

Ohjelmistokehittäjän arki suuryrityksessä

Miro Metsänheimo



Tekijä(t) Metsänheimo Miro	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Ohjelmistokehittäjän arki suuryrityksessä	Sivu- ja liite- sivumäärä 99
Opinnäytetyön otsikko englanniksi The everyday life of a software developer in an enterprise	
<p>Opinnäytetyön aiheena on ohjelmistokehittäjän arki suuryrityksessä. Hyvin henkilökohtaisessa työssä analysoidaan omaa oppimista pitkin ajanjaksoa, jonka aikana portfoliomaista päiväkirjaopinnäytetyötä kirjoitettiin. Työn päiväkirjaosuus kirjoitettiin 1.2. – 8.4.2016 välisenä aikana.</p> <p>Ohjelmistokehittäjän työnkuvaan kuuluu päivittäin jatkuva ongelmanratkonta ja uuden kehittäminen. Kehitys tapahtuu ison toimistorakennuksen avokonttorin tiiviissä tunnelmassa, jossa suurin osa tiimin jäsenistäkin työskentelee. Loput kollegoista ovat ympäri Suomea, pisimmillään jopa Portugalissa.</p> <p>Työn päiväkirjaosuus muodostuu tarkasta päivittäisestä tapahtumien kerronnasta. Päivän aikana lukija viedään tapahtumien ja ongelmien keskelle, sekä henkilökohtaiseen kehittymisen analyysiin. Viikoittain edeltävän viikkoon syvennyttään sen aikana muodostuneisiin ongelmiin ja niiden ratkaisuihin. Viikon päätelmissä avataan lukijalle myös edeltävillä viikoilla esille nousseita uusia termejä tai kehittäjän päivittäisessä työssä käyttämiä työkaluja – samalla analysoiden kirjallisuutta henkilökohtaiseen kokemukseen ja kehittymiseen.</p> <p>Seurantaviikkojen aikana työstä loistaa huomattava henkilökohtainen kehittyminen. Seurantaviikkojen alussa kirjoittaja on ollut kuukauden töissä perehtymässä, tätä ennen ainoastaan koulun penkillä. Viimeisillä viikoilla on huomattavissa jo nuoren ammattilaisen piirteitä. Muun muassa kehittämiseen vaadittujen työkalujen hallinta, ohjelmointikielien sisäistäminen ja vuorovaikutustaidot ovat kehittyneet selvästi verrattuna lähtötilanteeseen.</p>	
Asiasanat ohjelmistokehitys, analyysi, ongelmanratkaisu, kehitys	

Author(s) Metsänheimo Miro	
Degree programme Business Information Technology	
Report/thesis title The everyday life of a software developer in an enterprise	Number of pages and appendix pages 99
<p>The thesis focuses on the everyday work tasks of a software developer at an enterprise. In this very personal work, learning is analysed throughout the period. With the help of the portfolio-type diary format, which was written throughout the period between 1 of February 2016 and 8 April 2016.</p> <p>The job description of a software developer includes constant problem solving and the development of new features. The company premises are in a big building's open-plan office, where most of the team is also located. The rest of the colleagues are located around Finland – and some of them even in Portugal.</p> <p>The diary consists of meticulous daily event narration. In daily entries, the author introduces events, problems and personal progress analysis. The preceding week is analysed by the end of every week by elaborating the problems and solutions the developer faced. In the weekly analyses, the reader is also presented with clarification of terms or developer tools that have surfaced – while analysing personal experience and progress using literature.</p> <p>During the period of observation, the reader will notice a significant pace of personal development. At the beginning of the monitoring the writer had just been oriented for a month. Before this, only was the school bench. The last weeks' entries reveal the multi-faceted process of establishing professionalism and the required skillset. Among other things, the control of the tools required for software development, the internalization of programming languages and interaction skills have distinctly developed since the first weeks of maintaining the diary.</p>	
Keywords software development, analysis, problem solving, progress	

Sisällys

1	Johdanto	1
2	Lähtötilanteen kuvaus	3
2.1	Oman nykyisen työn analyysi	3
2.2	Sidosryhmät työpaikalla	5
2.3	Vuorovaikutustaidot työpaikalla	6
3	Päiväkirjaraportointi	7
3.1	Seurantaviikko 5	7
3.2	Seurantaviikko 6	13
3.3	Seurantaviikko 7	20
3.4	Seurantaviikko 8	27
3.5	Seurantaviikko 9	35
3.6	Seurantaviikko 10	46
3.7	Seurantaviikko 11	57
3.8	Seurantaviikko 12	67
3.9	Seurantaviikko 13	76
3.10	Seurantaviikko 14	84
4	Pohdinta ja päätelmät	94
	Lähteet	97

1 Johdanto

Opinnäytetyössä seurataan päiväkirjamaisesti aloittelevan ohjelmistokehittäjän arkea. Opinnäytetyön päiväkirjaosuus kirjoitetaan 01.02.2016 – 08.04.2016 välisenä aikana.

Opinnäytetyö koostuu kolmesta osasta – lähtötilanteesta, jossa kuvaillaan työtehtävät, arvioidaan, millä tasolla osaaminen on työtehtävän osaamisvaatimuksiin ja kerrotaan omasta kehittämisestä. Kuvaillaan myös työpaikan sidosryhmiä ja työn vaatimia vuorovai-
kutustaitoja.

Yritys jossa aloitin työskentelemään vuoden 2016 alussa on yksi Suomen, sekä yksi maailman suurimmista IT-alan toimijoista. Yrityksen asiakasryhmä koostuu maailmanlaajuisesti maiden hallituksista ja suurista teollisista yrityksistä. Suomessa isona asiakasryhmänä ovat myös pk-yritykset. Toimin yrityksessä nuorempana ohjelmistokehittäjänä.

Pääsen työskentelemään erittäin suurien datamäärien ja –järjestelmien kehittämisessä ja ylläpitotoimissa. Ohjelmistokehittäjällä tulee olla hyvä tuntemus erittäin laajasta ohjelmistojärjestelmästä, sen toimintatavoista, käsitteistä ja tietokantarakenteesta. Järjestelmä jonka parissa työskentelen, on pääosin ohjelmoitu Javalla. Seasta löytyy myös vanhaa koodia, joka on tehty C:llä ja uudempaa, johon ollaan käytetty AngularJS:ää. Työssä siis vaaditaan laaja-alaista ohjelmointitaitoa ja taitoa ymmärtää valtavan järjestelmän toimintatapoja. Kollegani suositteli minulle ensimmäisten viikkojen aikana Robert C. Martinin kirjaa Clean Code, joka auttaa lukijaa kehittämään helposti luettavissa olevaa laadukasta koodia. Olen tähän mennessä pitänyt hyvin paljon kyseisestä kirjasta.

Työympäristöni on isossa modernissa toimistorakennuksessa avokonttorissa. Iso osa tiimistäni työskentelee samassa avoimessa tilassa, mutta kuitenkin noin kolmannes tiimin jäsenistä sijaitsee toisilla paikkakunnilla tai jopa toisessa maassa. Teknisesti puhuen, yritys on antanut käyttöni tehokkaan tietokoneen ja useamman näytön, jotta työni olisi mahdollisimman tehokasta. Avoimen työympäristön huomioiden minulle on hankittu myös laadukkaat vastaavaan työhön kehitetyt ääniä suodattavat kuulokkeet. Itse kehitystyö suoritetaan virtuaalikoneessa VirtualBoxin kanssa helpottaakseen työympäristön asentamista ja ylläpitoa.

Keskeiset ammattikäsitteet

Tässä kappaleessa kuvaan työssä keskeisessä osassa esiintyviä ammattikäsitteitä.

Backlog	Lista ominaisuuksista tai teknisistä tehtävistä (<i>tiketeistä</i>), joita tiimi hallitsee ja toteuttaa. (Agile Alliance 2016).
Code review (pull request)	Arkkitehdin ja/tai kollegan koodin oikoluku ja kommentointi.
Daily	Päivittäin aamuisin pidettävä kokoontuminen, jonka aikana tiimin jäsenet kertovat mitä ovat tehneet, mitä tulevat tekemään ja onko jotain ongelmia. (Techtarget 2015).
Käyttötapaus	Kuvaus järjestelmän tekemästä vuorovai- kutuksesta yleensä käyttäjän tai toisen jär- jestelmän kanssa.
Master	Ohjelmiston juuri, johon tuodaan uudet ominaisuudet ja lopulta viedään tuotantoon.
Merge	Kehittäjän päivittäiseen työhön liittyvä Gitin ominaisuus, jolla yhdistetään yksi ohjelmis- ton haarautuma toiseen.
Retrospektiivi	Aina sprintin jälkeen pidettävä tilaisuus tiimin kesken, jossa mietitään mikä meni hyvin ja mikä huonosti sprintin aikana. Mietitään myös, mihin toimiin ryhdytään kehittääkseen omaa toimintaa.
Scrum	Holistinen lähestymistapa projektinhallintaan ja tuotekehitykseen, jota käytetään projektien hallinnoimisessa pilkkomalla iso kokonaisuus pieniin toteutettaviin <i>taskeihin</i> . (Kniberg 2009).
Sprint	Ajanjakso, jonka aikana toteutetaan tietty määrä tikettejä. Perinteisesti kestoltaan kahdesta neljään viikkoa.
Tiketti (Task)	Yksittäinen tehtävä, osa isompaa kokonaisuutta Scrumissa.

2 Lähtötilanteen kuvaus

Tässä kappaleessa kuvataan lähtötilanne ennen päiväkirjamerkintöjä, työn loppupuolella löytyy kuvaus, jossa peilataan omaa kehittymistä lähtötilanteeseen.

2.1 Oman nykyisen työn analyysi

Aloitin junior full-stack ohjelmistokehittäjänä suurella IT-palveluiden ja –järjestelmien toimittajalla. Vastaan ison järjestelmän kehittämisestä keskisuuren tiivishenkisen tiimin kanssa.

Projektin kehittämiseen yrityksessä käytetään Scrumin ketterän kehityksen viitekehystä. Tämän avuksi yrityksellä on käytössä Atlassianin projektinhallinnointijärjestelmä Jira, jota käytetään kehitystyön hallinnoimiseen, joka toteuttaa Scrumin pääperiaatteet. Järjestelmään pilkotaan backlogiin yksittäin toteutettavissa olevia tikettejä jotka lopulta yhdistyvät laajaksi kokonaisuudeksi.

Päivittäisiin työtehtäviini kuuluu tikettien toteuttaminen järjestelmään, nämä pitävät sisällään yleensä uuden ominaisuuden, mutta satunnaisesti myös ylläpitoon liittyviä tehtäviä, kuten tietokantapäivityksiä tai aikaisempien ominaisuuksien bugikorjauksia tai -muutoksia. Päivittäin tiimin kesken käydään myös daily, jossa tiimin jäsenet kertovat yksitellen mitä ovat tehneet, mitä tulevat tekemään ja ovatko kohdanneet työssään joitain ongelmia.

Tikettiä toteuttaessa järjestelmään minun täytyy suunnitella, mitä olen tekemässä määrittelijän tekemän käyttötapauksen tai -ratkaisun ja tiketin kuvauksen perusteella. Tämän jälkeen suunnittelen, mihin kohtaa järjestelmää minun olisi hyvä toteuttaa tämä ja voiko sen toteuttaa muuttamalla aikaisempia toteutuksia vai pitääkö tehdä kokonaan uusi toteutus. Tästä eteenpäin on toteutustyötä, joka pitää sisällään mm. tietokantahakuja, määrittelyä ja tietenkin itse ohjelmointia.

Viikoittaisiin työtehtäviini kuuluu lähes jokainen maanantai tiimin viikkopalaveri, jossa otetaan videoyhteys tiimin kesken muihin toimipisteisiin ympäri Suomea. Palaverissa käydään läpi viikoittainen työtilanne jokaisella projektilla, tiedotetaan tiimiä ajantasaisista ilmoituksista ja pidetään mahdollisia koulutuksia.

Työn toteuttamiseen tarvitsen taitoja monista eri ohjelmistokielistä ja teknologioista, kuten Java, AngularJS, C ja SQL. Työssä tarvitsen myös ongelmanratkottaitaitoja, sillä ohjelmointi on pääasiassa jatkuvaa ongelmanratkottaa.

Vaikka olenkin ollut vain kuukauden töissä, olen silti oppinut runsaasti muun muassa Javaa ja ohjelmiston yleisarkkitehtuurista, sekä ratkaissut yhdessä kollegoiden kanssa työympäristöön ja järjestelmään liittyviä ongelmia.

Tullessani työpaikalle, ensimmäisellä viikolla kun minulle esiteltiin projektia, jonka kanssa tulen työskentelemään – minulle kerrottiin, että järjestelmä on niin laaja, että tulen ymmärtämään järjestelmän toimintaa vasta puolen vuoden, tai ehkä jopa vasta vuoden jälkeen. Tämän lauseen jälkeen minulta lähti hartioilta iso paino, sillä sain kuulla, että minusta ei oleteta paljoa, vaan minulla on mahdollisuus näyttää kuinka nopeasti opin.

Työtehtäviä ja järjestelmän laajuutta silmällä pitäen olen selvästi vasta aloitteleva toimija, koska olen aloittanut täällä suoraan koulun penkiltä ja olen ollut töissä vain kuukauden – senkin lähes kokonaan vain orientoitumassa. Tarvitsen tällä hetkellä useamman kerran päivässä apua kollegalta jonkin ongelman kanssa tai pyydän mielipidettä ratkaisusta ennen, kun laitan tikettiä reviewiin. Tähän mennessä olen tehnyt toteutuksen kanteen tikitteihin, johon reviewissä annettiin iso määrä kommentteja siitä, kuinka voisin tehdä koodista parempaa tai jos järjestelmässä on ollut yleisiä toimintatapoja joista en ole tiennyt.

Kehittyminen on ikuinen projekti, jota pitää arvostaa. Pidän omaa kehittymistä erittäin tärkeänä. Ajattelutapani virheistä on hieman poikkeava yleisestä – minä pyrin tekemään virheitä, jotta voisin kehittyä paremmaksi. En masennu tekemistäni virheistä, vaan ilahdun – sillä tiedän jälleen kehittyneeni.

Ammatillisessa kehittämisessäni olen vasta alkuvaiheilla, vaikka koulua onkin takana jo lähes seitsemän vuotta. Ohjelmointipuolella olen vielä hieman epävarma omista taidoistani, mutta silti olen ensimmäisen kuukauden aikana onnistunut yllättämään itseni sillä, kuinka paljon ymmärrän ja pystyn jo tekemään.

Kehittymistasoni näkyy selkeästi siinä, kun rupean toteuttamaan jotain ominaisuutta – tarvitsen paljon yksityiskohtaisempia ohjeita kuin ammattilaiset, tarvitsen vähän väliä kollegalta apua tai mielipidettä ratkaisuun ja olen tietenkin paljon hitaampi toteuttamaan, sekä juutun ongelmiin pidemmäksi aikaa. Olen kuitenkin hyvä kysymään kysymyksiä heti kun ne mieleeni ilmaantuu ja sitä kautta olen päässyt nopeasti oppimaan tähän järjestelmään, vaikka vasta alkuvaiheilla ollaan.

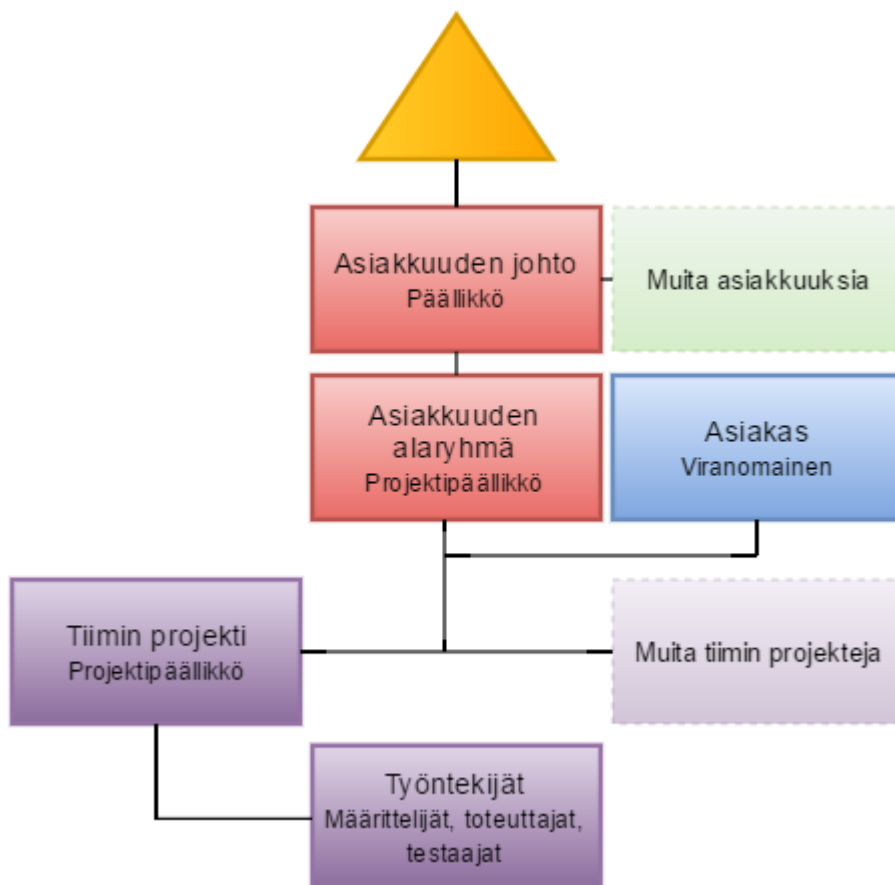
Minun täytyy vielä paljon oppia ja syventyä yrityksen ohjelmistopuolen käytäntöihin, sillä nyt kummassakin kahdessa tikeissä mitä olen toteuttanut, suurimmat virheet löytyvät sii-

tä, etten tiedä tietyntyyppisistä kirjoittamattomista säännöistä liittyen siihen, miten asiat yleisesti hoidetaan.

2.2 Sidosryhmät työpaikalla

Projektimme asiakkaana toimii viranomainen, jolle teemme ison järjestelmän kehittämistä ja ylläpitoa. Yrityksen sisäisiä sidosryhmiä lähimpänä on oman projektin tiimi, joka sisältää määrittelijöitä, toteuttajia, testaajia ja projektipäällikön. Seuraavalla askelmalla on asiakkuuden alaryhmän johto, jonka projektipäällikkö hoitaa monen eri projektin toimintaa. (Kuvio 1).

Asiakkaallamme on meiltä tilattuna useita eri projekteja, joilla on omat eri suuruiset tiimit kehittämässä niitä. Kuvion korkeimmalla tasolla on koko kyseisen viranomaisasiakkaan johtaja, joka hoitaa tämän asiakkaan monia eri isoja kokonaisuuksia. Tästä ylöspäin mentäessä päästään koko talon johtajaan, jonka jälkeen siirrytään pohjoismaiden johtajaan ja lopulta koko yrityksen johtoon. Näin korkealle tasolle kuitenkin päätin olla kuvaamatta.



Kuvio 1: Sidosryhmät työpaikalla

Asiakkaamme toimii valtiorin, eli valtion tieto- ja viestintäteknikkakeskuksen kautta. Valtori on ulkoistanut palvelimensa ylläpidon toiselle yritykselle, jonka palvelimilla kehitystyömme tapahtuu.

2.3 Vuorovaikutustaidot työpaikalla

Vuorovaikutusta tapahtuu työssä päivittäin, vaikka työ pääpiirteittäin tehdään yksin. Joka päivä aamulla klo 9 pidetään tiimin sisäinen daily, jossa käydään noin vartissa läpi mitä tiimin jäsenet ovat tehneet, tulevat tekemään ja onko heillä ongelmia. Dailyssä käydään myös läpi sprintin etenemistä ja Jiran avulla tikettien tilannetta.

Lähes viikoittain tiimimme pitää tiimipalaverin, johon osallistuu kaikki projektin tiimit ja niiden jäsenet ympäri Suomea videoneuvottelussa. Tiimipalaverissa käymme lävitse yleisiä uutisia, tiimin kehittymistä, innovointia, uusia ideoita ja jokaisen projektin tilanteen.

Viikoittain pidetään myös asiakkaan kanssa viikkopalaveri, jossa käydään asiakkaan kanssa läpi projektissa heränneitä kysymyksiä, ongelmia ja kehityksen tilaa. Näihin tapahtumiin en ole päässyt tähän mennessä osallistumaan, ne ovat yleensä projektin johdon ja asiakkaan välillä tapahtuvia palavereja.

Avokonttorissa työskennellessä kynnys mennä kollegalta kysymään kasvotusten tai keskustelemaan asioista on huomattavasti alempana ja sen takia keskustellaankin työn ohessa useita kertoja päivässä. Esimerkiksi kun työn aikana tulee jokin ongelma tai halutaan vain toisen mielipidettä omaan työhön – kysytään tietenkin vieruskaverilta. Avokonttorissa on myös helppo kollegalta kysyä kuulumisia ja tervehtiä.

Työpaikalla ajoittain järjestetään myös ideointitilaisuuksia, joissa mietitään uusia ratkaisuja ja innovaatioita, joita voitaisiin tarjota asiakkaalle. Näissä tilaisuuksissa tunnen olevani vahvoilla, sillä tunnen olevani hyvin innovatiivinen henkilö.

3 Päiväkirjaraportointi

3.1 Seurantaviikko 5

Maanantai 1.2.2016

Päiväkirjan raportointi on mukava aloittaa tällaisella maanantailla, klo 07:40 saavuin Pasi-
lan rautatieasemalle odottamaan junaa, jolla tiimimme lähti yrityksen Turun toimistolle
viettämään tiimipäivää ja samalla juhlistamaan yhden tiimin jäsenen syntymäpäivää. Ju-
nassa meillä oli ohjelmana alustavasti pienessä ryhmässä selata läpi jo viime viikolla esi-
telyä powerpoint -esitystä, johon oltiin lueteltu asiakkaamme tarpeita ja vaatimuksia. Tä-
män esityksen perusteella me lähdimme ideoimaan uusia lähestymistapoja ja ideoita, joita
saatettaisiin tulevaisuudessa esitellä itse asiakkaalle asti.

Kello 09:56 junamme saapui Kupittaan asemalle Turkuun, josta oli muutama sata metriä
matkaa kävellen yrityksen toimistolle, jossa sitten lähdettiin toteuttamaan päivän ohjel-
maa. Aluksi juuri toimeen astuva uusi tiimin johtajamme piti puheen yrityksen rakenteista,
sen muutoksista ja tiimin alkuvuoden tuloksista ja budjeteista.

Seuraavaksi jatkoimme jo junassa aloittamaamme ideointia nimellä ideariihi, joka jatkui
aina lounaaseen saakka ja sen yli. Tämän aikana huomasin itseni kehittyneeni hyvin pal-
jon vuorovaikutus- ja neuvottelutaidoiltani pystymällä tuomaan omia ideoitani hyvin avoi-
mesti ja perustellusti esiin.

Loppupäivästä oli kevyempää toimintaa – tiimimme jäsen, jonka työpiste sijaitsee paikan
päällä Turussa, piti meidän Helsingin porukalle esittelykierroksen toimipisteellensä. Tämä
sijaitsi vanhan tehdasrakennuksen sisällä nimeltä Old Mill. Tässä rakennuksessa ollaan
aikanaan tehty enimmäkseen sähkökeramiikkaa, kuten puhelin- ja suurjännite-eristimiä.
(Old Mill 2016).

Viimeiseksi toinen Turussa työskentelevä kollegani piti meille piiläisille humoristisen Tur-
kuun liittyvän tietovisan, jossa pärjäsinkin hyvin heikosti. Paluumatkalla ei enää työskennelty.

Tiistai 2.2.2016

Päivän tavoitteena minulla on selvittää tulevia työtehtäviä ja saada aikaisemmat toteutuk-
seni reviewin läpi.

Aamulla pääsin jatkamaan viime viikolla vielä hieman kesken jääneitä tikettejä, joihin olin saanut maanantain aikana kommentteja reviewissä, olin epähuomiossa esimerkiksi hakenut toiseen otteeseen tietokannasta samoja tietoja, vaikka olisin saanut haettua ne jo alkuperäisessä haussa samoilla resursseilla.

Hetki ennen tätä keskustelin kollegan kanssa seuraavasta tiketistä minkä ottaisin työn alle. Keskustelin toteutuksesta myös projektipäällikön ja arkkitehdin kanssa listattavista tiedoista ja käytännössä siitä, miten toteutus tehdään, koska en ole ennen tehnyt portaaliiin enkä rajapintaankaan toteutuksia.

Työn alle ottamani toteutus on siis käyttäjäportaaliin tuleva listaus käyttäjän aikaisemmista todistuksista ja reskontralaskuista, jotka haetaan rajapinnasta henkilötunnuksen perusteella. Minun tehtävään kuuluu tämä rajapintatoteutus, kun toinen kollega toteuttaa haettujen tietojen listauksen käyttöliittymään.

Myöhemmin päivällä pidettiin maanantaina loppuneen sprintin retrospektiivi, jossa käydään läpi mikä meni sprintin aikana hyvin ja mikä meni huonosti. Sitten keskustellaan mahdollisista kehityskohteista, joita lähdemme muuttamaan seuraavassa sprintissä. Seuraavan sprintin suunnitelma laaditaan huomenna.

Retrospektiiveissä usein käy niin, että positiivisia asioita ei juuri keksitä ja ihmiset ovat hyvin painottuneita negatiivisiin asioihin. Löysimme parannettavaa esimerkiksi Jiran käytössä. Moni kollega, mukaan lukien minä, ei sitä tiuhaan tahtiin päivittänyt. Itsellä tämä kuitenkin johtui siitä, etten vielä täysin osaa tai tiedä kuinka siellä toimitaan. Retron aikana yksi sprintin hyvistä asioista kollegani mainitsi olevan se, että minut saatiin yllättävän nopeasti perehdytettyä. Otin tämän isona positiivisena palautteena perehdyttäjilleni ja myös itselleni.

Päivän aikana sain molemmat minun tehtäväksi asetetut tiketit reviewin läpi mergettyä masteriin ja sitä mukaan myös testaukseen. Pääsin päivän aikana hyvin tavoitteisiini.

Keskiviikko 3.2.2016

Päivän tavoitteeksi asetin päästä eteenpäin toteutuksissani.

Tänään aamupäivästä jatkoin eilen mietinnän alle laittamaani tikettiä liittyen rajapintahaakuun käyttäjäportaalista. Kysyin dailyn jälkeen kollegalta vierestä myös apua selittämään, miten '.xsd' ja '.wsdl' tyyppiset tiedostot toimivat ja liittyvät Java toteutukseen rajapinnais-

sa. Osasin kuitenkin jo ennestään koulun XML kurssilta '.xsd' tiedostojen luontia, joissa määritellään XML tiedoston rakennetta ja vaatimuksia.

Ennen lounasta pidettiin seuraavan sprintin suunnittelutilaisuus, johon meni aikaa noin puolitoista tuntia. Suunnittelun aikana päätettiin mitä tikettejä tehdään seuraavan kolmen viikon sprintin aikana ja käytiin läpi asiakkaan asettamaa fataljia (1.5.2016) varten muutama tikettikokonaisuus. Näissä kokonaisuuksien määrittelijät esittelivät niihin liittyvät perustiedot, mm. miksi sellainen tehdään ja miten se aiotaan toteuttaa. Näillä toteutuksilla alkaa olla pian kiire, sillä vähintään kuukautta ennen tuota fataljia tuotteen pitäisi olla jo asiakkaalla testissä. Mainitsemani fatalji on hyvin tärkeä, koska sillä päivämäärällä astuvat näillä näkymin voimaan uudet lakimuutokset. Tällöin kaikki uudet toiminnot täytyy olla jo tuotannossa testattu ja valmiina käyttöön.

Lounaan jälkeen jatkoin toteutusta omaan tikettiin ja pääsin siinä yllättävän nopeasti eteenpäin ja laitoin sen myös välireviewiin, jotta saisin tietää, että olenko menossa oikeaan suuntaan toteutuksessa. Sain palautetta esimerkiksi metodien nimeämiseen ja yhteen rakenteelliseen virheeseen, missä olin hakemassa listauksen dokumentin jo listausvaiheessa, vaikka se pitää hakea erillisellä kyselyllä vasta, jos käyttäjä pyytää sitä. Pääsin tämän jälkeen hyvin jatkamaan toteutusta, johon kului sitten loppupäivä.

Päivän aikana opin paljon siitä, miten rajapinnat toimivat Javan kanssa. Opin myös uusia asioita projektin kokonaisuudesta, josta on vielä reilusti opittavaa. Pääsin mielestäni tavoitteisiini ja olen tyytyväinen päivän kulkuun.

Torstai 4.2.2016

Torstaina heti aamusta kysyin muutaman kysymyksen arkkitehdiltä tikettiin tarkentavia tietoja toteutusta varten. Tässä selvisi se, että reskontralasku liittyy todistukseen ja täten näytetään samalla rivillä ja tullaan hakemaan samasta luokasta, mutta hakua ei voida vielä toteuttaa, koska itse dokumentteja ei olla toteutettu.

Toisena asiana aamulla testaaja tuli luokseni kertomaan, että aikaisempi tekemäni eräajo ei toimi testiympäristössä – lähdin heti selvittämään miksi ei. Matkan varrella itselleni tuli ongelmia oman ympäristön kanssa. Tämän sain korjattua, tehtyäni useammalle projektin osalle Maven clean ja -install –komennot. Maven on avoimen lähdekoodin paketinhallintajärjestelmä, joka lataa ja ylläpitää automaattisesti projektin tarvitsemia kirjastoja ja pitää sisällään myös monia muita ominaisuuksia (Apache Maven 2016). Kun oma ympäristöni toimi, huomasin, että tämä eräajo toimii moitteetta siinä. Lopulta selvisi, että testiympäris-

tön yksi konfigurointitiedosto ei ymmärrä päivittää itseänsä masterin mukana, jolloin joudutaan käsin lisätä uusien eräajojen määritelmät.

Tämän jälkeen jatkoin rajapintakyselyn kehitystä ja sain sen siihen pisteeseen, että pyysin uutta reviewiä, josta löytyi vain muutama virhe. Tästä eteenpäin en enää pysty tätä tikettiä kehittämään, ennen kuin itse haettavat todistukset ja reskontralaskut ovat toteutettu. Tästä löytyykin jo erillinen tiketti, jota rupesin työstämään. Suunnittelimme myös muutaman kollegan kanssa, että missä branchissa ruvetaan kehittämään näitä ominaisuuksia, ettei tule liikaa ristiriitaisuuksia – sovittiin, että teen tämän ominaisuuden samaan branchiin aikaisemman kanssa. Tässä minulla meni miettiessä toetutusta loput päivästä.

Perjantai 5.2.2016

Varhaisesta aamusta lähdin jatkamaan uuden tiketin toteutusta. Tämä uusi toteutus on siis juridisen henkilön todistuksen ja reskontralaskun muodostaminen ja tallentaminen palvelimen levyn kulmalle, josta ne voi minun aikaisemmassa toteutuksessa sitten noutaa – tämä on vielä melko epämääräinen ohje ja selkeytyy myöhemmin.

Miettiessä ja koodia selatessa huomasin, että voin mahdollisesti käyttää jo aikaisemmin toteutettua vastaavaa luonnollisen henkilön todistuksen muodostamista kaikilta muilta osin paitsi reskontralaskun muodostamisessa, koska luonnollinen henkilö maksaa heti, mutta juridinen henkilö maksaa laskulla. Dailyssä sovin arkkitehdin kanssa, että käydään myöhemmin päivästä läpi mitä minun kannattaa lähteä ensimmäisenä työstämään.

Aamupäivästä testaaja palautti minulle aikaisemmin tekemäni tiketin, josta osan olin tehnyt parikoodauksena kollegan kanssa perehtyessä järjestelmään. Toteutuksessa on kyse eräajosta, joka ajetaan joka yö. Tarkoituksena on muodostaa tietyin ehdoin asiakirjan palautuskirjeelle kansilehti, joka myöhemmin postitettaisiin manuaalisesti eteenpäin asiakirjan mukana. Tuloste muodostetaan PDF muotoon ja tallennetaan levyille ennalta määritellyyn paikkaan.

PDF –tulosteesta puuttui testauksessa kokonaan yksi osio, vaikka niiden kuuluisi muutamassa kirjeessä ehtojen täytyttyä näkyä. Lähdin ratkaisemaan ongelmaa ja huomasin, että tämä johtuu kahdesta eri syystä – tiedoston muodostusluokassa oli if –lauseessa toinen muuttuja null, jonka korjasin eräajon tietojenkeruuvaiheesta. Toinen virhe oli koodistojärjestelmän XML –tuontitiedostossa, jossa oli väärä nimi koodilla. Tutkiessa löysin myös kolmannen virheen, jota testaaja ei ollut huomannut. Tämän virheen takia lopulliset tiedostot järjestäytyvät väärin kansioihin. Parantelin samalla koodia myös monesta eri kohdasta.

Iltapäivästä kävin keskustelemassa arkkitehdin kanssa ja asioita selvisi sen verran, että pääsin jatkamaan toteutusta. Jouduin kuitenkin ennen tätä tekemään Gitin rebasen masterista, sillä toinen kehittäjä on tehnyt ominaisuuden valmiiksi, jota tulen osin käyttämään toteutuksessani. Tässä kestikin isompi tovi, koska tuodessa muiden tekemiä muutoksia samalle alueelle jouduin yhdistelemään niitä minun omiin muutoksiini – eli korjaamaan konflikteja. Lopulta pääsin lähtemään, kun työpäivää oli kulunut jo yhdeksän ja puoli tuntia. Joku toinen päivä sitten lähdän aikaisemmin.

Yhteenveto viikosta 5

Viikon aikana sain huomata, kuinka tärkeätä testaus on isoissa ohjelmistoprojekteissa – kehittäjällä usein jää jokin asia huomaamatta kehitysvaiheessa ja uuden silmäparin näkökulma ratkaisuun on hyvä olla.

Omia toteutuksia palautettiin minulle takaisin kolme kertaa testauksesta, joista kaksi johtui omista virheistä ja huolimattomuudesta. Opin olemaan paljon varovaisempi ja varmistamaan itsekkin kunnolla sen, että onko kaikki niin kuin pitää.

Testausta kuuluisi tehdä sovelluskehityksen jokaisella vaiheella välttääkseen bugikorjauksia sovelluksen viimeisillä vaiheilla, jolloin korjaukset saattavat vaatia moninkertaisia henkilötyötunteja verrattuna siihen, jos bugi oltaisiin korjattu jo aikaisemmassa vaiheessa. Testauksen tarkoituksena sovelluskehityksessä on parantaa mm. luotettavuutta ja suorituskkyä. (Devi 2012).

Viikon aikana huomasin myös, että olen kehittynyt vuorovaikutustaidoissa. Minulla on ollut nuoruudessani vaikeuksia tuoda omia mielipiteitä esiin puhumattakaan niiden järkevistä perusteluista. Tiimipäivän aikana toin esille hyvin avoimesti omia ideoita, ratkaisuja ja parannusehdotuksia. Yksi näistä ehdotuksista oli järjestelmän loppukäyttäjältä palautteen vastaanottaminen. Nykyään byrokratia tulee vastaan, kun loppukäyttäjän palaute kiertää monen eri henkilön kautta lopulta kehittäjille – palautetta ei tule niin paljoa, kun olisi mielestäni tarve. Uskon myös, että suurin osa palautteesta ei koskaan saavu perille.

Innovatiivisen loppukäyttäjän palautteen vastaanotto tietojärjestelmissä on keskeistä uusien kehitysideoiden hankinnassa. Palautteen keruutavan tulisi rohkaista loppukäyttäjiä tuomaan vapaasti esille omia kehitysideoita. (Bragge, Merisalo-Rantanen, Hallikainen 2005).

