

Sovelluskehittäjän päiväkirja

Juha Palmu

Opinnäytetyö
Tietojenkäsittelyn koulutusohjelma
2019



Tekijä(t) Juha Palmu	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Sovelluskehittäjän päiväkirja	Sivu- ja liitesivumäärä 54
Opinnäytetyön otsikko englanniksi Software Developer Diary	
<p>Päiväkirjatyö on tehty Digia Oyj palveluksessa 10 viikon ajalta. Työssä kuvataan ohjelmistokehittäjän arkea, ongelmanratkaisutilanteita, uusia teknologioita, sekä kehittymistä IT-alan ammattilaisena.</p> <p>Pääasiallinen sisältö on Java-kehitystä, ja siihen liittyvien ratkaisujen kehittämistä. Isossa roolissa on versionhallintajärjestelmän vaihtaminen koko yksikön käyttöön.</p> <p>Työn aikana ilmenneet suurimmat haasteet tulevat vuosia vanhoista sovelluksista, vanhoista järjestelmistä ja vanhenneista käytänteistä. Järjestelmiin ja käytäntöihin on ollut mahdollista päästä vaikuttamaan suhteellisen uutenakin työntekijänä. Kirjoittajalla on tuoreempaa näkökulmaa uusiin ja hyväksi havaittuihin käytäntöihin.</p>	
Asiasanat Java, Backend, Software Development, Scrum	

Sisällys

1	Johdanto	1
1.1	Tietoperusta.....	1
1.2	Ammattitermit ja käsitteet.....	1
2	Lähtötilanteen kuvaus	4
2.1	Oman nykyisen työn analyysi.....	4
2.2	Sidosryhmät työpaikalla	5
2.3	Vuorovaikutustaidot työpaikalla.....	6
3	Päiväkirjaraportointi.....	7
3.1	Seurantaviikko 1	7
3.2	Seurantaviikko 2	13
3.3	Seurantaviikko 3	18
3.4	Seurantaviikko 4	22
3.5	Seurantaviikko 5	28
3.6	Seurantaviikko 6	33
3.7	Seurantaviikko 7	39
3.8	Seurantaviikko 8	42
3.9	Seurantaviikko 9	46
3.10	Seurantaviikko 10	49
4	Pohdinta ja päätelmät.....	53
4.1	Kehittyminen työn aikana	53
4.2	Jatkokehitys- ja tutkimusmahdollisuudet	53
4.3	Analysoinnin hyödyntäminen.....	54
	Lähteet	55

1 Johdanto

1.1 Tietoperusta

Opinnäytetyön toteutus alkaa 12.08.2019 ja loppuu 04.11.2019. Näistä ensimmäinen viikko on aloitusviikko, jonka tavoitteena on tuoda esiin nykyiset työtehtävät ja -ympäristö. Viimeinen viikko keskittyy työn aikana saatujen tulosten pohdintaan ja viimeistelyyn. Näiden väliin sijoittuva aika on varattu työn ohessa tehtävälle raportoinnille.

Työskentelen Digia Oyj:n palveluksessa tittelillä ”Junior Software Developer”. Vastaan työtehtävissäni ammattiliittojen jäsenrekisterin uudistusprojektissa palvelinpuolen Java-ohjelmoinnista.

Tehtävissäni vaaditaan osaamista Javasta ja sen Spring Frameworkista. Kehityksen tukena on hyötyä olla osaamista SVN ja Git versionhallinnoista, sekä virtuaaliympäristöistä paikallisessa kehityksessä. Kehitys tapahtuu Windows ympäristössä, mutta sovellukset julkaistaan Linux ympäristöihin.

Digia Oyj on saanut alkunsa vuonna 2005 SysOpen ja Digia Oy:n fuusiosta. Yritys laajenee koko ajan yritysostojen ja uuden henkilöstön palkkaamisen myötä. (Digia, Viitattu: 15.08.2019)

Digia on yksi Suomen johtavista ohjelmistoyhtiöistä, ja se työllistää lähes 1200 henkilöä erilaisilla toimialueilla.

Aloitin Digian palveluksessa 15.04.2018 jota ennen työskentelin Tilastokeskuksella 10 kuukautta. Työskentelen Digian Custom Solutions yksikössä osana Associations tiimiä, joka jakaantuu vielä kahteen erilliseen kehitystiimiin. Tiimi, jossa itse työskentelen, koostuu 12 henkilöstä, joihin sisältyy testaaaja, käyttöliittymä-, tietokanta- ja palvelinkehittäjiä. Kehitystä tehdään scrumin mukaisissa kahden viikon sprinteissä, jotka alkavat aina suunnittelulla ja päättyvät katselmointiin asiakkaan kanssa.

Tiimit toimivat samoissa avokonttoritiloissa, eli tarvittavat henkilöt ovat lähes aina helposti tavoitettavissa.

1.2 Ammattitermit ja käsitteet

Scrum. Kehitysmalli, jota yleensä harjoitetaan hieman sovellettuna. Omassa tapauksessa kahden viikon työsprintit, joiden lopuksi tulokset katselmoidaan asiakkaiden kanssa. Tä-

hän sisältyy myös työjonon siistimistä, jokaisen sprintin erillinen suunnittelu, sekä sprinttien lopuksi kehityksen aikana tehtyjen havaintojen läpikäynti.

Front end. Loppukäyttäjälle näkyvä osuus sovelluksesta.

Back end. Palvelinpuolen koodi. Käyttöliittymät usein hakevat datansa palvelinten rajapintasovelluksista.

REST (Representational state transfer). HTTP-protokollaan perustuva arkkitehtuurimalli erilaisten rajapintojen toteuttamiseen. Palauttaa useimmin dataa JSON-muodossa.

SOAP (Simple Object Access Protocol). Myös HTTP-protokollaan perustuva arkkitehtuurimalli. Palauttaa ja käyttää vain XML-dataa.

Repository. Koodin säilytyspaikka.

SVN (Subversion). Versionhallintasovellus, jolla on yksi jaettu repository.

Git. Hajautetun mallin versionhallintasovellus. Repositoryn voi joko säilyttää jonkun omalla koneella, palvelimella, tai kolmannen osapuolen palvelussa, esimerkiksi GitLab tai GitHub.

