryhmämme kesken. Käymme läpi sivuston lokalisointia ja siitä kuinka se aiotaan toteuttaa. Päätaavoite on kuitenkin saada palvelutehtävän kehitys mahdollisimman pitkälle.

Etenin palvelutehtävässä kiitettävästi. Tein palvelulle oman testausluokan, jonka avulla palvelua voi testata testidatan kanssa ennen käyttöönottoa ja sen aikana. Meillä oli muutamien työryhmäläisten ja designerin kanssa kokous lokalisoinnista ja siitä kuinka se toteutetaan meidän projektissamme. Pidimme osittaisen kehitysjonon työstön ja perään Git-kokouksen siitä, kuinka haluamme käyttää Gitiä työryhmässä.

14.3.2018 keskiviikko

Jatkan palvelutehtävää ja aion tehdä yksikkötestejä projektille lisää. Opettelen käyttämään IntelliJ IDEA-ohjelmaa Eclipsen lisäksi.

Sain aika hyvin käytyä IDEAn peruskomennot ja ominaisuudet läpi samalla, kun jatkoin palvelun työstämistä. Jatkoin yksikkötestejä sitä mukaa, kun huomasin siihen tarvittavia osia.

15.3.2018 torstai

Jatkan palvelun viemistä eteenpäin ja aion saada sen loppuun asti toimimaan tässä palvelussa. Opiskelen lisää IDEAn käyttöä ja lisään siihen tarvittavia lisäosia ja opiskelen testausta sillä. Projektista pitää poistaa myös vanha palvelu käytöstä sitten, kun uusi korvaava palvelu on viety loppuun asti.

Sain palvelun valmiiksi. Ongelmia ilmeni sekä Eclipsen, että IDEAn testaustyökaluissa, joten testausta täytyi siirtää, kunnes saan työkalut toimimaan. Uudessa palvelussa oli ongelmia API:n URL:ssa, mutta korjasin ongelman, ja sovelluksen koonti meni onnistuneesti läpi. Uudesta palvelusta pitää vielä käydä läpi tarvittavat id:t ja selvittää tämän jälkeen putki läpi front end -puolelle asti, mutta se ei kuulu tähän tehtävään.

16.3.2018 perjantai

Tänään poistan käytöstä vanhan palvelun, jonka korvaavan palvelun olen saanut laitettua projektiimme. Aion tarkastella back- ja front end -yhteyksiä, jotta testidata saadaan otettua pois käytöstä. Näiden jälkeen voisin tehdä yksikkötestit loppuun juuri käyttöön otetulle palvelulle.

Sain otettua vanhan palvelun pois käytöstä ja testasin yleisesti projektia. Testauksen kanssa oli aiemmin ongelmia, mutta sain niihin apua pääkehittäjältä. Laitoin testauksen toimimaan sekä Eclipsellä, että IntelliJ IDEAlla. Front end -yhteyttä ei vielä ole. Yksikkötestit ovat vielä jonkin verran kesken, mutta niitä voi jatkaa ja täydentää ajan mittaan.

Viikkoanalyysi

Tällä viikolla kehityin erityisesti back end -puolella. Kertailin hyvin Javan perusasioita ja tein perusfunktioita. Muilta kehittäjiltä sai myös paljon uusia vinkkejä ohjelmien käyttöön, yleiseen työskentelyyn ja asioiden helpottamiseksi. Hyvä ja tärkeä asia meidän työryhmässämme on, että kaikki auttavat toisiaan.

Debuggauksesta opin uutta asiaa, kuinka soapUI:lla lähetetään käskyjä ohjelmalle, jota ei ole vielä yhdistetty front end -puolelle. Minun pitäisi käyttää enemmän aikaa debuggaukseen ja yhteyksien tarkasteluun. Debuggauksen ja esimerkiksi komentojen opetteluun kannalta minun pitäisi valita alanko käyttämään jatkossa Eclipseä vai IDEAa. Debuggauksen osaaminen on pakollinen taito

Java-kehittäjälle varsinkin jatkossa, kun debuggausta käytetään enemmän ongelmien ratkomi-
seen, bugien etsimiseen ja korjaamiseen (Eclipse 2017, viitattu 18.3.2018).

Viikon alussa minun täytyi selvittää front end -kehittäjän kanssa animaatioasiaa. Tämä kuitenkin
siirrettiin silloin front end -kehittäjälle, koska tehtävämme menivät ristiin. Loppujen lopuksi front end
-kehittäjä joutuikin muuttamaan koko front end -logiikkaa, jotta mobiili, selain ja animaatio-osiot
saatiin toimimaan yhdessä sujuvasti ja luotettavasti.

Yksikkötestien kanssa oli ongelmana, etten muistanut, kuinka ne toteutetaan, joten etsin siitä tietoa
ja otin mallia jo toteutetuista projekteista. Jouduin jonkin verran pohtimaan, mitä kaikkea kannattaa
testata uuden palvelun toimivuudessa. Jotkut kehittäjät ovat sitä mieltä, että jokainen lauseke pitäisi
testata, mutta joka tapauksessa erityisesti kriittisimmät ja monimutkaisimmat osiot on kirjattava tes-
teihin (Vogella 2017, viitattu 18.3.2018).

4.3 Viikko 12

19.3.2018 maanantai

Uusi palvelumme täytyy ottaa erikseen käyttöön, ja käyttöönotto vaatii jonkin verran konfigurointia.
SoapUI:lla tarkastellaan minkälaista tietoa palvelu lähettää ja ottaa vastaan. Ainakin joissakin id-
hauissa saattaisi olla ongelmia, koska minulla oli vaikeuksia aiemmin tietyn id:n haussa. Pääkehittä-
jän mukaan meidän täytyisi tehdä muutoksia palveluun, mutta siihen täytyy kysyä lupa palvelun
tekijöiltä.

Olimme yhteydessä toisen palvelun kehittäjän kanssa, mutta emme vielä edenneet asiassa. Muok-
kasin jonkin verran projektia niin, että tiedot tulisivat oikeaan muotoon ja korjasin id-haun. Meillä oli
työryhmän kanssa suunnittelu ja retrospektiivi, joissa kävimme läpi projekteihin ja työryhmään liit-
tyviä asioita.

20.3.2018 tiistai

Tänään aion etsiä tietoa muista projekteista, kuinka niiden tekijät ovat käyttäneet palvelua ja saaneet tarvittavat tiedot ulos siitä ilman palvelun muokkaamista. Tämän jälkeen jatkan projektin debuggausta, jos sille on tarvetta. Jos saan tänään tehtävän tehtyä, niin otan uuden tehtävän Jirasta käsittelyyn.

Minun ei tarvinnut etsiä tietoa, koska ero löytyikin projektin xml-tiedostosta kun vertasin palvelun samaiseen tiedostoon. Sain palvelun toimimaan kokonaisuudessaan ja paransin samalla sovelluksen toimivuutta. Kaikki tiedot näyttäisivät tulevan oikein, ja yhdistäminen toisen palvelun tietojen kanssa onnistuu.

21.3.2018 keskiviikko

Tänään siivoan tekemääni palveluintegraatiota, minkä jälkeen otan työn alle jonkin uuden tehtävän. Päivällä olisi Skype-tilaisuus kehittämisen käytännöistä.

Sain siivottua tekemäni integraation. En osallistunut Skype-tilaisuuteen, koska en ehtinyt töiltäni. Nämä koulutustilaisuudet ovat yleensä vapaaehtoisia. Aloin muuttaa parametreja, joista tulee tällä hetkellä kovakoodattua tietoa niin, että niistä tulisi oikeaa tietoa. Katselmoin front end -kehittäjän tekemiä osioita, jotta ne pääsevät testaukseen.

22.3.2018 torstai

Jatkan parametrien ja niiden annotaatioiden muokkailua. Aion saada tämän tehtävän tänään valmiiksi ja tämän jälkeen otan kehitysjonosta uuden tehtävän. Meillä on tänään työryhmän kanssa kehitysjonon työstöä.