Viikon loppupuolella rupesin tutustumaan uuteen järjestelmän alueeseen. Aikaisemmin olin kehittänyt pääsovelluksen, logicin puolella eräajoja. Nyt pääsin kehittämään myös lähemmäksi pintaa – vaikka nämäkin toteutukset vaativat pääsovelluksen kehittämistä, pääsin kuitenkin näkemään, kuinka tätä isompaa kokonaisuutta voidaan käyttää rajapintakyselyiden kohteena ja kuinka näitä saatuja tietoja päästään käyttämään lopulta ihan lopukäyttäjänkin nähtäville.

Selvitystä ja uuden asian omaksumista viikon aikana vaati juuri tämä edellä mainittu rajapintatoteutus, johon en ollut aikaisemmin perehtynyt. Perehtymistä vaaditaan vielä ensiviikollakin, sillä en ole vielä sisäistänyt aihetta kunnolla – enkä edes testannut omaa toteutusta, koska en ole pystynyt vielä ilman muita toteutuksia.

Olen myös kehittynyt Gitin käytössä – en ollut aikaisemmin koskaan käyttänyt esimerkiksi Gitin stash- tai rebase toimintoja. Gitin stash –toiminnolla voidaan tallentaa sivuun omia muutoksia tulevaisuuden varalle nykyisestä toimintaympäristöstä ilman, että niitä muutoksia siirretään eteenpäin tiketille (commit). Gitin rebase –toiminnolla tuodaan palvelimelta uusi pohja omille tekemille muutoksille koodiin, jonka jälkeen yksi commit kerrallaan omat muutokset ajetaan uuden pohjan päälle. Tässä tulee usein konflikteja muiden tekemien muutoksien kanssa, jolloin joudutaan jäsentelemään koodia niin, että molempien tekemät muutokset toimivat virheettää.

Rebasea käytetään esimerkiksi tilanteissa, joissa useampi henkilö ohjelmoi yhtä toteutusta ja toisen ominaisuudet riippuvat toisesta. Jotta toinen voi testata omaa ominaisuutta, on hänen siis tuotava toisen tekemät muutokset omaan ympäristöönsä.

3.2 Seurantaviikko 6

Maanantai 8.2.2016

Maanantain tavoitteeksi asetin päästä eteenpäin viime viikolla aloittamassa rajapintaan liittyvässä tiketissäni ja saada selville, kuinka reskontralaskut toteutetaan.

Lähdin miettimään eteenpäin juridisten henkilöiden todistuksien tulostukseen liittyvää osaa tiketistä, sillä reskontralaskuista on vielä määrittelyn puolelta kysyttävää ja selvitettävää. Huomasin entistä vahvemmin, että todistusten toteuttamiseen ei vaadita kovin suuria muutoksia aikaisempaan luonnollisten henkilöiden todistuksien tulostukseen. Tätä en voinut kuitenkaan varman päälle todeta, ennen kun olen päässyt testaamaan, kuinka toteutus toimii portaalin puolelta.

Dailyssä käydessämme tikettejä läpi rupesimme keskustelemaan, kuinka todistusten reskontralaskut kannattaisi toteuttaa järjestelmään käyttämällä hyväksi jo aikaisemmin toteutettua koodia. Jatkettiin keskustelua dailyn jälkeen, mutta en vielä täysin ymmärtänyt, kuinka minun kuuluisi se toteuttaa.

Dailyn jälkeen lähdin miettimään yhdessä kollegan kanssa, kuinka lähtisin testaamaan rajapintaa ilman, että minulla on portaalista funktiokutsuja valmiina. Järjestelmään ollaan toteuttamassa juridiselle henkilölle myös tunnistautumista, jonka toteutuspuolen otin rebasena omien muutoksieni pohjalle perjantaina. Minun täytyi mahdollista testausta varten asentaa myös työympäristöni valmiiksi uutta juridisen henkilön KATSO -tunnistautumista varten – tämä vaati useiden paikallisen palvelimen kirjastojen ja asetusten päivittämistä, johon kului paljon aikaa.

Iltapäivän alkupuolella pidimme tiimin viikkopalaverin, johon ollaan tiimin kanssa valmisteltu wiki sivu täyteen tekstiä ja kuvia mm. ihmisistä ja tiimin lemmikeistä huomenna tiistaina eläköityvälle projektipäälliköllemme. Tämän esittelyn jälkeen jatkoimme normaalia viikkopalaveria. Palaverissa keskustelimme viime viikon isosta vuosittaisesta asiakastapahtumasta, Python –workshopista, johon kollega oli osallistunut ja toinen kollega oli valmistellut esitelmän nudge –teoriasta ja miten sitä voidaan mahdollisesti soveltaa sovelluskehityksessä ja käyttöliittymän suunnittelussa. Palaverin lopuksi käytiin myös läpi viime viikon tiimipäivän aikana käydyssä ideariihessä syntyneitä ideoita, joita lähdetään työstämään ja lopulta esittelemään asiakkaalle. Ilokseni huomasin, että mukana oli kolme ideaa, jotka

olivat tulleet minulta. Yksi näistä ideoista oli loppukäyttäjän palautteen vastaanotto sähköisesti – tätä ideaa myös muutama kollega kehui palaverin aikana.

Päivän aikana en juuri päässyt tavoitteisiin, sillä en päässyt selvittämään tarkasti, kuinka reskontralasku toteutetaan. En päässyt myöskään vielä testaamaan aikaisempaa toteutusta todistuksista. Päivän aikana kehityin oman työympäristön käytössä ja palaverissa opin uusia mielenkiintoisia asioita esimerkiksi tuosta nudge –teoriasta.

Tiistai 9.2.2016

Tiistain päivän tavoitteeksi asetin yksinkertaisesti päästä eteenpäin tiketissäni. Tavoite jäi näin pieneksi, koska iltapäiväksi oli kalenterissa muita asioita, joista kerron pian. Jätin tavoitteen pieneksi myös sen takia, koska en ollut juuri päässyt siinä eilen eteenpäin.

Varhaisesta aamusta päätin selvittää askel askeleelta mitä aikaisempi toteutus tekee ja missä vaiheessa – aloin vihdoinkin ymmärtää koodista jotain. Päätin lähteä rohkeasti vain toteuttamaan ja kirjoittamaan koodin rivien väliin omaa tekstiä. Päätin myös rohkeasti lähteä muuttamaan aikaisempaa toteutusta huomattavasti, että sen pystyy toteuttamaan järkevämmällä tavalla.

Pikkuhiljaa rupesin debug –tyyliin ohjelmoimaan vanhan päälle uutta ominaisuutta. Eli virhe virheeltä etenin koodin parissa syvemmälle lähtien alkumuutujista, jotka tässä vaiheessa vielä kovakoodasin, eli määrittelin ennalta. Ajoin koodin paikallisella palvelimella jokaisen korjauksen jälkeen ja siirryin seuraavaan paikkaan korjaamaan. Aamupäivästä kollegani liittyi mukaan parikoodaamaan tätä ominaisuutta ja edettiin hyvin ja opin erittäin paljon järjestelmästä.

Lounaan jälkeen meillä alkoi projektipäällikön eläkkeelle siirtymisen juhlistaminen. Tiimin jäseniä oli myös muilta paikkakunnilta matkustanut tekemään työpäivänsä meidän toimipisteeseen tätä varten. Keskustelimme kakun ja kahvin parissa hänen työelämästään, muistojen ja kuvien kera.

Työstämäni tiketti kuuluu isompaan kokonaisuuteen, jota työstää useampi toteuttaja ja määrittelijä samaa aikaa. Pidimme tämän pienen tiimin kesken iltapäivästä palaverin, jossa käytiin läpi ajankohtaisia asioita. Sain muun muassa tietää, että reskontralasku päätettiin jättää toteuttamatta todistuksen noudossa. Tämä helpottaa toteuttamista erittäin paljon ja uskon, että saan pian oman osuuteni valmiiksi.

Pääsin hyvin tavoitteeseeni, sillä etenin toteutuksessa enemmän kuin olin edennyt moneen päivään. Päivästä jäi hyvä fiilis etenkin viimeisen palaverin ansiosta, jossa jälleen kerran yllätyin, kuinka paljon olin itse äänessä ja koska siellä selvisi monta tärkeätä asiaa.

Loppuilta vietettiin ravintola Elitessä hyvässä seurassa ja hyvän ruoan parissa juhlistaa projektipäällikkömme uraa.

Keskiviikko 10.2.2016

Päivän tavoitteeksi aamulla asetin korjata viimeiset virheet tiketistä ja saada se siihen kuntoon, että toteutusta pystytään testaamaan.

Aamupäivästä lähdin selvittämään seuraavaa virhettä ja sen mukaan toteuttamaan tarvittavat korjaukset. Ilokseni totesin korjattuani ensimmäisen virheen, että toteutus lähti saman tien toimimaan. Pääsin tekemään maksutapahtuman, jonka jälkeen pääsin lataamaan luodun todistuksen. Toteutus siis toimii tältä osin juuri niin kuin sen pitää, en lähtenyt tämän pidemmälle sitä vielä testaamaan.

Tämän jälkeen jatkoin viime viikolla kesken jättämäni tikettiä, jonka toteutuksen tarkoituksena oli hakea tietyn luonnollisen- tai juridisen henkilön jokainen aikaisempi todistus tietyltä ajalta ja listata ne käyttöliittymään. Toteutukseen liittyy myös erillinen haku, jolla haetaan käyttäjän valitsema todistus PDF tiedostona ja lähetetään käyttäjän ladattavaksi.

Toteutin siihen tietokantalauseen, jonka yhteydessä täytyi myös lisätä enum –arvoja koodistoihin – näihin tallennetaan arvoja, joita käytetään kaikkialla järjestelmässä, jotta saadaan helposti haettua yhtenäiset arvot kaikkialle. Muutin myös rajapinnan vastaanotettavia tietoja ja vastauksen tietoja uusien tietojen mukaisesti.

Iltapäivästä minulle oli palautunut testauksesta uudemman kerran tuttu eräajototeutus, jossa haetaan asian liitteille palautuskirjettä. Tällä kertaa virhe oli se, että yksi tekstikenttä ei tulostu oikeille tulosteille ehtojen mukaisesti. Huomasin, että olin aikaisemmin lukenut käyttötapauksen kuvauksen väärin ja hakenut yhden liitteen tiedoista sen lajia, vaikka minun kuuluu selvittää, jos edes yksi koko asian kaikista liitteistä sisältää tietyn liitteen lajin. Sain tämän nopeasti korjattua ja samalla sain karsittua koodista yhden tietokantahaun pois, joka nopeuttaa eräajon ajamista.

Päivän päätteeksi käytiin myös verkkopuhelu, jossa kävimme läpi minun tekemiä muutoksia valmistuneisiin tiketteihin ja mietittiin, kuinka tästä jatketaan. Päätimme, että huomen-

na aamulla yhdistän kaksi toteutusta yhdeksi Gitin rebaseella ja jatketaan toteutusta yhdessä branchissa, jotta päästään myös testaamaan rajapintakutsua alusta loppuun saakka.

Pääsin päivän aikana tavoitteisiini ja reilusti niiden yli, sillä en osannut odottaa, että olin sisäistänyt tämän alueen koodista jo näin hyvin. Opin tänään, kuinka luodaan uusia enum-arvoja ja kuinka ne liitetään vastaaviin arvoihin tietokannassa. Kehityin myös tietokannan migraatiotyökalun käytössä, joka automaattisesti pitää tietokantaa ajan tasalla käyttäen toteuttajien luomia .sql tiedostoja.

Torstai 11.2.2016

Torstaina aamulla asetin päivän tavoitteeksi saada tikettien yhdistäminen valmiiksi ja päästä etenemään uudessa tiketissä.

Aamupäivästä lähdin tekemään ensimmäiseksi rebasea masterista omaan branchiin, joka sujui melko ongelmitta. Tämän jälkeen siirryin toisen tiketin branchiin ja otin sille branchille rebasen oman tiketin branchista. Tässä vaiheessa rupesi tulemaan isoja määriä konflikteja, koska masteriinkin oli tullut aika paljon muutoksia samalle alueelle. Sain kuitenkin lopulta rebasen valmiiksi ja muut pääsivät jatkamaan kehitystyötänsä samaan branchiin.

Myöhemmin aamupäivästä lähdin testaamaan, toimiiko minun tekemät toteutukset varmasti ja eri arvojen kanssa. Huomasin, että tietokanta antaa jotain virheitä. Tätä mietin hetken itsekseni ja hetken mietittyäni totesin, että en pääse lainkaan eteenpäin.

Lounaan jälkeen pyysin kollegan apua selvittämään ongelmaa ja lopulta seuraan liittyi toinenkin kollega. Aikaa kului useita tunteja ja lopulta päästiin yhdestä virheestä eteenpäin tämän arkkitehdin neuvoilla. Tästä alkoikin taas debuggaus, jota jatkoin kollegan kanssa päivän päätteeseen saakka. Viimeisillä minuuteilla saatiin vihdoinkin todistus tulostettua ja vielä oikeilla tiedoilla. Kello olikin vierinyt jo neljään ja lähdettiin kotiin päin.

Päivän aikana kehityin hyvin paljon koodin debuggauksessa ja ymmärsin jälleen isompaa osaa toteutuksesta. Tavoitteisiin en juuri päässyt, sillä en suunnitellut viettäväni koko päivää tuijottaen erilaisia virheitä ja selvittäen, mitä nämä tarkoittavat.

Perjantai 12.2.2016

Perjantain päivän tavoitteeksi päätin saada toteutettua eteenpäin eilen yhdistämiäni toteutuksia niin pitkälle, kun voin ja aloittaa myös uutta tikettiä.

Aamupäivästä jatkoin tikettiä ja huomasin, että en juuri pääse kovin pitkälle toteuttamaan sitä. Katsoin eilisen koodin lävitse ja löysin siitä muutamia parannuksen kohteita jotka committasin eteenpäin. Myöhemmin määrittelijä tuli kysymään minulta, kuinka järjestelmään aiotaan toteuttaa todistuksen pyytäjän nimi – päädyttiin yhteisymmärrykseen oikeasta tietokantataulusta ja kävin vielä varmistamassa asian arkkitehdiltä, joka oli myös samaa mieltä.

Pyytäjän nimen toteuttamisessa tuli eteen virhe, joka selvitettyäni johtui siitä, että minulla ei ollut oikeuksia tehdä rakenteellista muutosta tietokantatauluun. Päätin käyttää väliaikaisesti toista yleistä käyttäjätunnusta ja lähettää palvelimen ylläpitäjälle pyynnön oikeuksien lisäämisestä. Sain pyytäjän nimen tallennettua onnistuneesti ja koodinkin muutettua sen mukaisesti.

Lounaan jälkeen päätin aloittaa uuden tiketin, jonka kuvauksena oli meidän järjestelmän työpöytäversiosta poistaa todistuksen pyytäjän ammatti näkymistä ja tulosteista. En ollut ennen tehnyt työpöytäsovellukseen muutoksia, mutta aloittaessani selaamaan koodia huomasin, että se on toteutettu hyvin helppolukuisella MVC –arkkitehtuurilla. Löysin oikeat tiedostot nopeasti ja lähdin poistamaan kaikkea, mikä liittyy ammattiin ja korjaamaan virheitä, mitkä syntyivät funktiokutsuista, joiden riippuvuutena ammatti –muuttuja oli. Sain yllättävän nopeasti tehtyä tämän toteutuksen. Sain siihen myös reviewistä nopeasti hyväksynnän mergetä masteriin.

Loppupäivästä tuli eteen tilanne, jossa ei ollut enää hommaa jäljellä, joten päätin katsoa sprintin taulusta minulle sopivaa tekemistä jatkoksi. Löysin sieltä tiketin, jossa tehdään muutoksia jo voimassa olevaan eräajoon. Eräajo poistaa vanhoja tallennettuja muita käyttäjiä järjestelmästä ja poistaa niiden liitokset todistuksiin. Eräajoon pitää lisätä pääkäyttäjien vanhojen todistusten poisto levyiltä.

Eräajot olivat minulle jo ennestään tuttuja, joten pääsin nopeasti liikkeelle. Toteutus vaati uuden tietokantahaun ja tietojen jäsentelyn yhdeksi muiden tietojen kanssa. Lopuksi vielä muiden tietojen poiston yhteydessä käytin jo aikaisemmin olemassa olevaa metodia haakekseni tiedostot levyiltä ja niiden poiston. Sain lopulta laitettua tämänkin tiketin reviewiin.

Erittäin tuottoisa päivä tikettien suhteen. Sain yhtä tikettiä eteenpäin ja kaksi valmiiksi reviewiin, joista yksi sai jo hyväksynnän. Opin tänään, kuinka järjestelmän työpöytäsovelluksen koodi toimii ja kuinka käyttämäämme tietokantaan lisätään oikeaoppisesti uusia kenttiä. Pääsin myös hyvin päivän tavoitteisiin ja niiden yli. Hieman jäi häiritsemään se, etten päässyt päätiketissä kovin paljoa eteenpäin, mutta se odottaa myös toisen toteuttajan puolta.

Yhteenveto viikosta 6

Tällä viikolla olen oppinut järjestelmästä paljon uutta – tiedän nyt, kuinka järjestelmän pää rakenne toimii rajapintana ja miten sitä oikeaoppisesti käytetään. Tiedän myös, miten järjestelmän tietokantaan lisätään rakenteellisia muutoksia ilman, että jokin ominaisuus jossain muualla menee rikki.

Kehityin tehokkuudessani ja uskallan paljon paremmin jo tehdä omia päätöksiä koodin suhteen ja osaan hahmottaa, mikä olisi parempi ratkaisu tiettyyn kohtaan koodia – ja uskallan myös korjata sen.

Olin hyvin tyytyväinen, kun viikkopalaverissa tuotiin esille ideariihessä mietittyjä asioita ja huomasin, että usea niistä on minun aloitteesta tullut listalle. Moni kollega kehui ideaani palautejärjestelmästä, joka helpottaisi loppukäyttäjän palautteenantoa kehitettävistä asioista. Ihmeteltiin myös minkä takia tällaista järjestelmää ei ole jo olemassa.

Ongelmia tuli viikon aikana paljon, joista suurin tuli torstaina – ongelman ratkomiseen kului koko päivä. Ongelma ratkaistiin käyttämällä toisen kehittäjän tietoa ja taitoa, sekä Googlea. Javan Hibernate –ohjelmistokehys ei juuri antanut tarkkaa tietoa siitä, mikä on pielessä, koska tämä oli rajapinnan puolella eikä palauttanut virheitä portaalin puolelle SOAP -kutsun läpi. Tiedostossa oli usea tietokantakutsu, joista oli hankala tietää mikä niistä aiheuttaa virheen, koska virhe tulee vasta kun transaktio päättyy palattaessa luokasta. Selvitettiin lopulta Eclipsen debug –työkalulla tarkalleen missä kohtaa virhe tulee. Virhe johtui uuden henkilön lisäyksestä tietokantaan ja yksi tai useampi vaadituista tiedoista on tyhjänä. Hibernate palauttaa rajapinnan lokiin virheessään taulun- ja kentän numeron mistä virhe johtuu. Etsittiin internetistä tietokantahaku, jolla voidaan hakea mihin kenttään nämä numerot viittaavat. Lopulta saatiin selville mihin kenttään täytyisi saada jotain tietoa ja päästiin eteenpäin seuraaviin virheisiin, joissa ei kuitenkaan mennyt yhtä kauan aikaa.

Viikon alussa saatiin mielenkiintoinen uutinen siitä, että tikettiini ei toteutetakaan reskontalaskua, vaan juridisetkin henkilöt maksavat heti verkossa todistuksensa. Tämä päätös

helpotti huomattavasti toteutustaakkaa ja tämän ansiosta päästiin myös näin nopeasti eteenpäin, koska maksutapahtuma on jo toteutettu järjestelmään. Tämä opetti minulle, että asiakkaankin kanssa jo sovitutkin suunnitelmat saattavat muuttua jopa loppuvaiheilla alkuperäisen määrittelyn jälkeen.

3.3 Seurantaviikko 7

Maanantai 15.2.2016

Maanantain tavoitteeksi asetin saada selkeytystä tiketteihin ja niiden tehtäviin – kuka tekee minkäkin ja mitkä tiketit ovat tärkeysjärjestyksessä korkeammalla. Päätin myös päästä eteenpäin perjantaina reviewiin laittamassani tiketissä liittyen eräajoihin.

Varhaisesta aamusta huomasin, että perjantaina reviewiin laittamani tiketti oli saanut runsaasti kommentteja. Olin laittanut toteutukseni reviewiin hätiköiden ja testaamatta koodin ajoa edes paikallisesti. Viikonlopun aikana mietin, että ei olisi kannattanut tehdä näin, vaan olisi kuulunut miettiä pidemmälle eikä hoppuilla reviewiin laiton kanssa. Tämä tietenkin kommentteissa kostautui ja minulle mainittiin, että koodia ei ilmiselvästi ole testattu lainkaan.

Sain toki reviewistä myös muita kommentteja, joiden mukaisesti lähdin ensimmäisillä tunneilla parantelemaan toteutusta. Muutin tekemääni ratkaisua rajulla kädellä ja päätin, etten laita toteutusta eteenpäin ennen, kun pääsen testaamaan sitä. Testauksen voin aloittaa vasta, kun toinen tiketti on mergetty masteriin, sillä tässä tiketissä on ominaisuuksia, jotka riippuvat toisesta toteutuksesta.

Dailyssä keskustelimme toteutuksen kohdalla meidän pienen ryhmän kokonaisuutta – portaalin todistuksen tilausta ja listausta juridiselle henkilölle. Päätettiin jatkaa keskustelua dailyn jälkeen, jossa päätimme mitä kukin lähtee työstämään ja minkälaisessa tärkeysjärjestyksessä. Päätimme, että lähden parikoodaamaan kollegan kanssa yhtä osaa toteutuksesta. Tässä osassa käsittelemme uudesta KATSO –kirjautumisesta tulevia tietoja ja niiden mukaan luomme session ja haemme lisää tietoja käyttäjistä.

Eteemme tuli ongelma, joka oli jo aikaisemmin tiedossa. Jostain syystä KATSO –kirjautuminen ei palauta yrityksen y-tunnusta eikä –nimeä portaaliimme. Tästä syystä emme pysty luomaan onnistuneesti sessiota tai hakemaan yrityksestä lisää tietoja omasta tietokannasta. Lähdimme pikkuhiljaa selvittämään, mistä tämä ongelma johtuu. Selvitimme asiaa koko loppupäivän tuloksetta.

Saimme kuitenkin opittua paljon kyseisestä järjestelmän osasta, joka on hyödyllistä siinä vaiheessa, kun lähdemme tämän virheen metsästyksen jälkeen toteuttamaan muita ominaisuuksia.

Pääsin tänään tavoitteisiin. Tiedän nyt mitä tulen muutaman päivän työstämään ja missä mennään tämän osuuden kanssa. Pääsin myös viimeistelemään perjantaina reviewiin laittamaani tikettiä, jonka sain mielestäni siihen kuntoon, ettei sitä paljoka paremmin pysty enää tekemään. Opin tänään paljon portaalien puolel Wicket –toteutuksesta, joka on Javan kylkeen kehitetty ohjelmistokehys, jonka avulla HTML:ään voidaan helposti sulauttaa Javaa. Opin myös erityisesti sen, että en missään tapauksessa laita testaamatonta koodia eteenpäin.

Tiistai 16.2.2016

Asetin päivän tavoitteeksi valita jokin osuus tikettikokonaisuudesta ja kehittää sitä eteenpäin.

Aamupäivästä keskustelin kollegan kanssa siitä, että mistä minun olisi hyvä jatkaa. Päätettiin, että en lähde toteuttamaan liian lähelle hänen osuuttaan. Ehdotin, että lähden jatkamaan todistusten listauksen tekemistä, jossa järjestelmä listaa luonnolliselle ja juridiselle henkilölle hänen aikaisemmin tilaamiaan todistuksia.

Selvitin aluksi, että onko muilla kollegoilla julkistamattomia muutoksia tälle osa-alueelle. Selvisi, että yhdellä oli ja yhdistin sen toteutuksen omaani. Tässä vaiheessa pääsin vihdoin testaamaan minun toissa viikolla tekemää rajapintahakua. Huomasin myös, että olin toteuttanut kyseisen haun tukemaan vain henkilötunnuksen perusteella hakua, vaikka sen kuuluu tukea myös y-tunnusta. Huomasin myös monia muita parannuksia tässä muutama viikon vanhassa koodissa – yllätyin jälleen, kuinka paljon ja nopeasti olen oppinut.

Saatuani lounaan jälkeen rajapintahaun mieleiseksi, lähdin toteuttamaan käyttöliittymän puolella saatujen tietojen käsittelyä – itse listausta. Nykyinen työympäristöni ei tue rajapinnan pintapuolella Eclipsen debug –työkalua. Debug –työkalu toimii vain itse rajapinnan puolella. Päätin lähteä selvittämään, kuinka saisin sen toimimaan myös toisen palvelimen kautta. Lopulta päädyin rikkomaan toisen paikallisista palvelimista käyttökelttomaksi – syytä tälle ei koskaan selvinnyt.

Loppupäivä minulla meni korjatessa työympäristöäni, yritin montaa eri tapaa kollegan kanssa. Lopulta saimme ympäristön toimimaan kopioimalla verkon yli hänen omat konfiguraatitiedostot itselleni. Tämän jälkeen jatkoin debug –työkalun selvittämistä, kuitenkin hieman varovaisemmin. Onnistuinkin tämän jälkeen nopeasti selvittämään, kuinka saan sen toimimaan – lopulta se toimi paremmin, kun osasin odottaa. Pystyin nyt seuraamaan palvelimen ajoa rivi riviltä, myös rajapintakutsun läpi toiselle palvelimelle ja takaisin.

Loppupäivään ei jäänyt paljoa aikaa, mutta ehdin kuitenkin varmistamaan, että oma rajapintakutsu ja toteutus toimivat juuri niin kuin pitää.

Päivän aikana pääsin hyvin eteenpäin jatkamalla muutama viikko sitten aloittamaani kokonaisuutta. Pääsin hyvin tavoitteisiini. Opin tänään Apachen Tomcat –palvelimien konfiguraatiota ja debug –työkalun käyttöä.

Keskiviikko 17.2.2016

Tälle päivälle oli vaikeaa keksiä tavoitteita, koska kaikki oli aamulla vähän epäselvää.

Tänään meidän pääkehityspalvelin, eli palvelin numero kolme siirtyi uuden katon alle ja oli siitä syystä tavoittamattomissa. Tämä aiheutti päivän aikana useita ongelmia, koska esimerkiksi koodistot sijaitsevat tällä palvelimella.

Minulla oli päivän alussa sprint –taulun toteutuspinossa neljä tikettiä, joista yksi oli eilen uutena tullut muutos palautuskirjeiden eräajoon. Tämän eräajon toteutin muutama viikko sitten. Tässä tiketissä siihen tulee pieni muutos, joka korjaa tulosteiden kansioihin järjestäytymistä.

Dailyn aikana päätimme, että minun täytyy nyt muutama tiketti siirtää testaukseen omasta toteutuspinosta. Kyseiset tiketit on edellä mainittu eräajo ja toinen eräajo, jonka sain maanantaina mieleiseeni kuntoon. Tämä vaati vielä muutaman muutoksen ja paikallisen testauksen, ennen kun siirrän sitä eteenpäin. En voinut aikaisemmin testata tätä eräajoa, sillä se riippui toisen tiketin koodistomuutoksista, mutta päätin nyt toteuttaa sen tähänkin tickettiin.

Saatuani molemmat eräajot reviewin kautta testaukseen, lähdin kollegan kanssa selvittämään, kuinka saataisiin jatkettua portaalin puolen kehitystä. Ongelmana oli se, että olimme aikaisemmin kirjautuneet palveluun kehityspalvelimen kautta, joka on tänään poissa käytöstä. Päädyimme kovakoodaamaan portaaliin tiettyjä arvoja ja kommentoimaan muutama tarkistus pois, jotta pääsemme palveluun sisään jatkamaan kehitystä.

Samalla kun selvitimme kirjautumista palveluun, päätimme tehdä pienen skriptin '.bat' -tiedostoon, joka helpottaa meidän kehitystä. Tällä hetkellä joudumme jokaisen pienenkin muutoksen jälkeen tekemään Mavenin install –komennon ja siirtämään sen luoman .war –paketin Tomcatin deploy –kansioon. Täältä Tomcat ymmärtää itsestään hakea uuden päi-

vitetyn paketin käyttöön ja lopulta selaimeen. Meidän tekemässä skriptissä komentorivillä nyt riittää vain yksi komento tähän kaikkeen, eikä tarvitse enää siirrellä tiedostoja kansioiden välillä itse. Tässä säästää noin 30 sekuntia jokaisen koodimuutoksen kohdalla, joita tehdään päivässä useita kymmeniä kertoja.

Iltapäivästä laitoin myös oman kehitysympäristön siihen kuntoon, että pääsen sinne kirjautumaan ja keskustelin Turussa sijaitsevan kollegan kanssa seuraavasta toteutuksestani. Tämä toteutus on portaalin käyttöliittymän puolen listauksen jatkamista. Häneltä tuli paljon ohjeita siihen, kuinka tämä listaus kuuluu tehdä. Opin myös paljon siitä, kuinka Wicket toimii. Näiden ohjeiden kanssa pääsin pitkälle edistämään listausta ja uskon, että se saadaan pian valmiiksi.

Tavoitetta tänään en keksinyt itselleni aamulla, mutta sain kuitenkin päivän aikana enemmän aikaiseksi, kun ajattelin saavani. Opin tänään jälleen lisää työympäristöstä ja etenkin Wicket –ohjelmistokehyksestä, jolla portaalin käyttöliittymä on ohjelmoitu.

Torstai 18.2.2016

Päivän tavoitteena päätin saada todistusten listaus näkymään ja ehkä aloittelemaan todistusten latauslinkkien toteutusta.

Aamulla saatuani ympäristön kuntoon yllätyin iloisesti, sillä listaus oli yhtäkkiä alkanut toimia ja näin yhden todistuksen käyttäjällä, jolla olin kirjautunut. Ajattelin, että virheitä taisi aiheuttaa eilinen kehityspalvelimen käyttökatko.

Nyt kun listaus toimii, lähdin miettimään, kuinka saisin linkit todistusten lataukseen toimimaan. Todistusten listaus on toteutettu wicketillä niin, että ensin määritellään jokainen kenttä ja sen käyttämät luokka, nimi ja luokasta tulevat muuttujat. Tämän jälkeen tuodaan taulukkoon tiedot kerralla luokasta. Käyttötapauksessa on määritelty, että ensin haetaan tietokannasta jokainen todistus. Vasta siinä vaiheessa, kun käyttäjä painaa todistuksen latauslinkkiä, järjestelmä hakee todistuksen tietokannasta ja lähettää sen käyttäjälle.

Näiden ominaisuuksien rajapinnan puolen toteutukset olin jo tehnyt aikaisemmin valmiiksi, jäljellä on enää vain linkki. Sain tämän toteutettua siihen pisteeseen asti, että saan linkkiä painamalla haettua todistuksen tietokannasta todistuksen avaimella ja asetettua sen muuttujaan. Päivän ongelmaksi syntyi todistuksen lähettäminen tästä muuttujasta käyttäjän selaimelle.

Taulukon määrittelyssä linkin kohdalla on Wicketin oma onClick() –tapahtuma kumottu. Tämä tapahtuma määritellään itse uudelleen. En ihan täysin vielä ymmärrä Wickettiä, eikä myöskään kollegani. Vaikuttaa siltä, että se on melko yleisellä tasolla vihattu ohjelmistokehys, koska jo moni kollega on avoimesti ilmaissut negatiiviset tunteensa sitä kohtaan. Toteutin tiedoston latauksen monella eri tapaa ja mielestäni jokainen niistä pitäisi toimia, mutta jostain syystä mitään ei tapahdu. Järjestelmä hakee oikean todistuksen, mutta siinä vaiheessa, kun sitä ollaan lähettämässä käyttäjän selaimelle, mitään ei tapahdu.

Tämän miettimisessä minulla meni koko päivä. Pääsin päivän tavoitteisiin, koska toteutukseni lähti aamulla yllättäen toimimaan. Olen kuitenkin pettynyt etenemiseeni tänään. En voinut myöskään kysyä arkkitehdiltä apua, sillä he olivat palaverissa ison osan päivästä – tai sitten juuri silloin luulin pääseväni eteenpäin, kun olisi ollut mahdollista kysyä. Päätin heti huomenna aamusta kysyä apua.

Perjantai 19.2.2016

Perjantain tavoitteeksi asetin saada portaalin listauksen todistusten linkit toimimaan.

Aamupäivästä lähdin jatkamaan eilen kesken jäänyttä mysteeriä. Jostain syystä käyttäjälle ei tule tiedostoa selaimeen ladattavaksi. Ennen dailyä en juuri saanut mitään aikaiseksi tähän liittyen. Eilisen iltapäivän päätöstä en päässyt toteuttamaan aamupäivästä, sillä tämä henkilö joutui olemaan ison osan tästä päivästä palaverissa.

Dailyn jälkeen lähdin kollegan kanssa työstämään hänen osuuttaan, jossa oltiin hieman myös jumissa. Tässä tiketissä toteutetaan portaalin todistuksen tilaus –sivun alkuun valinta käytettävästä henkilöstä. Jos henkilöä ei ole listassa, se luodaan uutena – jolloin käyttäjä täyttää maa- ja paikkakuntatiedot.

Tässä oli ongelmia dropdown –valikon kanssa. Paikkakunta –valikon sisältö haetaan koodistopalvelusta, mutta jostain syystä wickettiin toteutettu hakumetodi ei pystynyt asettamaan valikon vaihtoehtojen avaimiksi sieltä löytyviä avaimia, vaan ne olivat numerojärjestyksessä nollasta ylöspäin. Tämä aiheutti ongelmia asettaessa valikkoon oikean arvon vaihdettaessa yritystä. Ongelma ratkaistiin kiertämällä toteutusta hieman, piilottamalla kentät silloin, kun valitaan jo tiedossa oleva henkilö.

Tänään kun on sprintin viimeinen työpäivä, pidettiin iltapäivällä myös sprintin retrospektiivi. Minut valittiin sihteeriksi, joten työni oli kuunnella tarkkaan ihmisten keskustelua ja kirjata ylös mikä meni hyvin, mikä meni huonosti ja mitä tullaan tekemään toisin tulevaisuu-

nessa. Keskustelun aikana nousi jälleen parannettavaa Jiran käytössä. Päätettiin, että aina kun tiketti siirtyy esimerkiksi toteutuspinosta testauspinoon, tiketille täytyy antaa myös kommentti asiasta. Päätettiin myös mm. laittaa linkit jokaisen tiketin kuvaukseen käyttötapaan, ratkaisukaavioihin, testitapauksiin ja muihin dokumentaatioon, jotta ne löytyisivät helposti.

Iltapäivästä jatkettiin kollegan kanssa hänen toteutusta eteenpäin. Tämän jälkeen vihdoin pääsin kysymään arkkitehdiltä apua oman toteutuksen linkkien toimintaan, kun hän oli palannut palaverista. Hän on valitettavasti myös ainoa, joka tässä asiassa osaisi auttaa. Paljon asioita selvisi ja lähdin muokkaamaan toteutustani sen mukaan. En kuitenkaan vielä linkkejä toimimaan.

Tiistaina esitellään asiakkaalle sprintin aikana valmiiksi saatuja ominaisuuksia, toivottavasti maanantaina saisin toteutukseni siihen kuntoon.

Yhteenveto viikosta 7

Tällä viikolla opin paljon uutta Apachen Wicket –ohjelmistokehyksen käytöstä ja pikkuhiljaa alan itsekin oppia vihaamaan sitä muiden joukossa. Nyt kun olen kolme työpäivää melkein kokonaan tehnyt yksinkertaista taulukkoa, johon listataan käyttäjän aikaisemmin tilaamat todistukset linkkeineen – huomaan, että toisilla teknologioilla tämä listaus olisi tehty alle puolessa päivässä.

Apachen Wicket on yksi suosituimmista Javan verkkosovelluskehysistä. Wicket pohjautuu uudelleenkäytettäviin komponentteihin ja niitä järjestelemällä verkkosovellukseksi. Jokaisella komponentilla on oma Java luokka ja HTML tiedosto. (Netbeans 2016).