GitLab. Palvelu Git repositoryjen säilyttämiseen. Tarjoaa myös erilaisia toimintoja työnkulun ja laadun parantamiseen, kuten esimerkiksi merge requestit.

Merge. Versionhallinnassa eri tilassa olevien tiedostojen yhdistäminen niin, että kaikki halutut muutokset säilyvät.

Build. Vaihe, jossa koodi käännetään ja paketoidaan.

Nightly build. Buildi, joka ajetaan joka yö. Näistä muodostuu nightly versio, joka sisältää aina edellisen päivän uudet muutokset.

CI (Continuous Integration). Suomeksi jatkuva integraatio. Järjestelmä, joka kääntää ohjelmakoodin ja ajaa testit. Näillä myös mahdollista hoitaa sovelluksen julkaisu.

Bamboo. Atlassianin toteuttama CI sovellus.

GitLab CI. GitLabin kanssa yhteensopiva CI sovellus.

XPath (XML Path Language). Kyselykieli XML-elementtien hakemiseen.

2 Lähtötilanteen kuvaus

2.1 Oman nykyisen työn analyysi

Nykyisiin työtehtäviini kuuluu Java sovelluskehitys, yksikkö- ja integraatiotestaus, jatkuvasta integraatiosta huolehtiminen ongelmatilanteissa, erilaisten raporttien luominen sovellusten käyttöön, ympäristöjen päivittäminen, sekä asiakkaiden versioitoimitukset.

Työtehtäväni ovat suurimmaksi osaksi Java-ohjelmointia. Tähän sisältyy uusien rajapintojen kehittäminen tietokannan ja käyttöliittymien väliin. Koodille tulee kirjoittaa testit, jotta sen toimivuudesta voidaan uusien toiminnallisuuksien myötä edelleen mennä takuuseen. Joskus yrityksestä huolimatta, koodi ei aina käänny tai versionhallintaan on päässyt koodia, joka rikkoo testit. Näissä tilanteissa on hyvä osata selvittämään mikä on mennyt vikaan ja miten se korjataan. Aina ei kuitenkaan oma osaaminen riitä, eli olisi hyvä pystyä kuitenkin paikantamaan henkilö, joka osaa selvittää ongelmat.

Ajoittain päivitän omia ja asiakkaiden testiympäristöjä. Tähän toimenpiteeseen on valmiit skriptit, joten valmiit paketit ja asennusohje tarvitsee viedä vain palvelimelle, jonka jälkeen ajetaan asennusskriptit.

Versioitoimitukset ovat käytännössä sama prosessi kuin testiympäristöjen päivitys, mutta niissä roolini on olla tukena, jos jokin menee vikaan.

Ongelmatilanteet on hyvin dokumentoitu tiimin sisäiseen wikiin ja niitä pyritään pitämään ajan tasalla. Näistä löytyy yleensä vastaukset, jos joku on epäselvää tai menee pahasti vikaan. Työssäni tarvitsen siis perusteellista kykyä ongelmanratkaisemisesta.

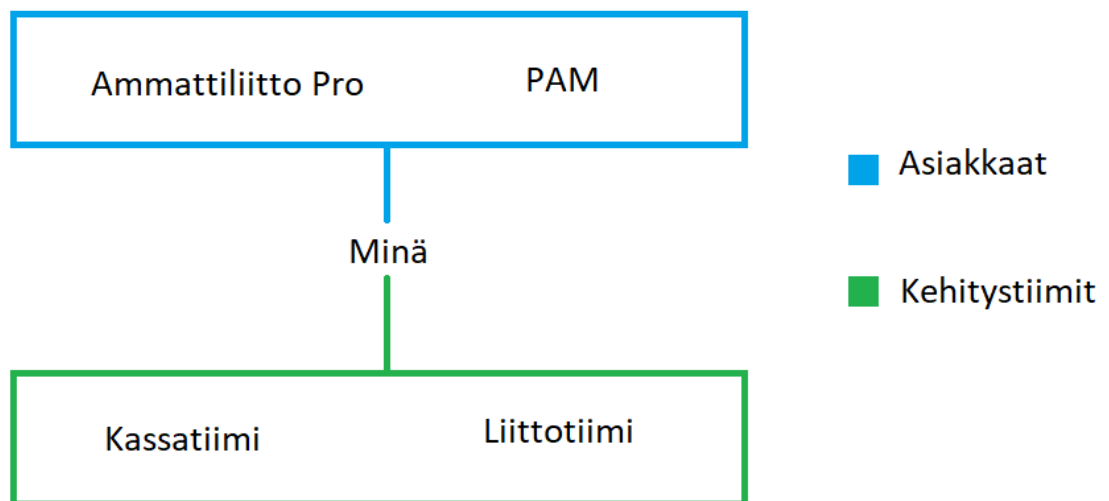
Pidän itseäni taitavana suoriutujana, ja tiimiltä saamani palaute tukee tätä luuloa. Olen kuitenkin ollut alalla sen verran pitkään, että omasta osaamisesta on ollut hyötyä myös kauemmin Digialla töissä olleille kehittäjille. Ehkä siis kuuluisin johonkin taitavan suoriutujan ja kokeneen asiantuntijan välimaastoon, vaikka tittelini ei puhukaan tämän puolesta.

Olen valmistunut datanomiksi jo vuonna 2012 ja toiminut alan töissä hieman yli 3 vuotta. Koen että kokemuksen myötä uskallan paremmin tuoda omia ajatuksia ilmi enkä vain mennä sillä mikä on paras ratkaisu jonkun muun mielestä. Pyrin tekemään siistiä jälkeä enkä välttämättä tyydy siihen mikä on helpoin ja nopein ratkaisu.

Jatkossa haluan oppia enemmän työni kannalta tarpeellisia teknologioita, että pystyn selvittämään niitä koskevat ongelmat paremmin itse. Oma osaamiseni perustuu enemmän suoraan koodaamiseen kuin vaadittaviin taustajärjestelmiin. Pitäisin järkevänä laajentaa omaa osaamista enemmän sysadmin suuntaan, jotta ymmärtäisin tätä paremmin.

Mielestäni nykymaailmassa ei kuitenkaan ohjelmoijalla enää pelkkä osaaminen varsinaisesta ohjelmoinnista riitä. Lisäksi tarvitaan osaamista jatkuvasta integraatiosta ja julkaisusta sekä näiden tukijärjestelmistä, pilvialustoista ja muistakin ylläpitoon liittyvistä teknologioista.