Sain muokattua sovelluksen niin, että se on valmis vastaanottamaan ja lähettämään oikeaa dataa. Kehitysjonossa ei ollut enempää tehtäviä, mitä tehdä tässä vaiheessa, joten tein tiedonhausta erilaisia versioita ja kirjoitin varmuuden vuoksi kyselyn, joka tallentaa ja päivittää mahdollista tietoa sivustolla.

23.3.2018 perjantai

Aion tehdä loppuun vaihtoehtoisen tiedonkäsittelyä ohjaavan toiminnon. Tämä on ainut tehtävä, jonka ehdin tänään tehdä.

Kirjoitin uuden erilaisen kyselytiedoston, joka hakee käyttäjän tiedot. Sovelluksella oli olemassa jo vastaava tiedosto, mutta halusin kokeilla, olisiko minun lähestymistapani toimivampi. Lisäksi kirjoitin varmuuden vuoksi tallennustoiminnon sovellukseen, vaikka todennäköisesti sovelluksemme ei tarvitse tällaista toimintoa.

Viikkoanalyysi

Tällä viikolla kehityin eniten Java-kehittämisen osalta. Huomasin, että kannattaa käyttää hyväksi meidän avointa lähdekirjastoamme ja etsiä uusia näkökulmia sen avulla. Kertasin monipuolisesti perusfunktioita ja tiedostoyhteyksiä. Opin käyttämään SoapUI-ohjelmaa monipuolisemmin ja opin lukemaan sen antamia ilmoituksia.

Olen aiemmin ajatellut, että vain front end -puolen tekeminen antaa onnistumisen tunteen, koska muutokset huomaa heti. Erityisesti viime ja tällä viikolla olen alkanut ymmärtää paremmin back end -puolen yhteyksien muodostumista ja oppinut lukemaan paremmin SoapUI:lla ja lokien kirjaamisella saatavia tuloksia. Kun nämä yhteydet ymmärtää ja näkee, mitä sovelluksen taustassa tapahtuu, saa samanlaisia onnistumisen tunteita kuin front end -puolellakin.

Viikon aikana aloin käyttää aktiivisemmin OpenGrok-työkalua, jonka avulla voi esimerkiksi etsiä tiettyjen sovellusten lähdekoodeja ja navigoida lähdeluettelossa. OpenGrok ymmärtää useita tiedostomuotoja ja kirjaa lähdekoodin historiaa talteen. Työkalu parantaa avointa kehitystä. (OpenGrok 2018, viitattu 25.3.2018.) Minä käytin tällä viikolla OpenGrok-työkalua erityisesti etsimään vaihtoehtoisia lähestymistapoja tiettyihin ominaisuuksiin. Pidän työkalusta myös siksi, että pystyn itse katsomaan tarvittavat asiat eikä minun tarvitse häiritä kyseistä kehittäjää tai ladata koko projektia omalle koneelle.

Minulla oli ongelmia viikon aikana SoapUI-ohjelman kanssa, koska sovellus on minulle uusi ja se antoi omituisia virheilmoituksia. Jouduin tämän vuoksi pari kertaa palauttamaan kaikki koodit ja aloittamaan alusta. Toisaalta alusta aloittaminen oli hyväkin asia, koska juuri se kannusti minua etsimään vaihtoehtoisia tapoja toteutukselle.

4.4 Viikko 13

26.3.2018 maanantai

Tänään aion refraktoroida koodia ja käydä läpi yleisesti UI-speksejä. En aloita vielä mitään uutta tehtävää, vaan katson rauhassa, missä mennään, ja käyn läpi yksityiskohtaisemmin front- ja back end -puolen yhteyksiä. Lisäksi teen tarvittaessa yksikkötestejä.

Päivä meni juuri niitä asioita tehdessä, joita aioinkin tehdä. Opin jonkin verran uutta projektin yhteyksistä.

27.3.2018 tiistai

Tänään katson pääkehittäjän kanssa tekemiäni kyselyitä ja päätämme, millaiseen muotoon haluamme kyselyt sovellukseen. Tämän jälkeen katson tulevia tehtäviä Jirassa ja katson onko siellä mitään, minkä voisi toteuttaa jo nyt. Jos en löydä sieltä mitään, teen uusia yksikkötestejä tai jatkan enkryptauksen kirjoittamista sovellukselle.

Päätimme kyselyiden muodon ja saimme lopullisen päätöksen sille, ettei tallennuskyselyä tarvita eikä erillistä kyselyn hakua tietyille parametreille. Aloitin front end -puolen tehtävän katselmointia, mutta kyseisen tehtävän kehittäjä ehti lähteä kotiin enkä ehtinyt kysyä virheistä. Samalla kun selasin sovellusta kehitysympäristössä, löysin sieltä front end -bugin, josta tein raportin. Pääkehittäjä pyysi minua katselmoimaan back end -tehtävän, mutta se jäi ajanpuutteen vuoksi seuraavaan päivään.

28.3.2018 keskiviikko

Katson kyselytoiminnon loppuun ja raportoin front end -kehittäjälle löytämistäni ongelmakohtista tehtävässä, jonka katselmoin eilen. Lisäksi raportoin kehittäjälle painike-bugista, jonka löysin. Näiden korjausten jälkeen katselmoin tehtävän uudestaan. Katselmoin back end -puolen tehtävän, joka minulta jäi eilen katselmoimatta. Jos jää aikaa, niin otan uuden tehtävän kehitysjonolta sprintin ulkopuolelta.

Front end -kehittäjä korjasi löytämäni bugit, joten katselmoin ne uudestaan ja kuittasin tehdyksi. Katsoimme back end -kehittäjän kanssa vähän tarkemmin hänen tekemäänsä tehtävää listojen lajittelusta ja logiikkaa kyselyparametreihin. Tein parametreja käsittelevän tehtäväni loppuun ja lisäsin ne projektiin. Loppupäivän kirjoittelin enkryptausta ja dekryptausta tietyille parametreille projektissa.

29.3.2018 torstai

Tänään teen palveluun kyselyn, jonka perusteella katsotaan, onko tietty palvelun ominaisuus käytössä asiakkaalla. Tarvittaessa jatkan yksikkötestejä, jos tiettyjä ominaisuuksia täytyy testata. Tutkin yhden parametrin tarkemmin, jotta saan tietää, minkä takia se oli pakollinen tieto hakea palvelusta ja mitä sillä tehdään. Korjailen projektin lokitusta, koska kiireessä se jäi parin viime viikon aikana vajavaiseksi.

Aloitin yksinkertaisimmasta tehtävästä eli lokituksen korjailusta. Poistin ylimääräisiä lokituksia ja lisäsin joihinkin lokeihin tekstejä, jotta niistä olisi helpompi ymmärtää, mitä ne koskevat. Lokitus-tehtävän jälkeen tarkastelin parametriasiaa ja muutin joidenkin parametrien nimiä kuvaavammiksi. Lisäsin kryptaustiedostot projektiin. Jatkoin yksikkötestejä päivän lopussa. Kyselytehtävä jää seuraavalle viikolle.

30.3.2018 perjantai

Pitkäperjantai

Viikkoanalyysi

Tämä viikko oli lyhyt, koska perjantai oli pitkäperjantai, joten meillä oli vapaapäivä. Tällä viikolla opin erityisesti yhteistyön merkityksestä katselmoinneilla, parikoodauksella ja tutkimalla erilaisia toteutuksia yhdessä muiden työryhmäläisten kanssa. Olen oppinut myös, että projekteissa tulee paljon tehtävää myös Jiraan merkittyjen tehtävien ulkopuolelle.

Tällä viikolla erityisesti katselmointien tärkeys on painottunut. Olen viikolla itse katselmoinut sekä minun tehtäviäni on katselmoitu. Katselmointi ehkäisee tehokkaasti bugien pääsyä kehitettävään kohteeseen ja helpottaa niiden korjaamista heti ennen kuin rinnalle laitetaan uusia ominaisuuksia. Välillä katselmoiteja on hyvä käydä läpi kehittäjän kanssa, jotta hän voi paremmin selittää, miksi asiat on tehty juuri siten. Samalla asiaan saattaa löytyä erilaisia ratkaisuja, korjauksia tai lisäyksiä.