Loppuviikosta kolusin Googlen aivan tyhjäksi tiedosta liittyen Wickettiin, sen taulukoihin ja linkkeihin yrittäessäni toteuttaa pdf –linkkien toimintaa. Tätä tutkiessa aloin jo loppupuolella turhautua, sillä ei minkään näin pienen ominaisuuden työstämiseen voi kulua yli kahta työpäivää. Ajattelin myös mitähän asiakas ajattelee, jos tietäisi, että hän maksaa työstä joka ei etene. Sain kuitenkin itseni pidettyä positiivisena ja lukiessani viikon tapahtumia olen saanut kuitenkin jälleen enemmän aikaiseksi, kun edeltävällä viikolla. Kehitystä on siis tälläkin viikolla huomattavissa.

Tämän viikon aikana käytin normaalia enemmän Eclipsen debug –työkalua. Konfiguroin sen myös toimimaan vanhemman Tomcatin kanssa, jota käytämme asiointiportaalin palvelimena. Tämä työkalu on osoittautunut erittäin käteväksi ja ihmettelen suuresti minkä

takia sitä ei ole opetettu lainkaan koulussa, vaikka työelämässä se on läsnä päivittäisessä työskentelyssä.

Debug työkalun avulla voit ajaa ohjelmasi ja samalla seurata lähdekoodista rivi riviltä mitä ohjelmassasi tapahtuu. Työkalun avulla näet myös muuttujien arvot. Koodiin asetetaan breakpointteja, joiden kohdalle ohjelman ajaminen pysähtyy. Näissä kohdissa voi tutkia onko muuttujissa oikeat tiedot sillä hetkellä ja jopa muuttaa niiden sisältöä testataksesi ohjelman toimintaa. Tässä tilassa voit halutessasi siirtyä rivi kerrallaan eteenpäin tai hypätä jonkun metodin sisään seuraamaan mitä siellä tapahtuu. Näin löytyy nopeasti tarkka kohta, missä päin koodia on vika. (Vogel 2013).

Debug työkalun käyttö tuli minulle aivan uutena asiana tullessani töihin. Aikaisemmin olen vain miettinyt, mitä tuo ötökkä -ikonina kantava painike tekee. Ennen olen hoitanut debuggauksen tulostamalla konsoliin muuttujia eri kohdista, joka on erittäin hidas ja kömpelö tapa debugata koodia.

Viikon aikana työtäni on hidastanut hieman se, etten ole päässyt ajoissa pyytämään apua oikealta henkilöltä. Muistan muutaman hetken, jolloin olisin voinut mennä kysymään ohjeita, mutta sillä hetkellä tunsin pääseväni eteenpäin. Tästä lähtien, jos minusta tuntuu siltä, etten saa aikaiseksi – menen heti seuraavan mahdollisuuden tullen kysymään, miten olisi hyvä edetä.

3.4 Seurantaviikko 8

Maanantai 22.2.2016

Maanantaina saavuin töihin myöhässä, koska aamulla oli muita henkilökohtaisia asioita hoidettavana. Päivän tavoitteiksi mietin saada viimein linkki todistusten listauksesta toimimaan.

Aamupäivästä saavuin kesken sprintin suunnittelun ja liityin mukaan suunnittelemaan. Saavuin aika loppuvaiheilla mukaan, mutta ehdin kuitenkin ilmaista oman tilanteen ja työntarpeen. Minulle oltiin suunniteltu vain yksi lisähomma, vaikka sprintti on kestoltaan kolme viikkoa. Tämä lisähomma oli juridisen henkilön todistuksen tilauksen maksutapah-tuma myös suoraan pankista. Totesin, että tämänhän minä olen jo toteuttanut toisessa tiketissä, ja ilmoitin asiasta.

Suunnittelun jälkeen pian käytiinkin jo lounaalla. Lounaan jälkeen lähdin kunnolla mietti-mään linkkien toimivuutta. Hetken mietittyäni onnistuin lataamaan PDF tiedoston, joka onnekseni vielä sisälsi oikean sisällön. Tämän voitonriemun kanssa pääsin etenemään vihdoin tiketilläni, mutta vain törmätäkseen seuraavaan ongelmaan.

Saan nyt todistusten listauksesta ladattua käyttäjän aikaisemmin tilaaman todistuksen. Seuraavaksi ongelmaksi muodostui itse linkin teksti. Linkissä kuuluisi lukea teksti 'Todis-tus', mutta sen sijaan siinä luki '[cell]'. Toisena ongelmana linkki ei myöskään vaikuttanut linkiltä, vaan se näytti ruudulla sinisen alleviivatun linkin sijaan ihan normaalilta tekstiltä.

Näitä ongelmia selvittäessä kollega testaili samalla todistusten listaussivua. Itse olin kehittänyt sivua kirjautuneena luonnollisena henkilönä. Hän yritti kirjautua juridisena henkilönä ja sivu näytti vain virhettä. Tämä ongelma korjattiin yhdessä mm. muuttamalla kovakoo-dattuja kirjautumistietoja. Huomattiin myös, että listaukseen tuodaan väärät henkilön tiedot ennen kovakoodauksia.

Minulla meni loppupäivä selvittäessä linkin tekstin näkymistä. Tämä ongelma olisi ehkä kierrettävissä esimerkiksi Javascriptin avulla, mutta ongelmana on se, että teksti kuuluu olla monella eri kielellä. Wicket tukee monikielisyyttä hyvin ja vaatii erillisiä '.properties' tiedostoja, joka sisältää käytettävät stringit halutuille kielille.

Pääsin päivän tavoitteisiin, joka oli hyvin positiivinen asia. Olin taistellut saman ongelman kanssa paikoillaan jo monta päivää. Syvensin tänään oppejani linkkien toiminnasta Wicketin kanssa.

Tiistai 23.2.2016

Päivän tavoitteeksi päätin päästä eteenpäin todistusten latauslinkkien teksteissä.

Aamupäivästä huomasin, että minulle oli tullut testauksesta takaisin tiketti. Tiketti koski todistuksen pyytäjän ammatin poistoa järjestelmästä työpöytäsovelluksen puolelta. Sovelluksen aiemmat todistukset –listauksessa oli mennyt taulukon otsikot ja kentät sekaisin, kun olin poistanut ammatin kentän taulukosta. Huomasin, että taulukon otsikot on määriteltä erillisessä '.properties' –tiedostossa. Otin sekä suomen, että ruotsinkielisistä tiedoista ammatti –otsikon pois ja toteutus toimi niin kuin pitää. Laitoin tiketin uudelleen testaukseen.

Myöhemmin jatkoin todistusten listauksen linkkien kanssa miettimistä. Kokeilin useita eri ratkaisuja, jotka olivat tulleet mieleeni eilisen päivän jälkeen – tuloksetta. Ennen lounasta kävin kysymässä arkkitehdiltä apua. Hän antoi hyviä ohjeita. Olin yrittänyt luoda Wicketin komponenttia linkin labelista ja itse linkin sisällöstä. Hän suositteli, että teen kokonaisen uuden paneelin näillä sisällöillä ja lisään tämän paneelin taulukkoon.

Pääsin lounaan jälkeen toteutuksessa siihen pisteeseen, että uskoisin toteutuksen olevan vain muutaman virheenkorjauksen päässä valmiista.

Iltapäivästä kollega oli järjestänyt tiimille ideointitilaisuuden, jossa käytiin läpi otsikoittain järjestelmän eri osa-alueita ja keksittiin niihin uusia ideoita. Näitä ideoita tullaan työstämään eteenpäin ja lopulta esittelemään asiakkaalle. Toimin osan tilaisuuden ajasta sihteerinä ja kirjasin ylös puheeksi nousseita asioita. Itse en valitettavasti kovin paljoa osallistunut keskusteluun, koska muut olivat paljon aktiivisempia kuin minä. Käytännössä veivät sanat suustani, ennen kuin kerkesin sanoa ne.

En saanut linkkien tekstejä vielä toimimaan, mutta tavoitteeni täyttyivät, koska pääsin mielestäni eteenpäin toteutuksessa. En kuitenkaan ole täysin tyytyväinen päivän etenemiseen, koska omalta osalta olen melkein samassa pisteessä kuin päivän alussakin.

Keskiviikko 24.2.2016

Keskiviikon tavoitteeksi asetin kunnianhimoisesti saada oma osuuteni kokonaan valmiiksi, koska olin eilen edennyt sen verran pitkälle.

Aamupäivästä lähdin jatkamaan linkin toteutusta. Eilen tekemäni Wicketin paneeli ei toiminut koska olin asettanut komponentin id:n ja labelin väärään paikkaan. Nämä korjattua ni pääsin etenemään ja sain vihdoin toimivan linkin näköisen linkin. Linkkiä painamalla selain avaa PDF –tiedoston uuteen ikkunaan käyttäen omaa sisäänrakennettua PDF lukijaa. Tarkoituksena kuitenkin olisi, että vanhemmatkin selaimet osaavat avata tämän tiedoston, joten minun täytyy asettaa linkkiin attachment header – tämä kertoo selaimelle, että kyseessä on ladattava tiedosto. Hetken mietittyä sain lisättyä tämän oikeaan paikkaan. Tajusin myös, että samaan kohtaan täytyy myös asettaa tiedostolle nimi. Kokosin tiedoston nimen käyttäen todistuksen kohteen nimeä.

Lounaan jälkeen lähdin miettimään toteutuksen viimeistelyä nyt kun toteutus toimii. Selasin kaikki muutetut tiedostot läpi. Korjasin mm. jääneitä TODO –merkintöjä, refaktoroin koodia selkeämmäksi, poistin kovakoodauksia ja kommentoituja rivejä.

Testatessani listaa huomasin, että jos käyttäjällä ei ole lainkaan aikaisemmin tilattuja todistuksia, näytetään vain tyhjä taulukko otsikoineen. Päätin, että laitan sen sijaan näkymään tekstin, joka ilmaisee käyttäjälle, että hänellä ei ole nähtävissä aikaisempia todistuksia. Toteutin tämän lisäämällä Wicketin HTML –tiedostoon uuden kentän ja .java –tiedostoon toteutin pientä logiikkaa, joka selvittää onko taulukkoon tulossa rivejä – jos ei ole, tulostetaan lisätty kenttä ja jos on, niin tulostetaan itse taulukko.

Myöhemmin iltapäivällä ilmoitin arkkitehdille, että minulta on loppumassa hommat kesken pian. Huomasin myös, että minulle oltiin palautettu testauksesta tekemäni toteutus, joka lisää eräajoon ominaisuuden. Tekemäni toteutus poistaa vanhoja tulosteita tietokannasta ja levyiltä. Testaaja oli kommentoinut, että hänen seuraamaansa tiedostoa eräajo ei ollut poistanut.

Hetken selviteltyäni päätin lähteä paikallisesti testaamaan toteutusta omalla koneellani. Juuri sopivasti oltiinkin saatu todistusten tilaus toteutettua, joten päätin sitä kautta tehdä uuden todistuksen ja ottaa eräajosta päivämäärärajaukset pois, jotta se poistaa jokaisen tiedoston, riippumatta tulostuspäivämäärästä.

Todistusta tilatessa törmäsin kummalliseen virheeseen, jota debug –työkalulla seurasin syvälle koodiin – paikkaan, johon ei olla edes koskettu toteutuksessa. Tämä toteutus on myös toiminut aikaisemmin. Koodissa yritetään hakea nykyistä päivämäärää tietokannasta – sieltä haettaessa päivämäärä pystytään pitämään tarkkana ja samana riippumatta eri palvelimista. Tietokantahaku jostakin syystä palauttaa arvon null. Tähän ei koskaan saatu selvitettyä syytä, vaan päätettiin korjata toteutus hakemaan nykyisen päivämäärän Date() –luokasta, joka sisältää nykyisen palvelimen paikallisen päivämäärän alustaessa. Päätin, että tämä ei vaadi synkronoitua aikaa tietokannasta, sillä haettavana oli asioita vuosien tarkkuudella, eikä minuuttien.

Saatuani todistuksen tilattua, ajoin eräajon. Eräajo toimi juuri niin kuin sen pitää. Tästä päätin, että testaaja on todennäköisesti yrittänyt itse luoda testidataa tietokantaan. Itse luomalla on vaikeata saada samat ja oikeat tiedot useaan tauluun niin, että ne ovat samat mitä järjestelmä asettaa. Päätin kokeilla saman asian seuraavaksi myös testiympäristössä, missä sitä aikaisemminkin yritettiin kokeilla – päätin kuitenkin jatkaa vasta huomenna.

Päivän aikana pääsin helposti tavoitteisiin. Sain aikaisempien todistusten listauksen toimimaan. Sain myös siistittyä koodia ja tarkistettua sen läpi niin, että kokonaisuus menee todennäköisesti reviewiin huomenna. Opin päivän aikana paljon uutta Wicketistä ja kuinka sitä käyttäen muokataan näkymiä. Opin myös uuden kirjoittamattoman säännön – oikean tavan hakea nykyinen päivämäärä niin, että se on synkronoitu eri tietokoneilla.

Torstai 25.2.2016

Päivän tavoitteena päätin selvittää mitä tulen tekemään seuraavien viikkojen aikana, koska minun tiketit näyttävät pikkuhiljaa valmiilta.

Aamupäivästä lähdin testaamaan eilen paikallisesti testaamaani eräajoa, joka toimi juuri niin kuin pitää omassa ympäristössäni. Muutaman tunnin selvitettyäni en huomannut mitään vikaa toteutuksessa tai testitiedoissa, mutta silti eräajo ei toiminut testiympäristössä. Testauksessa loin eräajolle poistettavan todistuksen portaalista omalla paikallisella ympäristöllä. Tilaus toimii tätä kautta myös testipalvelimelle, sillä ne käyttävät samaa tietokantaa. En voi vielä tehdä todistuksen tilausta testipalvelimen portaalissa, koska siellä ei ole vielä uusinta versiota. Tilaus luo tietokantaan kaikki oikeat tiedot. Tämän jälkeen minun piti luoda testipalvelimelle tiedosto kansioon johon tietokanta viittaa. Tarkistin vielä, että kaikki tietokantataulujen sisältämät viittaukset toimivat ja viittaavat myös tiedostoon. Silti ajaessani eräajon mitään ei tapahtunut tiedostolle. Päätettiin arkkitehdin kanssa, että tämä johtuu todennäköisimmin ympäristöjen eri käyttöjärjestelmistä. Oma paikallista ympäris-

töä joudun valitettavasti pitämään Windowsin päällä. Testipalvelin kun on Linuxin päällä. Eroja näillä on mm. kansiodien polkujen kauttaviivoissa, Windowsissa ne ovat taka- ja Linuxissa etukenoviivoja. Päätin, että jatkan testausta – tai delegoin sen testaajalle sitten, kun todistuksen pystyy tilaamaan testipalvelimen portaalista.

Huomattiin kollegan kanssa, että portaalin todistusten tilauksessa todistus muodostetaan ennen, kun käyttäjä maksaa sen. Maksun jälkeen todistus näytetään käyttäjälle sillä ehdolla, että se on maksettu. Tästä päättelimme, että jos keskeytämme maksutapahtuman ja menemme katsomaan juuri luomaa todistusten listaustani – siellä kuitenkin näkyy tämä maksamaton todistus käyttäjälle, koska maksulle tarkistusta ei ole toteutettu. Lähdin selvittämään, kuinka saan myös listaukseen tarkistuksen siitä, että todistus on myös maksettu. Selvitys loppui melko lyhyeen, sillä tämän sain hoidettua lisäämällä muutaman ehdon tietokantauseeseen, jolla todistukset haetaan. Löysin samalla yhden TODO –merkinnän tietokantahaun yhteydestä, jonka korjasin myös pois.

Päivän aikana pidettiin pienimuotoinen palaveri kehittäjien kesken keskustellen työnjaosta. Arkkitehti jakoi tikettejä reilulla kädellä sprinttitaululla meille omasta- ja myös määrittämättömien tikettien pinosta. Minulle hän antoi tikettejä liittyen mm. tulosteisiin ja järjestelmän verkkokäyttöliittymään – jota en ole vielä päässyt työstämään tähän mennessä.

Iltapäivästä lähdin selvittämään yhtä uusista tiketeistäni, joka liittyy verkkokäyttöliittymään. Tiketti on osin jo toteutettu, mutta siihen halutaan muutama lisäominaisuus. Tämän kanssa en päässyt juuri muuta tekemään kuin perehtymään, sillä talosta katkesi verkkoyhteydet. Päätin lähteä töistä ajoissa ja pitää samalla pois muutaman ylityötunnin.

Päivän aikana pääsin tavoitteisiini, sillä selvitettiin kehittäjien kesken seuraavien viikkojen tikettejä. Opin tänään hieman, kuinka järjestelmän verkkosovellus toimii AngularJS –sovelluskehityksen kanssa.

Perjantai 26.2.2016

Perjantain päivän tavoitteeksi asetin saada portaalin todistuksen tilaus ja listaus – kokonaisuuden code reviewiin ja sen jälkeen mergettyä masteriin. Oltiin sovittu kollegan kanssa, että hän laittaa tikettikokonaisuuden reviewiin heti, kun saa oman pienen osuutensa siihen vielä lisättyä.

Aamupäivästä lähdin jatkamaan eilen kesken jäänyttä verkkosovellukseen liittyvää tikettiä. Tehtäväksi muodostui modaali –ikkunaan puuttuvan sulje –painikkeen lisääminen ja päi-

vämäärän muotoilua – joka onnistuu erittäin jouhevasti Angularilla. Painikkeen lisäyksessä ongelmaksi koitui muutama puuttuva funktio, jotka kopioin lähes kokonaan muista toteutuksista. Päivämäärän muotoilussa en jostain syystä saa kellonaikaa tulostumaan ikkunalle. En ole vielä varma, että kuuluuko siinä näkyä tarkkaa kellonaikaa vai vain päivämäärä, joten jätin tämän selvittämisen myöhemmäksi.

Kollegani sai ennen lounasta tikettikokonaisuuden code reviewiin, johon hän merkitsi myös minut tarkastajaksi. Löysimme sieltä itse muutama paranneltava kohta ja saatiin nopeasti myös muutamalta muulta kehittäjältä kommentteja. Kommentit koskivat esimerkiksi koodirivejä, jotka voisi korvata paljon lyhyemmällä tai tehokkaammalla koodilla. Löydettiin vielä myös muutama TODO –merkintä koodista. Yksi niistä koski monen henkilön todistusten hakua samaan listaukseen, kun tällä hetkellä haetaan vain yhden. Listaukseen kuuluu hakea monen henkilön todistuksia, sillä yhdellä käyttäjällä saattaa olla monta eri henkilöä järjestelmässä. Toteutin tämän muuttamalla henkilön avaimen listaksi avaimia ja lisäämällä tietokantauseeseen tuen useammalle avaimelle.

Loppupäivästä jatkoin verkkosovelluksen kehittämistä. Tiketissä tarkoituksena on myös rajoittaa saatujen tietokantarivien määrä 500 kappaleeseen, ilmoittaa käyttäjälle viestillä, jos tämä luku ylittyy ja antaa käyttäjälle mahdollisuus rajata tuloksia päivämäärän mukaan. Sain toteutettua rivien rajoituksen ja viestin helposti katsomalla mallia muista toteutuksista. Toteutukseni jäi kesken vielä päivämäärän rajauksen kanssa, sillä siinä täytyy tehdä muutoksia myös rajapinnan taakse.

Viimeisillä minuuteilla saimme kollegan kanssa portaalin tikettikokonaisuuden siihen kuntoon, että mergesimme se masteriin. Siirrettiin samalla iso kasa tikettejä testauspinon. Ensi viikolla pääsen myös testaamaan muutama päivä sitten kesken jääneen eräajon loppuun, sillä viikonlopun aikana masterin versio ajetaan testipalvelimelle.

Pääsin tänään juuri tavoitteisiini, sillä saatiin kollegan kanssa mergettyä kokonaisuus ja siirrettyä tiketit testauspinon. Kerkesin myös siinä sivussa tekemään toista tikettiä ja opin paljon uutta Angularista.

Yhteenveto viikosta 8

Viikon aikana pääsin jokainen päivä tavoitteisiini. Olen oppinut rajaamaan päivän työtehtävät riittävän sopiviksi paloiksi, että saan ne hoidettua. Olen myös pääosin ollut tyytyväinen tehokkuuteeni.

Syvensin viikon aikana oppejani Wicketin kanssa, ymmärsin vihdoinkin miksi linkki ei toiminut aluksi. Tämä syy oli se, että Wicketissä täytyy olla tietynlaiset kentät jotka vastaavat toisiaan eri tiedostoissa. Yritin lisätä ison toiminnallisuuden yhteen linkkiin koittamalla tehdä siitä vain pientä komponenttia, vaikka siitä kuului tehdä kokonainen paneeli. Tämä tarkoittaa esimerkiksi sitä, että se vaatii omat '.html' ja '.properties' -tiedostot.

Viikkojen aikana olen törmännyt useaan kehittäjän jättämään merkintään koodissa. Näitä merkintöjä ovat mm. 'TODO' merkinnät, joita kehittäjä jättää taakseen. TODO – merkintöjä jätetään kohtiin, joita kehittäjän mielestä kuuluisi tehdä. Merkintöjä jätetään myös, kun tuntuu siltä, että sitä osuutta ei pysty jostain syystä tekemään sillä hetkellä. Merkintä voi olla myös muistutus poistaa jokin vanhentunut ominaisuus tai merkintä jotain toista kehittäjää varten. (Martin 2009, 58-59).

Olen itse ruvennut käyttämään TODO –merkintöjä selkeyttääkseni omia ajatuksia kehittäessä ominaisuutta. Etenkin tilanteissa, joissa on paljon erilaista tekemistä – jätän taakseen muutaman merkinnän itselleni myöhempää selvitystä tai kehitystä varten. Muutaman kerran olen jopa tehnyt niin, että teen jokaisen funktion valmiiksi ja kirjoitan niiden sisään TODO –kommentit kuvaillen mitä funktion kuuluu tehdä. Tämänlainen lähestymistapa auttaa minua saamaan tietynlaisen perspektiivin ja kokonaiskuvan tulevasta työstä. En myöskään joudu kesken kehitystä miettimään mitä seuraavaksi pitäikään tehdä.

Viikon aikana huomattiin yhdessä kollegan kanssa virhe meidän tekemässä toiminnallisuudessa. Huomattiin, että todistusta tilattaessa järjestelmä oli ennestään ohjelmoitu muodostamaan käyttäjälle todistus odottamaan levyille ennen maksua. Tämä on todennäköisesti tehty sen takia, että käyttäjä ei joutuisi maksun jälkeen odottamaan palvelimen vastausta liian pitkään. Jos käyttäjän pankkimaksun jälkeen sovellus tekee kauheasti aikaa vievää työtä, käyttäjällä saattaa tulla tarve painaa selaimen edellinen –painiketta, joka saattaa sotkea sessiota tai järjestelmä ei rekisteröikään maksua.

Tästä toteutustavasta pääteltiin, että jos meidän juuri lisäämä todistusten listaus ei tarkistamistään, onko todistus vielä maksettu – luotu todistus näkyy käyttäjälle listauksessa, vaikka sitä ei oltaisi maksettu. Tämä olisi voinut helposti livahtaa kenenkään huomaamatta tuotantoon, jota monet käyttäjät olisivat voineet käyttää hyväkseen. Olen hyvin tyytyväinen, että huomasimme tämän näinkin varhaisessa vaiheessa.

Viikon loppupuolella pääsin ensimmäistä kertaa työskentelemään AngularJS:n kanssa. Olin aikaisemmin ottanut yrityksen kustantaman verkkokoulutuksen aiheesta, sillä se on ollut minulle aivan uutta asiaa.

AngularJS on Googlen luoma Javascriptin ohjelmistokehys, joka laajentaa HTML:ää paljon ilmeisemmäksi ja luettavammaksi kieleksi. Angularilla voidaan lisätä toiminnallisuutta HTML –tageihin upottamalla sekaan huomaamattoman laajaa Javascript –toteutusta erittäin yksinkertaisesti. Yksi Angularin tärkeimmistä ominaisuuksista on kaksisuuntainen tiedon sitominen muuttujiin. Tämä tarkoittaa sitä, että yksi muuttuja on aina saman arvoisen eri puolilla koodia. Esimerkiksi voit määrittää HTML:n input –kentän käyttämään tiettyä 'ng-model' –muuttujaa, samalla määrittelet muuttujan tulostumaan tekstikentän alle. Nyt kirjoittamalla tekstikenttään teksti – kirjoittamasi teksti ilmestyy reaaliaikaisesti tekstikentän alle. (Jain, Mangal & Metha 2014).

Yllätyin henkilökohtaisesti siitä, kuinka pienellä määrällä koodia Angularilla pystyy tekemään isojakin asioita. Olen pitkään halunnut oppia ja ehkä kehittää joitain omiakin projekteja keskittyen Angulariin. Nyt töiden puolesta pääsen syventymään hyvin siihen ja tämän jälkeen mahdollisesti myös lähestyn sitä helpommin henkilökohtaisissakin projekteissani. Mielestäni Javascript on tällä hetkellä trendikieli ja siihen liittyvät ohjelmistokehykset ovat hyvin mielenkiintoisia ja ottavat varmasti ison osan tulevaisuuden kehitysprojekteista.

3.5 Seurantaviikko 9

Maanantai 29.2.2016

Maanantain tavoitteeksi asetin oppia enemmän Angularista ja edetä siihen liittyvässä toteutuksessa.

Aamupäivästä lähdin heti miettimään eteenpäin viime viikolla kesken jäänyttä toteutusta järjestelmän verkkosovellukseen liittyvässä tiketissä. Toteutuksesta puuttuu vielä modaali-ikkunan sulje –painikkeen ja todistusten listauksen rivimäärän rajoittamisen jälkeen tulosteiden haku päivämäärärajoituksella.

Toteutin käyttöliittymän puolelle päivämäärille kentät ja päivämäärän valitsimet, AngularJS tukee hienosti myös funktiokutsuja, jotka tapahtuvat kentän muuttuessa. Lisäsin Angularin controlleriin funktion, joka lähettää uuden kutsun rajapintaan uusilla päivämäärillä. Rajapinnan puolelle lisäsin käyttäjän valitsemat päivämäärät tulemaan parametrina. Muutin toteutusta lisäämällä luokan, joka sisältää uudet hakuehdot ja henkilön avaimen. Näillä tiedoilla muutin tietokantahakua hakemaan vain tietyn päivämäärärajoituksen sisällä tulostetut tulosteet.

Aamupäivästä selviteltiin myös minkä takia viikonlopun aikana olevat ajot eivät olleet asentaneet meidän perjantaina tekemää mergeä masteriin. Huomattiin samalla yksi virhe koodissa testaamalla toteutusta vielä kerran paikallisesti, korjasin tämän ja laitoin suoraan eteenpäin. Ajojen epäonnistumiseen löytyi syy, joka korjautui asentamalla uuden version manuaalisesti. Huomattiin kuitenkin myös, että uusi KATSO –kirjautuminen ei toimi testipalvelimella. Tähän löytyi syy siitä, ettei testipalvelimen VPN –yhteyteen olla avattu yhteyttä KATSO –kirjautumisen palvelimelle. Näiden virheiden takia päätin olla vielä tänään testaamatta viime viikolla kesken jäänyttä eräajoa testipalvelimella.

Lounaan jälkeen pidimme tiimin viikkopalaverin – joka alkaa vaikuttaa siltä, ettei se tule enää olemaan jokaisella viikolla tiimin johtajan vaihtumisen johdosta. Palaverissa kävimme läpi jokaisen projektin ajankohtaisia asioita. Keskustelimme myös tiimin johtajan Kiinan lomasta ja kollegan viime viikolla tapahtuneesta perheenlisäyksestä.

Päivän aikana sain tulosteiden listauksen toimimaan. Rupesin kuitenkin miettimään ratkaisuni kriittisesti, sillä jokaisen rajoituksen päivämäärämuunnoksen jälkeen tehdään uusi rajapintakutsu, vaikka voisin hyödyntää jo aiemmin hakemaani dataa jäsentämällä sitä

Javascriptilla ja Angularilla. Esimerkiksi jos käyttäjä avaa listauksen, haluaa asettaa listaukselle alku- ja loppupäivämäärät – järjestelmä tekee kolme rajapintakutsua, mukanaan kolme erillistä tietokantakutsua.

Poistin toteutuksestani kaikki rajapinnan takana olevat muutokset ja lähdin miettimään sitä alusta. Aluksi tarkistetaan, onko kaikki tulosteet jo haettu, jos ei ole niin ne haetaan. Tämän jälkeen, jos päivämääräkentät ovat tyhjiä, asetetaan muuttujien arvoiksi pienin- ja suurin mahdollinen päivämäärä. Seuraavaksi koko tulostetaulukko käydään läpi verraten päivämääriä tulosteen päivämäärään ja palautetaan lopputulos taulukolle. Tällä toteutustavalla tehdään aina vain yksi tietokantahaku, vaikka päivämääriä muutettaisiin useitakin kertoja. Tiedot myös säilyvät, jos vaikka käyttöliittymällä vaihdetaan toiseen välilehteen. Ongelmia tässä toteutuksessa minulle tuli päivämäärien vertailussa ja miten saan ne oikeaan muotoon. En aluksi osannut käyttää Javascriptin Moment ajanhallintakirjastoa, mutta tämä oli lopulta hyvinkin yksinkertainen.

Pääsin päivän aikana tavoitteisiin, ehkä voi jopa sanoa, että useaan kertaan. Onneksi lähdin ajattelemaan alkuperäisen toteutuksen vikoja, siitä olisi varmasti tullut sanomista code reviewissä. Päivän aikana opin paljon Angularin peruseriaatteista ja käytöstä. Opin myös käyttämään Javascriptin Moment –kirjastoa, jolla voidaan helposti hallita ja vertailla aikaa.

Tiistai 1.3.2016

Päivän tavoitteeksi asetin saada verkkosovelluksen tulosteiden rajausta valmiiksi ja testata vanhojen todistusten poisto –eräajon toiminta testiympäristössä.

Aamupäivällä lähdin jatkamaan verkkosovelluksen henkilön tulosteiden listauksen rajauksen toteutusta. Mielestäni eilen jätin sen sellaiseen kuntoon, että se kaipaisi vain viimeistelyä. Saatuaani kollegan kautta yli 500 riviä tulosteita yhdelle henkilölle pääsin testaamaan kunnolla toteutuksen toimintaa. Ongelmia tuli vastaan monesta suunnasta.

Suurimmiksi ongelmiksi muodostuivat jälleen päivämäärien vertailu ja Javascriptin perustoimintaperiaatteet. Javascript toimii eri lailla kuin moni muu ohjelmointikieli. Koodi ajetaan käyttäjän selaimen puolella, sekä se ajetaan kaikki kerralla. Kun selain ajaa Javascriptiä, se ei myöskään jää odottamaan, että jokin funktio palauttaa arvoja, vaan hyppää suoraan seuraavalle riville. Tässä tilanteessa siis Javascript lähettää kutsun rajapintaan hakeakseen tiedot tietokannasta ja hyppää suoraan eteenpäin käsittelemään näitä tietoja, vaikka kyseiset tiedot eivät ole vielä saapuneet. Jouduin erottamaan nämä toiminnallisuudet

kahdeksi eri funktioksi ja kutsumaan tietojen jäsentelyyn tarkoitettua funktiota samasta kohdasta, missä asetetaan tuodut tiedot muuttuun.

Toinen isoimmista ongelmista jatkui päivämäärien vertailun kanssa. Jostain syystä, kun muutin päivämäärärajuksesta arvoa, ohjelma ei käyttäydy niin kuin pitää. Lähdin selvittämään asiaa muuttamalla jokaisen päivämäärän unix epoch aikaan, eli kokonaislukuun siitä, kuinka monta millisekuntia on kulunut päivämäärästä 1.1.1970. Huomasin, että millisekuntimäärä heittää paljon siitä, mitä asetan. Pyysin kollegan myös ihmettelemään asiaa, löydettiin vika lopulta päivämäärämuunnoksesta, joka muuntaa päivämäärän stringistä Moment –objektiksi. Määriteltessä tulevaa dataa olin asettanut kaikki kirjaimet pienillä kirjaimilla, vaikka niiden kuului olla isoja kirjaimia.

Vielä yksi ongelma on mielestäni maininnan arvoinen. Angularissa on kenttiin asetettavia tapahtumien käsittelijöitä. Käytän päivämäärien rajauksessa käsittelijää, joka päivittää muuttujan arvon jokainen kerta, kun input –kenttään tulee 'blur' –tapahtuma. Tämä tapahtuu, kun käyttäjä ottaa kohdistuksen pois kentästä. Huomasin, että tällä hetkellä tämä tapahtuma jostain syystä tapahtuu jokainen kerta, kun päivämäärän valitsimesta vaihtaa kuukautta, vaikka sen kuuluisi tapahtua vasta kun käyttäjä valitsee päivämäärän. Muutin Angularin seuraamaan elementin modelia controllerista käsin. Asetin myös elementille uuden määritelmän, joka päivittää modelin vasta, kun käyttäjä on vaihtanut arvon ja siirtynyt kentästä pois.

Näiden ongelmien ratkomiseen meni suurin osa päivästä. Toteutus ei vielä toimi kaikilta osin ja siinä on hiottavaa. Päivän aikana testasin myös hieman eräajoa, jota en päässyt eilen testaamaan testipalvelimen ollessa vielä epäkunnossa. Tilasin todistuksen portaalin kautta, menin tietokannassa muuttamaan todistuksen tulostuspäivämäärän yli puoli vuotta sitten tulostetuksi. Ajaessani eräajon jostain syystä se ei vieläkään poistanut tätä todistusta levyiltä. Jätin selvitystyön tästä toiselle päivälle.

Päivän aikana en päässyt tavoitteisiini kokonaan. Verkkosovelluksen osuus jäi vielä kesken, mutta pääsin kuitenkin hyvin siinä eteenpäin. Onnistuin myös testaamaan eräajoa, mutta vain huomatakseni, että se ei toimi. Opin tänään paljon uutta päivämäärien käsittelystä, Angularista ja Javascriptistä. Vaikka en päässyt tavoitteisiin, olen silti päivän edistymiseen tyytyväinen.

Keskiviikko 2.3.2016

Keskiviikon tavoitteeksi asetin saada verkkosovelluksen tiketin reviewiin ja aloitella jotain toista tikettiä.

Aamupäivästä lähdin viimeistelemään verkkosovelluksen tulosteiden listausta. Henkilön tulosteita voidaan listata kahdesta eri paikasta, joten kopioin toiseen tiedostoon tekemäni muutokset toiseen paikkaan, johon jouduin kuitenkin tekemään hieman muutoksia tämän jälkeen. Keskustelin tästä tiketistä myös arkkitehdin kanssa ja ehdotin, että rajoitetaan tulosteiden enimmäismäärää entisestään viidestä sadasta kahteen sataan. Tein myös muutoksen päivämäärän rajauksen kenttiin niin, että jos jompikumpi ei ole tyhjä, kentät näytetään käyttäjälle. Tämä ehkäisee tilanteita, joissa päivämääräkentät katoavat listan mukana, kun tulosteiden määrä laskee alle kahden sadan.

Myöhemmin päivästä laitoin tämän tiketin code reviewiin, jossa sain kommentteja muuttujien nimien yksinkertaisuuteen. Nimien kuuluisi olla mahdollisimman yksilöllisiä. Olin myös alustanut muuttujia väärässä paikassa, sekä käyttämäni tekstieditori oli jättänyt whitespacia koodin riveille. Mieleenpainuvin kommentti kuitenkin liittyi toteutuksen toimintatapaan. Arkkitehdin mielestä minun kuuluisi tehdä päivämäärärajaus back-endin puolella – eli rajapinnan puolella, johon olin sen jo aikaisemmin tehnyt, mutta omasta päätöksestä muutin sen hakemaan kaikki tulosteet vain kerran. Tätä kommenttia en vielä totellut, vaan vastasin siihen, että mielestäni tämä toimintatapa on parempi, sillä se rasittaa tietokantaa paljon vähemmän.