2.2 Sidosryhmät työpaikalla



Sidosryhmät kuvana

Sisäisiä sidosryhmiä ovat yksikön kummatkin kehitys tiimit. Tiimit tekevät työtä ammattiliittoille ja työttömyyskassoille.

Työskentelen itse liittotiimille jäsenrekisterin uudistusprojektissa.

Kassatiimi on toinen Associations yksikön kehitystiimi, joka tekee sovellusta työttömyyskassoille. Kummatkin tiimit käyttävät jonkun verran samaa koodia, joten näihin tehdyt muutokset eivät saa heidän puolellaan rikkoa mitään.

Ammattiliitot maksavat siitä, että tuotetta kehitetään heidän tarpeiden mukaisesti. Sprinttien lopussa saadaan heiltä arvokasta palautetta ja kehitysideoita tuotteiden suhteen.

Heidän mielipiteensä ja intressit ovat erittäin tärkeitä työn kannalta, koska heidän tulee saada tarpeisiinsa sopiva tuote. Nämä ovat kuitenkin asioita, joita kehitystiimi ei itse pysty keksimään.

2.3 Vuorovaikutustaidot työpaikalla

Mielestäni tiimimme toimii erittäin hyvin avokonttorin ansiosta. Tarvittaessa saa nopeasti kysytyä tarpeellisia tietoja jostain toiminnallisuudesta, toteutustavasta tai ongelmatilanteesta. Osataan aina vähintäänkin osoittaa oikeaan suuntaan, jos suoraa vastausta ei ole heti saatavilla. Tiimi käy myös usein yhdessä syömässä.

Liittotiimin ja Kassatiimin yhteistyö ei ole ehkä ihan niin läpinäkyvää mitä itse toivoisi. Vaikka molempien tekemiset riippuvat toisistaan jonkun verran, ei ole aina täysin selvää mitä toisessa tiimissä ajoittain tapahtuu. Osaamista kuitenkin lainataan tarpeiden mukaan.

Läheskään kaikki tiimiläiset eivät ole suorassa yhteydessä asiakkaaseen, mutta tieto kyllä kulkee sujuvasti molempiin suuntiin tarvittaessa.

Omalle osaamiselle sidosryhmien välisessä vuorovaikutuksessa ongelmana pidän enimmäkseen sitä, kuinka teknisiä termejä uskaltaa kenenkin kanssa käyttää. Kaikille ei kuitenkaan samat asiat ole itsestäänselvyksiä.

3 Päiväkirjaraportointi

3.1 Seurantaviikko 1

Maanantai 19.08.2019

Päivän tavoitteena on saada enemmän selkoa viime viikolla esiintyneeseen ongelmaan, jossa Kassatiimin sovelluksen julkaisu uudelle ei onnistu Spring Frameworkin dependency injectionien takia. Kyseessä on uusi palvelinympäristö, johon sovellukset halutaan siirtää. Vanhassa ympäristössä sovellus on toiminut moitteitta. Perjantain perusteella näyttäisi siltä, että jostain syystä Springillä on ongelmia injektoida beania, joka on luotu perusluokasta, mutta dependency injenctionilta pyydetään siitä periytyvää luokkaa. Vaatii enemmän testausta, jotta voi sanoa varmaksi, että ongelma on tässä.

Tämän päivän tehtäviin kuuluu myös versionhallinnan haarojen mergeäminen. Ongelmallisen tästä tekee se, että aikaisemmat merget on tehty huonosti ja trunkista puuttuu committeja. Nähtäväksi jää minkälaisia ongelmia tämä aiheuttaa.

Tänään myös lopetetaan vanha sprintti ja suunnitellaan uusi. Kesän jälkeinen pitkä sprintti on tulossa päätökseen ja saadaan suunniteltua lisää tekemistä.

Sain selvitettyä dependency injection ongelman ja se johtui juuri siitä mistä luulinkin. Tämän vahvisti omat testini tyhjällä projektilla vastaavanlaisesta tilanteesta. En saanut kyllä selkoa mikä johtaisi siihen, että sovellus on tietyssä ympäristössä toiminut aikaisemmin. Ehkä kyseessä on ollut aikaisempi versio, jossa ei vielä ole tehty näin?

Versionhallinnan mergeämiset jäivät yllättäen odottamaan huomista. Tämän esteenä olivat toisen tiimin tekemät muutokset, jotka rikkoivat SVN trunk haaran ja joudun odottamaan tähän korjausta. Ehdin tähän kuitenkin käyttää sen verran aikaa, että sain katsottua mitkä revisiot trunkista puuttuu ja onko nämä syytä ottaa mukaan mergeen. Myös pientä testailua miten SVN on järkevä hoitaa mahdolliset conflictit.

Sprintin suunnittelussakaan ei varsinaisesti saatu käytyä läpi omalta osalta relevantteja asioita. Tällä hetkellä uuteen sprinttiin tuli vain edellisen sprintin keskeneräiset työt. Lisää olisi tarkoitus myöhemmin tulla groomauksessa. Tämä on viimeinen sprintti ennen normaalien sprinttien jatkamista, johon sisältyy myös katselmoinnit asiakkaiden kanssa.

Edellisen sprintin retrossa toin ilmi vähäisen työmäärän, mutta siihen ei silti saatu aivan välitöntä ratkaisua.

Backlogin groomauksessa on tarkoitus käydä läpi tiimin kanssa mahdolliset turhat tehtävät backlogilta, kuvata ja priorisoida uudet tarpeet, arvioida tehtäviin kuluva työmäärä, sekä isompien kokonaisuuksien pilkkominen pienemmiksi.

Agile Alliancen mukaan on pidettävä huoli siitä, että tiimille löytyy merkityksellistä työtä. On tärkeää, että tehtävät on priorisoitu oikein, ja että asiaankuuluvat henkilöt ovat antaneet parhaan osaamisensa mukaan työmääräarvion.

(Agile Alliance. Viitattu 15.09.2019)

Tiistai 20.08.2019

Tänään pääsen aloittamaan versionhallinnan merget. Nämä tulevat todennäköisesti olemaan hieman haastavia SVN takia, ja koska versionhallinnasta on aikaisemmin jäänyt puuttumaan tavaraa ja nyt ne pitäisi lisätä sinne jälkikäteen uusien muutoksien jälkeen. Nähtäväksi jää tuleeko tästä koko päivän homma, kun kaikkien buildien pitää näyttää vihreää ennen kuin tämä on ok.