Yrityksen kannalta katselmointi säästää aikaa ja rahaa sekä virtaviivaistaa kehitystyötä, jolloin laadun varmentamiseen tarvitsee käyttää vähemmän resursseja. Katselmoinnissa saattaa löytyä myös vikoja, jotka tulisivat ilmi vasta loppukäyttäjällä. Katselmointi myös parantaa työilmapiiriä, koska kehittäjät puhuvat enemmän toisilleen, mikä taas vähentää koodin omistajuuden tunnetta ja

auttaa erityisesti nuoria kehittäjiä kirjoittamaan puhtaampaa koodia. (Collaborator 2017, viitattu 2.4.2018.)

Selvitin viikon aikana vaihtoehtoisia tapoja koodin kryptaukselle, mutta loppujen lopuksi otin mallia olemassa olevasta toteutuksesta, jossa käsiteltiin tilinumeroita ja loput kirjoitin itse. Viikon aikana ei ehtinyt tulla suurempia ongelmia, vaan eheytin ja jatkoin vanhoja asioita ja kävin läpi paljon muiden koodeja.

4.5 Viikko 14

2.4. maanantai

2. pääsiäispäivä

3.4.2018 tiistai

Tänään kirjoitan tarkistuksen, joka minulla jäi tekemättä viime torstaina. Tarkoitus on tarkastaa palvelusta, onko tietty osio käytössä käyttäjällä. Teen muutoksia tekemääni kryptaustiedostoon ja muokkaan taas yksikkötestejä. Työryhmällämme on tänään myös demo, sprinttikatselmus ja sprintin suunnittelupalaveri. Näiden kokousten jälkeen saan uutta tekemistä.

Kävin läpi tarkistusta varten erilaisia päivämääräkonvertointeja, sillä integroidussa palvelussa parametrit olivat eri muotoa. Kirjoitin tarkistuksen loppuun, debuggasin ja laitoin Gitiin. Päivän palaverit sujuivat normaalisti, ja demossa esittelimme tuotetta oikealla datalla. Tarkistuksen viimeistely jää seuraavaan päivään.

4.4.2018 keskiviikko

Tänään viimeistelen tarkistuksen hienosäädön ja katson, tarvitseeko päivämääräparametri erilaisen konvertoinnin. Sovimme työryhmän kanssa, kuka alkaa tehdä mitäkin tehtävää. Aloitan uuden tehtäväni ja työstän sitä loppupäivän.

Sain tarkistuksen viimeistelyn valmiiksi. Jatkoin tämän jälkeen taas yksikkötestejä eteenpäin. En saanut otettua uutta tehtävää vielä, koska oli sen verran kiireinen päivä kaikilla, ettemme saaneet sovittua, kuka tekee mitään.

5.4.2018 torstai

Otan uuden tehtävän, kunhan muut tulevat töihin ja saamme sovittua tehtävienjaosta. Siihen asti katson, mitä tehtäviä sprintissä on, ja katson suunnilleen, mitä niiden eteen pitäisi tehdä. Myöhemmin alan työstää uutta tehtävää. Meillä on aamusta myös tehtäväjono siistimistä, jossa taas pisteitämme pisteyttämättömiä tehtäviä ja kertaamme, mitä tehtävät sisältävät.

Otin aamulla pari tehtävää työn alle. Toisessa tarkastetaan, että loppukäyttäjälle tulee tarvittavat virheilmoitukset, jotka on jo aiemmin tehty. Käyn vielä speksejä läpi, jotta kaikki tarvittavat ovat koodissa. Toinen tehtävä tulee olemaan ohjeistuksen tekeminen tiettyyn tilanteeseen käyttäjälle. Aloitin ensimmäiseksi virheiden tarkistuksen, jonka sain tehtyä suhteellisen nopeasti, kun sain sovelluksen riippuvuudet taas toimimaan.

6.4. perjantai

Tänään aion jatkaa tehtäviä, jotka aloitin eilen. Ensimmäiseksi minun pitää saada virheilmoitus siitä, kun serveripäässä menee jotakin pieleen ja ohjelma ei toimi. Oikeastaan tähän tehtävään ei pitäisi tehdä muuta, vaan toteutuksen pitäisi olla olemassa jo jossakin muodossa. Tämän jälkeen jatkan toisen tehtävän tekoa, jossa virheilmoitukset ovat vielä kesken.

Sain ensimmäisen tehtävän virheilmoituksen näkymään kehitysympäristössä muuttamalla koodia niin, että käyttäjän tulee olla kirjautunut, jotta näkee sovelluksen. Toisen tehtävän kehitys on vielä kesken. Front end -toteutus on jotakuinkin valmis, mutta sovellus vaatii vielä muutoksia back end -puolelle.

Viikkoanalyysi

Tälläkin viikolla huomasin, että vaikka kehitysjonossa ei olisi tehtävää, niin tällaisessa isossa palvelussa, jota kehitämme, on koko ajan jotakin pientä tehtävää – ainakin vähintään koodin refaktointia. Tällä hetkellä sovelluksessa on sellainen vaihe, jossa ei välttämättä ole kaikille tehtävää

kehitysjonossa. Esimerkiksi kaikki front end -puolen tehtävät liittyvät toisiinsa niin paljon, että front end -kehittäjällä on tehokkainta tehdä ne kaikki samalla kertaa.

Refaktoroinnin tärkein tehtävä on saada aikaan puhdasta ja yksinkertaista koodia. Tarkoitus on tehdä koodista muillekin kehittäjille helposti luettavaa. Muuttujien pitäisi olla nimetty kuvaavasti ja luokkien tulisi pysyä järkevissä mittasuhteissa. Koodissa tulisi välttää duplikointia ja sen tulisi pysyä lyhyenä. Puhdas koodi on kaikin puolin helpompi ja halvempi ylläpitää. (Refactoring Guru 2018, viitattu 8.4.2018.) Refaktoroin koodia säännöllisesti ja erityisesti tehtävien välissä, jolloin on eniten aikaa. Refaktorointia on helppo suorittaa esimerkiksi samalla, kun toinen kehittäjä katselmoi koodia.

Refaktoroinnin lisäksi voi tehdä muutakin lisätyötä, kun kehitysjonossa ei ole tehtäviä. Tehtäviä voi ottaa esimerkiksi muualta kuin meneillään olevasta sprintistä. Tämän lisäksi olen jutellut muiden kanssa kehitystyöryhmässä, koska aina joku keksii jotakin tehtävää, jos ei itse keksi. Näistä asioista on kätevää sanoa esimerkiksi päiväpalaverissa, jossa yleensä kaikki ovat paikalla. Tekeminen ei koskaan lopu kesken.

Viikolla minulla oli riippuvuusongelmia. Näin isossa palvelussa on paljon riippuvuuksia, joita pitää muistaa päivittää ja koota yhteen säännöllisesti. Tutkin riippuvuuksia Jboss Management -työkälulla, mutta en löytänyt ongelmaa, joten poistin riippuvuudet ja latasin ja otin käyttöön ne uudestaan. Vaihdoin myös käyttöönottokomentoa, jolloin riippuvuudet otettiin käyttöön eri elementin kautta. Riippuvuudet alkoivat toimia, ja pääsin piankin jatkamaan kehitystyötä.

4.6 Viikko 15

9.4.2018 maanantai

Tänään jatkan tehtävää, jossa kehitetään virheilmoitusten näkymistä käyttäjälle. Kuuntelen samalla töitä tehdessä OP:n arkkitehtuuripäivän broadcastia tavoitearkkitehtuurista ja broadmapeista. Ilta-päivällä meillä on infoa tietosuojahankkeesta.