Päivän aikana sain hyvin monta pienempää asiaa myös valmiiksi. Jo viikkojen ajan olen taistellut eräajon kanssa, jonka kuuluisi poistaa henkilön yli kuusi kuukautta vanhemmat todistukset tietokannasta ja levyiltä. Tämä eräajo ei jostain syystä tällä viikolla ole minulla toiminut lainkaan testiympäristössä. Sain tähän vihdoon selvitettyä syyn. Olin koko viikon yrittänyt ajaa eräajoa väärässä instanssissa. Seurasin tulosteita testi-instanssissa numero kolme, mutta olin ajanut niitä instanssissa numero yksi. Ajettuani eräajon oikeassa instanssissa niin, että olin myös muodostanut tulosteen testiympäristön portaalin kautta, eräajo toimi mallikkaasti. Laitoin tiketin eteenpäin viimeistelyyn.

Otin tänään myös uuden tiketin työn alle. Tiketin tarkoituksena on korjata bugi järjestelmästä, joka aiheutti tietyn tulosteen laskun eräpäivän asettumaan väärälle päivämäärälle, tässä tapauksessa pyhäpäivälle. Tikettiä ei ole määritelty lainkaan, joten kysyin kollegalta hieman määritystä. Sain paljon enemmän – itse ongelman sijainnin koodissa rivin tarkkuudella. Otin mallia muualta koodista ja toteutin pienimuotoisen funktiokutsun olemassa

olevaan funktioon, joka palauttaa päivämäärästä seuraavan pankkipäivän. Päätin tehdä tälle myös EasyMockin avulla yksikkötestit. EasyMock on Java –kirjasto, jonka avulla luodaan yksikkötestejä testatakseen koodin eheyttä. Tämän tiketin sain saman päivän aikana myös reviewin kautta mergettyä masteriin.

Kerkesin tänään tehdä vielä yhden tiketin lisää. Tiketin tehtävänä oli selvittää, minkä takia tietokantatauluun ei tullut tietyille tulosteille tulosten avaimia. Selvitin, että tulosten avain taulussa on sen vastaava koodiston avain, josta ko. tulosteeseen haetaan lokalisoituja tekstejä. Monilla tauluun tallennettavilla avaimilla ei ollut koodistossa vielä arvoja. Ongelman sain korjattua lisäämällä koodistoon puuttuvilta tulosteilta arvot ja lisäämällä vastaavat avaimet koodin enum –luokkaan. Sain myös tämän tiketin tänään reviewin kautta masteriin.

Tänään oli erittäin tuottoisa päivä. Olen hyvin tyytyväinen omaan toimintaani. Pääsin tavoitteisiini ja moninkertaisesti niiden yli. Sain kaksi tikettiä siirrettyä Scrum –taulun todo –pinosta testaukseen ja yhden tiketin viimeistelyyn. Syvensin oppejani tänään Javascriptin maailmaan ja opin mm. kokoamaan Gruntilla verkkosovelluksemme käyttämiä '.po' –tiedostoja, joissa määritellään sovelluksen lokalisaatio.

Torstai 3.3.2016

Torstain tavoitteeksi asetin saada verkkosovellukseen liittyvän toteutuksen valmiiksi ja ehkä aloittelemaan jotain muuta.

Eilen olin jättänyt kommentin code reviewiin toteutuksen tietojenhakutapaan liittyen. Olin ehdottanut, että minun tekemä toteutus jäisi ennalleen, eikä tietokannasta haettaisi jokaisella päivämäärärajoituksella erikseen uusia tuloksia. Aamupäivästä tarkistin kommentit tiketille. Arkkitehti oli ymmärtänyt mielipiteeni ja sanoi, että tehdään kuitenkin päivämäärärajaus rajapinnan takana, mutta ei lähetetä uutta pyyntöä jokaisella päivämäärän muunnoksella. Ratkaisuna tähän hän sanoi, että lisätään rajaa –painike kenttien viereen.

Lähdin toteuttamaan tätä muutosta – joka ei ollut kuitenkaan vaikea toteuttaa ajatellen, että olin jo kerran aikaisemmin sen käytännössä toteuttanut. Lisäsin uudelleen rajapinnan puolelle uuden luokan, joka sisältää alku- ja loppupäivämäärät. Muokkasin myös tietokantahakua rajaamaan päivämäärien mukaan. Lisäsin myös tällä kertaa ihan tietokantakuun tuloksille rajauksen, joka päätettiin muuttaa kahteen sataan. Arkkitehti huomasi myös, että tietokannasta tulosteiden taulusta puuttuu avaimesta indeksiviittaus päätau-

luun, joten lisäsin tämänkin muutoksen code reviewiin. Toteutus meni myöhemmin päivän aikana masteriin ilman sen kummempia kommentteja.

Päivällä sain yhden portaalin listaukseen liittyvän tiketin takaisin testauksesta, olin unoh-
tanut lisätä siihen ruotsinnokset stringeihin. Meidän tiimissä on yksi henkilö, joka osaa niin
kunnolla ruotsia, että sen uskaltaa laittaa julkiseksi. Kysyin tältä henkilöltä apua käännös-
ten tekemisessä ja sain tämän tiketin takaisin testaukseen.

Tänään aloin selvittää myös yhtä uutta tikettiä, jonka olin saanut toissa päivänä pinooni.
Asiakkaalta on tullut pyyntö lisätä viidelle eri tulosteelle mainosteksti liittyen tuleviin muu-
toksiin henkilön asiointitilillä. Toteutukseen liittyi PDF –luontitiedoston muokkauksen lisäk-
si koodistoon näiden tekstien lisäys käännöksineen. Huomasin toteuttaessa, että vain osa
viidestä tulosteesta on toteutettu hakemaan koodistosta tekstit. Keskustelin tästä tiketin
määrittelijän kanssa, ja hän sanoi, että asiasta täytyy keskustella asiakkaan kanssa. Koo-
distosta haettavat tekstit on helppo toteuttaa tulosteelle, mutta vanhempi tapa on paljon
monimutkaisempi – jolloin asiakas saattaa tulla toiseen tulokseen tulosteiden toteutuksen
suhteen.

Aamulla minun pinoon oli myös palautunut eilen testaukseen laittamani tiketti, jossa selvi-
tin, miksi tulosteiden tauluun ei tule tulosteille avaimia. Testaaja oli huomannut, että tulos-
teilla ei enää muutoksieni jälkeen ollut lainkaan otsikkoa. Testasin tätä myös itse paikalli-
sesti ja totesin saman. Muutaman hetken ihmeteltyäni totesin, että en saa selvitettyä min-
kä takia tulosteiden nimet eivät enää tulostu. Pyysin arkkitehtiä katsomaan asiaa ja syys-
täkin, sillä hän lopulta löysi syvältä vanhasta koodista bugin, joka estää tulosteiden nimien
näkyvän. Bugi löytyi funktiosta, joka selvittää onko tulosteelle asetettu arvoja koodistos-
sa vai ei, tässä tilanteessa oltiin asetettu koodistoon arvo, mutta ei ainuttakaan sisäistä
arvoa eli itse tekstejä. Tässä tapauksessa funktio palauttaa tyhjän stringin, joka aiheutti
tulosteen nimenkin näkyvän tyhjänä. Itse funktiota ei uskallettu lähteä muokkaamaan,
koska siinä tapauksessa täytyisi testata joka ikinen järjestelmän tuloste uudelleen. Löydet-
tiin helppo kiertoratkaisu tälle ongelmalle, joka jäi vielä odottamaan hyväksyntää.

Loppupäivästä rupesin huomaamaan, että minulta alkaa loppua työt kesken. Katselin
Scrum –tauluna, eikä sielläkään ole minulle tekemistä toteutus- tai testauspinoissa. Lähe-
tin sitten kyselyjä projektipäällikölle ja arkkitehdille, jos minulle keksittäisiin vielä hommia –
onhan sprinttiä kuitenkin jäljellä vielä neljä työpäivää.

Lopulta kollega pystyi luopumaan yhdestä tiketistänsä, jonka pariin siirryn huomenna.
Tämäkään ei pitäisi kuitenkaan olla kovin suuri homma. Olen jälleen hyvin tyytyväinen

päivän edistykseen, vaikka koodirivejä ei syntynyt kovin paljoa, silti sain monta tikettiä eteenpäin scrum –taulussa.

Perjantai 4.3.2016

Perjantapäivän tavoitteeksi asetin aloittaa eilen kollegan luovuttaman tiketin ja päästä siinä eteenpäin.

Aamusta lähdin valmistelemaan ympäristöäni eilen mainitsemaani tikettiä varten. Aina vaihtaessa branchia toiseen isompaan kokonaisuuteen minun täytyy kasata jokainen projekti Mavenin avulla saadakseni tuotua mm. päivitettyt koodistot ja uudet generoidut Java – luokat.

Dailyn jälkeen päätimme keskustella pienemmällä porukalla tulosteiden mainostekstiin liittyvästä tiketistä. Aikaisemmin olin huomannut, että kaikkien tulosteiden tekstit eivät tule koodistosta. Kysyimme toiselta kollegalta Turusta apua, jonka jälkeen lopulta keskustelin hänen kanssaan kahdestaan toteutuksesta. Tulosteille täytyy tekstin sekaan tulostaa myös voimaanastumispäivämäärä, joka ei ollut vielä varmaa mikä päivämäärä se on. Tästä kollegani sai idean toteutukseen – päätimme luoda uuden yhteisen teknisen koodin. Tekniset koodit ovat erillään yleisestä koodistosta, niitä voidaan käyttää kaikkialla. Tähän tekniseen koodiin asetamme voimaanastumispäivämäärän. Sitten jokaiselle tulosteelle haetaan tämä päivämäärä ja tarkistetaan jos se on joku muu kuin minimi päivämäärä, eli 01.01.0001. Jos on, niin näytetään tulosteella teksti päivämäärän kera.

Lähdin toteuttamaan tulosteiden mainostekstejä. Toteutus meni melko rutiininomaisesti, sillä olen ollut jo monesti eräajojen luomien tulosteiden parissa työskentelemässä. Ongelmia tuli ominaisuuden testaamisessa. Suurin ongelma tuli yhtä tulostetta tulostaessa. Ennen lounasta eräajo meni virheettä läpi, mutta lounaan jälkeen ajaessani uudelleen muutoksien kanssa – eräajo palautti virheitä. Mikään muu ei ollut muuttunut aikaisempaan ajoon verrattuna paitsi yksi tietokantakenttä, joka ei liity virheeseen millään tavalla. Hibernate valitti, että tulosteen osoite –taulussa on jo vastaavalla ID:llä varustettu kenttä. En saanut ongelmaa itse selvitettyä, vaan jouduin turvautumaan muiden toteuttajien apuun. Apua tuli todella nopeasti, sillä heillä tämä ilmeisesti oli jo aikaisemmin opittu rutiini, mistä en tiennyt. Ajettaessa eräajoa osoitteet –tauluun tallennetaan tulosteiden määränpään osoitteita, mutta jos tietty tuloste on aikaisemmin jo tulostettu – järjestelmä herjaa asiasta eikä lisää tulostetta tauluun. Korjatakseni virheen keräsin lokista jokaisen osoitteen ID:n ja poistin ne tietokannasta.

Sain tulosteiden mainostekstit päivän päätteeksi code reviewiin. Päivän aikana laitoin myös eilen arkkitehdin löytämän bugin code reviewiin – sain siihen muutaman kommentin. Olin huolimattomasti tarkistanut string –muuttujan tyhjän arvon yhtä kuin –merkeillä. Javassa tämä kuuluu tarkistaa erillisellä isEmpty() –funktiolla.

Ilmapäivästä keskustelin testiaan kanssa verkkosovelluksen tulosteiden listauksesta. Testatessaan hän huomasi, että toisessa paikassa päivämäärän rajauskenttiä ei näy. Tämän korjasin muuttamalla jälleen toisen huolimattomuusvirheen – olin kopioinut toisesta tiedostosta omia koodejani, mutta jättänyt yhden muuttujan nimen vääräksi. Tiketissä oli myös toinen mielenkiintoisempi ongelma. Päivämäärän rajauksessa alkukenttään laittaessa saman päivämäärän kuin tulosteen päivämäärä, listauksesta silti poistuu valitun päivämäärän tulokset. Nähdessäni tämän virheen ajattelin heti, että ottaakohan koodi kuluva päivästä kellonajan muuttujaan – olin oikeassa. Tutkiessani koodia huomasin, että rajapinnan takana, kun muunnetaan päivämäärä meidän omasta Pvm –muuttujasta Javan Date –muuttujaan järjestelmällä ei ole tiedossa kellonaikaa, joten se hakee kuluva päivän kellonajan GMT –muodossa muuttujaan, josta syystä valitun päivän tulosteita, jotka on tulostettu ennen tätä kellonaikaa ei näytetä. Korjasin tämän ongelman lisäämällä uuden get –funktion tulosteen ehdot –luokkaan. Funktiolle annetaan arvoksi Pvm –luokka ja se palauttaa Date –luokan, jossa on asetettu kellonajat nolnaan, eli keskiyöhön. Laitoin nämäkin muutokset code reviewiin päivän päätteeksi.

Päivän aikana sain aikaan paljon, mutta en silti päässyt tavoitteisiin. Tein päivän aikana kaikkea muuta, paitsi sitä mitä suunnittelin. Ei tämä kuitenkaan haitannut, sillä uskon, että sain nuo tiketit siihen kuntoon, etteivät ne enää palaa testauksesta takaisin. Opin tänään mm. määrittelemään lokalisaatiota vanhemmille tulosteille, jotka eivät ole koodistossa.

Yhteenveto viikosta 9

Viikon aikana en päässyt jokainen päivä tavoitteisiini, mutta olen silti hyvin tyytyväinen edistymiseeni. Ensi viikko on tämän sprintin viimeinen ja näillä näkymin minulla on vain yksi toteutustiketti viikon ajaksi. Tiketti ei tule todellakaan riittämään viikoksi, mutta projektipäällikkö on ilmaissutkin, että keskitytään pääosin testaamiseen, sillä tämä versioitoimitus on erittäin tärkeä ajatellen tulevaisuutta. Version täytyy olla mahdollisimman virheettömässä kunnossa, sekä siirtymävaihe täytyy olla saumaton.

Tällä viikolla olin paljon tekemisissä Javascriptin ja Angularin kanssa. Opin näistä hyvin paljon lisää, joista isoimpana ehkä Javascriptin toimintaperiaatteista. Javascriptillä kun kehittää, tulee vastaan hyvin mielenkiintoisia ongelmia. Ensimmäisenä kehittäjä huomaa

sen, että sitä on paljon vaikeampi debugata verrattuna esimerkiksi Javaan. Itse ohjelmointialusta on yleensä normaali tekstieditori, sillä isommat alustat, kuten Eclipse eivät pysty ilmoittamaan esimerkiksi syntaksivirheistä. Verrattuna back-endin ohjelmointikieliin, Javascript ajetaan käyttäjän tietokoneella – verkkoselaimessa.

Tästä syystä Javascriptiä pidetään myös tietoturvauehkanä. Esimerkiksi cross-site scripting (XSS) on yksi Javascriptin huonoista puolista. Huonolla tavalla tehdyt verkkosivustot voivat olla haavoittuvaisia tämänkaltaisille hyökkäyksille. Hyökkäykset toteutetaan injektoidalla omaa koodia toiselle käyttäjälle lähetettävälle sivustolle, jonka käyttäjän verkkoselain tämän jälkeen ajaa. (Vogt ym. 2007). Esimerkiksi jos verkkosivuston blogin kommenttikentissä on mahdollistettu HTML:n käyttö, mutta jätetty myös Javascripti mahdolliseksi – kuka vain käyttäjä voi esimerkiksi ladata kommenttikenttään toisesta verkkosivustosta lähtöisin olevan kuvan, joka ladattaessa ajaa haittakoodia käyttäjän selaimessa.

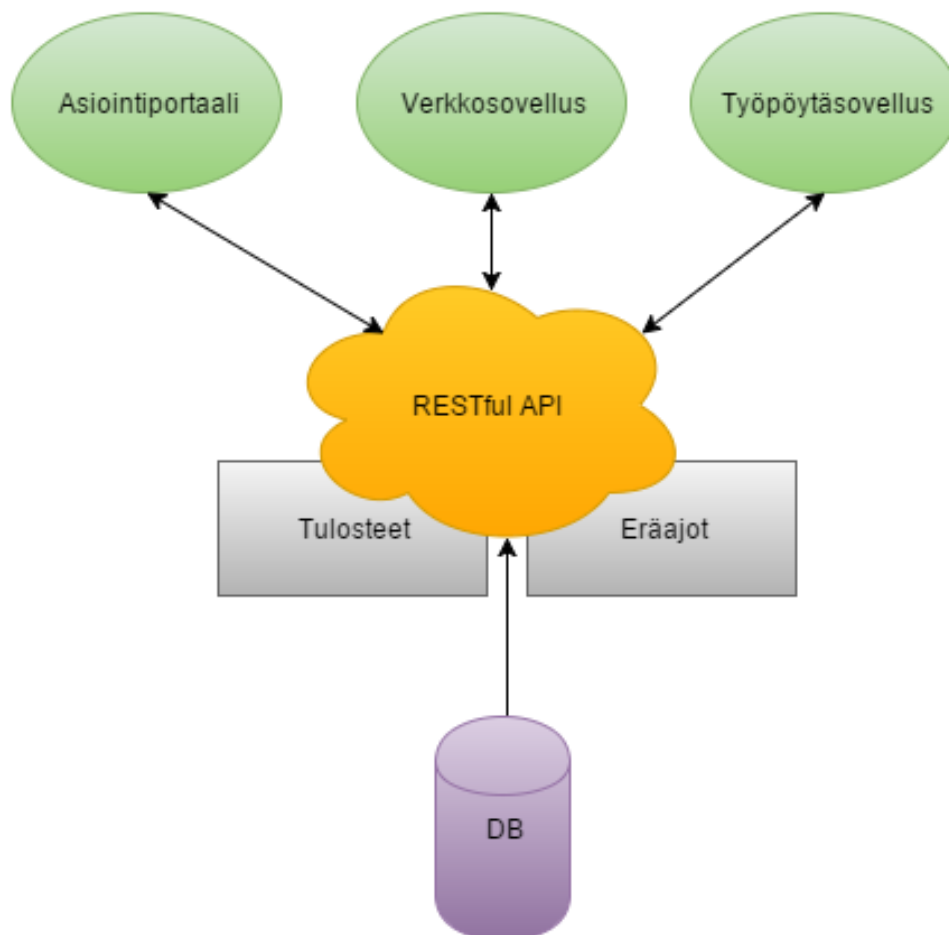
Käyttäjän selaimessa ajettavalla koodilla on myös kehittäjälle omanlaisensa ongelmat esimerkiksi debuggauksen kanssa. Useimmissa moderneissa verkkoselaimissa on sisäänrakennettu tai lisäosana saatavilla debug –työkalu, joka mahdollistaa samantyyllisen koodin debuggauksen, kun Eclipse'n debug –työkalu. Tämä debugkaus kuitenkin täytyy tehdä verkkoselaimessa, eikä palvelimen avulla. Debuggauksen kannalta Javascriptissä on myös huonoa se, että koodin kääntäjä ei anna syntaksista virheitä. Jos Javascriptissä on jokin syntaksivirhe, verkkosivusto latautuu, mutta mikään skripti ei toimi.

Kehittäjälle Javascriptin kanssa työskenteleminen aiheuttaa myös muita ongelmia, mitä back-endin puolella ei tule vastaan. Esimerkiksi Javalla koodia ajettaessa ajaminen menee rivi kerrallaan eteenpäin koodissa tehden kaiken yksitellen. Palvelin myös lakkaa toimimasta ja tulostaa virheen, jos koodissa on jokin pielessä. Javascriptin kanssa tämä toimintaperiaate ei päde. Koodia ajettaessa selain käy läpi jokaisen funktion yksitellen järjestyksessä, mutta ei silti jää odottamaan mitään paluuarvoa, vaan jatkaa heti eteenpäin. Tässä syntyy ongelmia esimerkiksi rajapintakutsujen kanssa, kuten minulla tämän viikon alkupuolella. Javascript kutsui rajapinnan takaa tietoja arvoilla, mutta rajapinnalla kestää muutama sekunti niiden palauttamisessa. Tämän aikana Javascript –koodi onkin keretty jo ajaa loppuun, eikä palautettuja tietoja koskaan käsitelty.

Olen viikkojen aikana oppinut myös aidosti mitä tarkoittaa RESTful rajapinta ja kuinka sellaista kehitetään ja käytetään. Järjestelmässämme kutsutaan rajapintaa useasta eri paikasta, joihin jokaiseen palautetaan eri tietoja. Rajapinnassa ajetaan myös kaikki eräajot ja muodostetaan tulosteet. Järjestelmässämme tiedot haetaan yhteisestä tietokannasta.

Asiointiportaalista rajapintoja käytetään xml ja xsd –tiedostojen määritelmien mukaisesti. Tiedostoissa määritellään ennalta mitä tietoa järjestelmien välillä liikkuu. Tiedostojen perusteella luodaan Java –luokat, joita käytetään tietojen siirtoon rajapinnan yli, sekä käytetään myös kummallakin puolella rajapintaa.

Verkkosovelluksesta kutsutaan suoraan rajapintaa URI:en avulla, joihin upotetaan mukaan avaimia ja annetaan parametreina muita tarvittavia tietoja. Tiedot palautetaan JSON –muodossa käyttöön. (Kuvio 2).



Kuvio 2: RESTful sovelluksen kuvaus

REST on lyhennetty sanasta Representational State Transfer. Se on verkkoarkkitehtuurin malli, joka perustuu erilaisiin URI –pyyntöihin. Esimerkiksi haettaessa tiettyjä tietoja funktion kautta, määritellään ennalta rajapintaan URI –osoite, jota pyytäessä ja annettaessa oikeat parametrit voidaan kutsua tätä funktiota ja palauttaa yleensä JSON –muodossa pyydetty tiedot kutsujalle. RESTissä tehdään GET, POST, DELETE ja PUT –rajapintoja käsittelemään tietoja. Esimerkiksi GET –rajapinta palauttaa tietoja tietokannasta ja POST –rajapinnalla yleensä luodaan tai laajennetaan tietokannan tietoja. Nämä kaksi ovat yleisimmässä käytössä, koska rajapintoja käytetään yleisimmin tietojen hakuun tai niiden lisäämiseen. (Hongjun 2011).

Eniten ongelmia viikon aikana koitui päivämäärien käsittelystä. Päivämäärille on useita eri muuttujia, joita täytyy muuttaa muodosta toiseen. Tällä viikolla olen käyttänyt kolmea – meidän omaa Pvm, Javan util.Date ja Javascriptin Moment.js –luokkia. Ongelmia tuli esimerkiksi päivämäärien vertailuissa toisiin ja päivämäärän muodostamisessa stringistä, esimerkiksi date pickerin antamasta '01.03.2016' –muuttujasta luokkaan. Mielenkiintoisin kaikista oli ongelma, jossa luodessa uutta Date –luokan muuttujaa, päivämäärä tuli oikein menneisyydestä, mutta kellonajaksi tuli kuluvan päivän GMT –aika. Sain kuitenkin ongelman ratkaistua luomalla uuden Calendar –muuttujan, jonka kellonajat asetin nolleen ja palautin siitä muodostetun Date –muuttujan.

3.6 Seurantaviikko 10

Maanantai 7.3.2016

Maanantain päivän tavoitteeksi asetin aloittaa ja edetä kollegalta saamassani tiketissä. Tämä tiketti liittyy asiointiportaalin hakemuksen valintaruudun toiminnallisuuteen.

Aamupäivästä huomasin, että olin saanut viikonlopun aikana kommentteja aikaisemmin mergettyihin tiketteihin. Arkkitehti oli antanut negatiivista palautetta kahteen tikettiin. Toisessa olin kirjoittanut tietokannan taulun nimen väärin, enkä ollut testannut muutosta ennen mergeä – jälleen kerran opin kuinka tärkeätä on testata tiketti ennen, kun se menee eteenpäin. Ehkä tällä kertaa toivottavasti opin sen pysyvästi. Toinen negatiivinen kommentti koski käyttämäni Javan isEmpty() -funktiota. Meillä on oma samankaltainen funktio, jota minun olisi kuulunut käyttää. Hän oli kommentoinut asiasta jo aiemmin ja tästä syystä kommentti oli enemmän negatiivinen. Nämä pitivät pienen masennuksen yllä aamupäivästä, mutta päätin kuitenkin piristyä ja aloitella uutta.

Dailyn jälkeen keskustelimme todistuksen tilauksen ja -listauksen tilannetta. Toteutukseen päätettiin muutama muutos koskien muodostettavan todistuksen kieltä ja tekstejä selkeyttääkseen käyttöliittymää.

Aloin miettiä uutta toteutusta. Tarkoituksena on tietysin ehdoin näyttää valintaruksi hakemuksen sivulla. Jos käyttäjä on kirjautunut uutena käyttäjänä – näytetään ruksi, mutta jos käyttäjällä ei ole olemassa asiointitiliä, valintaruksia ei näytetä. Jos käyttäjä on jo olemassa oleva asiakas, valintaruksi ja sen arvo näytetään, mutta arvoa ei pysty muokkaamaan. Toteutukseen liittyy näiden lisäksi monia muita ehtoja liittyen käyttäjän rooleihin ja hakemuksen tyyppiin. Tämä toteutus on hieman jopa humoristista, sillä yhden pienen valintaruksin toiminnallisuuteen käytetään monta henkilötyöpäivää.

Lähdin toteuttamaan tikettiä lisäämällä henkilön tuotuihin tietoihin hänen aikaisemmin valitseman valintaruksin arvon. Lisäsin tämän jälkeen uuden rajapintakutsun, joka selvittää käyttäjän henkilötunnuksen perusteella onko hänellä jo olemassa asiointitiliä. Yrityksen y-tunnukselle ei ole vielä tällä hetkellä tukea kyseiselle kyselylle, joten jätän sen selvittämisen myöhemmäksi. Lisäsin tuotuu luokkaan myös tiedon siitä, että löytyikö hakijan henkilötietoja tietokannasta vai ei. Näillä tiedoilla minun pitäisi saada kaikki ehdot täytettyä.

Seuraavaksi minulle tuli eteen jälleen tutustumista Wicketin kanssa. Jostain syystä en saanut Wicketin komponenttia, tässä tapauksessa valintaruksia ja sen tekstejä näkymään ehtojen mukaisesti. Jostain syystä Wicketin oma `setVisible()` –metodi ei toimi näillä. Tämän kanssa mietinkin loppupäivän.

Päivän aikana pääsin tavoitteisiini, sillä aloitin uuden tiketin ja pääsin siinä mielestäni myös hyvin eteenpäin. Aamupäivällä koin iskun itsetunnolle, kun luin tekemiäni virheitä, mutta kummatkin kuitenkin onneksi huomattiin ja sain korjattua tänään. Minulla on myös sellainen tunne tämänpäiväisestä toteutuksestani, että olen tehnyt sen hieman hutaisten. Tavoitteenani kuitenkin on saada se aluksi toimimaan, jonka jälkeen rupean miettimään, kuinka voisin refaktoroida koodia.

Tiistai 8.3.2016

Päivän aluksi asetin tavoitteeksi toteuttaa eilen sovitut muutokset todistuksen tilauksen kokonaisuuteen asiointiportaalista.

Ennen dailyä en kerennyt juuri muuta, kun valmistella ympäristöä, sillä saavuin tänään hieman myöhemmin töihin. Dailyn aikana keskusteltiin muutamista tiketeistä, joiden parissa työskentelen tällä hetkellä. Sovittiin myös, että jatketaan molemmista keskustelua myöhemmin.

Eilen oltiin yhdessä mietitty mahdollisia parannuksen aiheita todistuksen tilauksen ja listauksen kokonaisuuteen. Tänään yksi projektimme parissa kauiten työskennelleistä henkilöistä antoi mielipiteensä näihin muutoksiin. Hänen mielestä meidän ei kannata tällä hetkellä lisätä enempää hommia työpinoon, sillä ominaisuus toimii tällä hetkellä. Päätettiin luoda uusia tikettejä backlogiin näistä aiheista ja kysyä asiakkaalta mahdollisesta toteutuksesta versioitoimituksen jälkeen. Olin hieman jo aloitellut näitä eilen keskustelemiamme muutoksia, mutta en onneksi kovin paljoa.

Sain tehtyä muutoksen yrityksen valintaan KATSO –kirjautumisen kautta tulevista yrityksistä. Aikaisemmin valintakentässä oli oletuksena valittuna ensimmäinen yritys, mutta tämä haluttiin muuttaa niin, että siinä sen sijaan kehoitetaan valitsemaan yritys. Löysin myös samalla yhden puuttuvan ruotsinnoksen toteutuksesta. Sain nämä muutokset reviewiin.

Myöhemmin keskusteltiin myös jo eilen aloittamastani toteutuksesta – asiointitilin kieltorastista tämän saman henkilön kanssa. Hän kertoi tällekin porukalle melkein samat asiat. Muutaman muutoksen hän halusi kuitenkin pitää. Valintarastia ei enää haluta piilottaa, jos

käyttäjällä on olemassa asiointitili, sillä siihen menisi taas lisää aikaa. Näiden päätösten jälkeen huomasin, että eilen tekemäni toteutus jää turhaksi ja siirsimme tiketin takaisin sen alkuperäiselle toteuttajalle.

Näiden päätösten jälkeen aloin huomata, että minulla ei ole enää kovin paljoa tehtävää. Sprintti kuitenkin jatkuu perjantaihin asti. Testauspinossa on jonkin verran tikettejä, joita varmaan rupean tekemään seuraavina päivinä. Kysellessäni ympäriinsä sain kuulla, että toteutukseni tulosteiden listauksesta ei vielä kukaan toimi verkkosovelluksessa oikein. Jos päivämäärävalitsimista molemmista asettaa saman päivämäärän, kun tulosteen tulostuspäivämäärä, mitään tulosteita ei silti näy. Tämä johtuu jälleen samasta ongelmasta, kun aikaisemminkin, mutta olin laittanut molemmat päivämäärät nollaamaan aikansa. Tällä hetkellä ne siis olivat täysin saman arvoisia. Lisäsin aikaisemmin tekemääni funktioon parametriksi boolean -arvon, jonka lähetän päivämäärän mukana. Tämä arvo kertoo funktiolle, palauttaako funktio minimiajan vai maksimiajan. Tässä kohtasin myös outoon ongelmaan – tietokanta tukee enintään vuosilukemaa 9999. Olin kuitenkin antanut verkkosovelluksen puolella rajapintakutsulle arvoksi 31.12.9999, mutta silti jostain syystä rajapinnan puolella tarkasteltaessa arvoa jossain välissä siihen on lisääntynyt yksi vuosi. Lukema oli 31.12.10000. Tästä syystä tietokanta heitti virhettä. Korjasin tämän muuttamalla käyttöliittymästä palautuvan maksimiarvon vuoteen 9998.

Olin saanut päivän aikana myös muutaman kommentin tekemääni toteutukseen asiointitilin mainostekstistä. Kommentit koskivat jälleen if -lausetta, jossa tarkistin, onko päivämäärä minimi vai ei. Sain myös kommentin, että minun olisi hyvä alustaa muuttujat luokissa, sillä niitä ei alusteta konstruktorissa. Korjattuani toteutuksen ja mergettyäni sen masteriin sain pian sähköpostia, että kaikki yksikkötestit eivät enää mene läpi. Lähdin selvittämään aluksi, että tuleeko minulla paikallisesti sama virhe ja huomasin, että tulee. Virhe koski funktiokutsua, joka hakee teknisen koodin tietokannasta. Tämä funktiokutsu ei sijaitse itse testiluokassa, joten virhettä oli vaikeampi paikallistaa. Virhe tulee sen takia, koska se on testiluokasta kutsuttavan luokan konstruktorissa, joten senkin virheen funktiolle täytyy tehdä mockaus. Mockauksella määritellään millä arvolla funktiota kutsutaan ja mitä odotetaan sen palauttavan. Lisäsin tämän testiluokkaan ja lisäsin muutoksen myös toiseen toteutuksen mukaan reviewiin.

Päivän aikana pääsin tavoitteisiini, mutta tämä ei ollutkaan kovin kummoinen tavoite, sillä pääsin siihen jo ensimmäisten tuntien aikana, koska toteutukseen tuli muutoksia. Työpöytäni väheni tänään monta tikettiä ja se parhaimmillaan kävi jo ihan tyhjänä. Voi olla, että loppuviikosta tulee vaikeuksia keksiä tekemistä. Päivän aikana eteen tuli muutama virhe, jotka sain hyvin mutkitta myös korjattua. Opin tänään, että testiluokan toteutumi-

seen saattaa aiheuttaa muutokset, jotka tapahtuvat sen ulkopuolella – vaikka ei näytä siltä, että niillä olisi minkäänlaista vaikutusta siihen.

Keskiviikko 9.3.2016

Keskiviikon tavoitteeksi asetin saada omat toteutustiketit pois alta ja testata joitakin muiden tekemiä toteutuksia.

Päivän aluksi valitsin testauspinosta kaksi tikettiä, jotka kollega oli saanut valmiiksi. Tiketit koskevat perusteen liittämistä portaalin hakemukseen. Nämä kaksi koskivat hakemuksen viimeistä vaihetta, jossa hakemuksen tiedot lähetetään rajapintaan tallennettavaksi.

Dailyn jälkeen keskustelimme pienen porukan kesken versionhallinnasta löytyvistä koodistopäivityksistä. Huolenaiheena oli kuinka koodiston import –työkalu tuo tietueita tiedostosta. Testaaja oli viikon aikana tehnyt koodistoon pikkuhiljaa muutoksia ja oli eilen ottanut viikon takaisesta päivämäärästä kaikista muutoksista yhden tuontitiedoston. Hän oli huolissaan siitä, että jos se sisältää päällekkäisiä arvoja niin kuinka koodisto käsittelee tuodessa niitä. Päädyttiin siihen, että se ylikirjoittaa ne uusimmilla arvoilla.

Päivästä iso osa meni aamulla ottamieni kahden tiketin testaamiseen. Testatessani löysin muutaman paranneltavan kohteen – esimerkiksi muutamassa tekstikentässä, kun käyttäjä kirjoittaa tekstiä ja asettaa siihen myös rivinvaihtoja, rivinvaihdot katoavat näyttäessä saman tekstin yhteenveto –ikkunalla. Löysin myös yhden kriittisen virheen toteutuksesta, nämä samat kyseisten tekstikenttien arvot eivät tallentuneet tietokantaan lainkaan. Huomasin myös yhden bugin, jos hakemuksen viimeisellä sivulla tulee tietty virhe ja käyttäjä yrittää palata taaksepäin – järjestelmä näyttää käyttäjälle edellisen sivun, mutta lenkkimakkaraksi kutsuttu statuspalkki näyttää, että käyttäjä olisi siirtynyt eteenpäin. Näkyvistä katosi myös eteen- ja taaksepäin –painikkeet.

Iltapäivästä saatiin sähköpostia projektipäälliköltä, että meillä on nyt kova tarve saada loput tiketit valmiiksi ja tästä syystä meillä on nyt lupa tehdä palkallista ylityötä. Uuden version tuotantoon siirto lähenee ja joitakin toteutuksia ja useampia testauksia puuttuu vielä.