Mergejen lisäksi suunnittelin kysyväni esimieheltä, jos voisimme ottaa one to one keskustelun ennen ensi viikkoa, koska asiat, joista tahtoisin keskustella ei mielellään odottaisi ensi maanantaihin. Tarkoitus olisi puhua mahdollisista lisätöistä, koska näköjään kaksi kehitystiimiä ei pysty minua tarpeeksi työllistämään. Pidemmän päälle todella rasittava tilanne.

Lähes koko päivä meni mergejen kanssa. Ensin trunk kehityshaaraan, kehityshaara trunkkiin ja vielä kerran trunk kehityshaaraan. Suurimmaksi ongelmaksi koitui toisessa vaiheessa conflictien hoito aikaisempien huolimattomuuksien takia, sekä se että SVN automerge teki jotain ihan omia muutoksia ja niitä piti metsästä. Tietokantaskripteihin oli SVN automergen jäljiltä tullut ylimääräisiä rivejä, jotka aiheuttivat sen, että gradle build mene läpi. Eikä ongelmat loppuneet tähän, kun jostakin toisesta tietokantaskriptistä oli jäänyt mergen yhteydessä rivejä uupumaan, jonka takia trunkin buildi ei mennyt läpi. Näihin sain onneksi apua kankoodarilta, joka osasi paremmin selvittää miksi skriptin ajaminen epäonnistuu. Skriptistä oli jäänyt puuttumaan alter table rivi, jossa lisättiin uusi kenttä.

En saanut esimiehen kanssa varattua aikaa tälle viikolle koska hänen kalenterinsa on täynnä. Ainoa järkevä aika olisi ollut huomenna lounasaikaan. Saatiin sovittu esimiehen

etäpäivälle perjantaiksi Skype-palaveri one to onen lisäksi, jossa käydään läpi töitä ja muita fiiliksiä.

Keskiviikko 21.08.2019

Eilisten mergejen jäljiltä oli CI buildit menneet aamulla läpi. Joistakin tiedostoista oli kuitenkin tullut väärät versiot, jotka ehdittiin heti aamusta jo palauttaa aikaisempiin versioihin. Tämän päivän ohjelmassa muuten on parin indeksointiongelman korjaaminen. Sovelluksessa uutta henkilöä luodessa nimi ei päädy hakuindeksiin, ja yrityksen tietoja päivittäessä uusi nimi ei päivity indeksiin heti. Kumpikaan näistä ei todennäköisesti osoittaudu kovinkaan isoksi ongelmaksi. Näiden jälkeen joudun kysyä lisää töitä muualta koska backlogi on aivan tyhjänä. Jos kaikki työt loppuvat niin todennäköisesti käytän aikaa Dockerin opiskeluun ja kokeiluun.

Indeksointiongelmat johtuivat siitä, että kaikkia indeksiin vietäviä kenttiä ei täytetty uuden jäsenen luonnin yhteydessä. Yritysten osalta indeksointiongelma johtui siitä, että hakukenttään ei päivitetty uutta nimeä. Indeksistä hakemalla vanhalla nimellä löytyi yritys uudella nimellä, koska tämä päivitettiin vain osaan kentistä.

Loput työpäivästä meni Docker opiskeluun. Katsoin tähän vähän pohjaa yhdestä sovelluksesta, joka on tehty dockerin päälle. Omalla työkoneella koituu vähän haasteelliseksi dockerin pyörittäminen, koska muu kehitys tapahtuu virtuaalikoneiden avulla ja dockerin ajaminen sellaisenaan vaatisi Windowsista Hyper-V päälle. Tämä ei tietenkään ole mahdollista samaan aikaan kun pyörittää VirtualBoxia. Olisi luultavasti mahdollista ajaa Dockeria virtuaalikoneessa.

Sain päivän lopuksi pystytettyä itselleni Docker-ympäristön.

Torstai 22.08.2019

Tälle päivälle ei ole ollenkaan suunniteltuja töitä, joka on sinänsä erittäin harmi. Joudun todennäköisesti jatkaa eilisiä Docker opiskelua, ellei uusia bugitikettejä ilmesty. Dockerin osalta haluaisin demoapplikaation pyörimään oikealla palvelimella, enkä Flaskin omalla kehityspalvelimella. Nyt Docker imageni vain suorittaa komennon `py app.py`, joka oikeassa tuotannossa estäisi esimerkiksi sen, että useampi käyttäjä voisi käyttää sovellusta yhtä aikaa. Flaskin development server mahdollistaa vain yhden samanaikaisen käyttäjän.

Lähes koko päivä meni dockerin opiskelussa. Nopeasti selvitin muutaman bugitikkiin liittyen sen, mitä tietokantaproseduuria niissä kutsutaan, jotta tietokantakoodari voi tehdä asialle jotain.

Dockeriin käytetyllä ajalla selvitin, kuinka saan kunnollisen tuotantoympäristön niin, että Flask-applikaatio olisi useamman kuin yhden käyttäjän käytettävissä saman aikaisesti. Gunicorn vaatii, että python sovelluksesta löytyy Flask-applikaatio nimellä application eikä yleisemmin käytetty app, joten tämän muuttujan nimi piti vaihtaa. Tämän olisi tietenkin myös voinut ratkaista erillisellä tiedostolla, jossa app importataan application nimellä. Dockerfileen myös lisätty rivi, joka asentaa Docker imagelle Gunicorn sovelluksen. Lisäksi myös dockerin ajama komento vaihdettu käyttämään Gunicornia sen sijaan, että python-tiedostoa ajettaisiin suoraan. Gunicornin olisi myös voinut lisätä riippuvuudeksi requirements.txt tiedostoon, mutta laitoin sen siksi erillisenä rivinä koska sovellus ei toimiakseen sitä vaadi. Ratkaisuksi tein docker-compose.yml tiedoston, jossa määrittelin jaetun yhteyden Nginx- ja Flask-konttien välille. Nginx-kontille oma config-tiedosto, jossa määritellään mihin lokaaliin osoitteeseen pyynnöt ohjataan. Flask-kontille laitoin sovelluksen pyörimään Gunicorn serverille, ja määritin sen käyttöön useamman workerin, jotta yhtäaikaisia käyttäjiä voi olla enemmän. Tähän pitäisi vielä todennäköisesti lisätä jonkinlainen asynkronisuus, jotta toimisi sujuvammin.