Arkkitehtuuripäivän broadcast jäi kuulematta aamupäivän osalta yhteyksien pätkimisen vuoksi. Onneksi siitä on tallenne myöhemmin kuunneltavaksi. Etenin tehtävässäni kiitettävästi ja sain selvitettyä yhteyksiä asioiden välillä. Osallistuin kuitenkin toiseen Skype-kokoukseen. Sain jatkettua tehtävääni eteenpäin, ja jatkan sitä vielä huomenna.

10.4.2018 tiistai

Tänään aion saada toisenkin virheilmoitustehtävän tehdyksi. Siihen pitäisi olla lokitus kunnossa, että näkee lokilta, mitä haussa menee pieleen, mutta tarvitaan vielä erilaisia epäonnistumisviestejä loppukäyttäjälle ja ohjeet, kuinka toimia tällaisissa tilanteissa.

En saanut virheilmoitustehtävää tehtyä kokonaan, koska siihen tuli paljon lisää sisältöä, joka vaati viisi uutta Java-luokkaa avuksi. Laajensin virheilmoitusluokkia käsittämään samalla muutkin tarpeelliset http-käsittelyt ja lisäsin erilaisia ehtoja.

11.4.2018 keskiviikko

Tänään jatkan virheilmoitustehtävää ja parannan koontifunktiota, joka siihen liittyy. Meillä on työryhmän kanssa tehtäväjonon siistimistä aamulla. Tämän jälkeen jatkan tehtävääni ja jos saan sen valmiiksi, katselmoitan sen jollakin työryhmästämmä. Aion tutkia Splunk-työkalun toimivuutta loppupäivästä, jos minulla on vielä noidenkin jälkeen aikaa.

Virheilmoitustehtävän tekeminen jatkuu vielä vähintään viimeistelyllä, jos ei vielä tule uusia ongelmia vastaan. En ihan päässyt tavoitteeseen saada tehtävää valmiiksi, mutta aika lähellä jo ollaan. Opin paljon uutta http-tilan käsittelemisestä, niiden ehdoista ja kehityspotkesta front end -puolelta back end -puolelle. Koska tässä nyt kestikin oletettua kauemmin, joku tiimiläisistäni ehti ottaa Splunk-tehtävän ennen minua.

12.4.2018 torstai

Tänään aion taas jatkaa virheilmoitustehtävää, ja toivon mukaan saan sen tänään valmiiksi ja saan laittaa sen katselmointiin. Meillä on iltapäivällä kokous liittyen arkkitehtuuriin.

Virheilmoitustehtävässä tuli vieläkin uusia asioita, joita en ollut vielä tehnyt, joten tehtävän teko jatkuu vielä perjantain puolelle. Arkkitehtuurikokous peruttiin.

13.4.2018 perjantai

Tänään saan virheilmoitustehtävän valmiiksi ja sen jälkeen alan refaktoroida tiettyjä osioita, joita ei ole vielä ehditty siistiä. Meillä on tietosuojainfoa Skypen välityksellä aamupäivällä.

Sain virheilmoitustehtävän valmiiksi, ja se meni katselmoitavaksi. Tehtävää voi joutua korjaamaan, jos siitä löytyy jotakin katselmoinnissa.

Viikkoanalyysi

Viikko sujui kaiken kaikkiaan hyvin normaalisti. Käsittelin koko viikon virheilmoituksia ja http-statuk-sia. Opinkin niistä paljon uutta asiaa tekemällä ja lukemalla. Scrumissa meillä oli normaalit viikko-kokoukset ja käytänteet, joita pohdin tämän viikon analyysissä. Lisäksi olen opiskellut Splunk-käy-täntöjä, vaikka loppujen lopuksi en päässytkään toteuttamaan Splunk-tehtävää.

Rest- ja http-asiat olivat minulle ennestään tuttuja, mutta kertasin niitä samalla, kun tein tehtävää. Opin myös paljon uutta ja pystyin näin tekemään monimutkaisemman funktion, joka käsittelee eri-laisia virheilmoituksia. Päättavoite tehtävässä oli saada ulos koodi "200 OK", joka on yleinen onnis-tumiskoodi (REST API Tutorial, viitattu 15.4.2018). Lisäksi hain statukset 400 ja 201 sekä tein niille erilaisia ehtoja. Tein tässä tehtävässä enemmän kuin pyydettiin, mutta se ei haittaa, sillä jatkokehityksestä on tässä tilanteessa vain hyötyä.

Tällä viikolla ei tullut monta Scrumiin liittyvää kokousta, mutta olemme työryhmän ulkopuolisten työkavereiden kanssa pohtineet Scrumin käytäntöjä ja muun muassa sitä, mikä on Scrumin tärkein kokous. Mielestäni juuri päiväpalaverit ovat todella tärkeitä työryhmän kannalta, koska viimeistään niissä voi sanoa kaikkien kuullen, jos on ongelmia kehityksessä. Päiväpalavereissa myös kuulee, mitä muut ovat tekemässä, ja kaikki pysyvät jatkuvasti perillä asioista. Scrumiin liittyvässä sertifikaattikoulutuksessa oli sanottu, että Scrumin tärkein kokous on retrospektiivi, jonka tarkoituksena on käydä läpi, mitä hyvää sprintissä oli ja mitä voisi parantaa (Wikipedia 2018a, viitattu 15.4.2018).

Splunk on työkalu, joka kaappaa, indeksoi ja korjaa reaaliaikaista tietoa, josta se voi luoda kaavi-oita, raportteja, hälytyksiä ja visualisointeja. Splunk tarjoaa liiketoiminnalle tietoa diagnosoimalla sitä. Splunk-työkalua voi käyttää sovellusten hallintaan, turvallisuus- ja vaatimusmukauttamiseen sekä liiketoiminnan ja web-analyysin toteuttamiseen. (Wikipedia 2018c, viitattu 15.4.2018.) Näin Splunk-työkalusta on hyötyä sekä meille kehittäjille että koko liiketoiminnalle.

4.7 Viikko 16

16.4.2018 maanantai

Hoidan aamupäivän sähköposteja ja muita pikkujuttuja. Lounaan jälkeen meillä on sovelluksen demo, sprinttikatselmus ja sprintin suunnittelupalaveri. Tämän kokouksen jälkeen aloitetaan uusi sprintti ja saan uusia tehtäviä.

Kokoukset pidettiin, mutta en vielä ottanut uusia tehtäviä, vaan iltapäivä meni kaikkien pienten asioiden hoitamisessa. Katsoin vielä kertaalleen virheilmoitustehtävän läpi, ettei sitä tarvitsisi enää korjailla jälkeenkään.

17.4.2018 tiistai

Tänään kuuntelen Skype-kokousta lähes koko päivän. Yritän samalla aloittaa uutta tehtävää. Tämä päivä on minulla lyhyempi.

Lisäilin vielä virheilmoitukseen osia, jotka olivat tarpeellisia. Teen virheilmoitukselle vielä yhden erilaisen toteutuksen, jota voimme käyttää tässä vaiheessa - sellaisen virheilmoituksen, joka palauttaa pelkästään 200 OK, koska monipuolisempi virheilmoitus vaatisi vielä pitkän listan vakioita.

18.4.2018 keskiviikko

Tänään kirjoitan toisen tai kolmannen virheilmoitustoteutuksen, joka palauttaa vain yhden tiedon tarkastusfunktion avulla. Tämän jälkeen pitää vielä muokata front end -puolta vastaamaan tätä uutta back end -toteutusta. Tämän jälkeen testaan toimivuuden.

Sain kirjoitettua erilaisen virheilmoitusluokan, joka palauttaa ainoastaan tietynlaisen vastauksen. Muut vastaukset tehdään tarvittaessa myöhemmin. Lisäksi katselmoin front end -kehittäjän koodin, joka käsitteli saavutettavuutta. Opiskelin saavutettavuutta enemmänkin, jotta muistin miten saavutettavuus tulisi tehdä koodissa. Katselmoin vielä toisenkin tehtävän, joka käsitteli sitä, onko käytäjällä tietty ominaisuus käytössä. Katselmoitavissa tehtävissä ei ollut huomioitavaa.