Loppupäivästä olin saanut muutaman tiketin takaisin testauksesta koskien todistusten tilaamista. Prosessin ensimmäisellä sivulla käyttäjä voi valita KATSOsta tulevista y-tunnuksista haluamansa. Tämän jälkeen järjestelmä näyttää y-tunnuksen kentän alla listan yritykseen liitetyistä henkilöistä. Lista tällä hetkellä valitsee oletusarvoisesti ensimmäi-

sen henkilön listasta. Toiminto haluttiin muuttaa samanlaiseksi kuin ylempi kenttä, eli kehottamaan käyttäjää valitsemaan henkilön. Toinen muutos koski sivulla olevia kunta- ja maa –kenttiä. Nämä kentät kuuluisi olla näkyvissä vain silloin, jos käyttäjän kirjautunutta henkilöä ei ole vielä tallennettu tietokantaan. Kolmantena huomiona oli korjaus, jossa seuraavalla sivulla näkyvässä varmistuksessa, jossa listataan samat tiedot, on saattanut näkyä väärä tietue kunnan kohdalla. Se on myös saattanut näkyä tyhjänä. Ensimmäinen virhe johtui siitä, että luokkaan, jossa hakijan tiedot siirretään ei ole asetettu kunnan lajia. Toinen virhe johtui siitä, että tietokannassa ei ole pakko olla arvoa kunnalla. Lisäsin tarkistuksen, että jos tietoa ei tule tietokannasta niin ei näytetä kenttää ollenkaan.

Päivän aikana sain laitettua eteenpäin myös muutaman puuttuvan ruotsinnoksen verkkosovelluksen tulosteet –listasta. Tein myös pienen tiketin, jonka otin kollegalta. Tiketti liittyi KATSOSTa tulevan kirjautumisen logitukseen. Tieto haluttiin laittaa tulostumaan palvelimen logille jokaisesta kirjautumisesta.

Päivän lopussa keskustelin minun työtilanteesta projektipäällikön kanssa. Olin saanut jokaisen tikettini siihen pisteeseen, etten voinut itse enää niiden etenemiseen vaikuttaa. Sovittiin, että katsotaan seuraavan päivän aamuna minulle jotain hommaa.

Pääsin päivän aikana hyvin tavoitteisiin. Sain omat toteutukset kaikki reviewiin ja sain testattua kokonaan myös kaksi toisen toteuttajan tikettiä. Voi olla, että tulen vielä testaamaan samoja tikettejä uudestaan ihan varmuuden vuoksi. Yllätyin tänään siitä, kuinka tarkka osaan itse olla testatessani – vaikka näidenkin viikkojen aikana olen ollut muutamissa oman toteutuksen testauksessa huolimaton. Olen pikkuhiljaa oppimassa tekemään työni erittäin huolellisesti.

Torstai 10.3.2016

Torstain tavoitteeksi mietin aamulla selvittää itselleni hommaa ja saada sitä tehtyä.

Saavuin tänään hyvin aikaisin töihin ja tästä syystä olin ensimmäisenä paikalla. Hetken käynnistelyäni konetta kollega saapui paikalle. Kysyin häneltä, että olisiko hänellä jakaa tai antaa jotain tikettiä omasta pinostaan. Valitettavasti hänellä ei ollut. Kysyin samaa seuraavalta saapuvalta kollegalta, projektipäälliköltä, kenen kanssa eilen keskustelinkin. Hän antoi minulle omasta pinostaan tekemistä. Pääsin ensimmäistä kertaa tekemään testitapauksia, eli määrittelemään mitä kuuluu testata tietystä toiminnallisuudesta.

Toiminnallisuus, josta keksin testitapauksia on perusteen liittäminen hakemukseen portaalissa. Tästä osuudesta teen testitapaukset muutamalle kohdalle, KATSO –tunnuksen toiminnallisuudelle ja hakijan tiedot –näkyhälle. Testitapaukset yritin luoda mahdollisimman tarkasti niin, että jokainen virhe tulisi esille.

Tämän jälkeen lähdin itse testaamaan omia testitapauksia. Testasin KATSO –tunnuksen liittämistä hakemukselle. Tarkoituksena on, että käyttäjä voi kirjautua KATSO:n kautta portaaliin ja tilata hakemuksen. Hakemuksen näkyhälle on myös listausvalinta KATSOsta tulevista yrityksistä. Testauksessa huomasin, että käyttöliittymällä ei pääse hakemuksessa taaksepäin tai keskeyttämään hakemuksen tilausta, jos KATSOsta tulevaa yritystä ei olla valittu. Huomasin myös puuttuvan virheviestin listauskentästä, jos sitä ei olla valittu. Kenttää ei myöskään rajattu punaisella niin kuin muita virheellisiä kenttiä.

Testattuani nämä keskustelin projektipäällikön kanssa mitä tekisin seuraavaksi ja hän ehdotti, että jos siirryn suoraan korjaamaan löytämiäni virheitä. Tässä tapauksessa käy kuitenkin niin, että en voi enää jatkaa toteutuksen jälkeen testausta, koska testaajan kuuluu olla eri henkilö, kun toteuttajan. Lähdin kuitenkin toteuttamaan löytämiäni virheitä. Taaksepäin- ja keskeytä –napit eivät toimineet siitä syystä, että Wicketissä pakolliseksi kentäksi oltiin asetettu dropdown –valikon komponentti, eikä pääkomponenttia, jossa tämä valintalista on. Puuttuvan virheviestin sain lisättyä lisäämällä Wicketin ' .html' tiedostoon puuttuvan span –kentän. Kentän rajaus punaisella korjautui tässä samalla.

Löysin korjatessa muutaman kosmeettisen virheen. Huomasin yrityksen valintakentän alla ylimääräisen rivinvaihdon. Tämä johtui siitä, että jostain syystä jokaisen input –kentän jälkeen oli lisätty ' ' –merkki. Tämä tarkoittaa non breaking space, ja on käytännössä välilyönti. Huomasin myös, että yrityksen valintakenttää ei ole lihavoitu niin, kuin muut pakolliset kentät. Tämä korjautui lisäämällä yhden css –luokan elementille. Viimeiseksi, huomasin, että yhdessä sivun radio button elementin labelissa oli pakollinen merkki – vaikka radio button –elementti on aina valittuna. Nämä muutokset korjattua toteutus toimi juuri niin kuin pitää ja sain muutokset tämän päivän aikana myös reviewin kautta mergettyä.

Päivän aikana testaaja huomautti, että asiointitilin mainostekstin teknistä koodia ei pysty muokkaamaan työpöytäsovelluksen kautta, josta sitä on loppupeleissä tarkoitus asiakkaan muokata. Sovellus heittää virhettä, jos kenttää yrittää lähteä muokkaamaan. Sovelluksen teknisten koodien listaus oli toteutettu niin, kun tietokantataulussa on vain joko numerotieto tai merkkitieto, ja päivämäärät aina tallennetaan merkkitietoon. Sovelluksessa löytyy myös päivämäärälle oma tietokenttä. Päättelin, että sovellukseen on varmaan tehty

ominaisuus, joka asettaa tähän kenttään kaikki merkkitiedosta löytävät päivämäärät. Päätelin myös, että kyseinen funktio on varmaankin asetettu olemaan huomioimatta minimi- ja maksimipäivämääriä, kuten asettamaani '01.01.0001' –päivämäärää. Tästä syystä sovellus näyttää tämän merkkitieto –kentässä ja heittää virhettä sitä muokattaessa. Kysyin varmistusta vielä vanhemmalta toteuttajalta ja olin päätellyt oikein. Hänen mielestään me emme lähde tällä aikataululla korjaamaan tätä ominaisuutta, vaan ilmoitamme asiakkaalle, että teemme tietokantapäivityksen teknisen koodin muokkaamisesta käsin sitten kun on sen aika.

Loppupäivästä toinen testaja palautti minulle tiketin liittyen tulosteiden lähetys –tauluun lisättävistä tulosteiden avaimista. Yhden tulosteen kohdalla tauluun on tullut väärän tulosteen avain. Lähdin selvittämään syytä. Törmäsin kyseisen tulosteen PDF – luontitiedostossa funktiokutsuun, johon oli jostain syystä määritelty juuri tämä toisen tulosteen enum –luokan arvo. Tämän korjattuani ajoin eräajon uudelleen ja totesin, että nyt tauluun tallentuu oikea avain. Laitoin tiketin reviewiin pyytäen kommenttia, että oliko tällä väärään tulosteeseen viittauksella jokin tarkoitus, vai onko se ollut vain huolimattomuusvirhe.

Pääsin päivän tavoitteisiin, sillä keksin itselleni aamulla kollegan avulla töitä. Sain niitä myös eteenpäin. Opin päivän aikana yrityksen testaamiskäytäntöjä ja järjestelmää, kun jouduin sitä tutkimaan melkein kuin suurennuslasin kanssa.

Perjantai 11.3.2016

Päivän tavoitteiden päättäminen rupeaa olemaan hyvin hankalaa, koska oma työpino on niin pieni, ettei tiedä riittääkö hommaa edes koko päivälle. Asetan nyt kuitenkin tavoitteeksi saada omasta pinosta tikettejä eteenpäin.

Aamupäivästä toteutuksesta oli palautunut toissa päivänä testaukseen laittamani tiketti. Tiketissä olin kommentoinut muutamiin tekstikentän rinvaihtoihin, tietokantaan tallentamattomiin tietoihin ja virheen jälkeisen edellinen –painikkeen painamiseen liittyen. Olin testannut tietokantaan tallentamattomat tiedot väärin, olin hakenut tietokannasta väärillä avaimilla. Kumpaakaan näistä toisista ongelmista toteuttaja ei korjannut, totesi vain, että nämä ovat vanhaa toteutusta. Tämä laittoi minut ihmettelemään, että jos törmätään virheeseen tai outoon toteutukseen, eikö sitä korjata tai yritetä tehdä paremmaksi. Ymmärsin kuitenkin, että emme voi kuluttaa työtunteja vanhojen ominaisuuksien kanssa ilman, että kysymme asiakkaalta luvan käyttää budjettia. Tein siis tuosta virheestä uuden tiketin backlogille, jotta se joskus sieltä korjattaisiin.

Aamupäivästä pidettiin sprintin retrospektiivi. Omalla puhevuorollani en keksinyt mitään negatiivista sanottavaa sprintistä, mutta sanoin, että edellisessä retrossa sopimamme tikettien liputus on toiminut hyvin ja tikettien jaottelu on ollut parempaa. Palaverissa pääaiheeksi osoittautui kysymykset käyttötapausten ja testausten reviewistä. Keskeisiä kysymyksiä olivat missä vaiheessa näitä tehdään, mihin näitä dokumentoidaan ja kuka näitä tekee. Itselläkin oli täysin auki nämä prosessit. Päätettiin ainakin, että testauksen reviewin tekee toinen testaja, eikä tikettiä siirretä viimeistelyyn ennen, kun se on käynyt sen läpi. Toisena aiheena keskusteltiin pidempään sprintin kesken uusien tikettien lisäämistä sprinttiin. Näin ei saisi tehdä ilman, että asiasta keskustellaan useamman henkilön kanssa. Palaverin loppupuolella kollegani Oulusta, joka on ollut projektin parissa kenties alusta lähtien, ehdotti, että pidennettäisiin tänään loppuvaa sprinttiä viikolla. Ehdotus meni läpi, itse olin tätä ehdotusta vastaan, mutta en kuitenkaan ilmaissut muille omaa kantaani asiaan. Nyt töiden pula jatkuu minun osalta vielä ensi viikkoon.

Iltaapäivästä lähdin tekemään testitapauksia asiointiportaalin hakemuksen asiointitilin kieltöön ja hakijan tietoihin yhteenvedossa liittyen. Sain nämä myös testattua ja siirrettyä retrossa keskustelemaamme testireviewiin, eli kommentoin tiketille, että tämän testit tarvitsevat reviewin ja siirsin avoimien tikettien testauspinoon.

Asiointiportaalissa on myös hakemuksella sellainen ominaisuus, että siihen voidaan tietysin parametrein tulla suoraan toisesta hakemuksesta. Hakemukseen tehtäessä muutoksia, hakemuksen jokainen eri käyttötapa täytyy testata uudelleen. Lähdin testaamaan tätä käyttötappaa. Tälle oltiin kirjoitettu valmiiksi testitapaukset joskus aikaisemmin dokumentaatioon. Tämän tyyppin hakemuksella järjestelmä hakee valmiiksi käyttäjää varten kenttiin tietoja. Jos johonkin kenttään ei löydy tietoja, käyttäjä pystyy siihen sen lisäämään. Huomasin testatessani, että yrityksen y-tunnuksen kenttää pystyy muokkaamaan, vaikka siihen tulee tieto järjestelmästä. Huomasin myös muutaman muun yritykseen liittyvän ongelman – joissain tapauksissa yrityksen nimikenttiä ei näytetty ollenkaan. Kirjasin näistä ongelmista tiketille ja testitapauksiin havainnot. Päätämme maanantaina, kuka kehittäjästä lähtee korjaamaan löytämiäni ongelmia.

Päivän loppupuolella pääsin myös ihmettelemään kollegan kanssa hänen toteutukseensa syntyvää ongelmaa. Hän oli tehnyt rajapintakutsun verkkosovelluksesta ja tuloksista puuttuu yrityksen y-tunnus kokonaan. Hän näytti minulle koko prosessin, jossa hän lähettää kutsun ja vastaanottaa tiedot. Rajapinnan puolella viimeisellä koodirivillä y-tunnus oli vielä tallella, mutta heti seuraavassa vaiheessa verkkoselaimen network –työkalulla katsottuna palautetuissa tiedoissa ei ollut y-tunnusta. Ratkaistiin ongelma huomattuamme,

että tiedot sisältävässä luokassa getterin ja setterin nimissä oli yksi iso kirjain, mikä itse muuttujan määritelmässä ei ollut. Javassa tämä ei aiheuta mitään ongelmaa, mutta jostain syystä käyttämämme Java –kirjasto, joka muuntaa luokan sisältämät tiedot JSON –muotoon ei ymmärtänyt, että nämä funktiot kuuluvat tälle muuttujalle.

Päivän aikana pääsin tavoitteisiini, vaikka ne olivatkin hyvin epämääräiset. Sain muutamaa tiketin omasta pinostani pois. Maanantaina minulla ei olekaan muuta, kun vain tämä yksi viimeisenä testaamani tiketti omassa pinossa ja sekin on mahdollisesti menemässä toiselle toteuttajalle. Pääsin onneksi päivän aikana myös katomaan hetkeksi kollegani kirjoittamaa koodia. Kaipaen jo muutaman koodittoman päivän jälkeen omia toteutustöitä.

Yhteenveto viikosta 10

Tämä viikko kului paljon hitaammin kuin muut. Uskon, että tämä johtuu siitä, että työpinnossani on viikon aikana ollut epämääräinen määrä tikettejä ja työn määrä on ollut myös tästä syystä vaihtelevaa.

Viikon aikana opin toimimaan entistä paremmin Wicketin kanssa. Vielä se kuitenkin yllättää toimintatavoillansa ja monimutkaisilla komponenteillaan. Opin myös tekemään testitapauksia ja testaamaan perusteellisesti. Olen hieman epävarma kuitenkin siitä, että kuinka perusteellisesti teen testitapauksia ja testauksia. Näiden viikkojen aikana on tullut huomattua, että olen melko äkkipikainen ja laitan tikettejä eteenpäin ennen, kun pitäisi. Yritän kovasti päästä eroon tästä tavasta. Etenkin testauksessa, kun pitää olla erityisen tarkka, että kaikki toimii juuri niin kuin pitää.

Testaus meillä hoituu niin, että yleensä määrittelijä on ennalta määritellyt testitapaukset, jotka testataan perusteellisesti ja kuitataan testatuksi. Tietenkin samalla löytyy muita pikkukorjauksia, joita voi ehdottaa korjattavaksi. Testaaja myös tarkistaa, että toteutus toimii niin kuin käyttötapauksissa on määritelty.

Näiden mukaan toimiminen on ollut muutaman tiketin kohdalla vaikeaa, sillä testitapauksia ei ole määritelty etukäteen, vaan minun täytyi ne keksiä. En voinut myöskään tarkistaa käyttötapauksesta, miten toteutuksen kuuluisi toimia, koska toteutukseen on tehty sovittuja muutoksia, joita ei olla muutettu käyttötapauksissa. Eli minun siis täytyy vaan luottaa, että toteutus toimii niin, kun sen pitää.

Ohjelmistojen testaus on tärkeää, sillä jokainen meistä tekee virheitä. Virheitä on pienempiä ja isompia – jokainen ominaisuus täytyy kuitenkin perusteellisesti testata ennen tuo-

tantoa, koska jokin virhe saattaa tulla hyvin kalliiksi. Testaamalla varmistetaan, että asiakas saa juuri sitä mitä tilaa – usein määrittelyvaiheessa ollaan ymmärretty asiakasta väärin ja tästä syystä toimitetaan asiakkaalle tuote, jota hän ei ole edes halunnut. Testaamalla varmistetaan myös työn laatu, jos ohjelmisto sisältää useita virheitä, se ei todellakaan ole laadukas ohjelmisto. (ISTQB 2016).

Viikon aikana oma työpäiväni on huventunut melkein olemattomiin muutaman päivän päätteeksi. Onneksi aina seuraavan päivän aamuna olen kuitenkin saanut tekemistä. Tämä on aiheuttanut minulle hieman stressiä, kun en ole varma mitä tulen seuraavina päivinä tekemään. Etenkin nyt, kun sprintti kestää vielä ensi viikon ajan. Keskusteltiin viime viikon päätteeksi alustavasti, että voisin jatkaa kollegan yhden ticketin edistämistä – eli mahdollisesti pääsen ensi viikolla taas koodin pariin.

Järjestelmässämme käytämme yksikkötestejä, jotka automaattisesti ajetaan jokaisen commitin jälkeen masteriin. Yksikkötestejä nykyaikana yleensä käytetään niin sanotussa 'Test Driven Development' (TDD) –ohjelmointityylissä. TDD:ssä on kyse siitä, että yksikkötestit kirjoitetaan ennen tuotantokoodia. Tuotantokoodia kirjoitetaan vasta, kun yksikkötesti on ajettu ja se antaa virheen, että tuotantokoodissa ei löydy tämän nimistä funktiota. Tällä tavalla saadaan mahdollisimman puhtaasti kirjoitettua koodia. Tällä tavalla koodista löytyvät turhat rivit saadaan minimoitua. TDD:n avulla saadaan ohjelmiston testikattavuutta nostettua – tämä on prosenttimäärä, kuinka iso osa ohjelmiston funktiosta on kirjoitettu yksikkötesti. Kun ohjelmisto on hyvin testikattava, tämä tarkoittaa sitä, että jos jokin menee rikki jostain muutoksesta – se näkyy heti rikki menneellä testillä rivin tarkkuudella. (Martin 2009, 122-123).

Projektimme kehitys aloitettiin yli kymmenen vuotta sitten, jolloin kukaan ei melkein ollut edes kuullut TDD:stä, jonka takia projektissamme näin ei tehdä. Yksikkötestejä kirjoitetaan vain tärkeimmistä ominaisuuksista. Niitä kirjoitetaan myös ominaisuuksista, jotka käsittelevät paljon logiikkaa tai algoritmeja varmistaakseen niiden oikeanlainen toimivuus.

Koodin refaktorointi, josta mainitsin viikon alkupuolella, tarkoittaa toimivan koodin muuttamista enemmän luettavaksi ja tehokkaammaksi. Tämä kuuluisi tehdä vasta, kun on toimiva koodi ja testitapaukset kirjoitettuna koodille, koska tässä vaiheessa koodi saattaa mennä myös rikki. Refaktoroidessa kehittäjä pikkuhiljaa siivoaa koodia, vähentää käytettyjä rivejä koodissa, nimeää funktioita uudelleen kuvatakseen paremmin niiden toimintaa yms. Jokaisen muutaman muutoksen jälkeen kehittäjä voi ajaa yksikkötestit uudelleen tarkistaakseen, että koodin toiminta on vielä eheä. (Martin 2009, 172-173).

Olen itse päättänyt kehittäessä ohjelmoida ensin toimivan ratkaisun toiminnallisuudelle, jonka jälkeen refaktoroin koodia, jotta se on luettavaa, helposti laajennettavissa ja tehokasta. Itse olen ollut melko huono yksikkötestien tekemisessä, olen kuitenkin päättänyt, että rupean niitä tekemään enemmän. Hyvän kehittäjän erottaa siitä, että yksikkötestejä kirjoitetaan paljon.

3.7 Seurantaviikko 11

Maanantai 14.3.2016

Viikonlopun aikana mietin, että tälle viikolle on pakko selvittää jotain tehtävää. Etenkin, kun meillä on lupa tehdä ylitöitä projektia kohden, eikä minua pieni lisätulo haittaisi. Tästä syystä asetin päivän tavoitteeksi selvittää itselleni tälle viikolle hommaa. Asetin myös aamupäiväksi tavoitteeksi saada oma osuuteni portaalin hakemuksen testauksesta valmiiksi.

Aamupäivästä lähdin testaamaan viime viikolla kesken jäänyttä tikettiä, joka koskee portaalista tehtyä hakemusta tietyllä tavalla, jolloin saadaan tiedot etukäteen, haetaan henkilörekisteristä ja asetetaan niille kuuluviin kenttiin. Viime viikolla testasin jo käyttöliittymän osuuden ja löysin siitä huomioitavaa, joita kollegani lähti jo tänään korjaamaan. Lähdin testaamaan tietojen tallentamista luonnollisella- ja juridisella henkilöllä erikseen. Tarkistin jokaisen tietokantaan tallentuvan tiedon, että se on oikeassa paikassa ja –muodossa. Löysin muutamia huomioitavia asioita, mutta ne eivät ole tärkeitä, sillä toteutus toimii ilman niitä. Esimerkiksi kun henkilön osoite on ulkomailla, tietokantaan tallennetaan eri kohtaan paikkakuntatieto kuin Suomessa asuvalla henkilöllä. Jostain syystä paikkakuntatieto ei tallentunut hakemuksen tauluun lainkaan, mutta lopulta huomasin, että se kuitenkin tallentuu toiseen paikkaan – josta sitä myös myöhemmin käytetään.

Aamupäivästä keskustelin myös projektin päällikön kanssa muistuttaakseni, että kaipaan lisää töitä. Pyysin häntä myös tarkistamaan, että olin tehnyt riittävän tarkasti testitapaukset asiointitilin hakemukseen.

Lounaan jälkeen pidimme tiimin palaverin, tällä kertaa poikkeuksellisesti englanniksi, koska joukossamme oli myös portugalilaisia tiimin jäseniä. Palaverissa käytiin läpi ajankohtaisia asioita ja uutisia. Tämä palaveri on koko asiakkuuden tiimin jäsenten kesken pidettävä palaveri. Monet tiimin jäsenistä eivät tiedä muiden projekteista kovin paljoa, esimerkiksi edes miltä niiden käyttöliittymä näyttää. Tästä syystä on päätetty, että palavereissa ruvetaan esittelemään tiimin eri projekteja. Tänään oli tarkoitus esitellä meidän tiimin projektia, mutta videojärjestelmämme ei toiminut. Siirrettiin tätä ensi palaveriin.

Päivän aikana sain myös testauksesta takaisin asiointitilin mainokseen liittyvän tiketin. Testaaja oli huomannut, että kahdesta tulosteesta puuttuu itse mainos. Toiseen tulosteeseen en ollut itse huomannut laittaa ollenkaan mainosta. Lisäsin tämän nopeasti. Ongelmia tuli toisen tulosteen kanssa, koska siihen olin kuitenkin jo lisännyt mainoksen.

Jotta saan tulosteen tulostettua, minun täytyy luoda järjestelmään kokonaan uusi asia ja liittää tähän asiaan monta eri muuttujaa. Tämän jälkeen minun täytyy ajaa kolme eri eräajoa, joiden ajamisessa kestää erittäin kauan paikallisesti. Tämän takia debuggaus on hyvin hankalaa, koska joudun odottamaan testaamista noin vartin verran eräajojen pituuden vuoksi. Tämän ongelman kanssa taistelin loppupäivän.

Ennen kun päivä päättyi, kollegani tarjosi minulle omasta pinostaan yhtä tikettiä toteutettavaksi. Otin sen ilomielin vastaan ja toivottavasti pääsen huomenna aloittelemaan sen toteuttamista, kunhan pääsen tämänhetkisen virheen yli.

Päivän aikana pääsin tavoitteisiini, sillä sain tekemistä, jolla toivottavasti pidän itseni kiireisenä loppuviikon ajan. Sain myös testattua tikettini pois omasta pinosta. Opin tänään enemmän laajan tietokannan rakenteesta, kun jouduin testatessani tarkistamaan kymmenestä eri tietokantataulusta tietoja.

Tiistai 15.3.2016

Aamupäivällä saapuessani töihin oli hauska näky – siivoojat olivat nostaneet koko maise-makonttorista tuolit, roskakorit ja naulakot pöydille puhdistessaan lattiaita. Mietin päivän tavoitteeksi aloitella eilen kollegalta saatua tikettiä ja viimeistellä korjaus liittyen asiointitilin mainostekstiin.

Laitoin heti aamusta paikallisesti eräajot ajamaan testatakseni, että toimiiko eilen tekemäni muutokset toteutukseen. Hetken odoteltuani päättelin, että tässä ajossa saattaa kestää jokunen tovi, joten nappasin testauspinosta tiketin ja lähdin testaamaan kollegan tekemää toteutusta. Toteutus liittyy portaalista tehtyjen hakemusten tarkastukseen ja tietojen täyttöön viranomaisen toimesta verkkokäyttöliittymässä. Tämän vaiheen jälkeen hakemukselta muodostuu lopullinen asia.

Dailyssä keskusteltiin edellä mainitusta toteutuksesta ja päätettiin, että eilen kollegalta saatua toteutusta ei toteutettaisi vielä tällä viikolla, vaan keskitytään testauksiin. Tässä koin hieman pettymystä, sillä olin odottanut kunnon toteutustehtäviä jo jonkin aikaa.

Dailyn jälkeen huomasin, että eräajot olivat valmistuneet. Lopulta sain muutaman tuhat sivua tulosteita ja ilokseni huomasin hetken selailtuani, että mainostekstit toimivat. Korjasin vielä yhden ylimääräisen rivinvaihdon pois tulosteilta. Huomasin myös, että yhdellä

tulosteella puuttuu kokonaan otsikko. Epäilen, että tämä johtuu omasta ympäristöstäni – tämä testataan vielä testausympäristössä.

Jatkoin web-liittymän hakemuksen testausta. Tämä oli vielä hyvin alustavaa testausta, joten en kirjoittanut vielä mitään testitapauksia ylös. Aloin kirjata löydöksiäni ylös, ja huomasin ison liudan puutteita, virheitä ja parannuksia. Löysin muutaman kriittisen virheen, henkilörekisteriin pystyi tämän hakemuksen kautta lisäämään lähes tyhjiä henkilöitä, joilla ei ole muita tunnistetietoja, kun avain. Huomasin myös, että henkilö- ja y-tunnuksen kohdalle pystyy asettamaan minkä tahansa stringin. Huomasin, että jokaiseen päivämääräkenttään pystyy asettamaan päivämäärän tulevaisuuteen – myös syntymäpäivän. Löysin myös muutaman kentän, jota ei näy ollenkaan hakemuksella ja muutaman kentän, jotka eivät tallennu lopullisen asian tietoihin, vaikka niiden pitäisi.

Loppupäivästä käytiin pienellä porukalla läpi tätä verkkosovelluksen hakemuksen toteutusta. Mukana oli sitä toteuttamassa ollut kollega ja projektipäällikkö. Käytiin läpi koko hakemusprosessi portaalista lähtien ja mietittiin minun löytämiäni tapauksia, sekä löydettiin uusia paranneltavia. Tämän keskustelun yhteydessä sain tietää, että on tiedossa muutama ominaisuus lisää, mitä hakemukseen pitää toteuttaa. Esimerkiksi on kokonaan jätetty huomiotta sellainen mahdollisuus, että hakijan esitetyt hakemuksia kuuluisi pystyä myös hylkäämään ilman, että siitä muodostuu ensin asia. Kävimme alustavasti läpi tämän mahdollisia toteutustapoja. Puhun huomenna vielä arkkitehdin kanssa tämän toteuttamisesta ja toivon mukaan pääsen jo aloittamaan sen toteuttamista. Palaverimme jälkeen projektipäällikkö meni vielä keskustelemaan tämän arkkitehdin kanssa muun muassa minun löydöksistä ja muista ongelmista, mitkä todettiin palaverissa.

Palaverissa annoin hetken muhimassa olleen ideani projektipäällikölle. Ideani liittyy nykyisiin web-käyttöliittymän modaali-ikkunoihin – niitä ei pysty liikuttelemaan. Löysin netistä hyvältä näyttäjän Javascript –kirjaston, jota käyttäen tämän pystyisi hienosti toteuttamaan. Alkuperäisen työpöytäsovelluksen käyttäjät ovat tottuneet ikkuna –tyyppiseen sovellukseen ja olisivat varmasti tyytyväisiä, jos voisivat liikutella modaali-ikkunaa. Lähetin päivän lopuksi projektipäällikölle sähköpostin tästä kirjastosta.

Päivän aikana en päässyt tavoitteisiini, koska tikettiä jota minun piti aloitella, päätettiin lykätä. Sain kuitenkin asiointitilin mainostekstin takaisin testaukseen ja toivottavasti tällä kertaa myös viimeistelyyn.

Keskiviikko 16.3.2016

Päivän tavoitteeksi asetin aloitella eilen sovittua hakemuksen hylkäyspainiketta ja testata hakemuksen tallennus loppuun.

Lähdin aamupäivästä testaamaan hakemuksen tallentamista. Eilen testasin pääosin käyttöliittymän puolta, joten nyt lähdän tarkistamaan, että jokainen tieto menee oikein tietokantaan. Testatessa löysin, että viesti hakemuksen käsittelijälle –kenttä löytyy lopullisen asian lisätiedon lajeista, joten sen voi tallentaa sinne. Eilen oltiin mietitty sitä, että se näytettäisiin vain hakemuksella. Samalla huomasin, että toinen lisätietoihin kuuluva tieto ei ollut tallentunut. Tästä päätin, että todennäköisesti mikään asian lisätieto ei tallennu kantaan – olin oikeassa.

Sain ennen dailyä viestin arkkitehdiltä, että hän tekee itse hakemuksen poiston toiminnallisuuden samalla, kun hän tekee muuta vastaavaa työtä. En siis vielä kukaan saanut itselleni kunnan kehitystyötä.

Dailyn jälkeen keskustelin kollegan kanssa testauksessa löytämiäni ongelmia. Korjattiin nämä ongelmat saman tien, tähän ei kulunut paljon aikaa. Pian tämän jälkeen saatiin projektipäälliköltä sähköpostia, jossa kerrottiin yksityiskohtaisemmin, mitä muutoksia täytyy vielä tehdä. Keskusteltiin nämä läpi pienessä palaverissa. Jäätin kollegan kanssa vielä palaverin jälkeen keskustelemaan työnjaosta – olen siis vihdoinkin saamassa jotain kunnan työtä.

Päätettiin kollegan kanssa, että lähdän toteuttamaan lisätiedoista listausta hakemuksen käsittelyyn. Toteutuksessa käytän tietokannasta tulevia tietoja ja luon niiden perusteella hakemuksen käsittelyn ensimmäiselle sivulle listan hakemuksen lisätiedoista. Tämän toteuttamisessa ei kestänyt kauaa – sain sen valmiiksi juuri ennen lounasta.

Lounaan jälkeen testasin vielä hetken ja tein muutaman pienen muutoksen listaukseen. Tämän jälkeen selviteltiin seuraavaa tehtävää ja huomattiin, että joudumme kysymään määrittystä projektipäälliköltä yhteen osuuteen, joten en ottanut työn alle vielä mitään lisää. Lähdin parikoodaamaan kollegan kanssa hänen toteutusta.

Saatuamme muutama muutos valmiiksi lähdimme miettimään hakemuksen formin validointia, eli asettamaan pakollisia kenttiä ja varmistamaan, että annetut tiedot ovat oikeassa muodossa ennen, kun ne tallennetaan henkilörekisteriin.

Ennen, kun päästiin edes alkuun, huomasimme hakemuksen ensimmäisellä sivulla puutteen – portaalista tulleella hakemuksella oli tyhjä valintakenttä, vaikka se oli asetettu portaalissa ja se myös löytyi tietokannasta. Huomasimme, että verkkosovellus ja asiointiportaali käyttävät eri koodistoja, toinen hakee tiedot koodistopalvelusta ja toinen tietokannasta. Vertaillessamme tietoja, huomasimme, että kaikki tiedot eivät täsmää. Eli portaalista pystytään valitsemaan arvo, jota ei löydy verkkosovelluksen koodistosta ja tästä syystä kenttä jää tyhjäksi. Ehdotimme muutamaa korjausvaihtoehtoa projektipäällikölle – joko lisätään puuttuvia tietoja toiseen koodistoon, tai merkitään tuntemattomat arvot yleisellä 'muu' -arvolla. Tämän selvittämiseen menikin suurin osa päivästä.

Päivän viimeisenä asiana keskustelin kollegan ja projektipäällikön kanssa työnjaosta ja päätimme, että ryhdyn huomenna tekemään asian hakemuksen liitteille omaa modaaliikkunaa. Tällä hetkellä hakemuksen liitetiedostoja ei pysty katselemaan enää sen jälkeen, kun siitä on luotu asia.

Päivän aikana en päässyt kaikkiin tavoitteisiin, koska en päässytkään toteuttamaan eilen sovittua hommaa, vaan suunnitelmat muuttuivat. Sain kuitenkin alustavan testauksen loppuun hakemukseen liittyen. Yllätyin päivän aikana jälleen, kuinka pienellä määrällä koodia Angularilla saa koodattua toiminnallisuuksia, ja opin siinä samalla hieman lisää Angularista.

Torstai 17.3.2016

Torstain päivän tavoitteeksi asetin saada eilen sovitun hakemusten liitteiden modaaliikkunan toteutettua.

Aamupäivästä lähdin toteuttamaan liitteiden listausta. Toteutuksessa on kyse asiointiportaalien kautta tehtyjen hakemusten liitteiden listaus lopullisella asialla. Ikkuna aukeaa asian liitteet –välilehdellä erillisestä painikkeesta modaali-ikkunaan. Ikkunasta pystyy myös lataamaan tiedostot palvelimen levyltä.

Dailyssä sain kuulla, että kollegani, joka on tähän mennessä ollut minulle läheisin, on irtisanoutunut ja lähtee toisen yrityksen palvelukseen kuun lopussa. Hänen kanssaan olen tehnyt useita ominaisuuksia ja hän on myös se henkilö, joka minut pääosin perehdytti. Tämä uutinen oli hieman haikea kuulla, mutta ei kuitenkaan masentanut loppupäiväksi.

Liitteiden listauksen toteutin lisäämällä aluksi painikkeen asialle ja sen viittaamaan funktioon, joka kutsuu uutta Angularin controlleria. Tämä controller on liitetty html –tiedostoon,

johon kopioin muutaman viikon takaisesta todistusten listauksesta rakenteen ja muokkasin sen sopivaksi. Toteutin myös rajapintakutsun ja rajapinnan puolelle toiminnallisuuden, joka hakee asiaan liitetyt hakemuksen liitteet ja niiden tulosteen avaimen. Tällä avaimella kutsun linkkiä painettaessa uutta funktiota, joka hakee itse tiedoston ladattavaksi. Toteutuksessa ei tullut juuri lainkaan ongelmia. Pääosin toteutus oli muualta esimerkkiä katsoen nopeasti tehty.

Lounaan jälkeen keskustelin kollegan kanssa jäljellä olevista toteutuksista ja päädyttiin siihen, että otan listasta kaksi pienempää muutosta ja toteutan ne hakemukselle. Toinen näistä liittyy hakemuksen käsittelijään. Hakemuksella kuuluu aina olla käsittelijä, emmekä halua päästää käyttäjää käsittelemään hakemusta, ennen kun se on asetettu sen käsittelijäksi. Tähän keksin ratkaisun liittämällä ensimmäisen yleistietosivun seuraava – painikkeeseen painettaessa tarkistuksen, että jos hakemuksella ei ole vielä käsittelijää, niin käyttäjältä kysytään, haluaako hän ryhtyä käsittelijäksi. Jos käyttäjä ryhtyy käsittelijäksi, hän pääsee eteenpäin hakemuksella.