Perjantai 23.08.2019

Päivän työt ovat alkuvaiheessa selvitystä vaille. Eilisen perusteella tulossa on todennäköisesti vain yksi tiketti korjattavaksi. Kyseisessä toiminnossa esiintyy ongelmia, kun postinumerotietoja päivitetään sille luodun rajapinnan kautta. Rajapinta on tehty Springillä käyttäen repository patternia. Thorben Janssenin kuvauksen mukaan repository patternin hyödyt tulevat siitä, että ei tarvitse muuttaa bisneslogiikkaa mahdollisten tietokantamuutoksien takia. Koodia on myös helpompi testata ja käyttää uudelleen.

(Thorben J. Viitattu 15.09.2019)

Olen tämän suhteen samaa mieltä, sillä koodin testaamisen helpottaminen ja uudelleenkäyttäminen ovat tärkeitä asioita. Näin voidaan myös varmistua paremmin siitä, että kaikkea koodia ei tarvitse tehdä uusiksi, kun halutaan muuttaa jotain.

Rajapinnan endpoint on määritetty controllerissa, joka kutsuu serviceä, joka kutsuu repositoryä päivityksen yhteydessä. Tähän on kuitenkin tehty älyttömiä ratkaisuja niin sanotun legacy tietokannan takia. Kaikki sovelluksen rajapinnat ottavat vastaan vain GET ja PUT requesteja. PUT requestien kautta tapahtuu siis myös uusien arvojen lisääminen ja poisto.

Nämä määritetään lähetettävän request bodyn kentissä "poisto" ja "lisays", joita käsittelevä logiikka on toteutettu tietokannassa. Nämä kentät ottavat tapauskohtaisesti arvoiksi false/true tai 0/1. Näiden takia tarvitaan aika paljon turhia erikoissääntöjä, ja nyt näyttäisi siltä, että postinumerodataa on myös replikoituna useammassa paikassa. Ongelma tulee siitä, että replikoitua dataa ei tietenkään voi päivittää kutsumalla repositoryn save metodia. Ratkaisuksi tähän tulee todennäköisesti niin, että jos käsiteltävä postinumero ei ole uusi, kutsutaan sen sijaan tietokantaproseduuria, joka käy päivittämässä nämä kaikki. Sovelluksella on oma frameworkinsa, joka generoi luokkia tietokantaproseduurien ja objektien perusteella. Mielestäni tämä on sellainen asia, josta olisi pitänyt luopua ihan ylläpidettävyyden ja kehittäjien mielenterveyden kannalta.

Esimieheni oli varannut tälle päivälle Skype-palaverin, jossa olisi tarkoitus käydä läpi työtehtäviäni, jotka ovat nyt kesälomienkin jälkeen olleet täysin auki. Jos tähän hommaan saataisiin nyt jotain järkeä viimeistään ensi viikon aikana.

Puhuttiin aamupäivällä esimiehen kanssa läpi mahdollisia tulevia projekteja, joista löytyisi minulle sopivaa tekemistä. Kuulosti näin alkuun ihan lupaavalta, ettei mene kaikki päivät peukaloita pyöritellessä. Muilta osin päivä on töiden puolesta ollut erittäin antoisa. Sain tehtäväksi uuden REST palvelun liittojen jäsenen kurssihistorian hakemiseen, joka nyt ei varsinaisesti älytöntä tekemistä vaatinut. Tätä varten tuli luoda tietokantaa vastaavat Java-entitit, DTO-luokka, repository, service ja controller. Pienimuotoisena ongelmana oli se, että nämä tiedot haetaan henkilötunnuksella. Rajapinnan kautta kuitenkin lähetetään vain jäsennumero, joten tähän tarvittiin erillistä serviceä jäsennumeron muuttamiseksi henkilötunnukseksi. Tämä palvelu kuitenkin antoi aluksi jäsennumerolle pelkkää nullia. Muunnospalvelu on toteutettu käyttämään tietokantaproseduuria, joka ottaa jäsennumeron lisäksi sisäänsä sovellustunnuksen. Jouduin laittamaan erikseen käyttämään liittojen sovellustunnusta tarkan sovellustunnuksen sijaan, jotta sain tämän toimimaan.

Myös kassapuolen asianhallintasovellukseen löytyi hieman tekemistä. Ilmeisesti aikaisemmin tekemäni muutokset eivät olleet ihan riittänyt siihen, että sovellus lähtisi uudessa ympäristössä toimimaan. Virheilmoitus oli tällä kertaa erilainen kuin viimeksi, eli jotain järkevää todennäköisesti sain tehtyä tähän liittyen maanantaina. Uusi virhe olikin vastaavanlainen mutta eri luokasta. Korjasin tähän injektoitavan riippuvuuden oikeaksi. Seuraavassa toimituksessa sovelluksen pitäisi kaiken järjen mukaan käynnistyä.

Viikkoanalyysi

Viikosta jäi yleisesti ehkä vähän ristiriitaiset tunnelmat. Itsenäinen Docker opiskelu oli mielestäni antoisaa, kun taas töiden puute kävi melko raskaaksi viikonlopun lähestyessä. Sovelluksen versiotoimituksen lähestyessä uutta tekemistä ei myöskään ollut järkevää keksiä liikaa. Korjasin siis sovelluksen bugeja sitä mukaa kun testaajat niitä löysivät. Mielestäni osaan hyvin selvittää mistä erilaiset bugit johtuvat ilman sen suurempaa taistelua. Kassatiimin sovelluksen ongelmat tuntuivat selviävän heti kun sain heiltä tarvittavat lokitiedostot.

Erityisen hankaliksi osoittautuvat aina sellaiset ongelmat, jotka koskevat Solr indeksointijärjestelmää. Viikon aikana ilmaantuneet indeksointivirheet johtuivat kaikki siitä, että uusia henkilöitä luodessa indeksiin lisättävältä henkilöltä puuttui pakollisia tietoja. Esimerkiksi se missä sovelluksissa tämä henkilö on käytössä. Ongelmallisen tästä tekee se, että en Java-koodista löytänyt mitkä kentät olivat pakollisia. Nämä olisivat varmaan ollut jossakin XML-mäppäyksissä. Tätä kenttää ei myöskään oman käsityksen mukaan edes käytetä missään, ja uudelleenindeksoinnin ajaminen hakee tähän tietokantaan kovakoodatun arvon, joten tässä ei muutenkaan ole mielestäni järkeä.