19.4.2018 torstai

Virheilmoitustehtävässä on front end -päässä jokin ongelma, jonka aion tänään korjata. Meillä on myös työryhmän tehtävälisan järjestämistä ja siivousta aamupäivällä. Katson vielä pääkehittäjän kanssa tätä tehtävää, koska tehtävä on aika laaja.

Kokous peruttiin, koska tehtävälisalla ei ollut pisteytettävää. Tein uuden toteutuksen virheilmoitukseen, mutta siinä on vielä jotakin ongelmaa. Minun piti hakea uusia oikeuksia tiettyihin palveluihin, joita alamme tehdä tämän pääsivun jälkeen. Tehtävän teko jatkuu vielä perjantaina.

20.4.2018 perjantai

Tänään teen tehtävän back end -puolen loppuun. Tämän jälkeen teentehtävän front end -puolen loppuun. Aion osallistua iltapäivällä myös Demo Friday -tapahtumaan, jossa esitellään eri sovellusten demoja.

Teimme osan tehtävästä yhdessä pääkehittäjän kanssa, ja sen jälkeen hän opetti minulle lisää yksikkötestien tekemisestä ja Mockitosta yleensäkin. Opiskelin Mockittoa enemmänkin ja aion vielä myöhemmin jatkaa opiskelua. Kävin kuuntelemassa demot kahvilassa. Aloin kirjoittaa front end -puolelle virrehakua ja samalla funktiota, joka paremmin syrjäyttää testidatan.

Viikkoanalyysi

Tällä viikolla opin taas, että tehtävät saattavat vaikuttaa yksinkertaisilta, mutta niissä onkin odotettua enemmän työtä, kuten esimerkiksi virheilmoitustehtävä, jota olen jatkanut koko viikon ja joka on koko ajan laajentunut ja laajentunut. Loppujen lopuksi päätettiin tehdä mahdollisimman yksinkertainen toteutus. Poistin projektistamme suurimman osan kirjoittamastani koodista ja tallensin ne myöhempää käyttöä varten.

Meillä on käytössä Spring Framework, jonka kautta tulee ymmärtää myös RESTiä. RESTiä käytetään hajautettujen järjestelmien suunnitteluun. Tämän avulla toteutin meidänkin projektiimme http-kutsuja käsittelemään back- ja front end -yhteyttä, kun taas Spring on avoimen lähdekoodin sovel-luskehitysjärjestelmä, joka kehystää koko projektia. Spring-järjestelmässä REST-työkalun rakentamiseen http-pyyntöjä hallinnoi ohjain, jonka tunnistaa @RestController-annotaatiosta (Spring 2018, viitattu 22.4.2018), eli kontrollerissa käsitellään tämänkin viikon virhetehtävän toiminta ja suoritus.

Sain uusia vinkkejä pääkehittäjältä yksikkötestien kirjoittamiseen. Kirjoitimme yhdessä yhden yksikkötestin virheilmoitustehtävän funktiolle, ja tämän jälkeen jatkoin itsenäisesti testien kirjoittamista. Yksikkötesteihin käytämme Mockito-työkalua, joka auttaa kirjoittamaan puhdasta ja yksinkertaista APla. Mockito-ohjelmaa käytin annotaatioiden avulla. Yksikkötestiluokka oli jo aloitettu, ja initialisointiluokkana oli käytetty eri toteutusta kuin mitä me olisimme käyttäneet. Tämä luokka laitaa "mokittamaan" myös staattiset metodit.

Front end -puoli – jota ehdin aloittaa loppuviikosta – käsittelee http-kutsuja Ajaxin avulla. Kuten muutkin tämänkaltaiset ominaisuudet, http-kutsukin käsitellään Reactissa sille määrättyssä tilassa. Tässä tehtävässä kutsu on hyvin yksinkertainen ja oppikirjamainen toteutus, joka vain hakee datan ja käsittelee ja ilmoittaa käyttäjälle, jos jotakin epäonnistuu. Tähän olikin jo suurin osa toteutuksesta tehty aiemmin, joten minun piti vain muokata Ajax-kutsua.

4.8 Viikko 17

23.4.2018 maanantai

Tänään aion hoitaa aamulla työsähköpostit pois alta. Tämän jälkeen hoidan muitakin pikkujuttuja, joita on jäänyt kiireiden takia hoitamatta. Sitten alan katsoa virheilmoituksen front end -puolen loppuun säätöä ja siistin koko ketjua, jonka olen tehnyt tehtävän aikana.

Sain toteutettua kaikki päivän tavoitteet. Refaktoroin koodia. Virheilmoituksen putki toimii nyt, ja loppukäyttäjälle tulee oikeanlainen virheilmoitus.

24.4.2018 tiistai

Tänään aion ottaa jonkin uuden tehtävän käsittelyyn. Aion katselmoida muiden tekemiä tehtäviä ja käydä läpi yleisesti käyttöliittymän UI-puolta.

Virheilmoituksessa ilmeni jotakin ongelmaa, jota pitää vielä korjata. Etäyhteyksissä oli ongelmia, joten virheilmoitustehtävän korjaaminen täytyy tehdä seuraavana päivänä toimistolla. Sain kuitenkin katselmoinnit tehtyä ja käytyä läpi yleisellä tasolla toimivuutta, joten opin siitä paljon uutta.

25.4.2018 keskiviikko

Korjailen tänään bugin, joka ilmeni virheilmoitustehtävässä katselmoinnin jälkeen. Sitten meillä on tiimipäivä tänään, joten tuskin saan tehtyä korjausta loppuun.

Sain selvitettyä, missä virheilmoituksessa menevät asiat vikaan, ja opin taas uutta debuggauksesta. Ehdin tehdä noin tunnin töitä, ennen kuin lähdimme viettämään tiimipäivää.

26.4.2018 torstai

Tänään meillä on aamupäivän kestävä Skype-katsaus, jossa käydään läpi tuotannon nykytilannetta. Lisäksi meillä on kokous, jossa käymme läpi tuotteen kehitysjonon työstöä. Näiden kokousten lisäksi jatkan virheilmoitus-bugin korjausta.

Sain luvan osallistua huomenna saavutettavuuskoulutukseen, joten asensin koneelle valmiiksi NVDA-ruudunlukuohjelman ja harjoittelin sen käyttöä. Kehitysjonon työstö meni normaalisti. Ehdin osallistua vain noin tunnin ajan Skype-katsaukseen. Opin jälleen, miten priorisoida tehtäviä ja kokouksia. Iltapäivällä osallistuin vielä arkkitehtikokoukseen sekä Opsetin henkilöstöpulssin purkuun.

Koko päivä oli aika lailla kokouksia, mutta ehdin silti työstää bugia eteenpäin. Debuggasin siis paljon löytääkseni, missä jotakin menee pieleen.

27.4.2018 perjantai

Tänään minulla on lähes koko päivän kestävä saavutettavuuskoulutus, jossa käymme läpi saavutettavuuden linjausta ja ruudunlukuohjelman käyttöä. Välissä jatkan taas omia tehtäviäni.

Ehdin aamulla tunnin verran katsoa bugia ja asensin koneelle lisäosan, joka helpottaa debuggausta selaimessa. Tänään kuulin ensimmäistä kertaa kyseisestä ohjelmasta. Lisäksi opin todella paljon uutta saavutettavuudesta ja NVDA-ruudunlukuohjelman käytöstä.