Toinen keskustelemistamme toteutuksista liittyy ongelmaan, jonka huomasin alustavassa testauksessa. Jos käyttäjä painaa edellinen –painiketta selaimesta, hänet ohjataan hakemuksen sivulle takaisin, vaikka hän olisi siitä jo muodostanut asian. Samaan tilanteeseen päästään myös hakemusten listauksesta, sillä se päivittyy vain tietyin väliajoin. Ratkaisuna tähän päätin asettaa tarkistuksen, jos hakemuksen tila ei ole kesken. Jos tarkistuksen mukaan hakemuksen teko ei ole kesken, seuraava nappi poistetaan käytöstä ja ruudulle tulostetaan virheviesti, jossa ilmoitetaan käyttäjälle asiasta.

Näissä toteutuksissa ei ilmennyt minulla juuri ollenkaan normaalista poikkeavaa ongelmaa, joka ei pienellä debuggauksella olisi selvinnyt. Sain myös päivän päätteeksi laitettua alustavat ruotsinnokset hakemukselle. Näistä kaikista muutoksista saimme tehtyä pull requestin, joka on toivottavasti huomiseksi mergetty.

Päivän aikana oli puhetta siitä, että mitä tehdään, jos lakimuutokset eivät astukaan voimaan ajallaan tai osa peruttaisiin kokonaan. Tässä vaiheessa voidaan jo sanoa, että kaikki niihin liittyvä työ on jo toteutettu, mutta eduskunta ei ole kuitenkaan vielä tehnyt päätöstä. Minultakin kysyttiin, että kuinka iso työ olisi tietyn ominaisuuden palauttamisella. Toivottavasti tähän ei kuitenkaan päädytä.

Päivän aikana pääsin tavoitteisiini, ja tein hommia siitä vielä ylikin. Olen tyytyväinen päivän etenemiseen. Olen oppinut paljon lisää Angularin käytöstä ja sen kanssa toteuttaminen näköjään käy minulta jo erittäin jouhevasti.

Perjantai 18.3.2016

Aamulla päivän tavoitteeksi asetin testilla verkkosovelluksen hakemuksen tarkistamisen prosessia pidemmälle ja kehittää tarvittavia ominaisuuksia.

Päivän aluksi huomasin heti, että minulla ei paikallisesti käynnisty pääpalvelin lainkaan. Virheviestinä antoi, että tietokantaan ei voida lisätä toista samanarvoista avainta. Tiesin heti, että kyse on minun omasta mokasta. Olin eilen testannut muutamaa tietokantalausetta paikallisesti, ennen kun mergeän ne masteriin. Olin unohtanut poistaa nämä rivit tietokannasta testauksen jälkeen. Tämän takia palvelimen flyway –automaattinen tietokannan migraatio työkalu ei pystynyt tekemään kantapäivitys – tiedostoni mukaisia käskyjä. Sama ongelma kävi myös testiympäristössä, joten käynnistin senkin palvelimen uudelleen, kun sain vian korjattua.

Aamupäivästä keskustelin projektipäällikön kanssa asiointitilin kiellon toteutuksesta hakemuksen tarkistuksessa. Hän sanoi, että toteutetaan se vain niin, että tieto tallentuu suoraan hakemuksesta tietokantaan. Tämän toteuttaminen vaati hakemuksen tarkistus – sivulla varmistuksen, että kieltonappi ei päivity, jos tietokannasta löydetään jo hakemuksen henkilö ja hakemuksen tiedot korvataan niillä tiedoilla. Tein myös toteutuksen rajapinnan puolelle, jossa muutin tallennuksen päivittämään asiointitilin kiellon myös silloin, kun kyseessä on jo olemassa oleva henkilö.

Huomasin asiointitilin tallennusta toteuttaessa, että hakemuksen tiedot saattavat toisinaan muuttua, jos tietokannasta haetaan henkilön tiedot. päättelin, että tässä saattaa tulla sellainen tilanne, että hakemuksen tiedoissa on ajantasaisempaa tietoa, kun tietokannassa. Emme voi kuitenkaan suoraan laittaa henkilörekisteriin mitä joku on hakemuksessa kirjoittanut. Keksin tähän mielestäni hyvin hienon ratkaisun, ja onnistuin toteuttamaankin sen erittäin hienolla tavalla. Haettaessa tietoja tietokannasta, koodi ohjautuu uuteen funktioon, johon annetaan parametrina hakemuksessa olleet henkilön tiedot ja henkilörekisteristä palautuneet tiedot. Funktiossa käydään loopissa läpi jokainen hakijan Javascript –objektin avain ja verrataan sen arvoa haettujen tietojen avainten arvoihin. Jos arvot eivät täsmää, lisään uuteen taulukko –muuttujaan muuttuneen kentän nimi. Tämä sama prosessi toistetaan kaikille eri henkilöiden tiedoille, kun niiden tietoja haetaan hakemuksen eri vaiheilla.

Nyt minulla on taulukossa jokaisen tiedon avain, joka on muuttunut. Tässä seuraavassa funktiossa käydään läpi minkä tahansa henkilön muuttuneita avaimia. Formin kenttien modelien nimet ovat seuraavaa mallia: 'hakemus.data.hakija.hetu'. Tästä nimestä kohta 'hetu' on yksi muuttuneiden avainten taulukon arvo. Tämän arvon avulla rakensin loopin,

joka hakee jokaisen taulukon avaimen muuttuneen elementin Angularin avulla HTML:stä perustuen sen ng-model –attribuuttiin. Tämän jälkeen jQuery:n avulla lisään kyseisen tekstikenttä –elementin sisälle ikonin, joka ilmaisee, että jotain on vialla. Lopputuloksena jokaisen muuttuneen kentän perässä näkyy kellertävä pyöreä ikoni, jonka keskellä on huutomerkki. Tämän lisäksi, jos sivulla on yksi tai enemmän muuttunutta kenttää, lisään tekstin pohjalle ilmoituksen, joka selittää asiaa käyttäjälle.

Ongelmia tämän toteutuksen kanssa tuli hieman, mutta en kuitenkaan kovin pitkään ollut missään niistä paikoillaan. Suurin ongelma liittyi takaisin- ja seuraava painikkeisiin. Näitä painamalla ilmoitukset kentistä kuuluisi kadota ja tulla takaisin riippuen näkymästä. Toisessa tilanteessa samoihin kenttiin pitää tulla eri kohtiin eri henkilön tietoihin liittyvät muuttuneet kentät. Sain ongelman kuitenkin ratkaistua. Esittelin ratkaisuani projektipäällikölle, ja hän piti erityisen paljon näkemästään. Hän sanoi toteutusta erittäin hienoksi. Olen hyvin tyytyväinen siihen itsekin.

Toteutin päivän aikana myös pienemmän korjauksen toteutukseen. Aikaisemmin tallenna –painiketta hakemuksen päätteeksi pystyi painamaan useaan kertaan. Muutin painikkeen pois käytöstä, kun sitä painetaan.

Päivän aikana pääsin tavoitteisiini ja olen hyvin tyytyväinen päivän kulkuun ja omaan toimintaani. Opin paljon uutta Angularista. En voi sanoa sitä riittävän vahvasti, mutta pidän Angularin kanssa työskentelemisestä erittäin paljon. Tänäa juuri hieman vitsailin kollegan kanssa, että kuinkahan monta viikkoa olisi mennyt, jos oltaisiin tehty kaikki muutaman päivän aikana tehdyt ominaisuudet Wicketillä.

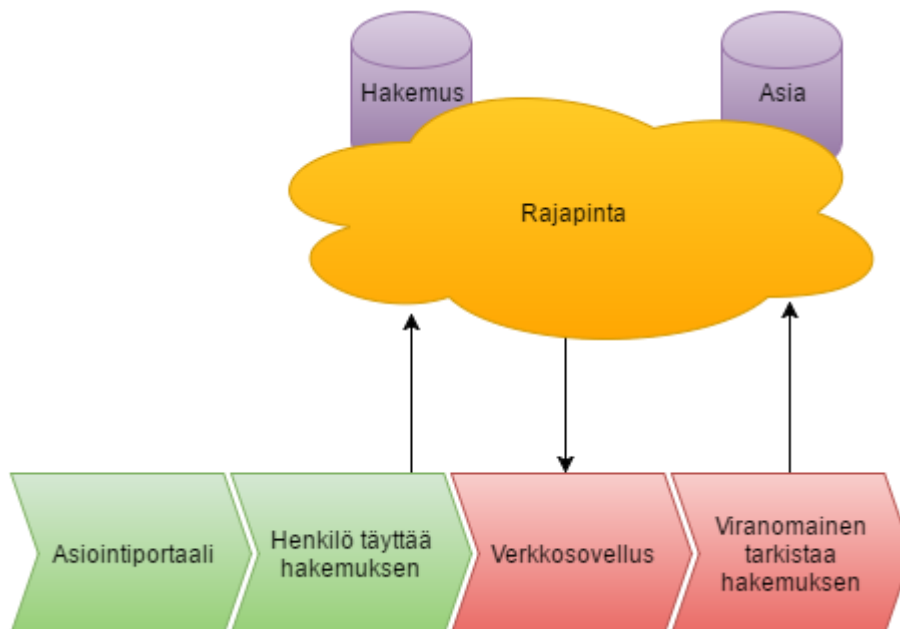
Yhteenveto viikosta 11

Viikon aikana kehitin taitojani eniten Angularin kanssa. Olen alkanut nauttia erittäin paljon sen kanssa kehittämisestä. Tästä eteenpäin käytän varmasti sitä ensisijaisesti omissakin projekteissani. Olen vieläkin erittäin tyytyväinen perjantaina tekemääni ratkaisuun muuttuneiden tietojen ilmaisuun käyttäjälle. Tälle tarkistukselle olisi ollut monta erilaista ratkaisua, jotka olisivat vaatineet paljon enemmän rivejä koodia. Kolleganikin ehdotti minulle, että kannattaa varmaan lähteä lisäämään jokaisen kentän perään uusi piilotettu kenttä, jota sitten näytetään tarvittaessa. Keksinkin ehdottomasti parhaimman ratkaisun, sillä yhdessä pienessä loopissa saan tulostettua HTML:n sekaan kaikki tarvittavat rivit. Ainoa tekemäni muutos Angularin controllerin ulkopuolelle oli ilmoitus käyttäjälle siitä, mitä nämä huutomerkki –ikonit tarkoittavat input –kenttien lopussa. Tein myös keskiviikkona ja tors-

taina muutaman toteutuksen Angularilla, mutta nämä olivat enemmän kopiointia muista toteutuksista, kun omaa toteutusta.

Viikon alkupuolella tein myös tarkempaa tutkimusta tietokantaan tallennetuista tiedoista, kävin yli kymmenen tietokantataulun läpi portaalin hakemuksesta tallentuvista tiedoista. Tämän ansiosta minulla on nyt paljon selkeämpi kuva tietokannan rakenteesta henkilöiden ja asioiden osalta, sillä hakemuksen tallennuksessa käytetään molempia suuria kokonaisuuksia.

Olen muutamien viikkojen aikana puhunut paljon asiointiportaalin hakemuksista ja verkkosovelluksen hakemuksista. Selkeyttääkseni asioita käyn prosessin vielä lävitse. Asiointiportaalin käyttäjä kirjautuu sisään – käyttäjä voi olla joko luonnollinen tai juridinen henkilö. Käyttäjä tekee asiointiportaalissa hakemuksen täyttämällä omat- ja haettavan henkilön tiedot ja muita vaadittavia tietoja hakemukselle. Tiedot lähetetään rajapintaan ja sieltä tallennetaan tietokannan hakemus –tauluihin. Verkkosovelluksen käyttäjänä toimii aina viranomaisen. Kun käyttäjä päättää käydä läpi hakemustyöpinoa, järjestelmä hakee rajapinnasta kaikkien hakemusten tiedot. Käsitellessä tietoja käyttäjä liittää hakemuksen henkilöt henkilörekisteristä löytyviin henkilöihin, jos sellaisia löytyy. Käyttäjä myös muuttaa tarvittaessa hakemuksen tietoja, jonka jälkeen käyttäjä tallentaa hakemuksen. Hakemuksesta tallentuu järjestelmään uusi asia. (Kuvio 3).



Kuvio 3: Hakemuksen prosessin kuvaus

Testipalvelimemme tekee kasauksen joka yö. Kasauksessa järjestelmä hakee uusimmat toteutukset versionhallinnasta ja ajaa siihen liittyvät tietokantapäivitykset ja tarkistaa, että

yksikkötestit toimivat. Tietokantapäivitykset meillä hoituvat Flywayn avulla. Flyway on avoimen lähdekoodin tietokantamigraatioon käytetty suosittu Javan sovelluskehys. Projektissamme käytämme sitä tietokannan versionhallintaan. Flyway luo tietokantaan taulun 'SCHEMA_VERSION', joka sisältää jokaisella rivillä version ja viittauksen skriptiin, joka on useimmiten .sql –tiedosto. Flyway seuraa tietyn kansion sisältöä ja ajaa aina version tietokantamuutokset järjestyksessä pienimmästä järjestysnumerosta suurimpaan. Luodessa uutta skriptiä, pitää tietokannasta tarkistaa nykyinen versionumero ja nimetä tiedosto tämän mukaisesti seuraavalla numerolla. (Flyway 2016).

3.8 Seurantaviikko 12

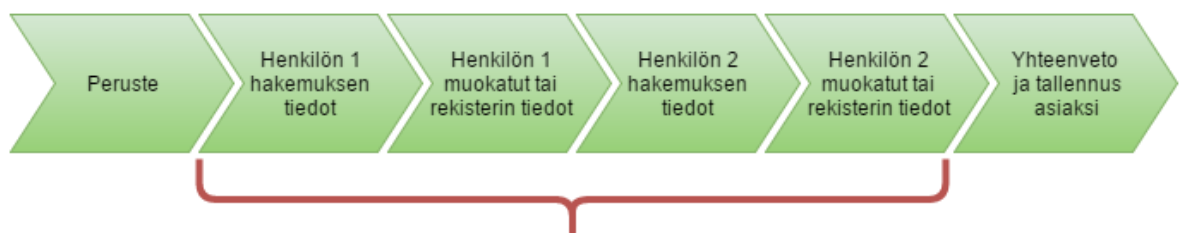
Maanantai 21.3.2016

Maanantain tavoitteeksi asetin saada jatkettua viime viikolla kesken jäänyttä hakemuksen tarkistusta verkkosovelluksen puolella.

Aamupäivästä lähdin toteuttamaan viime viikon lopulla huomaamiani puutteita hakemuksesta. Olin esimerkiksi huomannut, että käyttöliittymä ei näytä niin hyvältä, kun jotkin kentät ovat pidempiä kuin toiset ja muita pieniä viilauksia. Huomasin myös, että hakemuksen kohteen henkilöllä ei ole tarvetta näyttää jokaista kenttää, joten siivosin niitäkin pois. Huomasin samalla, että viime viikolla toteuttamani hakemuksen ja rekisterin tietoja vertaileva ja eroista ilmoittava funktio ei vertaile nimiä. Tämä johtui siitä, että henkilön nimi on erillisessä objektissa. Lisäsin toisen loopin aikaisemman sisälle, joka looppaa nimien kentät läpi.

Dailyn jälkeen lähdin toteuttamaan laajempaa ratkaisua. Hakemuksessa on kuvion 4 mukainen wizard –tyyppinen rakenne. Huomasin, että niitä painamalla ei tapahdu mitään, vaikka ne vilkkuvat ja hiiren osoittimeksi vaihtuu sormi liikuttaessa hiirtä niiden päälle. Tarkoituksena olisi siis toteuttaa toimiva ratkaisu, jonka avulla voitaisiin siirtyä näkymästä toiseen niin, että käyttäjän muuttamat tiedot tallentuvat hakemukselle.

Hankalan tästä toteutuksesta, niin kuin aikaisemmastakin hakemuksen- ja rekisterin tietojen vertailusta tekee hakemuksen rakenne. Hakemuksen alussa käyttäjä tarkistaa portaalista tulleen hakemuksen perusteen. Seuraavilla neljällä sivulla HTML:ää katsottuna koodi on sama, mutta kenttien määrä, sisältö ja muokattavuus vaihtelevat näkymästä toiseen. Hakemuksen tiedot –näkymissä on myös painikkeet, joilla haetaan rekisteristä henkilön tietoja tietyin ehdoin – jos ei löydy, voidaan myös asettaa uusi henkilö rekisteriin. (Kuvio 4).



Käytävät samoja elementtejä

Kuvio 4: Hakemuksen käsittelyn sivurakenne

Aikaisemmassa toteutuksessa seuraava- ja edellinen –painikkeille oli tehty erilliset funktiot. Yhdistin nämä yhdeksi ja lisäsin siihen toiminnallisuuden myös wizardin etenemispaikkin klikkauksille. Funktio ottaa parametrina käyttäjän haluaman askeleen, tai jos esimerkiksi seuraava –painiketta painetaan, funktiolle annetaan parametrina nykyinen sivu plus yksi.

Ensimmäiseksi ongelmaksi ilmaantui se, että pystyin painamaan itseni mille tahansa sivulle ennen, kun olin edes edennyt hakemuksella niin pitkälle. Ratkaisin tämän lisäämällä uuden taulukko muuttujan, joka sisälsi alustavasti avaimet 1-6 arvoilla false. Käyttäjän siirtyessä seuraavaan ruutuun, kyseisen ruudun arvo muuttuu arvoon true.

Aikaisemmassa toteutuksessa oli asetettu koko prosessin tiedot henkilön yhteen muuttujaan, jonka tiedot vaihtuivat eri näkymissä. Muutin tämän niin, että jokaisella henkilöllä on oma muuttujansa ja näytettävillä tiedoilla on omansa. Näin koodista saa paljon luettavampaa ja kenttien arvojen muuttuminen saadaan käsiteltyä paremmin vaihdettaessa näkymiä.

Päivän aikana käytiin myös uuden sprintin suunnittelu, sillä edellinen sprintti loppui perjantaihin. Suunnittelu eteni erittäin nopeasti, sillä ei voitu ottaa kovin paljoa hommaa lisää, koska asiakkaalle toimitetaan testattavaksi versio 31.3. Tämä tarkoittaa sitä, että tämä sprintti tulee olemaan erittäin lyhyt, kuluvan päivän lisäksi viisi työpäivää, koska väliin tulee pääsiäisen pyhät.

Päivän päätteeksi sain muutokseni mergettyä masteriin. Yllätyin, että en saanut juuri lainkaan kommentteja arkkitehdiltä liittyen toteutukseeni. Aikaisemmin olen saanut aina useampia parannuksia. Keskustelin projektipäällikön kanssa toteutuksesta vielä puuttuvista asioista, joita lähden huomenna työstämään. Pääsin tavoitteisiini, sillä sain hyvin tehtyä hakemusta eteenpäin. Olen hyvin tyytyväinen tästä päivästä.

Tiistai 22.3.2016

Tiistain päivän tavoitteeksi asetin viimeistellä verkkosovelluksen hakemuksen tarkistusta.

Lähdin heti aamusta toteuttamaan eilen iltapäivällä huomattuja puutteita hakemukselta. Portaalin hakemuksella on muutamia kenttiä, kuten vahvistuspäivämäärä, tp-tapa ja p-pyyynnön valintaruksi, joiden kentät lisäsin hakemuksen tarkistukselle. Huomasin, että asiointiportalissa p-pyyynnön valintaruksi asettuu arvoon false ja menee pois käytöstä, jos

tp-tapa on jokin muu, kuin normaali. Toteutin tämän saman myös tarkistukselle. Saman tyyppisen ratkaisun tein henkilön tiedoista liittyville henkilön tyyppille ja –lajille. Jos henkilön tyyppi on luonnollinen henkilö, niin henkilön laji poistuu käytöstä ja asettuu myös luonnolliseksi. Juridisella henkilöllä voi olla eri lajeja. Huomasin testatessani myös vielä muuttaman liian pitkän kentän, jotka korjasin.

Jouduin tekemään muutoksia vielä portaalin hakemuksen tallennus –osioon. Huomasin esimerkiksi, että haettavalta henkilöltä puuttuu kansalaisuustieto kokonaan. Tämän lisäsin oletuksena tulemaan suomen kansalaisena, sillä järjestelmään ei tämän tyyppisestä henkilöstä voida edes tallentaa muun maan kansalaista.

Puhuttiin eilen projektipäällikön kanssa, että hakijan tilinumeron BIC –tunnus ei näy hakemuksella. Hän mainitsi myös, että haettavan henkilön lisätiedot, mm. ammatti ja omaisuus eivät näy hakemuksen tarkistuksessa. Päätettiin kuitenkin, että näitä kenttiä käsittelijä ei pääse muokkaamaan. Nämä tiedot kuuluvat molemmat haettavan henkilön lisätietoihin, joten ratkaisin tämän loopilla, jossa käyn läpi henkilön jokaisen lisätiedon ja tämän tyyppistä muodostan otsikon kentälle ja arvosta kentän sisällön. Asetin kentät myös disabled –tilaan, jolloin niitä ei pystytä muokkaamaan.

Lounaan jälkeen lähdin toteuttamaan tilinumeron BIC –tunnuksen generointia käyttäjän antamasta tilinumerosta. Käyttäjä voi antaa järjestelmään joko kotimaisen- tai ulkomaisen tilinumeron. Käyttäjän antaessa ulkomaista tilinumeroa, näytetään BIC –kenttä pakollisena täytettävänä. Jos kenttää ei olla täytetty hakemusta lähettäessä, käytän jo olemassa olevaa funktiota, joka parsii tilinumerosta 'FIXX' jälkeen tulevat 1-3 numeroa, joiden perusteella haetaan koodistosta niitä vastaava pankin BIC –tunnus. Tässä kohtaa törmäsin ongelmaan. Aikaisemmin olin taistellut jokainen kerta tehdessäni Mavenin install –komennon projektille, olin joutunut kommentoimaan yhden JUnit –testitapauksen, joka sattumoisin liittyy juuri tähän BIC –generointiin, jotta saan testit menemään läpi. Nyt jouduin vihdoinkin lähteä selvittämään syytä, miksi funktio ei toimi minun ympäristössä. Selvitettyäni hetken kollegan kanssa ongelmaa, hän muisti, että koodistosta generoidut .xml –tiedostot kuuluvat olla kopioituna myös toiseen paikkaan, jotta funktio osaisi ne hakea. Tämän jälkeen testit menivät läpi, mutta sama ongelma jatkui. Hetken selvitettyään kollega totesi, etten pysty kutsumaan kyseisiä koodistoja asiointiportaalin koodin puolelta, vaan minun täytyy siirtää toteutus rajapinnan puolelle. Tehtyäni näin totesin, että toteutus toimii. Itse toteutus oli yksinkertainen – tarkistetaan vain, että jos BIC –tietoa ei löydy ennestään, kutsutaan funktiota tilinumerolla ja asetetaan paluuarvo muuttujaan. Huomasin samassa luokassa, jossa BIC –generointi tehdään, että sieltä löytyy myös tilinumeron formatointifunktio. Tämä

funktio asettaa välilyönnit oikeisiin paikkoihin tilinumeron varrella saadakseen siitä selkokuiskemman. Päätin käyttää tätä myös tilinumeron kohdalla.

Päivän aikana opin, että en voi kutsua koodistoon liittyviä funktioita ihan mistä vain. Olen myös tyytyväinen, ettei minun täydy enää joka ikinen kerta kasatessani projektia – jota kuitenkin tehdään päivittäin – kommentoida pois testiluokasta yhtä funktiota. Pääsin päivän aikana tavoitteisiini. Hakemuksen käsittelyosuus on jo lähes testausta vaille valmis.

Keskiviikko 23.3.2016

Tavoite tälle päivälle ei kovin paljoa eroa eilisestä. Päätin yleisesti tehdä viimeisteltäviä asioita tähän versioon. Ollaan kuitenkin niin lähellä toimitusta.

Aamupäivästä lähdin korjaamaan aikaisemmin löytämiäni muutamia ongelmia. Olin huomannut, että hakemuksen tarkistuksessa ilmoitusteksti muuttuneista kentistä näkyy, vaikka ainuttakaan huutomerkki –ikonin ei löydy tekstikenttien perältä. Selvittäessäni syytä huomasin, että muuttuneet kentät –taulukkomuuttujaan tallentuu myös kentät, joita ei löydy DOM:ista. Korjatakseni tämän, kutsuin jQuery:n avulla muuttujaan jokaisen DOM:in sisältämän aikaisemmin merkitsemällä luokalla merkitsemäni elementin. Jos elementtejä ei löydy; ei tulosteta myöskään ilmoitustekstiä.

Eilen keskusteltiin projektipäällikön kanssa. Esille tuli tilanne, jossa haetun henkilön postinumero vaihtuu toiseen, jolloin myös vaihtuu henkilön alue. Lopullinen asia siirtyy siis haetun henkilön postinumeron mukaiseen alueeseen. Päätettiin, että tästä olisi hyvä ilmoittaa erikseen muuttuneen kentän huutomerkki –ikonin lisäksi. Lisäsin elementin, joka sisältää käyttäjälle ilmoitettavan tekstin ja säädän sen näkyvyyttä Angularin avulla riippuen siitä, onko postinumero muuttunut.

Ennen lounasta taloomme oli tullut vierailemaan Valtiokonttorin pääjohtaja Timo Laitinen, joka tuli puhumaan minun asiakaskohtaiselle tiimille ja muutamille muille tiimeille valtionhallinnon digitalisaatiosta. Puhe oli hyvin mielenkiintoinen ja seurasin sitä mielenkiinnolla.

Lounaan jälkeen mietittiin jäljellä olevia tehtäviä ja työnjakoa. Mietin lounaan aikana, että me ollaan nyt ensimmäistä kertaa verkkosovellukseen tekemässä validaatioita henkilötunnukseen ja y-tunnukseen. Nyt kun on siihen mahdollisuus, tehdään validointi kerralla kuntoon. Kollega oli aikaisemmin aamupäivästä kopioinut aikaisempaa Java –toteutusta, jossa ei olla tehty oikeanlaisia tarkistuksia aiheesta ja tehnyt siitä oman Javascript versioon. Päätettiin saman tien tehdä tämä kerralla kuntoon. Aikaisemmin henkilötunnuksen

tarkistuksessa ei ole tarkistettu, että sen sisältämä päivämäärä on oikea päivämäärä. Henkilötunnuksesta ei ole myöskään tarkistettu tarkistusmerkkiä lopusta. Y-tunnuksen validoinnista on puuttunut myös tarkistusmerkin tarkistus. Näitä oli mielenkiintoista toteuttaa ja testata. Saatiin työ yhdessä toteutettua ilman ongelmia.

Lopulta päädyttiin luopumaan työnjaosta ja tekemään parikoodaamalla loppuja puutoksia. Asiakkaalle esiteltiin eilen tulevaa versiota, jossa huomattiin, että yhdeltä vasta luodulta henkilöltä puuttuu osoitteesta kokonaan kunta, joka on pakollinen tieto. Tunnistin tämän henkilön tiedot ja tiesin, että tämä luodaan minun tekemästä todistuksen tilauksesta. Tämän selvittämiseen meni aikaa. Aluksi ajateltiin, että ongelma tuskin on kovin vakava ja saattaa tapahtua vain erittäin harvalla osalla tapauksia. Lopulta huomasimme, että jos käyttäjä on kirjautunut sisään KATSO:n kautta ja hänellä on tiedoissaan vain yksi y-tunnus – joka on kaikista yleisin tapaus – ei paikkakunnan valintakenttää koskaan näy, eikä tämän takia myöskään tallennu. Lisättiin yksi funktiokutsu sivun latautuessa ja ongelma oli korjattu.

Samalla, kun selvitettiin edellistä ongelmaa, huomasimme, että luonnollista henkilöä tallennettaessa tietokantaan todistuksen tilauksessa ei tallenneta henkilön syntymäpäivää tai sukupuolta, vaikka molemmat tiedot ovat saatavissa henkilötunnuksesta. Asetettiin nämä tiedot tallentumaan tietokantaan.

Loppupäivän aikana Helsinkiin päiväksi Turusta matkustanut kollega kutsui minut hänen ja toisen projektin päällikön kanssa keskustelemaan. Todettiin, että tämän version jälkeen ei ole juuri jäljellä kovin paljon töitä nykyisen projektin parissa. Minut ollaan päätetty siirtää toiseen projektiin työskentelemään, kuitenkin saman tiimin ja asiakkuuden alle. Työskentelen todennäköisesti vielä huomisen nykyisen projektin parissa, jonka jälkeen lähdän perehtymään uuteen projektiin. Mielenkiintoista tästä uudesta työnkuvasta tekee se, että olen ainoa toteuttaja, joka asuu Suomessa. Muut projektin kehittäjät asuvat Portugalissa. Suomessa on minun lisäksi arkkitehti – sama henkilö, joka on myös aikaisemmassa projektissa ja muutama testaja, sekä itse projektipäällikkö.

Päivän aikana pääsin hyvin tavoitteisiini, sillä saatiin ominaisuuksia toteutettua, korjattua ja paranneltua erilaisia toimintoja ympäri monia toteutuksia. Opin tänään mm. henkilö- ja y-tunnuksen rakenteet ja validointimenetelmät. Odotan innolla, mitä tämä uusi projekti tuo tullessaan.

Torstai 24.3.2016

Tämä on tämän viikon viimeinen päivä, koska viikonloppuna on pääsiäisen pyhiä. Päivän tavoitteeksi asetin vielä tänään viimeistellä versioon tulevia toteutuksia ja saada viimeinen testaustiketti pois omasta pinosta.

Aamupäivästä lähdin selvittämään ongelmaa, jossa portaalin todistuksen tilauksessa valittaessa hakijalle kuntaa ja palatessa vahvistusruudulta takaisin, kunnan ja maan valintakentät ovat piilotettu. Tämä johtui ominaisuudesta, joka piilottaa valintakentät, jos käyttäjällä on jo tiedot rekisterissä ja kunta on valittu. Muutin toteutusta niin, että käytin luokassa jo aikaisemmin olevaa muuttujaa, jonka arvon muutun todeksi, jos käyttäjää ei ole vielä rekisterissä. Tästä eteenpäin riippumatta kenttien arvoista, kentät näytetään aina kyseisellä henkilöllä. Korjasin myös samalta sivulta muutaman pakollisen kentän pakolliseksi.

Huomasin korjatessani, että vahvistussivulta siirtyessä maksusivulle ja sieltä jos kerkeää painamaan takaisin ennen, kun käyttäjä ohjataan maksusivulle – muodostuu virhe. Koitin hetken selvittää ongelman taustaa, mutta päätin kuitenkin ratkaista ongelman piilottamalla edellinen –painikkeen kokonaan näkymältä.

Seuraava korjaus liittyy myös asiointiportaaliin, mutta tällä kertaa hakemukseen. Ensimmäisellä hakemuksen sivulla on muutama kenttä, joihin täytyy syöttää tietyssä muodossa tuomioistuimen päätöksen numeroita. Näihin kenttiin ei ole lisätty validointia. Lisäsin kenttien alle ohjetekstit kertomaan käyttäjälle missä muodossa tieto kuuluu syöttää kenttään.

Testatessa pidemmälle hakemuksen tarkistusta verkkosovelluksesta, huomattiin, että eilen toteuttamamme henkilötunnuksen validaattori ei toimi jokaisella henkilötunnuksella. Seurasin debuggerilla koodin ajoa ja huomasin, että virhe tulee Javascriptin Moment kirjaston muuttujan konstruktorissa. Jostain syystä objektilla luotu konstruktori ei toiminut, mutta lisäämäni stringistä muodostettu Moment –objekti toimii. Korvasin aikaisemman objektin luonnin uudella ratkaisulla.

Samalla kun testasin henkilötunnuksen validaattoria ja henkilötunnuksesta generoituja syntymäpäiviä ja sukupuoliä, huomasin, että eilen lisäämäni hakemuksen peruste ei tallennu asialle oikein. Pienen hetken tutkittuani löysin rajapinnan puolelta kohdan missä se tallennetaan – tieto tallentui jokainen kerta oletusarvolla, ja hakemuksessa syötetty arvo jätettiin huomioimatta. Tämän korjattuani huomasin ongelman myös omassa toteutuksessani. Jos ensimmäisellä sivulla muutetaan perustetta ja tämän jälkeen, jos hakee uuden henkilön tiedot myöhemmässä vaiheessa, ensimmäisessä vaiheessa tehty perusteen

muutos korvaantuu uudella tiedolla. Päätin asettaa ensimmäisessä vaiheessa muutetun tiedon omaan muuttujaan ja myöhemmässä vaiheessa aina käytän tätä muuttujaa tiedon korvaamiseen uuden henkilön tiedoissa.

Huomasin samalla myös, että jos henkilön lisätiedoissa ei ole ainuttakaan riviä – hänen tiedoissaan silti näytetään lisätiedot –otsikko. Lisäsin tähän elementtiin tarkistuksen, jossa koko elementti piilotetaan, jos henkilöllä ei ole lisätietoja.

Loppupäivästä tuli vielä muutama muutos hakemusten listaukseen verkkosovelluksessa. Määrittelijä pyysi, että lisäisin listauksen kentiksi hakemuksen tulevan asianumeron ja alueen. Hän halusi myös yhden kentän pois, joka tällä hetkellä ei missään tapauksessa tulostu listaan muutenkaan.

Viimeisenä korjasin vielä aikaisemmin tekemääni dialogia, jossa kysytään käyttäjältä, haluaako hän ryhtyä hakemuksen käsittelijäksi. En ollut huomionnut toteutuksessa, että käsittelijää voi myös vaihtaa. Lisäsin dialogiin tekstin, jossa ilmoitetaan käyttäjälle, jos hakemuksella on jo toinen käsittelijä ja kuka hän on.

Päivän aikana sain myös testauspinostani viimeisen tiketin siirrettyä viimeistelyyn. Tein myös muutaman selvityksen tulosteiden toimintatavoista ja muutin muutaman tulosteen tekstejä. Päivän aikana tein hyvin monia erilaisia hommia viimeistellen toteutuksia. Pääsin tavoitteisiini ja odotan innolla ensi viikkoa, kun pääsen työskentelemään toiseen projektiin.

Perjantai 25.3.2016

Tänään olen vapaalla viettämässä pääsiäistä.

Yhteenveto viikosta 12

Tämä lyhyt viikko kului suurimmilta osin version viimeistelyyn ja viimeisten puuttuvien ominaisuuksien toteuttamiseen. Alkuvuikosta toteutin mielestäni hyvin näkymien välillä vaihtuvien tietojen käsittelyn. Hakemuksella kuuluu olla tallessa alkuperäisten kahden henkilön tiedot, joiden lisäksi tallessa kuuluu olla nykyisessä välilehdessä näkyvät tiedot ja molempien henkilön lopulliseen lähetykseen menevät tiedot. Aina siirryessä muokattavasta näkymästä – joko seuraava- tai edellinen painikkeiden kautta tai navigaatiopalkista, tiedot vaihtavat paikkaa muuttujista toisiin. Olen hyvin tyytyväinen toteutukseen, mitä sain aikaiseksi. Sain sen myös testattua niin, että kaiken pitäisi toimia juuri niin kuin pitää. Opin tämän toteutuksen kautta entistä syvemmillä Javascriptin ja Angularin maailmoihin.

Viikon aikana puhuin HTML:n DOM –objektista. DOM on lyhenne sanoista 'Document Object Model'. Se luotiin mm. Javascriptiä varten salliakseen sen manipuloida sitä. Tämän ansiosta verkkosivumme ovat nykyään dynaamisia. DOM määrittelee dokumentin (esim. verkkosivun) rakenteen objektiksi, jota selaamalla voidaan luoda, muokata tai poistaa dokumentin sisältöä. (W3C 2016).

Itse käytin Angularin kautta DOM –objektia selvittääkseni, sisältääkö dokumentti sinne itse lisäämiäni ilmoituksia kenttien eroavaisuuksista. Jos dokumentti sisältää niitä, syötetään dokumenttiin uusi elementti, joka ilmoittaa käyttäjälle, mitä nämä ilmoitukset tarkoittavat.

Viikon keskivaiheilla sain kuulla mielenkiintoisia uutisia. Pääsen tutustumaan uuteen projektiin ja uusiin kollegoihin. Entistä mielenkiintoisempaa tästä tekee se fakta, että suurin osa uusista kollegoista asuvat Portugalissa. Tulen olemaan arkkitehdin lisäksi ainoa to-
teuttaja, joka työskentelee Helsingissä. Odotan innolla, kuinka englannin kielen taitoni kehittyvät ja miten hyvin yhteistyö sujuu, kun välissä on tuhansia kilometrejä ja kielimuuri.