Docker opiskelusta pidin tällä viikolla suuresti. En koe, että itseopiskeluun käytetty aika olisi heitetty hukkaan, vaikka Docker ei yksikössämme olekaan suuressa käytössä. Ensimmäinen haaste oli Dockerin ja VirtualBoxin yhteensopimattomuus. Ongelma kuitenkin ratkesi helposti luomalla uusi virtuaalikone Dockeria varten. Olin tätä tarkoitusta varten luonut simppelein webbisovelluksen Pythonilla ja Flaskilla. Kirjoitin tästä ohjeistuksen muistiin, koska tästä on varmasti hyötyä tulevaisuudessa itselleni ja muille

```

1. Luodaan docker machine:
docker-machine create --driver=virtualbox dockerbox
Docker luo uuden virtualbox koneen, jossa imageja voidaan suorittaa.
Docker ja virtualbox eivät siinä mielessä ole yhteensopiva toistensa kanssa,
koska Docker vaatii Windowsista Hyper-V päälle, mutta virtualbox ei toimi tämän kanssa.

2. Otetaan machine käyttöön:
docker-machine env dockerbox

3. Edellinen komento kertoo mikä komento pitää suorittaa, jotta machine otetaan käyttöön sen hetkessä komentorivissä
Powershell:
& "C:\Program Files\Docker\Docker\Resources\bin\docker-machine.exe" env dockerbox | Invoke-Expression

(Stepit 2 ja 3 pitää ajaa uudestaan per komentorivi)

4. Nyt voidaan tehdä docker image projektin juuressa. Ajoin komennon:
docker build --tag=flasktest .
--tag antaa imagelle nimen, jolla sen voi myöhemmin ajaa.

5. Aja buildattu image komennolla:
docker run -d -p 8080:80 flasktest
-d eli detach. Ei näytetä sovelluksen outputtia ajokonsolissa.
-p eli publish. Määrittää mihin porttiin containerin portti määpätään.
Demoapplikaatiossa määritän sovelluksen pyörimään osoitteessa 0.0.0.0:80

6. Tarkista machinen ip:
docker-machine ip dockerbox
192.168.99.101

7. Demosovellus vastaa osoitteesta:
http://192.168.99.101:8080/hello

Optional:
Jotta saa VS Coden docker lisäosan toimimaan,
tarvitsee VS Coden asetuksiin syöttää docker-machine env dockerbox komennolla saadut arvot seuraaviin asetuksiin:
docker.host
docker.certPath
docker.tlsVerify
docker.machineName

```

Docker ohjeistus

Microsoftin Visual Studio Coden Docker laajennuksen ohjeissa on selitetty nämä hyvin. Asetukset on myös mahdollista antaa käyttöjärjestelmän ympäristömuuttujina. (Microsoft. 2019. Viitattu 15.09.2019)

Dependency injectionilla viitataan objekteille annettaviin instanssimuuttujiin. Erilaisissa frameworkeissa tämä on yleensä hoidettu niin, että luokissa kerrotaan erikseen mitkä muuttujat injektoidaan. Näin säästytään siltä, että kaikki pitäisi luoda itse. Esimerkiksi rajapintasovelluksissa controllerille injektoidaan jokin service, ja servicelle injektoidaan repository. Tällä myös varmistetaan se, että sovelluksen eri osat eivät ole riippuvaisia toisistaan, ja sovellus toimii myös uusien toteutusten ympärillä.

James Shore on jo vuosia sitten kuvannut riippuvuuksien injektointia lyhyesti sanoin ”Dependency injection means giving an object its instance variables. Really. That’s it.” (James S. 2006. Viitattu 15.09.2019)

3.2 Seurantaviikko 2

Maanantai 26.08.2019

Uusi viikko alkaa näillä näkymin jonkinlaisella suunnittelulla sovelluksemme käyttöliittymän testaamiseen. Olisi hyvä senkin osalta saada tietää, jos jokin toiminnallisuus hajoaa. Ra-

japintoja meillä testataan jo Robot Frameworkia käyttäen, ja tätä todennäköisesti hyödynnettäisiin myös käyttöliittymien osalta. Kasaan varmaan jonkun pienen demosetin testejä, jotka käyn läpi testaajamme kanssa.

Iltapäivästä on esimiehen kanssa one to one keskustelu, jonka alussa käydään ensin arkkitehdin kanssa läpi tulevia työtehtäviä. Jos nyt viimeistään tänään loppuisi näiden hommien kanssa arpominen. Tarkoitukseni olisi näiltä osin myös tuoda esiin nykyinen palkka, titteli ja nykyisten työtehtävien mielekkyys. En välttämättä koe, että nykyiset tehtäväni ovat sellaisia joihin osaamiseni parhaiten sopii. Tuntuu siltä, ettei näissä ole mahdollisesti varaa kehittyä, ja tulevaisuuden kannalta näissä ei varsinaisesti ole mitään hyötyä siksi, että käytössä olevat teknologiat ovat jo niin vanhoja. Tuntuu että olen urani aikana saanut tehdä ihan tarpeeksi legacysovelluksia. Olisi ehkä kiva päästä tekemään vaihteeksi jotain uutta.

Yllättävää oli, että osalle käytössä olevista sovelluksista löytyi jo robottitestejä, mutta ne eivät ole ajossa missään. Yllätykseksi huomasin, että testit eivät lähde edes käyntiin eri kirjastojen versioiden yhteensopimattomuuden takia. Tarkoitus olisi nyt kuitenkin varmaan saada korjattua nämä olemassa olevat testit ja keksiä sitten lisää uusia.

Esimiehen kanssa saatiin käytyä läpi tulevia työtehtäviä. Vastuulleni tulee ainakin enemmän yksikön SVN Git siirtymän tehtäviä, sekä "tehtävien siirto"-projektin tehtäviä. Jälkimmäinen ei ollut vielä täysin selvä itselle, mitä se tulee sisältämään, mutta toiveet sen suhteen on ainakin kovat. Lisäksi pääsisin myös mukaan yksikön arkkitehtuuria korjaavaan projektiin, jonka ideana on ottaa arkkitehteja yksikön ulkopuolelta antamaan mielipiteitä siitä mitä täällä tehdään väärin.