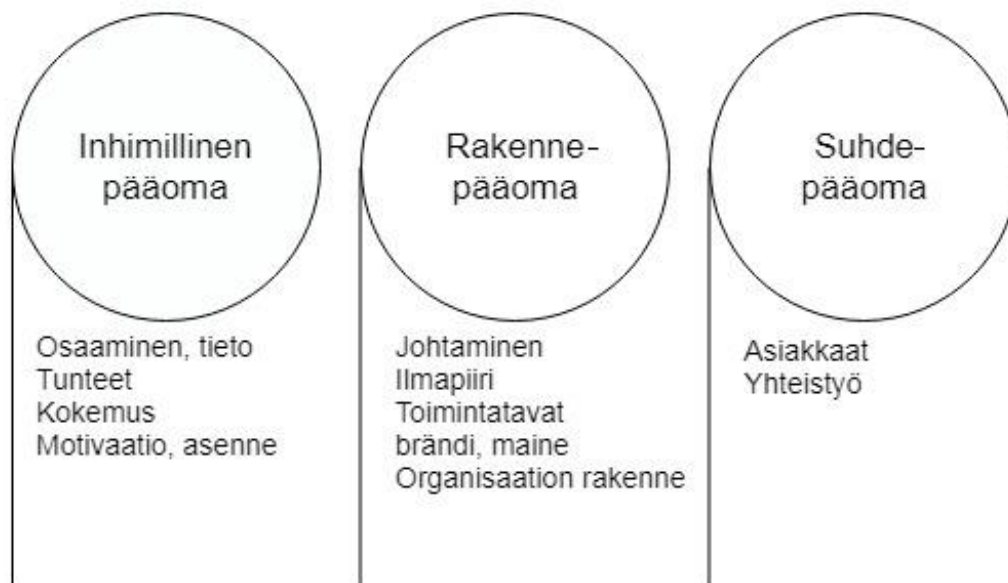
Viikkoanalyysi

Tällä viikolla olen ehtinyt tehdä kehitysjonon tehtäviä vähemmän kuin yleensä, sillä meillä on ollut paljon kokouksia ja koulutuksia. Kokouksista ja Skype-infoista kuitenkin oppii aina uutta meneillään olevasta projektista ja muista ajankohtaisista asioista, kun asioita pohditaan yhdessä. Hyvä asia itsensä kehittämisen kannalta on myös koulutukset, joihin pääsee joskus mukaan. Tällä viikolla

opin myös kahden uuden lisäosan käytön kehittämisen tueksi: NVDA-ohjelman sekä Modify Headers for Google Chromen.

Työntekijän älyllinen pääoma lisää organisaation tuottavuutta siinä missä niin sanotut oikeat työtehtävätkin. Tätä asiaa olen tutkinut tällä viikolla ja piirsin kuvion ajatuksen tueksi (Kuvio 2.). Tämä luokittelu on yleisesti käytetty käsitteiden havainnollistamiseksi. Älyllisellä pääomalla on suuri merkitys liiketoiminnalle, mutta sen syy-seuraussuhteita on hankala arvioida (Mäki 2017, viitattu 28.4.2018) Mäen mukaan ICT-alalla älyllinen pääoma on osallisena kaikissa yrityksen toiminnoissa aina palvelun luonnista johtamiseen. Jäin pohtimaan viikon aikana älyllisen pääoman merkitystä varsinkin siksi, että usein viikoista, joissa ei varsinaisesti kehitä tehtäviä, tulee olo, ettei ole saanut mitään aikaiseksi.

Älyllinen pääoma



KUVIO 2. Älyllinen pääoma

Toinen aihe, josta opin tällä viikolla, oli front end -debuggaus, ja siihen liittyi Chromen lisäosa Modify Headers for Google Chrome. Aikaisemmin olenkin jo pohtinut debuggausta yleisesti ja painottanut enemmän back end -puolen debuggausta. Front end -debuggaukseen liittyvät yleisesti käsitteet konsoli ja lokittaminen. Minun työskentelyssäni konsoli on Chromen Developer Tools, jonka

avulla hoidan suurimman osan front end -debuggauksesta. Tällä viikolla työkaverini vinkkasi kuitenkin Modify Headers for Google Chromesta, jonka käytössä hän auttoi minut alkuun, ja ehdimme tutkailla testiympäristöä. Lisäosa on tarkoitettu lisäämään, muokkaamaan ja filtteröimään http-tunnisteita. Aion jatkaa siihen perehtymistä vielä myöhemmin.

Kolmas isompi aihealue tällä viikolla oli saavutettavuuskoulutus sekä siihen liittyvä NVDA-ruudunlukuojelma. Ohjelma mahdollistaa tietokoneen ja meidän tapauksessamme projektimme käytön näkövammaisille sekä muuten rajoittuneille ihmisille. Opin saavutettavuudesta ja ohjelman käytöstä todella paljon uutta ja toivon, että näitä oppeja tullaan pitämään tärkeinä myös liiketoiminnan kannalta, jotta näitä asioita kehitettäisiin ja pidettäisiin yllä. Olisin toivonut, että tämä koulutus olisi ollut tarjolla kaikille, koska nämä ovat tärkeitä asioita.

4.9 Viikko 18

30.4.2018 maanantai

Tänään katson bugia, mutta uskoisin, että bugi on front end -puolella, jota toinen kehittäjäme on käsitellyt. Kuuntelen kehittämisen malleja käsittelevää infoa ja käyn läpi sähköposteja. Jatkan uutta tehtävää, jos bugi saadaan selvitettyä.

Sain selvitettyä bugia. Opin uusia asioita organisaatiosta ja liiketoiminnasta. Katselin saavutettavuuden muistiinpanot, jotta voin ohjeistaa toista kehittäjäme koulutuksen pohjalta.

1.5.2018 tiistai

Vappu.

2.5.2018 keskiviikko

Tänään meillä on projektin demo, katselmointi, suunnittelu ja retrospektiivi. Jatkan bugi-tehtävää ja katson, onko aloittamani tehtävä jo tehtävissä.

Sain aika pitkälti debugattua, mistä bugi tulee, mutta korjausta ei ole vielä tehty. Kokoukset menivät normaalisti. Saimme uutta ohjeistusta saavutettavuuteen, jonka kävin läpi iltapäivällä.

3.5.2018 torstai

Meillä alkoi uusi sprintti, ja siinä on paljon katselmointitehtäviä, joita aion tänään alkaa katsoa. Niiden välissä katson myös, että löydänkö bugitehtävään ratkaisun vai miten asian kanssa kannattaa edetä.

Katselmoin paljon muiden tehtäviä. Kysyin muilta, mitä mieltä he olivat bugista, ja saimme selvitettyä, että bugi on Ajax-kutsussa. Korjaaminen jäisi kuitenkin seuraavalle päivälle. Opin, että kannattaa joka tapauksessa etsiä bugin syytä joka puolelta, vaikka osan koodista onkin kirjoittanut joku muu, ja hän on varma, ettei kyse ole hänen tekemästään osasta.

4.5.2018 perjantai

Tänään korjaan bugin, ja sen jälkeen aloitan uuden tehtävän. Sprintissä ei ole nyt paljon tehtävää, koska sprintti on lyhyempi, mutta katsomme muiden kanssa, mitä he haluavat tehdä ja mitä minä sitten teen. Meillä on halukkaille muutaman tunnin Git-koulutus, jonne aion osallistua.

Sain korjattua bugin, ja se vaati vain Ajax-kutsun muokkaamista niin, että se saa otettua virheilmoituksen vastaan back end -puolelta. Sain pari uutta tehtävää, joista aloitin toista saamatta sitä vielä valmiiksi. En osallistunutkaan Git-koulutukseen, koska kyse oli ainoastaan Git-työkalun perusosaamisesta ja jatkokurssi on vasta myöhemmin. Kävin kuitenkin läpi tärkeimmät asiat Git-työkalusta koulutuksen jälkeen ja opin muutaman uuden komennon.

Viikkoanalyysi

Tällä viikolla minulla ei ollut kovin isoja ongelmia ja sain aika nopeasti selvitettyä tarvittavat asiat. Opin paljon uutta kehittämisen kannalta muun muassa erilaisista bugeista ja niiden korjauksesta, Git-käytännöistä ja Ajax-kutsuista. Minulla oli viikon aikana joitakin työn ulkopuolisia asioita hoidettavana. Tällaisten asioiden hoitaminen on tehty työpaikallamme helpoksi, sillä pystyn tekemään työni etänä ja liukuvien työaikojen ansiosta voin esimerkiksi tarvittaessa pidentää päivää parilla tunnilla, jotta tunnit tulevat täyteen.