Viikon aikana meillä kävi myös keskustelemassa valtiokonttorin pääjohtaja Timo Laitinen aiheesta valtionhallinnon digitalisaatio. Pääjohtaja kertoi pääosin selvityksestä, jonka Valtiovarainministeriö oli antanut Valtiokonttorille tehtäväksi. Selvityksen kohteena on valtion virastot ja niiden digitalisaatio ja tuottavuuspotentiaali. Selvitystä tehdessä viraston työntekijöiltä ja johtajilta kysyttiin ideoita siitä, kuinka heidän toimintaa voidaan kehittää digitalisoimalla palveluja. Kehitysehdotuksia saatiin yhteensä 1024 kappaletta. Nämä ehdotukset liittyvät mm. palveluketjujen selkeyttämiseen, asioiden käsittelyn sujuvuuteen ja hallinnon ketteröitymiseen. Pääjohtaja mainitsi, että tällaisia tuloksia osattiin monelta taholta odottaa ja hänelle oltiinkin todettu, että nämä asiat ovat olleet jo aikaisemmin tiedossa ja niistä ollaan jo keskusteltukin. Selvityksen tarkoituksena olikin saada ensimmäistä kertaa mustaa valkoiselle ja saada asioita muutettua, eikä vain keskusteltua. Hän sanoi muutamaksi isoimmaksi ongelmaksi muutoksen tiellä virastojen välisen yhteistyön puutteen ja lakisäädökset. Esitelmän aikana pääjohtaja kertoi myös esimerkkejä siitä, kuinka näitä asioita ollaan hoidettu muissa maissa ja peilasi näitä kokemuksia Suomen epäonnistumisiin. (Valtiokonttori 2015).

Omasta mielestäni tällainen selvitys tuli tarpeeseen. Valtion virastot tarvitsevat pientä rohkaisua digitalisaatiossa ja etenkin yhteistyössä. Itse pidän lainsäädäntöä yhtenä isoimmista ongelmista. Tällä hetkellä virastojen välillä ei juuri saa liikkua tietoa, muutoin kuin luval-
la. Mielestäni lain kuuluisi sallia melkein minkä vain tiedonvaihdon virastojen järjestelmien sisällä. Olen tyytyväinen tilaisuuteen ja opin paljon uutta.

Ongelmia viikon aikana ei tullut juuri lainkaan. Yksi isompi ongelma ympäristööni liittyen, jossa minulta puuttui koodiston koodit toisesta paikasta. Huomasin itseni selvästi kehittyneen viikon aikana, sillä en saanut melkein ollenkaan kommentteja arkkitehdiltä, tai muiltakaan toteuttajilta code reviewissä.

3.9 Seurantaviikko 13

Maanantai 28.3.2016

Tänään olen vapaalla viettämässä pääsiäistä.

Tiistai 29.3.2016

Tiistain päivän tavoitteeksi asetin saada asennettua koneelleni uuden projektin kehitysympäristö, tutustua hieman siihen ja viimeistellä vielä edellisen projektin hommia.

Aamupäivästä huomasin, että viikonloppuna on lähetelty sähköposteja rikki menneestä ominaisuudesta. Verkkosovelluksen uuden asian tallennus oli mennyt rikki ja syyksi epäiltiin minun tekemää muutosta yli viikon takaa. Virhe johtui siitä, että olin oletanut, että yksi objektin sisältämä muuttuja on turha ja käyttänyt sen sijaan toista muuttujaa. En ollut ottanut huomioon sitä, että tätä muuttujaa käytetään uuden asian tallennuksessa. Olin ajatellut vain asian tallennusta hakemuksesta. Tässä on käytännössä kyse kahdesta saman nimisestä muuttujasta, jotka ovat jostain syystä eri nimisiä kahdessa eri toteutuksessa. Lisäsin toiseen näistä toteutuksista yksinkertaisen rivin, jossa kopioidaan toisen muuttujan tiedot toiseen, lähettäen molemmat näistä samoilla tiedoilla eteenpäin. Tämä oli nopein ja yksinkertaisin ratkaisu.

Samalla huomasin, että uuden asian tallennuksesta asialle tulee myös tieto, että asian kirjaamistapa on internet. Kirjaamistavan kuuluisi tällä tavalla olla manuaalinen ja hakemuksesta tallentaessa internet. Korjasin tämän selvittämällä, onko asian tallennusvaiheessa olemassa hakemuksen avainta, jota manuaalisessa tallennuksessa ei ole. Jos ei ole, asetetaan kirjaamistapa manuaaliseksi.

Huomasin samalla vielä kolmannen asian. En voinut asettaa uutta luotua asiaa toimeenpanoon, koska asian vastaava –kenttä oli tyhjä. Sain arkkitehdiltä ohimennen vihjeen korjata tämän suoraan paikkaan, josta virhe aiheutuu. Lisäsin if –lauseeseen ehdon, jossa se hyväksyy myös null –arvon. Asialle asetetaan vastaava tässä samassa funktiossa, mutta siihen pisteeseen ei vielä päästy.

Arkkitehti kertoi samalla muutoksen asian siirtoon toiseen alueeseen, jos henkilöllä on eri postinumero, kun nykyisellä käsittelijällä. Tällä hetkellä toteutus toimii niin, että aina kun asia siirretään, sen tila asetetaan toimeenpanoon. Tämä ei ole toivottu tilanne jokaisessa

tapauksessa. Toteutusta haluttiin muuttaa niin, että asian tila pysyy aina samana, paitsi jos sen tilana on tarkastettavana, jolloin se asetetaan suoraan toimeenpanoon.

Kyselin päivän aikana uuden projektini tiimiltä, että onko kehitysympäristön siirto palvelimelle onnistunut. Sain vastauksen, ja harmikseni kuulemma VPN –yhteys on ollut hyvin epävakaa, jolloin siirto katkeilee. Hänen mukaan saattaa pahimmillaan kestää jopa monta päivää. Päätin tämän jälkeen katsoa läpi esitysmateriaalia uudesta projektista.

Päivän aikana korjasin myös päivämäärien validointia. Aikaisemmin validointi tarkisti vain, että päivämäärä on validi. Tämä tarkoittaa kuitenkin sitä, että vuosiluvuksi voidaan asettaa jopa kymmeniä tuhansia vuosia. Asetin Angularin direktiiviin tarkistuksen, joka katsoo, että vuosiluku on alle 9999 ja yli 0.

Lopuksi katsoin kollegan kanssa kahdestaan hakemuksesta ja sen ympäröivistä osista puuttuvia ruotsinnoksia. Kirjoitimme valmiit määritelmät teksteille, jotka jäävät odottamaan meidän ruotsintajan työtä.

Päivän aikana pääsin osittain tavoitteisiin, sillä yritin edistää kehitysympäristön asennusta ja sain kuitenkin monia ominaisuuksia korjattua. Sain myös hieman tutustuttua tulevaan työhön.

Keskiviikko 30.3.2016

Keskiviikoksi päätin olla asettamatta tavoitetta, koska en osannut aamulla sanoa yhtään mitä tulen tekemään päivän aikana. Yksikään tiketti ei ole minulla, eikä taulultakaan löydy vapaita tikettejä.

Päivän aluksi huomasin verkkosovelluksen hakemuksen tarkistuksella, että ensimmäisellä sivulla olevat asian lisätiedot otsikko ja tyhjä taulukko ovat näkyvissä, jos asialla ei ole lisätietoja lainkaan. Henkilön lisätiedot poistin jo aikaisemmin näkyvistä tällä ehdolla, joten päätin poistaa tämänkin.

Yön aikana eilen tekemämme muutokset on ajettu testipalvelimelle. Olimme päättäneet, että tänään testaajat ottavat hakemuksen tarkistuksen käsiinsä ja testaavat sitä rankalla kädellä. Ruvettiinkin saamaan nopealla tahdilla pitkin päivää korjattavaa ja muutettavaa hakemuksella.

Verkkosovelluksessa ohjeet ovat ikkuna- tai välilehtikohtaiset. Määrittelijä kuitenkin huomasi, että ohjesivu ei ymmärrä hakemuksen eri vaiheita. Ohjeet olisi hyvä saada myös

hakemuksen eri vaiheille, jotta niistä voitaisiin tehdä mahdollisimman hyvät käyttäjälle. Tämä saatiin korjattua erittäin yksinkertaisesti lisäämällä nykyisen askeleen sisältämä Angularin scope –muuttuja if –lauseeseen.

Monet verkkosovelluksen näkymistä voi avata joko koko ruudulle tai vain modaali –ikkunaan. Testaaja oli huomannut, että hakemuksen avaamalla modaali –ikkunassa, käyttäjä ei pysty painamaan seuraava –painiketta ensimmäisellä sivulla. Tämä ongelma muodostui, kun seuraava –painikkeen yhteyteen olin lisännyt dialogin, missä kysytään käyttäjältä, tahtooko hän ryhtyä hakemuksen käsittelijäksi. Dialogi ei toiminut modaali –ikkunalla. Tämäkin ongelma saatiin yksinkertaisella tavalla korjattua.

Näiden muutosten jälkeen saatiin isompi lista puutteista, joista lähes jokainen oli pieniä muutoksia. Esimerkiksi y-tunnuksen validaattori ei ymmärtänyt tilannetta, jossa tarkistusmerkki on nolla, uudelle henkilölle ei saisi tallentaa ulkomaista osoitetta, ja henkilön tili-numero ei näkynyt yhteenveto –sivulla. Saatiin kaikki ongelmitta korjattua.

Yksi isompi ongelma ilmeni korjatessa henkilön tyyppin kenttää. Korjaus koski sitä, että juridiselle henkilötyypille pystyi asettamaan luonnollisen henkilön lajin. Muutettiin toteutusta piilottamaan valintaruudusta väärät lajit, jos juridinen on valittuna. Tästä huomattiin ongelma, että jos juridinen henkilö muutetaan luonnolliseksi henkilöksi, käyttöliittymä ei päästä eteenpäin. Huomattiin, että etunimi kenttä on pakollinen, mutta ei kuitenkaan ilmoita asiasta käyttäjälle maalaamalla kenttää punaisella. Pitkän taistelun jälkeen päädyttiin tekemään ruma väliaikaiskorjaus – eli todennäköisesti ruma lopullinen korjaus. Lisättiin etunimi –kentälle ID ja haettiin tämän avulla ko. elementti DOM:ista ja lähetettiin sille blur –tapahtuma. Blur tapahtuma tarkoittaa tapahtumaa, jossa käyttäjä lähtee kentästä, joka triggeroi kentän validoinnin.

Loppupäivästä tein yksin yhden muutoksen, josta asiakas oli huomauttanut. Aikaisemmin olin lisännyt virheilmoituksen, joka ilmoittaa käyttäjälle, jos kyseinen hakemus on hylätty tai muodostettu jo asiaksi. Tämä on huono yhdistelmävirheilmoitus, koska nämä kaksi asiaa ovat käytännössä vastakohtia. Huomasin, että minulla on käytössä hakemuksen tila –muuttuja, joka sisältää tiedon siitä, onko hakemus hylätty, kesken vai valmis. Tätä käyttämällä loin erilliset virheviestit. Kirsikkana kakun päälle lisäsin 'hakemuksesta on muodostettu jo asia' –virheeseen linkin jo muodostettuun asiaan.

Päivän aikana en kuullut mitään toisen projektin kehitysympäristön tilanteesta. Tänään oli mukava tekemisen meininki toimistolla. Kello 14 kollega kasasi asiakkaalle paketin hyväksymistestausta varten. Tästä eteenpäin virheidenkorjaus jatkuu.

Torstai 31.3.2016

Torstain tavoitteeksi jälleen viimeistellä hakemukseen liittyviä toteutuksia. Tänään tiedän kuitenkin jo aamusta, millä hommilla aloittelen päivän.

Aamupäivästä lähdin ensimmäisenä korjaamaan eilen ilmennyttä ongelmaa, joka liittyy hakemuksen ja henkilörekisterin tietojen vertailuun, jossa ilmoitetaan käyttäjälle eroavista kentistä. Testaaja oli huomannut, että haettua ensimmäisen henkilön ja palatessa takaisin hakemaan samaan kohtaan uudelleen henkilöä – järjestelmä vertaa aikaisemman haetun henkilön tietoja uuteen. Järjestelmän kuuluisi aina verrata hakemuksen alkuperäisiä tietoja toisiinsa. Tämä korjautui yksinkertaisesti muuttamalla parametreja, joita asetan funktiolle.

Huomasin ympäristössäni ongelmia yrittäessäni käynnistää paikallista versiota asiointiportaalista. Ongelma johtuu siitä, että versionhallinnasta minulle tulee uudet päivitetty teemaan liittyvät tiedostot, jotka on rakennettu uuden Liferay –version päälle. Paikallisesti minulla on vielä siitä vanha versio. Liferay on avoimen lähdekoodin portaaliohjelmisto, joka on kirjoitettu Javalla. Sain ongelman korjattua kysymällä apua kollegalta, joka on tämän päivituksen tehnyt. Päätettiin vielä olla asentamatta minulle paikallisesti uutta versiota – muutettiin muutamaa konfiguraatiodiedostoa niin, että ne viittaavat vielä vanhaan Liferayn versioon.

Testaaja oli myös huomionnut asiointiportaalin puolella hakemuksen liitteet –osiossa puutteita. Siinä pyydettiin käyttäjää lisäämään liite, mutta ei selitetty lainkaan mitä liitteitä. Korjasin samalla hieman muotoilua, koska lisää liite –kenttä oli paljon enemmän vasemmalla päin ruudulla kuin muut kentät. Huomasin samalla, että asiakkaalle ilmoitetaan vasta virheviestissä, että liite kuuluu olla PDF –muodossa. Lisäsin kentän alle pienen viitteen vaadittuun tiedostomuotoon. Ongelmia tässä tuli tietenkin, onhan kuitenkin kyse Wicketistä. Viitatessani käännöksiin '.properties' –tiedostossa, jostain syystä Wicketissä täytyy labelin nimi kirjoittaa väliviivoilla, vaikka itse käännöksen määritelmässä sanat ovat eroteltu pisteillä. Tähän kuului myös tehdä viittaus Javan puolella.

Päivän aikana pidimme tänään viimeistä päivää viettävälle kollegalleni kakkukahvit. Siellä tuli käymään myös jokin aika sitten eläköitynyt projektipäällikkömme, joka myös viettää tänään virallisesti viimeistä päivänsä. Hän oli lahjana tiimille tuonut jokaiselle tiimiläiselle itse kutomansa villasukat. Tiimiläisiä on siis reilu tusina, hänellä on käynyt jopa työstä tämä lahja. Keskustelimme ja vaihdoin kuulumisia tiimin kesken. Hyvästelimme yhdessä vielä lähtevät kollegamme.

Loppupäivästä lähdin korjaamaan testaaajan huomaamaa vikaa. Henkilöä haettaessa rekisteristä, käyttäjä voi valita henkilön, jonka rooli on eri kuin vaadittu henkilön rooli tässä kohdassa. Tämä aiheuttaa virheen järjestelmään ja myös vältettävän ongelman käyttäjälle. Käyttäjä ei kuitenkaan kovin helposti pääse valitsemaan henkilöä, jolla on vääränlainen rooli, mutta se on kuitenkin mahdollista. Lisäsin haun sisältämään funktion AJAX –kutsuun virhetilanteissa palautuvan funktion, joka hakee koodistosta kaikki mahdolliset henkilön roolit ja valitsee sieltä tällä hetkellä vaaditun roolin perustuen hakemuksen henkilön rooliin. Tätä tietoa käyttäen tein uuden virheilmoitusviestin näytettäväksi käyttäjälle, joka ilmoittaa, että valittu henkilö ei ole vaadittua roolia. Virheessä lukee myös, mikä rooli on vaadittu. Tässä kohtaa minulla tuli pieni ongelma, koska roolit sisältävä koodisto –objekti sisältää kaksi eri kenttää suomenkieliselle ja ruotsinkieliselle roolin nimelle. Minun täytyi selvittää tähän lisäksi session –muuttujasta käyttäjän kieli ja tästä päätellä kumpaa muuttujaa haluan virheviestissä käyttää.

Päivän aikana portugalilaiset lähtivät yrittämään siirtää uuden projektin kehitysympäristöä uudella tavalla. He pakkasivat sen 63 –osaiseen .zip –tiedostoon ja osa kerrallaan siirtävät niitä palvelimelle. Tämä näyttää siltä, että se vihdoinkin onnistuisi. Sain päivän aikana siirrettyä noin kolmasosan tiedostoista omalle koneelle. Enempää he eivät olleet kerenneet siirtää vielä.

Viimeisenä korjasin vielä asiakkaan testauksessa ilmenneen kirjoitusvirheen virheviestissä. Päivän aikana pääsin hyvin tavoiteisiini, koska sain korjattua useita virheitä. Päivän päätteeksi hyvästelin vielä kollegan ja toivotin hänelle onnea uusiin haasteisiin.

Perjantai 1.4.2016

Perjantain tavoitteena päätin jatkaa eilistä tavoitetta – viimeistellä hakemuksen eri vaiheita.

Aamupäivästä huomasin, että testaaaja oli kommentoinut ongelmasta. Haettaessa henkilön tietoja rekisteristä, kaikki muut tiedot päivittyvät kentissä, mutta syntymäpäivä jää samaksi kuin aikaisemman henkilön tiedot. Seurasin ongelmaa debuggerilla rajapinnan puolella hakemuksen henkilön luokan konstruktoriin, jossa huomasin, että syntymä- ja kuolinpäiväksi virheellisesti asetetaan hakemuksesta tulleita tietoja. Korjasin ongelman muuttamalla tietolähteeksi juuri haetun henkilön tiedot.

Testaaaja oli huomannut myös ongelman, jossa henkilön syntymäaika ei tallennu tietokantaan, vaikka se on asetettu. Tähän ongelmaan löysin syyn myös rajapinnan puolelta. Löy-

sin tietoja käsitellessä if –lauseen, jossa oli väärä logiikka. Se toimi tällä hetkellä täysin päinvastoin kuin pitäisi. Toteutus toimi muutettuani sen oikein päin.

Edellisen korjauksen jälkeen huomasin uuden ongelman. Jos käyttäjän syntymäpäivä muuttuu haettaessa tietoja henkilörekisteristä, kentän reunaan silti ilmestyy päivämäärän valitsin. Tämän valitsimen kautta päivämäärää pystyy muokkaamaan, vaikka kenttä on poistettu käytöstä. Tämän korjausta jouduin hetken miettimään. Lopulta päädyin jälleen hieman rumempaan ratkaisuun ja päätin jQueryllä valita date pickerin valitsimen – painikkeen ja piilottaa sen käytöstä, jos tietyt ehdot täyttyvät. Toteutukseen täytyi lisätä vielä tämän päinvastainen toiminto, eli näyttää valitsin uudestaan, jos ehdot eivät täyty.

Seuraavaksi toteutin muutaman pienemmän käyttöliittymään liittyvän muutoksen. Asiakas oli viestissä halunnut eilen lisäämäni tekstin pois liitteiden päältä, jossa kerrottiin käyttäjälle, että hän voi syöttää enintään kymmenen liitettä. Keksimäni pääteksti kuitenkin jäi paikalleen.

Huomasin hakemuksen tarkistuksen puolella, että liitteiden avaa –linkkien taulukkosolut ovat jostain syystä eri kokoiset. Lisäsin näihin bootstrapin col –luokat määrittämään molemmille samat pituudet. Muistin muutoksia tehdessäni myös, että määrittelijä oli ohimennen sanonut huomanneensa, että lisäämäni velallisen lisätiedot – eli kaksi tekstikenttää yhdellä sivuista vaihtelevat välillä paikkaa. Tiesin heti, mistä tämä johtuu. Lisätiedot tuodaan rajapinnasta Javan Set –muuttujassa. Set –listamuuttujan ero List –muuttujaan on se, että List –muuttujassa jokaisella listan objektilla löytyy indeksi ja muuttujan järjestys pysyy aina samana. Set –muuttujassa taas ei ole näistä kumpaakaan. Tiedot ovat tallella, mutta järjestys muuttuu jokainen kerta, kun muuttuja luodaan tai sinne lisätään lisää tietoa. Tästä syystä välillä kentät olivat toisin päin. Korjasin ongelman lajittelemalla tulostuvat kentät lajin mukaan aakkosjärjestyksessä.

Loppupäivästä etsittiin yhdessä määrittelijän ja testaajan kanssa uusia ongelmia tietojen tallennuksessa. Luotiin uusia asioita vanhasta työpöytäsovelluksesta – joka suoraan sanottuna on kiviaikainen ja hyvin vaikea käyttää. Verrattiin näitä luotuja asioita hakemuksesta luotuihin asioihin. Ainoana virheenä huomattiin, että asioille ei asetettu määräaikaa eikä vanhentumispäivämäärää. Molemmat näistä ovat kuitenkin pakollisia tietoja. Luettiin käyttötapauksesta, kuinka näiden kuuluisi toimia ja toteutin ne näiden mukaisesti. Tämä vaati muutaman if –lauseen ja funktion, joka lisää tietyn määrän vuoria päivämäärään ja asettaa tuloksen luokkaan – joka lopulta menee kantaan.

Päivän aikana sain jo suurimman osan kehitysympäristöstäkin siirrettyä, mutta vieläkin on siirrot meneillään loppuun. Osia on jäljellä reilu kymmenen kappaletta. Tänään pääsin tavoitteisiini. En kuitenkaan saanut aikaiseksi niin paljon, kuin olisin toivonut. Tästä en voi kyllä itseäni syyttää. Tämä johtuu siitä, ettei juuri ollut tekemistä jäljellä.

Yhteenveto viikosta 13

Viikon aikana keskityttiin huomattavan paljon hakemuksen tarkistuksen sisältämiin virheisiin. Jotkut löydettyistä virheistä olivat kuitenkin kaukaa haettuja ja hyvin todennäköisesti tilanteita, joihin käyttäjä ei koskaan ajaudu. Olemme kuitenkin kehittämässä laadukasta virheetöntä järjestelmää, sen pitää kestää myös virheellisen käytön läpi.

Isoimpana ongelmana viikon aikana pidän uuden projektin kehitysympäristön siirtoa. Tarkoituksena on siis siirtää virtuaalinen kehitysympäristö Portugalista minun koneelle Suomeen. Minun piti tämän viikon alussa jo ruveta osallistumaan toisen projektin kehittämiseen, mutta en voinut ilman kehitysympäristöä. Ongelman aiheuttaa tietoturva. Kehitysympäristö sisältää asiakkaan arkaluontoista ja salassa pidettävää tietoa. Tähän ratkaisuna päätettiin siirtää ympäristö asiakkaan palvelimen tiedostojärjestelmän kautta. Palvelin sijaitsee Suomessa toisen yrityksen tiloissa, olemme molemmat sinne yhteydessä VPN – yhteyden kautta. Ongelmaksi koitui VPN –yhteyden epävakaus ja hitaus Portugalissa. Lopulta päätettiin pilkkoa yli 30 gigatavun tiedosto 63 osaan ja siirtää osa kerrallaan asiakkaan palvelimelle noin 100 kilotavua sekunnissa, josta minä siirrän osa kerrallaan koneelleni. Tähän mennessä olen saanut reilun kaksi kolmasosaa siirrettyä omalle koneelleni.

Jokaisessa asiakkuudemme projektissa käytämme versionhallintaan Gittiä. Git on erinomainen versionhallintatyökalu, jonka kehityksessä on suomalaista kädenjälkeä. Linuxin luoja Linus Torvalds kehitti Gitin viikonlopun aikana BitKeeperin tehtyä päätöksen lopettaa ilmaisen lisenssinsä. Gitin kehityksessä haluttiin tavoitella mm. nopeutta, yksinkertaisuutta ja vahvaa rinnakkaista kehitystä. Syntymästään saakka Git on onnistunut ylläpitämään kaikkia alkuperäisiä tavoitteita. Git on hyvin tehokas pienissä- sekä suurissa projekteissa sen mahtavan branch –ominaisuuksien avulla. (Chacon & Straub 2014, 5).

Gittiä käytetään yleisesti pitämällä 'master' –branchissa aina täysin toimiva versio. Masterista luodaan brancheja, jossa varsinainen kehitystyö tehdään. Kehitystyön jälkeen kehittäjä committaa muutoksensa omaan branchiin ja tämän jälkeen ne mergetään masteriin. Masteriin mergettäessä täytyy olla täysin varma, että toteutus on toimiva ja vakaa, sekä täysin valmis. (Chacon & Straub 2014, 126).

Olen tullut Gitin käyttöön onneksi tutuksi jo koulussa. Vaikka sen käyttäminen onkin yksinkertaista, välillä kuitenkin tapahtuu joitain vahinkoja ja tekemäsi työ on vaarassa kadota. Viikon aikana minulle sattui ongelma Gitin kanssa. Olin luonut uuden branchin, johon minun oli tarkoitus tehdä muutokseni. En ollut kuitenkaan muistanut vaihtaa branchia ennen muutoksieni tekemistä. Committia tehdessäni huomasin, että olen tekemässä sitä väärään branchiin. Päätin tehdä Gitillä stash –komennon, joka tallentaa muutokseni paikallisesti talteen myöhempää käyttöä varten. Seuraavaksi muutin branchia, otin omat muutokset stashista takaisin ja tein commitin. Tässä vaiheessa huomasin, että jostain syystä branchin vaihdos ei ollut toteutunut kunnolla, vaan olin Gitin 'detached head' –tilassa. Hetken mietittyä löysin committini, vaihdoin branchia onnistuneesti ja mergesin committini uuteen branchiin. Erittäin hyvä puoli Gitissä on mielestäni se, että sinun on hyvin vaikea kadottaa tekemääsi työtä.

Olen monesti aikaisemmin miettinyt, että kuinka pieniltä vaikuttavat ohjelmistoprojektit tai muutokset olemassa oleviin projekteihin tulevat aina niin kalliiksi, kun on kyse valtavasta projektista. Minulle on pikkuhiljaa selkeytynyt, kuinka vaikeata on kehittää erittäin laajaa järjestelmää rikkomatta muita osia. Olen myös huomannut, että testaukseen ja dokumentointiin menee huomattavan paljon aikaa.

Viikon aikana opin ongelmanratkontaa selvittämällä syitä ongelmille, joita testaajat olivat huomanneet hakemuksen tarkistuksessa. Tein myös ensimmäisiä kertoja Angularin direktiiveihin muutoksia. Direktiivit ovat HTML:n DOM elementin artibuteina kutsuttavaa toiminnallisuutta kyseiselle elementille. Direktiiviin voi määritellä lähes kaikenlaista erilaista toimintaa, mikä liittyy esimerkiksi tiedon näyttämiseen. (AngularJS Documentation 2016).

Verkkosovelluksessamme direktiivejä käytetään esimerkiksi erilaisten kenttien, kuten päivämäärän, henkilö- ja y-tunnuksen validointiin. Viikon aikana tein lisäyksiä mm. päivämäärän validointiin liittyen. Edellisinä viikkoina tein parikoodauksena henkilö- ja y-tunnukselle validoinnit.

3.10 Seurantaviikko 14

Maanantai 4.4.2016

Vielä täksi päiväksi minulla riittää töitä tässä projektissa. Loppuviikosta minun täytyy viimeistään ruveta selvittämään lisää hommia uudesta projektista. Maanantain tavoitteeksi asetin vielä viimeistellä hakemusta ja päästä eroon viimeisistäkin ongelmista.

Viime viikolta tiketille oli jäänyt vielä yksi ongelma, johon tarvitsen hieman määrittelyä arkkitehdilta. Ongelmana on asian näkymällä oleva painike, josta saadaan tulostettua hakemuksen mukaiset tiedot PDF –tiedostoon. Tällä hetkellä asialle kirjatut asiat eivät näy tulosteella. Arkkitehti kävi kanssani prosessin läpi ja kertoi, että järjestelmän kuuluisi luoda asialle hakemus pohja joka tapauksessa, kun asia siirretään eteenpäin. Tästä syystä hän ehdotti, että poistetaan alkuperäinen hakemus sen jälkeen, kun siitä on tallennettu asia. Jotta poiston jälkeen päästään vielä käsiksi alkuperäisen hakemuksen tietoihin, luodaan uusi liitteen laji hakemuksen liitteisiin lisättävään alkuperäisen hakemuksen tietoihin. Hakemuksen alkuperäiset tiedot menevät jo ennestään hakemusten liitelistaan, jonka toteutin joitakin viikkoja takaperin.

Piirsin tästä itselleni vielä selkeyttääkseni prosessin vaiheita pienimuotoisen prosessikaavion. Lähdin sen pohjalta toteuttamaan. Lisäsin liitteille uuden lajin nimeltä alkuperäisen hakemuksen mukaiset tiedot. Tein tähän samalla Flywaylle migraatitiedoston tietokantaa varten. Seuraavaksi muutin hakemuksen tietojen tallennuksesta ja –latauksesta liitetiedoston lajiksi juuri luomani uuden liitteen lajin. Yllätyin tämän jälkeen, että toteutus toimi juuri niin kuin pitää. Oletin, että joudun tekemään vielä joitain muita pienempiä korjauksia.

Seuraavaksi pidimme tiimipalaverin pitkästä ajasta. Palaverin aikana täytimme yrityksen jokaiselle henkilökunnan jäsenelle lähetettyä vuosittaista strategiakyselyä. Kyselyssä kysyttiin omia mielipiteitä ja parannuksen aiheita yrityksen toimintaan ja nykyisiin strategioihin, ja kuinka nämä mielestäni toteutuvat. Palaverissa käytiin läpi myös nopeasti tiimin projektien tilanteet.

Ennen lounasta sain viimein siirrettyä viimeiset uuden projektin kehitysympäristön osat koneelleni ja purettua paketit. En vielä lähtenyt käynnistelemaan virtuaalikonetta, vaan keskityin viimeisiin korjauksiin vielä tänään.

Lounaan jälkeen keskustelin testajaansa kanssa. Hän oli huomannut muutamia ongelmia lisää tyhjiin työpinoon. Tallentaessa uutta henkilöä tietokantaan, henkilön tiedoissa kuuluu olla joko henkilötunnus tai syntymäaika. Huomattiin, että tietokannassa saattaa tietyn eräajon jäljiltä löytyä useita henkilöitä, joilla ei ole kumpaakaan tietoa. Huomattiin myös, että tietokantaan voi tallentaa henkilöitä, joiden syntymäaika on vain vuosiluku. Ongelma syntyi siinä, kun henkilön tietoja haettiin henkilörekisteristä. Järjestelmä ei osannut käsitellä sitä, jos syntymäpäivä on vain nelinumeroinen vuosi tai täysin tyhjä. Lisäsin Javan Hibernate luokan konstruktoriin if-lauseet, jossa selvitetään, onko rekisteristä tuleva syntymäaika validi päivämäärä, jos ei ole, niin tarkistetaan, onko se neljän merkin pituinen. Tässä vaiheessa minulle tuli ongelma, kun koodi ei koskaan mennyt pelkän vuosiluvun kohdalla if-lauseeseen sisään, vaikka se oli vain neljä numeroa pitkä. Huomasin, että jostain syystä stringin perässä on läjä välilyöntejä. Poistin nämä Javan trim()-funktiolla, jonka jälkeen toteutus toimi. Ongelmaksi muodostui myös se, että hakemuksen henkilöllä syntymäpäivä on meidän omassa Pvm-luokassa, mutta henkilörekisterin henkilön syntymäpäivä on stringinä. Tähän helppona ratkaisuna päätettiin arkkitehdin kanssa, että muunnetaan tämä string väliaikaisesti Pvm-muuttujaksi, ja asetetaan vaan kuukausi ja päivämäärä ykköseksi. Tähän liittyen päätettiin myös, että hakemuksella ei sittenkään tule pystymään tallentamaan henkilöä antaen ainoastaan sille syntymävuoden, poistin tämän toiminnallisuuden.

Päivän ensimmäisen homman yhteydessä huomasin itse uuden ongelman. Nyt kun hakemus poistetaan luodessa asiaa ja avataan uudelleen sama hakemus, vaikka sitä ei ole enää olemassa, järjestelmä heittää virheen. Virhe tulee rajapinnan puolelta asti, joten päätin asettaa rajapinnan puolen yhden rivin tietokantakutsun try-catch-metodin sisään, joka estää järjestelmää kaatumasta virheen tullen – tässä tapauksessa siitä syystä, että haettua hakemuksen avainta ei löydy enää. Catch-osaan, johon tullaan virheen syntyessä, käskin sovellusta palauttamaan arvon null. Angularin puolella lisäsin selvityksen, että jos hakemuksen avain on 'undefined', heitetään virheviesti ja palataan edelliselle sivulle, mistä käyttäjä onkaan sitten tullut.

Päivän aikana sain paljon asioita tehtyä ja olen tyytyväinen päivän etenemiseen. Lähes kaikki päivän hommat tuntuivat jo rutiinilta ja tiesin lähes heti, kuinka aion ne tehdä. Päivän lopuksi minulle jäi sellainen tunne, että hakemus on siinä kunnossa, että sitä voi jo sanoa valmiiksi. Siirsin siihen liittyvän tiketin vielä lähdeittäessäni testajan työpinoon.

Tiistai 5.4.2016

Tiistain päivän tavoitteeksi mietin aamulla uuden projektin ympäristöön tutustumista ja ehkä jotain toteutustakin, mutta heti dailyssä sain kuulla, että minulle on luvassa enemmän hommia. Joten tavoitteena tälle päivälle lopulta on vieläkin hakemuksen viimeistelyä ja uuden tiketin aloittelua.

Aamupäivästä huomasin, että arkkitehti oli kommentoinut code reviewiin laittamiani muutoksia eiliseltä. Olin käyttänyt päivämäärän konstruktoria väärällä tavalla ja Angularin puolella if –lauseessa pieni muotoiluvirhe. Hän mietti myös toteutukseni luettavuuden epäselvyyttä, joten lisäsin kommentin koodin sekaan selkeyttämään sitä.

Testaaja oli huomannut vielä yhden vian hakemuksen tallentamisessa asiaksi. Jos vahvistamispäivämäärää muokkaa hakemuksen tarkistuksessa, muokattu päivämäärä ei päädy asialle saakka. Huomasin, että tallentaessa uutta asiaa, tällä hetkellä käytetään hakemuksen alkuperäisiä tietoja. Muokkasin toteutusta lisäämällä luokkaan uuden konstruktorin, joka täyttää luokan muuttujat hakemuksen tarkistukselta tulleilla tiedoilla. Tämä virhe on saattanut estää muitakin tietoja päivittymästä, mutta nyt niidenkin pitäisi toimia.

Keskustelin arkkitehdin kanssa ja sivulauseessa mainitsin, että hakemusten lista ei päivity, kun hakemus hylätään tai muodostetaan asiaksi, vaan vasta kun käyttäjä päivittää listan. Hän sanoi, että sen kuuluisi päivittyä. Lähdin selvittämään, miksi se ei toimi. Huomasin, että hakemusten toisessa listauksessa tämä ominaisuus toimii, mutta toisessa ei. Tähän toiseen sitä ei ole vielä edes toteutettu. Tässä listauksessa on kaksi eri välilehteä, joita molempia käsitellään samassa muuttujassa, joka vaikeuttaa toteutusta. Sain lopulta pitkän testailun jälkeen tehtyä kaksinkertaisen loopin, joka looppaa molemmista välilehdistä läpi kaikki hakemukset ja selvittää täsmääkö asiaksi muodostettava hakemus jotain listan hakemusta. Jos täsmää, se poistetaan listasta. Sain nämä kaikki muutokset tänään code reviewin läpi mergettyä masteriin.