Tiistai 26.08.2019

Päivä jatkuu robottitestien viilaamisella. Ongelmaksi on koitunut pitkälti versioiden yhteensopimattomuudesta johtuvat ongelmat. Liikaa ei uskalla päivittää versioita sen takia, että se vaatisi vähän enemmän muutoksia projektien koodeihin. Kaikki projektit ovat kuitenkin yhden master projektin alla, ja näin ollen käyttävät samoja versioita kirjastoista. Myöhemmin pitäisi myös tehdä merge prodfix haaraan.

Merge siirtyy huomiselle aikaisempien ongelmien takia. Testien suhteen pääsin vähän eteenpäin. Löysin kirjastoista sellaiset versiot, joihin päivittämällä tarvitsee korjata vain aikaisemmin tehdyt testit. Olisin tietenkin mieluummin ottanut paljon uudemmat versiot mutta tässä vaiheessa se ei liene järkevää. Esimerkiksi Google Guavan päivittäminen olisi

ollut Robot Frameworkia varten välttämätöntä, ja Guavan päivittäminen olisi vaatinut joko päivityksen Javan versioon 8, tai kaikkiin Predicate-luokkaa käyttäviin luokkiin olisi tarvinnut toteuttaa oletusmetodiksi merkitty metodi. Java 7 ei tue interfacejen oletusmetodeja, joka koitui hieman ongelmalliseksi. Näitä lisäyksiä olisi pitänyt tehdä niin moneen paikkaan, etten kokenut sitä järkeväksi tässä tilanteessa

Keskiviikko 28.08.2019

Tänään päivä menee suurimmaksi osaksi palaverissa. Ehdin aamulla käyttää hieman aikaa robottitestien korjaamiseen ennen kuin käydään jotakin asioita läpi arkkitehtuurin suhteen. Palaverin sisällöstä en ole täysin varma, mutta kokousvaraus on tehty nimellä Tuotekehitys (kassa/liitto/tekninen). Prodfix merge pitäisi luultavammin tehdä myös tänään, ja iltapäivästä on backlogin groomaus, jossa mahdollisesti tulee lisähommaa. Tähän on toimitettu etukäteen lista groomattavista tehtävistä, ja listalla näyttäisi olevan ainakin pari hommaa, jotka vaativat Java-työtä.

Groomauksen jäljiltä tuli muutama Java-työ. Työmääräarviot näihin oli tosin vain muutamman tunnin luokkaa. Merget jäi taas tekemättä koska CI buildi näytti punaista, joskin tähän oli syynä palvelimen levytilan loppuminen.

Testejä korjattaessa törmäsin moneen ongelmaan. Sovelluksen sivu on sen verran muuttunut ajan myötä, että vanhat CSS-selectorit eivät toimineet enää. Vaihdoin nämä uusiin XPath-selectoreihin ihan vaan siksi koska itselle helpompi niin, ja saa ehkä vähän tarkemmin sen elementin minkä oikeasti haluaa. Toinen ongelma oli se, että uuden henkilön luovassa testissä haetaan kolmannen osapuolen palvelusta henkilötunnus. Tämä generoitu henkilötunnus voi olla käytännössä miltä tahansa aikaväliltä, ja sovelluksemme testi, jossa liitetään jäsenelle jäsenyys ei mene läpi osaa ajasta siksi, että generoitu henkilö on liian vanha tai nuori. Kolmannen osapuolen rajapinta ei suoraan mahdollista tätä. Pitäisi käyttää rajapintaa, jota tämä sivu itsessään käyttää. Siihen olisi mahdollista syöttää nimi ja maksimi syntymävuodet POST kutsulla. Tämä vaatii lisää testaamista huomenna.

Torstai 29.08.2019

Aamu alkaa tänään mergejen hoitamisella, koska buildi näyttää vihdoinkin vihreää. Suunnitelma on mergetä trunk prod haaraan, jonka jälkeen vielä prod pitäisi mergetä prodfix haaraan. Näiden vaiheiden välissä pitäisi vielä lokaalisti ajaa tunnin kestävä buildi. Mergejen jälkeen siirryn ratkomaan eilen ilmenneitä ongelmia robottitestien suhteen ja mahdollisesti luomaan uusia testejä. Meillä on wikiin kirjattu testitapaukset, jotka tulisi suorittaa aina

asiakkaille asennuksen yhteydessä. Nämä smoke-testit sisältävät ihan perus testejä sille, ettei mikään ole hajonnut. Alunperinkin ideana taisi olla se, että nämä testit saataisiin automatisoitua.

Merget osoittautuivat taas jokseenkin ongelmallisiksi. Trunkista prodiin onnistui helposti, joten oletin, että prodista prodixiin menisi myös helpolla. SVN on ilmeisesti keksinyt jotain omaa, ja jälkimmäisessä mergessä saatiin älyttömiä conflicteja, joita ei varmasti olisi pitänyt edes tulla. Näitä pähkäillessä menikin käytännössä loppupäivä, kun eri vaihtoehtoja kokeilemalla osa tiedostoista oli aina väärässä tilassa ja jouduin aloittamaan mergen alusta. Päivän loppuksi luulen saaneeni mergen siihen tilaan, että sen uskaltaisi commitoida, mutta buildin ajaminen sai jäädä huomiseksi.

Sain tehtäväksi pistää asiakkaiden REST-sovellukseen autentikoinnin päälle. Huomasin asiakkaiden ympäristöjen olevan ehkä vähän huonosti dokumentoitu, sillä en meinannut helposti löytää mistä ympäristöstä on kyse ja millä tunnuksilla pääsen sinne sisään. Näiden sotkujen jälkeen jouduin käynnistellä Tomcatteja uudestaan pariinkin kertaan, koska asiakkaan ympäristöön ei ilmeisesti ollut järkeviä käyttäjätunnuksia palveluiden testaamiseen.