Periaatteessa bugi, josta olen puhunut viikon aikana, ei ole oikeasti bugi, koska ongelma esiintyi jo ennen kuin testaaja sulki kyseisen tehtävän. Näin sitä ei tarvinnut merkitä bugiksi, vaan tehtävän kehitysosion sai aukaista uudelleen. Sanotaan, että ensimmäinen löydetty bugi oli 1947 Grace

Hopperin lokikirjaan kirjaama koi, joka löydettiin tietokoneen sisältä. Tämän myötä myös nykyinen ohjelmointibugi on saanut nimensä. Virheiden korjaamisprosessia sanotaan debuggaukseksi, ja jo 1950-luvulta lähtien tietokonejärjestelmät on suunniteltu myös estämään, löytämään tai korjaamaan virheitä toiminnan aikana. Bugeja on olemassa paljon erilaisia, ja monesti syynä ovat kirjoitusvirheet koodissa, kehitysmenetelmät, ohjelmointikielien heikko tuki ja analyysivirheet (Wikipedia 2018b, viitattu 6.5.2018). Kun ohjelmoijan osaaminen kehittyy, kirjoittaa hän enemmän koodia, jossa on vähemmän bugeja.

Viikolla korjasin bugin käyttäen Ajax-työkalua. Ajax kommunikoi palvelimen kanssa ilman, että käyttäjän täytyy päivittää sivustoa. Ajax viittaa XMLHttpRequest-objektien käyttämiseen palvelimien kanssa JavaScriptin avulla. Käsittelin aiemmin virheilmoitusten tulemistä palvelimelta ja nyt otin virheilmoituksen käsittelyyn front end -puolella Ajaxin avulla, jolloin sain käyttäjälle virheilmoituksen näkymään suoraan ilman sivun päivittämistä. Ajaxin käyttö myös nopeuttaa palvelinkutsuja, koska se tallentaa tietoja lähettämättä koko sivua. Sovellukset ovat näin responsiivisempia, nopeampia ja käyttäjäystävällisempiä (Segue Technologies 2013, viitattu 6.5.2018).

Tälläkin viikolla sain mahdollistettua joustavan työnteon etätyöskentelymahdollisuuden avulla. Etätyöskentely jakaa paljon mielipiteitä, ja joillekin se sopii hyvin ja toisille ei. Etätyöskentely mahdollistaa sen, että työntekijä pystyy keskittymään asioihin rauhassa ilman keskeytyksiä, ja hänen voi olla helpompi keskittyä ajatusta vaativaan työhön. Toisaalta työskentely saattaa venyä liian pitkäksi huomaamatta tai tauotus voi unohtua. Etätyöskentely voi säästää työntekijän aikaa, kun ei tarvitse erikseen siirtyä työpaikalle. Meillä on työssä käytössä myös monia välineitä, joilla ottaa yhteyttä muihin etänäkin, kuten Skype. Välillä etäyhteyksien kanssa saattaa olla ongelmia, jolloin työskentely tai yhteydenpito estyy. Hyödyllinen asia etätyöskentelyssä on, että voi työskennellä siellä, missä sillä hetkellä tuntuu hyvältä. Toimistolla työskentelyllä saa hyvin erotettua työ- ja vapaa-ajan. Olenkin toimistolla suurimman osan työajastani viikoittain.

4.10 Viikko 19

7.5.2018 maanantai

Tänään jatkan uutta tehtävää, jonka aloitin perjantaina. Toivon mukaan saan sen tänään valmiiksi, jos siihen ei tule uusia asioita, jotka tulee huomioida. Sain tehtävän lähes kokonaan valmiiksi, joten

sovelluksen yhden osan ikonit toimivat nyt, ja sen osan tiedot kulkevat oikein back end -puolelta front end -puolelle.

8.5.2018 tiistai

Poissa.

9.5.2018 keskiviikko

Muokkaan tänään logiikkaa edellisestä tehtävästäni ja katselmoin toisen kehittäjän koodia. Meillä on kehitysjonon läpikäyntiä ja sen siistimistä. Näiden jälkeen etsin itselleni uuden tehtävän tehtäväjonosta, koska meneillään olevassa sprintissä ei ole enää tehtäviä.

Sain katselmoinnit tehtyä ja aloin kirjoittaa kehittäjän pyynnöstä yksikkötestejä kyseiseen funktioon perustuen. Sain pääkehittäjältä lisäkoulutusta yksikkötesteihin, sillä en ole vielä monta kertaa niitä tehnyt. Jatkoin tämän jälkeen myös muiden yksikkötestien kirjoittamista funktiolle, jota olen aiemmin tehnyt.

10.5. torstai

Tänään olisi ollut vapaapäivä kaikilla, mutta teen tänään töitä, jotta saan kerrytettyä tunteja. Jatkan yksikkötestien tekemistä. Sain myös pääkehittäjältä listan asioista, joita voisin tehdä tänään, ja toisen ehdotuksen toiselta kehittäjältä. Teen tarvittavat testit loppuun ja otan kehitysjonosta uusia tehtäviä.

Sain tehtyä noin puolet yksikkötesteistä, joissa menikin koko päivä. Ehdin hieman katsoa mitä tulevat tehtävät ovat. Opin paljon uutta yksikkötestien tekemisestä.

11.5. perjantai

Aion jatkaa yksikkötestejä ja aloittaa uutta tehtävää. Yksikkötestejä voi jatkaa aina, kun ei ole tärkeämpää tehtävää sillä hetkellä työn alla. Ensi viikolla viimeistään aloitan ihan uusia tehtäviä, kun sprintti vaihtuu, ja sprintissä on silloin oikeasti tehtäviä tehtäväksi.

Sain vietyä yksikkötestejä eteenpäin ja aloitin tehtävän, jossa tehdään loppukäyttäjälle virheilmoituksia uudesta ominaisuudesta, jonka aikaisemmin integroin projektiin. Tämä vaati myös Ajax-muutoksia sekä jonkin verran back end -muutoksia, jotta virheilmoitus saadaan näkymään oikeaan aikaan.

Viikkoanalyysi

Suurin osa viikosta meni yksikkötestien kertaamiseen ja osaksi uuden opetteluun testien osalta. Viikko oli hyvin hiljainen, ja meillä oli rauhallinen työtahti. Suurin osa tällä viikolla tekemistäni töistä ei ole kirjattuja työtehtäviä, eli niitä ei ole kirjattu mihinkään eikä niistä jää lopuksi kirjallista merkintää.

TDD eli Test Driven Development eli testivetoinen kehitys tarkoittaa lyhykäisyydessään sitä, että yksikkötestit kirjoitetaan osissa ennen varsinaista ohjelmakoodia. Tällä pyritään takaamaan ohjelmiston oikeanlainen toiminta, ja tämä parantaa rajapintojen suunnittelua. Meillä kuitenkin yksikkötestit useimmiten kirjoitetaan vasta ohjelmakoodin jälkeen, ja niihin ei yleensä käytetä paljoakaan aikaa, varsinkaan, jos kehitysjonossa on jotakin kiireellistä. Testivetoinen kehitys on kuitenkin usein osa ketteriä menetelmiä, joita meillä yritetään parhaamme mukaan toteuttaa.

Yksikkötestaus on siis ohjelmakoodin yksittäisten osien testaamista. Usein yksikkötestauksessa testataan ohjelman eri ominaisuuksia. Meillä käytetään testaustyökaluna Junitia, joka on tunnetuin testaustyökalu Javalle. Jokaiselle ohjelmointikielelle on olemassa omat yksikkötestaustyökalunsa.