Lounaan jälkeen keskustelin projektipäällikön kanssa seuraavasta tiketistä. Kyseessä on asiointiportaalin uuteen teemaan liittyviä muutoksia. Lähden toteuttamaan linkkiä, jota painamalla aukeaa eri hakemuksiin liittyvät ohjeet modaali-ikkunaan. Aikaisemmin ohjeet löytyivät eri hakemusten oikeasta laidasta. Tämä ominaisuus on puoliiksi jo toteutettu toiseen projektiin, joten voin vähän ottaa mallia sieltä. Projektit käyttävät samaa Liferayn portaalia.

Tähän ei siis enää riittänyt viime viikolla tekemäni väliaikaiset muutokset uuden Liferay – version takia, vaan joudun päivittämään itselleni paikallisesti myös tämän uuden version ja sen mukana uuden teeman. Kollega, joka oli asentanut tämän jo muutamalle testipalvelimelle, tuli auttamaan minunkin asennusta. Asennus oli kohtuullisen helppo. Aluksi hain hänen tekemän asennuspaketin, joka minun täytyi vain kopioida oikeaan kansioon. Tämän jälkeen opin uutta – koneellani pyörii paikallisesti MySQL –tietokanta, joka sisältää Liferayn vaatimaa dataa. Minun täytyi tehdä sinne uusi tietokantataulu ja tuoda uudet tiedot sinne tiedostosta, ja lisätä Liferayn käyttäjälle oikeudet. Tämän jälkeen lisäsin Tomcatin startup –muuttujiin muutaman lisäyksen, ajoin Mavenilla .war –paketin uusimmasta portaalista, kopioin Tomcatin konfigurointitiedostot aikaisemmasta Tomcatista ja käynnistin sen.

Asennukseen meni lopulta yllättävän paljon aikaa, enkä kerennyt päivän aikana lähteä edistämään uutta toteutusta. Päivän aikana pääsin tavoitteisiini, koska sain hakemuksen kaikki raportoidut ongelmat korjattua ja luokittelen ympäristön konfiguroinnin uuden tiketin aloitteluksi. Opin tänään myös uutta ympäristöön ja Liferayhin liittyen.

Keskiviikko 6.4.2016

Keskiviikon tavoitteeksi asetin aloitella ja päästä eteenpäin uudella toteutuksella. Eilen sain kehitysympäristön siihen pisteeseen, että voin lähteä toteuttamaan.

Aamupäivästä lähdin heti miettimään, kuinka saan nykyisen ohjeet sisältävän div – elementin sisällön näkymään linkistä avautuvan modaali-ikkunan sisään. Katsoin hieman mallia toisesta projektista ja sovelsin siihen myös vanhaa toteutusta. Sain äkkiä linkin oikeaan paikkaan ja sitä painamalla avautumaan ikkunan. Ensimmäiseksi ongelmaksi muodostui itse tekstien saaminen ikkunan sisään. Aikaisemmassa toteutuksessa oli linkki, josta ohjeet avataan uuteen selainikkunaan – päätin käyttää samaa toteutusta. Yllätyin, kun toteutus toimi ilman sen suurempia mutkia vanhalla toteutuksella. Ainoana ongelmana linkissä on se, että se avaa modaali-ikkunan lisäksi uuden selaimen ikkunan, jossa on samat ohjeet. Tähän löysin ratkaisun Wicketin komponentin luontifunktiosta, siellä määriteltiin linkkiä painettaessa avautuva ikkuna ja sen määrittelyt. Poistin nämä rivit koodista, jonka jälkeen toteutus toimi niin kuin pitää.

Tiketissä oli mainittuna tämän toteutuksen lisäksi uuden teeman ja Liferayn version aiheuttamat käyttöliittymän viat ja teeman viimeistely. Valitettavasti en pääse itse teemaan käsiksi, mainitsin asiasta myös dailyssä, johon arkkitehdin ja toisen kollegan kanssa päätettiin tehdä uusi toteutustiketti. Tiketille listaan kaikki löytämäni ongelmat ja parannukset

teemaan liittyen. Tähän kulutin muutaman seuraavan tunnin ja löysin lähes kymmenen korjattavaa kohtaa, jotka vaikuttavat yleisesti teemaan. Esimerkiksi monien erilaisten kenttien labelin ja tekstin korkeus ei täsmännyt ja muutamalle elementille oli jostain syystä tullut ylimääräistä tilaa. Kirjasin kaikki löydökseni englanniksi tiketille portugalilaisen kollegan korjattavaksi.

Seuraavaksi rupesin korjaamaan samalla selailulla löytämiäni ongelmia itse käyttöliittymän puolelta. Isoimpana näistä hakemusten uusi progress –palkki näkyi välillä formin sisällä ja välillä siellä missä kuuluukin – formin yläpuolella. Tähän ongelmaan en hetkeen keksinyt ratkaisua, enkä lopultakaan keksinyt mitään järkevää ratkaisua. Päätelin ongelman syyn vertaamalla näkymää, jossa se toimi ja näkymää, jossa se ei toiminut. Ainoana erona näissä näkymissä on ylimääräinen tekstikenttä nimeltä caption formin yläpuolella kertomassa käyttäjälle, mitä tähdellä merkityt kentät tarkoittavat. Tämä kenttä näytettiin vain joillain näkymistä, joten päädyin lisäämään niihin näkymiin, joissa sitä ei näy pelkän tyhjän elementin. Tähän kului yllättävän paljon aikaa Wicketin syystä. Tämä teksti tulee tiedostosta, joka on jokaisen hakemuksen vaiheen yhteisesti käytävä tiedosto, ns. 'parent'. Tästä syystä toteutus oli hankalampaa, kuin olettaisi.

Nämä kaksi muutosta veivät suurimman osan päivästäni, koska samat muutokset piti tehdä asiointiportaalin jokaiselle Liferayn portletille. Näistä jokainen on erilainen, myös parentin kautta lisätyn tekstin ehdot ovat erilaiset.

Aamupäivästä kollega oli huomannut, että todistuksen tilaamisen maksun paluulinkki ei toiminut uuden Liferayn version kanssa niin kuin pitää. Tästä syystä todistusta ei voinut lainkaan tilata. Toinen kollega kerkesikin jo selvittää tähän syyn, jonka minä lopulta korjasin. Maksujärjestelmälle lähetetään maksutietojen mukana paluulinkit, joihin järjestelmä ohjaa käyttäjän maksun onnistuessa ja epäonnistuessa. Näiden paluulinkkien eteen uusi Liferayn versio oli opetellut tulostamaan itse palvelimen nimen. Aikaisemmin jouduimme hakemaan palvelimen nimen erikseen linkille, joten ennen korjaustani paluulinkissä oli palvelimen osoite kahteen kertaan.

Loppupäivästä korjasin myös ongelman, joka oli tullut asiakkaalta hyväksymistestauksesta. Verkkosovelluksen asian saatavan lajin alavetovalikossa oli aikaisemmin ollut erillinen kenttä nimeltä 'Saaminen'. Tämä kenttä on jossain vaiheessa sieltä kadonnut. Tähän ongelmaan selvisi syy selaamalla Gitin vanhoja committeja tälle tiedostolle. Löysin viime vuoden syyskuulta muutoksia, jossa kollega oli refaktoroinut koodia ja samalla muuttanut funktion nimeä, mutta unohtanut muuttaa funktiokutsun nimeä. Tästä syystä funktio ei korvannut valikosta yhden stringin nimeä toiseksi.

Päivän aikana sain hyvin paljon aikaiseksi, on mukavaa pitkästä ajasta tehdä käyttöliittymäpuolelle muutoksia. Vaikka en päässytäkään muokkaamaan teeman määrittäjiä itse. Pääsin tavoitteisiini ja sain jopa tiketin melkein valmiiksi. Riippuen tietenkin pitääkö arkki-tehti ratkaisuistani esimerkiksi progress-palkin korjauksessa. Käyttöliittymällä riittää kyllä vielä pientä viilattavaa huomiseksi.

Torstai 7.4.2016

Torstain tavoitteeksi asetin saada valmiiksi asiointiportaalin vaatimat käyttöliittymän korjaukset ja siihen liittyvät teeman muutosten raportointi eteenpäin. Enempää hommia minulla ei ole vielä tiedossa. Uskon, että näissä minulla ei kestä koko päivää.

Aamupäivästä lähdin jatkamaan käyttöliittymän muutoksia. Ensimmäiseksi lähdin ratkaisemaan ongelmaa, jossa progress-palkista jää näkyviin tyhjä palkki, kun hakemus on valmis. Päätin piilottaa sen askelineen kokonaan, kun hakemus valmistuu. Jokaisessa eri hakemuksessa oli hieman eri tavalla, mutta kuitenkin toteutettu muuttuja, joka sisältää tiedon siitä, onko hakemus valmistunut. Käytin tätä muuttujaa progress-palkin setVisibility-funktiossa.

Samalla kun korjasin edellistä korjausta, tajusin paremman tavan eiliselle ratkaisulleni, jossa lisäsin puuttuvan tyhjän tekstikentän hakemuksen yläpuolelle, jonka puuttuminen rikkoi näkymää. Olin eilen tehnyt asettanut tämän uuden komponentin setVisibility – funktioon päinvastaisen arvon, kuin caption tekstikentässä, jossa on sisältöä. Muutin toteutuksen yksinkertaisesti vain selvittämään, onko caption tekstikenttä näkyvässä, jonka perusteella joko näytän tai en näytä sen sijaan placeholderia.

Muita aamupäivän aikana tekemiäni korjauksia olivat ruotsinnosten puuttumiset, otsikoiden muotoilua ja muita pienempiä muutoksia. Sain päivän aikana kaikki nämä mergettyä masteriin ja sitä myötä huomiseksi testattavaksi.

Minulle tuli mieleen aamupäivällä myös, että alkuperäisessä suunnitelmassa hakemuksella oli kenttien perässä kysymysmerkkejä tarvittaessa. Näiden päälle, kun käyttäjä vie cursorinsa, ruudulle ilmestyy tooltipin tyyppinen pieni ohje selittäen tiettyyn kenttään liittyviä asioita. Lopulliseen versioon tämä ominaisuus ei ole täysin ehjänä selvinnyt. Sen takia, että tekstejä ei pysty Liferayn puolelta tuoda tooltippeihin, niiden sijaan toteutettiin saman tyyppinen ohje modaali-ikkuna. Rupesin ihmettelemään, että miksi näihin ei voida tuoda tekstejä samalla tavalla, kuin muihinkin kenttiin. Nämä tekstit kuitenkin ovat hyvin lyhyitä ja ytimekkäitä. Tällä tavalla voitaisiin helposti toteuttaa paljon hienompi ratkaisu. Keskus-

telin asiasta arkkitehdin kanssa. Hän oli samaa mieltä, että tämä ratkaisu olisi parempi, mutta lopulta päädyttiin siihen, että ei lähdetä toteuttamaan, koska siitä olisi liian paljon uutta selvitettävää asiakkaalta – ollaan kuitenkin niin lähellä tuotantoon siirtoa.

Iltapäivästä keskustelin projektipäällikön kanssa seuraavista tehtävistäni. Mainitsin jälleen, että voisin jo siirtyä tekemään uutta projektia, mutta minua haluttiin vielä käyttää tähän projektiin. Tämän projektin tiimi on kutistunut päivä päivältä, eilen lähti ainoa minun lisäksi oleva toteuttaja ja tänään lähti yksi määrittelijöistä. Huomenna tulen olemaan lähes ainoa työntekijä projektipäällikön lisäksi, joka edistää vielä tätä projektia. Keskustelun jälkeen olin jälleen selvillä, mitä tulen tekemään. Rupean jatkamaan järjestelmätestausta. Järjestelmätestauksessa on kyse suuresta prosessista, jossa testataan järjestelmän prosessit läpi alusta loppuun käyttäen uusia ja vanhoja ominaisuuksia, joihin version muutokset ovat vaikuttaneet.

Testaukseen minulle tuli aluksi tulosteiden testaus. Kävin läpi muutaman tulosteen, johon kummallekaan ei tullut oletettua tulosta. Tulosteella kuuluisi näkyä henkilön asioista lista, mutta sieltä puuttui ainakin tämä testauksessa aikaisemmin luotu asia. Tähän saattaa olla syynä käyttötapauksesta löytyvä ehto, johon kyseiset asiat saattavat sopia. Kolmannen kohdalla minulle tuli sovelluksessa virhe. Tässä vaiheessa rupesin epäilemään, että minun ympäristössä on jotain vikaa. Kokeilin kollegan koneella samoilla parametreilla ja siellä tuloste tulostuu ongelmitta. Tästä sainkin loppupäiväksi hommaa. Kokeilin kasata uudelleen koko järjestelmän Mavenilla ja käynnistellä asioita uudelleen tuloksetta. Loppupäivästä päätin, etten lähde jatkamaan edes selvittelyä. Päätin, että huomenna pyydän kollegalta työpöytä clientin, jolla kaikki toimii hänen koneellaan. Tämä client keskustelee suoraan testipalvelimen kanssa.

Päivän aikana sain toteutukset valmiiksi ja huomenna testaukseen. Pääsin siis tavoitteisiini. Hieman harmittelen, etten päässyt tekemään paljon hienompia tooltippejä hakemuksille, ehkä vielä joskus.

Perjantai 8.4.2016

Perjantain tavoitteeksi asetin saada oma osuuteni järjestelmätestauksesta valmiiksi ja korjata mahdolliset ilmestyvät käyttöliittymän uudet virheet.

Aamupäivästä keskustelin projektipäällikön kanssa siitä, että haluan koneelleni työpöytä clientin. Ihmeteltiin yhdessä, miksi minulla ei sitä ole kolmen kuukauden työskentelyn jälkeen. Hän antoi minulle linkin verkkolevyille, josta löydän asennuspaketin.

Lähdin heti jatkamaan järjestelmätestausta, mutta aloitan kuitenkin alusta. En luota oman kehitysversioni antamiin tuloksiin. Syystäkin, sillä ensimmäisen henkilön tulosteen kohdalla huomasin jo ongelman. Järjestelmä antoi virheen, että raporttia ei voida tulostaa. Kirjautuin palvelimelle ja menin etsimään lokista vialle tarkempaa syytä. Löysin vialle virhepinon lokin perältä. En itse tiennyt yhtään, mistä aloittaa tämän selvittäminen, joten delegoin sen heti vanhemmalle kehittäjälle ja jatkoin itse testausta.

Vanhempi kehittäjä vastasi pian, että törmäsin nyt melko vakavaan ongelmaan. Virhe johtui siitä, että henkilölle ei ole lainkaan asetettu tallentaessa vireilläolotietoja. Tämä on vakava virhe ja aiheuttaa monissa eri toiminnoissa virheen. Vika löytyi verkkosovelluksen uuden asian lisäyksestä, jossa on unohdettu luoda henkilölle vireilläolo.

Jatkettaessa testausta viiden henkilön asioista, vain yhdellä henkilöllä näkyi järjestelmätestiä varten luoma asia tulosteellansa. En löytänyt käyttötapausta tälle selitystä, ihmettelin asiaa myös pidemmän aikaa projektipäällikön kanssa. Päätettiin, että hän soittaa kollegalle, joka saattaisi osata sanoa ulkomuistista, miten tämän kuuluisi toimia. Saatiin vastaus, että jos henkilölle on lähetetty asiasta maksukehotus, ja sen eräpäivä on tulevaisuudessa, henkilön kyseistä asiaa ei näytetä tulosteella. Ilmoitin, että tämän kuuluisi sitten myös lukea käyttötapauksessa. Vahvistin tämän vielä vaihtamalla henkilön maksukehotuksen eräpäivää tietokannassa menneisyyteen, jonka jälkeen tulosteelle ilmestyi kyseinen asia.

Samalla kun kirjasin löytämiäni ongelmia wikiin, huomasin samassa paikassa eilen toisella testaajalla ilmenneen virheen. Ihmettelin, että asiaa ei keskusteltu dailyssä tai missään muuallakaan sen löytymisen jälkeen. Eilen testaajalla oli eräajon aikana ilmennyt yhden postitettavan tulosteen kohdalla virhe. Tämä virhe korjattiin, mutta testaaja huomasi, että tämän yhden virheen takia mitään muita tulosteita ei muodostunut. Jokaiselle tulosteelle oli kuitenkin tietokantaan päivittynyt tieto, että se on tulostettu. Päätelin tästä, että tässä hän voi tulla aika vakavakin virhe. Tällä hetkellä siis yli 5000 kirjettä olisi ollut lähtemättä postiin sen takia, että yhdessä tulosteessa tuli virhe – ja järjestelmä on sitä mieltä, että kirjeet ovat postitettu. Aloitin keskustelun virheestä, minun kanssa oltiin samaa mieltä, että tässä saattaa tulla isojakin ongelmia. Päätettiin kuitenkin, että ei korjata vielä tähän versioon, vaan luotetaan, että virheitä ei tule sattumaan.

Iltapäivästä päivitettiin uusi teeman kehitysversio testipalvelimelle ja huomattiin muutama korjattava tämän lisää. Esimerkiksi lisää liite –painike oli siirtynyt ulos lomakkeesta. Sain myös viestiä portugalilaiselta kehittäjältä, että hän tarvitsee lisää määrittystä minun eilen

raportoimiin ongelmiin. Todettiin, että hän ei pääse mitenkään testaamaan näitä ongelmia, koska ne ovat toisen projektin puolella. Päätettiin, että minä toteutan ne hänen puolesta. Samalla, kun nyt tuon teeman paketin sain kehitettäväkseni, korjasin sieltä myös muutamman muun vian, mitä minulla tuli vastaan. Korjasin myös käyttöliittymän rakenteesta muutamia pieniä ongelmia.

Päivän aikana pääsin tavoitteisiini, sillä sain oman osuuteni testattua järjestelmätestauksesta. Onneksi teemme näinkin laajaa testausta, koska jos tällainen vika olisi selvinnyt tuotantoon asti, siitä olisi syntynyt erittäin iso ongelma. Sain myös käyttöliittymän puolta hiottua niin, että se on jo lähes virheetön.

Yhteenveto viikosta 14

Viikon alussa toivoin jo pääseväni seuraavan projektin pariin, mutta yllättäen minulle ropisi viikon aikana lisää hommia. Kunnan kehitystyöt sain valmiiksi jo alkuvuikosta. Pääsin väliin löytämään virheitä toteutuksista ja lopuksi vielä hiomaan teemaa, jota viikon aikana jo kerkesin harmittelemaan, etten pääse tekemään. Itse pidin järjestelmätestausta lähes turhana, kun sitä suunniteltiin. Näitä virheitä katsoessa se on kaikkea muuta kuin turhaa. Järjestelmä on niin riippuvainen oikeista tiedoista, että jos tieto puuttuu tai toisessa tapauksessa, on liian pitkä – jotain hyvin pahaa tapahtuu. Ilman järjestelmätestausta, jossa käydään käytännössä järjestelmään tallennetun asian koko elinkaari läpi, tuotantoon olisi todennäköisesti päässyt muutama isompi virhe.

Asiointiportaalin kehitys hankaloitui paljon uuden Liferay –versio myötä. Aikaisemmin pystyin sivun rakenteeseen pienen muutoksen tehtyäni ajamaan vain yksinkertaisen tekemäni skriptin, joka pakkaa portaalista uuden .war –paketin ja siirtää sen Tomcatin deploy –kansioon. Tämän jälkeen muutokset ovat nähtävissä hetken kuluttua. Uuden version myötä minun joutuu tämän prosessin lisäksi kopioida oikein määritelty web.xml –tiedosto korvaamaan toista ja käynnistämään vielä Tomcat kokonaan uudelleen. Tämä vie parhaimmillaan muutaman minuutin työajasta, vaikka lisäsin skriptiin tuonkin tiedoston siirron. .war –paketti on Mavenin luoma asennuspaketti projektista, jonka pystyy asentamaan helposti verkkopalvelimelle.

Olen viikkojen aikana puhunut paljon virtuaalisista kehitysympäristöistä ja niiden tuottamista ongelmista. Projektissa on ollut myös ongelmia, joissa kaikki toimii ongelmitta paikallisessa ympäristössä, mutta ei taas toimi jollain testipalvelimella, koska jokin konfiguraatitiedosto sisältää yhden kirjoitusvirheen. Näihin kaikkiin ongelmiin on ratkaisu, jota valitettavasti ei ole vielä meillä käytössä. Docker on vähän kuin hyvin kevyt virtuaalikone.

Dockerin tavoitteena on helpottaa kehittäjien elämää tarjoamalla kehittäjille mahdollisuuden luoda 'kontteja', joihin voidaan asentaa jokin tietty ympäristö oikeilla konfiguraatioilla. Tätä konttia voidaan käyttää jokaisen kehittäjän omalla koneella parantaakseen sen tehokkuutta verrattuna virtuaalikoneeseen, joka käytännössä vie enintään puolet koneen tehoista. Samalla helpottuu uusien konfiguraatioiden päivitys, sama konfiguraatio päivittyy jokaiselle yhden komennon jälkeen. Dockerin avulla kehittäjät voivat lyhentää tuotantoon siirtoon kestävää aikaa jopa olemattomiin, sillä Docker pitää huolta siitä, että sovelluksen tuotantoon siirrossakaan ei tule mitään ympäristöön liittyviä ongelmia. (Anderson 2015).

Toinen työtä helpottava asia olisi jatkuva integraatio, jonka suomensin suoraan sanasta 'continuous integration', eli CI. CI on ratkaisu käytännössä korvata testajat kehitystiimistä parhaimmillaan jopa kokonaan, mutta se vaatii kehittäjiltä paljon enemmän työtä. Kehittäjien pitää tehdä jokaiselle isommalle funktiolle automaattinen testi joko JUnitilla tai vastaavalla, joka testaa sen oikean toiminnan. Kehittäjän tehdessä pushia masteriin, CI ajaa jokaisen testin ja huomauttaa heti, jos jokin menee rikki. CI:n lopullinen tavoite on antaa tiimille mahdollisuus tehdä jatkuvaa version integraatiota suoraan tuotantoon. Jos järjestelmän testikattavuus on riittävä, voidaan sovellukseen luottaa siinä vaiheessa, kun testit menevät kaikki läpi. (Atlassian 2016).

Yhdistämällä automaattitestit, CI:n ja Dockerin – voitaisiin jokaista tuotetta viedä muutosmuutokselta suoraan tuotantoon, ilman minkäänlaista viivettä. Tämä on ketterää kehitystä parhaimmillaan. Tämän asiakkuuden projekteissa tästä tilanteesta ollaan vielä hyvin kaukana, mutta ehkä joku päivä. Docker on tullut jo useampaan otteeseen puheeksi ja on ollutkin jo käytössä muutamassa projektissa tietyllä tasolla, mutta automaattitesteistä ja CI:tä todennäköisesti ei tulla näkemään ainakaan nykyisissä projekteissa. Menisi kuukausia, jotta saataisiin kirjoitettua testit jälkikäteen projekteille.

Viimeinen viikkoni jonka kirjoitan päiväkirjaan, meni vielä saman projektin hommissa. Ensi viikolla todennäköisimmin pääsen tekemään hommia uuteen projektiin. Odottelen innolla uusia töitä!

4 Pohdinta ja päätelmät

Opinnäytetyön kirjoittaminen on sujunut seurantaviikkojen aikana hyvin. Loppuvaiheilla tämä alkoi jo tuntua rutiinilta. Tässä vaiheessa kuitenkin hartioiltani lähti iso paino. Etenkin huomattessani, ettei minun tarvitse enää työpäivän aikana kirjata muistiinpanoja eikä työpäivän jälkeen kuluttaa vapaa-aikaani työn kirjoittamiseen.

Työn kirjoittamisen aikana kerkesi tapahtua paljon muutakin, kun pelkkää ohjelmointia. Tiimistämme lähti jäseniä, toinen eläkkeelle ja toinen toisen yrityksen palvelukseen. Jälkimmäistä harmittelen edelleen. Hänen kanssaan otin ensiaskeleet yrityksessä ja tein yhdessä useamman ominaisuuden – työympäristö ei ole ollut sama hänen lähdön jälkeen.

Ennen viikkoseurannan aloittamista olin ollut orientoitumassa tässä työssä vasta kuukauden ajan, muun muassa tehden jokaiselle työntekijälle tarkoitettuja orientoivia tehtäviä ja käyttötapauksen linkkien päivittämistä. Varsinaiseen työnkuvan mukaiseen työhön pääsin vasta aloitettuani kirjoittamaan työtä. En myöskään ollut aikaisemmin missään ohjelmistokehittäjän työssä. Taitoinani ainoastaan ne, mitä kouluista sain irti – ja mitä itse on tullut opittua.

Peilattuna lähtötilanteeseen olen kehittynyt enemmän, kuin pystyin kuvittelemaankaan. Ajattelin, että tässä vaiheessa rupeaisin pikkuhiljaa kykenemään aidosti itsenäiseen työhön ja ymmärtämään järjestelmänkin toimintaa. Olen kuitenkin ylittänyt tuon kynnyksen jo reilusti seurantaviikkojen edetessä. Lähtötilanteessa olin vielä epävarma omista ohjelmistokehitystaidoista, vaikka olin työhaastattelussa jo syksyllä kehunut niitä. Tämä ongelma minulla on ollut jo pitkään. Olen ollut melko varma, että tulen tekemään ohjelmistokehittäjän töitä jo jonkin aikaa, mutta olen silti kantanut mukani pientä epävarmuutta omista taidoistani. Olen mielestäni nyt päässyt täysin eroon tästä epävarmuudesta. Olen tällä hetkellä varma, että minusta tulee alan ammattilainen – ellen jo sitä ole.

Viikkojen aikana eniten kehityin ohjelmoinnin ja ongelmanratkonnassa taidoissani. Loppuvaiheilla toteutin jo täysin itsenäisesti omia ratkaisuja järjestelmään. Annan rohkeasti omien mielipiteitteni kuulua, jos minulla niitä on ja iso osa toteutuksistani menevät jo ilman kommentteja suoraan mergettäväksi.

Tärkein oppimani uusi menetelmä, jos en sitä jo riittävästi ylistänyt, on Eclipsen debug – työkalu. Tämä nopeuttaa ongelmien löytämistä koodista ja auttaa minua yleisestikin kehittämisessä huomattavasti. Olen muutenkin oppinut käyttämään tätä kehittäjän päätyökalua

Eclipseä paljon tehokkaammin – pystyn ison järjestelmän tietyn koodirivin löytämään yllättävänkin nopeasti.

Pääsin viikkojen loppupuolella tutustumaan asiakkaamme toiseen järjestelmään, johon tulen tulevaisuudessa tekemään enemmänkin töitä. On virkistävää vaihtelua päästä tekemään muita projekteja ja tekemään myös monenlaista kehitystyötä. Esimerkiksi seuranta- viikkojen loppupuolella ilahduin päästessäni tekemään käyttöliittymämuutoksia. Pääsin seurantaviikkojen jälkeen suunnittelemaan myös tietokantaa – täysin itsenäisesti. Tässä vaiheessa minulle muistui mieleen hyvin kouluajoilta oppimat UML –kaaviot, joita silloin pidin melko turhina. Huomasin kuitenkin sen olevan itsellenikin selkeämpää piirtää tietokannan rakenne ennen sen luontilauseiden kirjoittamista, etenkin kun on kyse isommista muutoksista.

Toimenkuvani on alussa mainitsemani 'full stack' –kehittäjä. Tämä tarkoittaa sitä, että kehitän järjestelmiä muiltakin osin, kun vain joko back- tai front-endistä – eli itse logiikkaa tai käyttöliittymää. Tämä on käynyt toteen paremmin kuin uskoin. Olen päässyt tekemään suurimmilta osin sitä, mistä pidän eniten; back-endiä. Tämän ohella olen tehnyt paljon käyttöliittymämuutoksia, testausta ja jopa hieman määrittelyä. On virkistävää päästä tekemään monenlaisia asioita ja ehkä jopa hieman pois omalta mukavuusalueelta.

Opinnäytetyön kirjoittamisen aikana tutustuin moniin eri aiheisiin, joihin en todennäköisesti olisi syvemmin muuten tutustunut. Tämä on auttanut minua työssäni, sillä olen opetellut näitä osa-alueita syvemmin myös teoriapohjalta. Muun muassa automaattitestauksesta ja JUnit testeistä minun on pitänyt jo kouluaikoina opiskella, mutta nyt pääsin syventymään niihin työni ja opinnäytetyön kirjoittamisen yhteydessä.

Kulutin työn kirjoittamiseen päivittäin heti töiden jälkeen aikaa muistiinpanojen kanssa. Toisinaan nämä muistiinpanot olivat tekstimuodossa ja toisinaan olin ottanut kuvan Gittiin tekemistäni committeista. Huomasin, että olen seuraavana päivänä paljon valmiimpi työhön ja pääsen tehokkaammin alkuun, kun olen edeltävänä iltana vielä kerrannut päivien tapahtumia. Viikoittain työn seuraaminen on antanut näkökulmaa työhön uppoavasta ajasta ja rahasta. Aikaisemmin olen ihmetellyt, kuinka suuret järjestelmät voivat maksaa näin suuria summia. Nyt kuitenkin huomaan sen, kuinka tarkkaa työtä tämä vaatii. Iso osa työstä menee järjestelmän dokumentointiin ja testaukseen. Nämä kaksi asiaa ovat isoissa järjestelmissä tärkeämpiä, kuin itse kehitystyö. Koska iso järjestelmä, joka ei ole dokumentoitu, ei ole laadukas. Tai niin kuin Yourdon (1988, 73) sanoi, "There is nothing in the programming field more despicable than an undocumented program."

Työn kirjoittamisen aikana huomasin itseni myös kehittyvän erittäin positiivisesti kirjoittamisessa. Päivä päivältä kirjoittaminen tuntuu luontevammalta ja lauseet muodostuvat paremmin. Loppua kohden tuntui jopa, että minun täytyy yrittää rajoittaa tekstin ulostuloa, sillä viikkojen merkinnät jatkuivat vain pitenemistään. Tuntui siltä, että pystyisin kirjoittamaan aiheesta vain enemmän ja enemmän.

Opinnäytetyön seurantaviikkojen ajankohdat asettuivat täydellisesti kohdilleen. Kirjoittaminen alkaa tiimipäivällämme Turun toimistolla. Samoihin aikoihin aloitin kunnolla kehitystyöt projektiin. Projektin kehitystyöt pyörivät siihen liittyvien lakimuutosten ympärillä. Ajanjakso loppuu sopivasti juuri, kun projektista loppuu minun osalta työt ja pääsen jatkamaan uuden projektin parissa. Työnkuva on päivittäin niin erilainen, että kerrottavaa riitti jokaiselle päivälle usein hyvinkin paljon, tämä on yksi iso syy, minkä takia pidän tästä työstä. Joka päivä on erilainen, joka päivä tunnen kehittyväni ja ennen kaikkea – joka päivä on kiva käydä töissä, koska koodaus on intohimoni.

Lähteet

Agile Alliance 2016. Backlog. Luettavissa: <http://guide.agilealliance.org/guide/backlog.html>
Luettu: 30.01.2016

AngularJS Documentation 2016. Directives. Luettavissa:
<https://docs.angularjs.org/guide/directive>
Luettu: 3.4.2016

Apache Maven 2016. Tervetuloa –ruutu. Luettavissa: <https://maven.apache.org/>
Luettu: 4.2.2016

Atlassian 2016. Reaching true agility with continuous integration. Luettavissa:
<https://www.atlassian.com/agile/continuous-integration>
Luettu: 11.4.2016

Bragge J, Merisalo-Rantanen H & Hallikainen P. 2005. Gathering Innovative End-User Feedback for Continuous Development of Information Systems: A Repeatable and Transferable E-Collaboration Process. Luettavissa:
https://www.researchgate.net/profile/Johanna_Bragge/publication/3230353_Gathering_Innovative_End-User_Feedback_for_ContinuousDevelopment_of_Information_Systems_A_Repeatableand_Transferable_E-Collaboration_Process/links/004635235dd2aebfa0000000.pdf
Luettu: 7.2.2016.

Charles Anderson 2015. Docker. Luettavissa:
<http://www.computer.org/csdl/mags/so/2015/03/mso2015030102.pdf>
Luettu: 11.4.2016

Devi R. 2012. Importance of Testing in Software Development Life Cycle. Luettavissa:
<http://www.ijser.org/researchpaper%5CImportance-of-Testing-in-Software-Development-Life-Cycle.pdf>
Luettu: 7.2.2016.

Flyway 2016. Documentation, Get started. Luettavissa: <https://flywaydb.org/>
Luettu: 20.3.2016

Henrik Kniberg 2009. Kanban vs Scrum. Luettavissa: <https://www.crisp.se/file-uploads/Kanban-vs-Scrum.pdf>
Luettu: 30.01.2016

Hongjun Li 2011. RESTful Web Service Frameworks in Java. Luettavissa: http://ieeexplore.ieee.org/sci-hub.io/xpl/login.jsp?tp=&arnumber=6061739&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D6061739
Luettu: 6.3.2016

ISTQB 2016. Why is software testing necessary?. Luettavissa: <http://istqbexamcertification.com/why-is-testing-necessary/>
Luettu: 13.3.2016

Lars Vogel 2013. Java debugging with Eclipse – Tutorial. Luettavissa: <http://www.vogella.com/tutorials/EclipseDebugging/article.html>
Luettu: 20.2.2016

Techtarget 2015. Definition: sprint. Luettavissa: <http://searchsoftwarequality.techtarget.com/definition/Scrum-sprint>
Luettu: 30.01.2016

Old Mill 2016. Esittely. Luettavissa: <http://www.oldmill.fi/esittely.html>
Luettu: 01.02.2016

Netbeans 2016. Introduction to the Wicket Web Framework. Luettavissa: <https://netbeans.org/kb/docs/web/quickstart-webapps-wicket.html>
Luettu: 20.2.2016.

Nilesh Jain, Priyanka Mangal & Deepak Metha 2014. AngularJS: A Modern MVC Framework in Javascript. Luettavissa: <http://www.jgrcs.info/index.php/jgrcs/article/viewFile/952/610>
Luettu: 28.2.2016.

Philipp Vogt, Florian Nentwich, Nenad Jovankovic, Engin Kirja, Cristopher Kruegel & Giovanni Vigna 2007. Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis. Luettavissa: http://publik.tuwien.ac.at/files/pub-inf_5310.pdf
Luettu: 6.3.2016

Robert C. Martin 2009. Clean Code, A Handbook of Agile Software Craftmanship. Viides-toista painos. Robert C. Martin Series. Prentice Hall. New Jersey.

Scorr Chacon & Ben Straub 2014. Pro Git. Luettavissa:

[https://books.google.fi/books?hl=en&lr=&id=jVYnCgAAQBAJ&oi=fnd&pg=PP3&dq=git+lin](https://books.google.fi/books?hl=en&lr=&id=jVYnCgAAQBAJ&oi=fnd&pg=PP3&dq=git+linux&ots=vN2bs9bug9&sig=HWVH-)

[us&ots=vN2bs9bug9&sig=HWVH-](https://books.google.fi/books?hl=en&lr=&id=jVYnCgAAQBAJ&oi=fnd&pg=PP3&dq=git+linux&ots=vN2bs9bug9&sig=HWVH-SqttDhcYE_p2f74ch3gv0&redir_esc=y#v=onepage&q&f=false)

[SqttDhcYE_p2f74ch3gv0&redir_esc=y#v=onepage&q&f=false](https://books.google.fi/books?hl=en&lr=&id=jVYnCgAAQBAJ&oi=fnd&pg=PP3&dq=git+linux&ots=vN2bs9bug9&sig=HWVH-SqttDhcYE_p2f74ch3gv0&redir_esc=y#v=onepage&q&f=false)
Luettu: 3.4.2016

Valtiokonttori 2015. Valmiina digikiriin – Digitalisaatio ja virastojen tuottavuuspotentiaali.

Luettavissa: <http://www.valtiokonttori.fi/fi->

[FI/Virastoille_ja_laitoksille/Digitalisaatio/Loppuraportti_Valmiina_digikiriin](http://www.valtiokonttori.fi/fi-)

Luettu: 28.3.2016

W3C 2016. What is the Document Object Model?. Luettavissa:

<https://www.w3.org/TR/WD-DOM/introduction.html>

Luettu: 28.3.2016

Edward Yourdon 1988. Techniques of Program Structure and Design. Prentice Hall.