Muita hyötyjä oikeanlaisen suunnittelun lisäksi yksikkötestauksessa on ylläpidon helpottaminen. Testit ilmoittavat jonkin muutoksen aiheuttamasta viasta jokaisessa toiminnallisuudessa. Yksikkötestissä pääpiirteittäin luodaan tiettyyn ominaisuuteen toiminnallisuus, joka läpäisee testin. Testissä on tarkoitus testata vain niitä toiminnallisuuksia, joita käsitellään testattavassa osiossa. (CS Helsinki 2018, viitattu 16.5.2018)

5 POHDINTA

Olen oppinut paljon uutta tämän tarkastelujakson aikana ja kehittynyt monella osa-alueella. Päiväkirjan kirjoittaminen on edistänyt oppimista, sillä sen avulla olen voinut syventää pohdintojani tekemistäni työtehtävistä. Lisäksi päiväkirjan avulla on helppo pitää muistissa tehdyt työtehtävät, ja seuraavana päivänä ei mene niin paljon aikaa sen muisteluun, mitä on tehnyt ja mitä aikoo tehdä. Aion jatkossakin pitää pienimuotoista päiväkirjaa työtehtävistäni. Päiväkirja auttoi myös huomamaan, että olen tehnyt enemmän asioita päivän aikana kuin itse huomaankaan. Voisin jopa sanoa, että tällaisen päiväkirjan kirjoittaminen laskee minun tapauksessani stressiä, sillä asiat on kirjoitettu ylös eivätkä ne ole ainoastaan sekaisin pääni sisällä.

Tarkastelujakson aikana minulla oli monenlaista ongelmaa esimerkiksi uuden frameworkin ja uusien ympäristöjen kanssa. Selvisin monesta ongelmasta tiimiläisten ja muiden työkavereiden avulla sekä selvittämällä asioita paljon itsenäisesti erilaisista lähteistä. Opin, että apua pitää pyytää, ja siihen kannustettiin. Avunpyynnön kynnykseni madaltui, ja huomasin, että tiimiläisten kanssa on luontevaa pohtia yhdessä ja keskustella meneillään olevista ongelmista. Myös minulta kysytään rohkeammin apua ja kysymyksiä kehittämäämme sovellusta koskien.

Olen oppinut enemmän organisaation sisäisistä toimintatavoista jakson aikana. Päiväkirja on myös kannustanut toimintatapojen ja sidosryhmien pohtimiseen, minkä avulla olen ymmärtänyt monia asioita paremmin, kuten oman sijoittumiseni organisaatiossa. Olen myös puhunut paljon näistä asioista tiimiläisten, kuten meidän Scrum Masterimme kanssa, joka on mielellään kertonut kaiken, minkä tietää. Olen myös löytänyt uusia sisäisiä prosesseja, joiden olemassaolosta minulla ei ollut aikaisemmin tietoa. Lisäksi olen saanut olla mukana parantamassa niitä.

Olen joiltakin osin kehittynyt myös ohjelmointitaidoiltani. Reactin olen opetellut kokonaan tämän työjakson myötä, ja Reactiin liittyvät lisäosat ovat tulleet tutuiksi. Olen kerrannut Javaa ja Ajaxia ja syventänyt taitojani niiden osalta. Opin myös paljon uutta yksikkötesteistä, mutta koen, että niissä on vielä työsarkaa. Olen kehittynyt myös debuggauksessa ja ongelmien selvittämisessä jakson aikana kohtalaisen hyvin, sillä olen saanut paljon uusia vinkkejä debuggaukseen ja ongelmien lähestymistavat ovat kehittyneet parempaan suuntaan.

Opinnäytetyön viikkoanalyysiosiot ovat olleet mielestäni todella antoisia. Viikon päätteeksi on tullut pohdittua syvällisemmin useaa tärkeää asiaa ja, kun nämä asiat on kirjoittanut ylös, on todella huomannut oman kehityksen. Lisäksi viikkoanalyysien avulla olen huomannut monta tärkeää seikkaa työhöni liittyen ja oppinut jopa itsestäni uutta. Olen voinut antaa työlleni enemmän, koska olen oikeasti pohtinut sen hetkisiä ongelmia ja varannut aikaa pohtimiseen.

Muita analysoimiani asioita ovat esimerkiksi saavutettavuus ja älyllisen pääoman merkitys. Nämä ovat molemmat tärkeitä asioita, joita tuskin olisin pohtinut näin syvällisesti ilman tätä opinnäytetyötä. Saavutettavuus nousi esille koulutuksen myötä, joka järjestettiin halukkaille työpaikallamme. Opinnäytetyötä varten tein muistiinpanoja koulutuksesta, mikä taas johti siihen, että asiat jäivät paremmin mieleen. Älyllisen pääoman merkitys taas ei ollut missään kohtaa työpaikalla esillä, mutta sen pohtiminen antoi kannusteen jälkeen päin keskustella aiheesta tiimin ja työkavereiden kanssa.

Kaiken kaikkiaan opinnäytetyöprosessi on antanut enemmän kuin ottanut. Olen kehittynyt monella osa-alueella ja aion jatkaa kehitystä. Jopa henkistä kehitystä on tapahtunut, vaikka sen huomaa minen vaatiikin enemmän asioiden analysointia. Kuten aiemmin sanoin, tästä oppineena aion jatkaa pienimuotoisen päiväkirjan kirjoittamista ja kehityksen seuraamista. Toivottavasti tulevaisuudessa voin sanoa osaavani full stack -kehityksen takaperin, silmät kiinni ja nenästä kiinni pitäen.

LÄHTEET

Collaborator 2017. What is code Review? Viitattu 2.4.2018, <https://smartbear.com/learn/code-review/what-is-code-review/>.

CS Helsinki 2018. Yksikkötestaus. Viitattu 16.5.2018, <https://www.cs.helsinki.fi/u/avihavai/edutain>

Eclipse 2017. Debugging the Eclipse IDE for Java Developers. Viitattu 18.3.2018, https://www.eclipse.org/community/eclipse_newsletter/2017/june/article1.php.

Finto 2018. YSA - Yleinen suomalainen asiasanasto. Viitattu 24.2.2018, <http://finto.fi/ysa/fi/index/F>.

GitHub Stanko 2018. React-animate-height. Viitattu 6.3.2018, <https://github.com/Stanko/react-animate-height>.

Mäki, V 2017. Älyllisen pääoman mittaaminen ja kehitys ICT-alan yrityksissä. Lappeenrannan yliopisto. Talousjohtaminen. Kauppatieteiden kandidaatintutkielma. Viitattu 28.4.2018, http://www.doria.fi/bitstream/handle/10024/136255/kandidaatintutkielma_vilma_maki.pdf?sequence=2&isAllowed=y.

OpenGrok 2018. OpenGrok. Viitattu 25.3.2018, <http://oracle.github.io/opengrok/>.

React 2018. Animation add-Ons. Viitattu 11.3.2018, <https://reactjs.org/docs/animation.html>.

Refactoring Guru 2018. Clean code. Viitattu 8.4.2018, <https://refactoring.guru/refactoring/what-is-refactoring>.

REST API Tutorial 2018. REST API Quick Tips. Viitattu 15.4.2018, <http://www.restapitutorial.com/lessons/restquicktips.html>.

Segue technologies 2013. What is Ajax and Where is it Used in Technology? Viitattu 6.5.2018, <https://www.seguetech.com/ajax-technology/>.

Spring 2018. Building a RESTful Web Service. Viitattu 22.4.2018, <https://spring.io/guides/gs/rest-service/>.

Vogella 2017. Unit Testing with JUnit – Tutorial. Viitattu 18.3.2018, <http://www.vogella.com/tutorials/JUnit/article.html>.

Wikipedia 2018a. Scrum (software development). Viitattu 15.4.2018, https://en.wikipedia.org/wiki/Scrum/software_development).

Wikipedia 2018b. Software bug. Viitattu 6.5.2018, https://en.wikipedia.org/wiki/Software_bug.

Wikipedia 2018c. Splunk. Viitattu 15.4.2018, <https://en.wikipedia.org/wiki/Splunk>.