Perjantai 30.08.2019

Heti aamusta oli buildinäytössä punaista eilisten prod mergejen jäljiltä. En ymmärrä miksi merge ei muka ollut ok, kun se meni lokaalisti läpi. Ehdin selvittää tiedoston, jota ajaessa buildi hajoaa. Näyttäisi että SVN ei osaa tehdä monen revisionin mergeä tämän tiedoston osalta oikein. Esimerkiksi jos tein mergen vain revision 1 ja HEAD välillä, tiedosto ei vastannut ollenkaan sitä mikä lähdehaarassa oli. Tein merget pienemmissä osissa ja nyt se näyttäisi olevan kunnossa. Jos tästä sotkusta selviäisi tämän päivän aikana.

Tuntui että koko päivä meni ihan hukkaan mergesotkuja selvittäessä. Töissä ei ollut paikalla oikein osaavia henkilöitä, jotka ovat SVN ongelmia kanssa joutuneet ennen taistelemaan. Yritin parhaani selvittääkseni tämän tilanteen, mutta yhden ongelman jälkeen vastassa oli aina toinen. En pysty ymmärtämään miten yksisuuntaisessa mergessä tulee conflicteja ikinä, ja miksi mergen lopputulos ei aina ole sama, vaikka suoritettu operaatio on täysin sama. Sillä ei pitäisi olla mitään merkitystä mergetäänkö kaikki revisiot kerralla vai osissa. Päädyin lähtemään aikaisemmin kotiin säästääkseen omaa mielenterveyttä. Maanantaina osaavien henkilöiden kanssa katsotaan tämä läpi, ja todennäköisesti nyt yritän aloittaa viemään sitä Git siirtymää eteenpäin. Nämä SVN hommat ovat koko työssäoloikani olleet käsittämättömiä haasteita, eikä tällainen säätäminen ole kenenkään eduksi.

Viikkoanalyysi

Tuntuu siltä, ettei koko viikosta jäänyt mitään käteen. Joka päivä joko piti miettiä mergejä tai sitten yrittää suoriutua niistä. Päästiin keskiviikkona kuitenkin ymmärrykseen siitä, että tulevaisuuden kannalta kolme isointa ongelmaa ovat SVN, Bamboo ja Java 7. Näistä olisi tarkoitus ilmeisesti tämän syksyn aikana siirtyä pois ja minun pitäisi olla tässä tiiviisti mukana. SVN Git siirtymä aloitetaan todella pian, ja nuo muut sitten siihen perään. Bamboo koettiin ongelmaksi, koska sen build agentit eivät tue isompaa Java versiota, eli Java jää tämän takia jälkeen. Java 7 on taas ongelma siksi että emme voi hyödyntää mitään uutta. Viikon aikana yritin päivitellä niitä testejä, ja ongelma oli oikeiden versioiden löytäminen niin, että varsinaista bisneslogiikkaa ei tarvitse päivittää. Yhdet versiot olivat sellaisia, että olisi tarvittu Java 8, koska Java 7 ei tue interfacejen oletusmetodeita.

Todettiin viikolla tilanteeni olevan siinä mielessä hyvä, että aikaisemmin yksikössä ei ole ollut mahdollista irrottaa projektista henkilöitä tekemään tällaisia siirtotöitä. Olisihan se tietenkin kiva, jos pääsisi alusta asti tekemään töitä sellaisilla välineillä mitkä olisivat niin sanotusti alan standardeja, mutta toistaiseksi olen tyytyväinen siihen, että saan olla mukana tärkeiden teknologioiden käyttöönotossa. Ehkä minulla on enemmän varaa lähteä haastamaan muita tässä, kun olen uutena työntekijänä talossa, kun taas muut ovat olleen töissä parikymmentä vuotta ja ehkä vähän käsitys hyvistä käytännöistä muuttunut.

Mergeongelmia ei saatu selvitettyä millään järkevällä tavalla. Talon sisällä kukaan ei tuntunut ymmärtävän, miksi SVN mergessä kaikista tiedostoista ei tule oikea versio. Omalta osalta yritin saada mergeä läpi esimerkiksi erilaisilla asetuksilla ja valinnoilla. Vaikka valitsin kaikista accept incoming, tiedosto tuli silti joissain tilanteissa väärin. Vastaavanlaisia tapauksia tapahtui useammassa haarassa ja näiden eroavaisuuksien selvittämiseen meni älyttömästi aikaa. Git siirtymästä olisi siksi hyötyä, että mergeäminen on tehty siellä niin helpoksi. Stackoverflowissa kysymykseen "How and/or why is merging in Git better than in SVN?" vastannut käyttäjä Spoike kertoo pitkän kaavan kautta, miksi Subversionin merget menevät helpommin solmuun. Subversionin mergeongelmat johtuvat sen suunnitteluongelmista. Subversion ei ota kantaa siihen, mitä on mergetty ja milloin, vaan on ainoastaan kiinnostunut siitä, että jotain on mergetty. (Käyttäjä Spoike. 2010. Viitattu 16.09.2019) Tämä on todennäköisesti ollut myös syynä siihen, että revisioiden mergeäminen osissa pääsi lähemmäs lopputulosta.

Toivoisin että nämä asiat olisi ratkottu paljon aikaisemmin. Kuulemma siirtymä Gittiin olisi pitänyt tehdä jo kesän aikana, mutta se on syystä tai toisesta koko aika vain siirtynyt. En

pidä sitä kovin järkevänä, että niin sanotut avainhenkilöt eivät halua tehdä näitä asioita. On tietenkin hyvä saada itse jotain töitä, mutta tällainen muiden töiden siirtäminen tuntuu hölmöltä. Käydyissä palavereissa ei oikeastaan tullut esille sitä, miksi nämä kaikki ovat siirtyneet tällä tavalla. Muuta kuin jotain naureskelua.

3.3 Seurantaviikko 3

Maanantai 02.09.2019

Aamulla oli onneksi perjantain jälkeisten mergejen jäljiltä builditaulu odottamassa vihreänä. Jos tämän päivän aikana saisi tehtyä vielä viimeisen mergen. Tähän tulee todennäköisesti menemään tänään käytettävissä oleva työaika, sillä iltapäivästä on asiakasdemo ja seuraavan sprintin suunnittelu.

Mergen osalta menee varmaan vielä pieneksi taisteluksi, sillä puhtaan virtuaalikoneen pystyttämässä on aina jonkun verran säätämistä. Luulisi että koneen pohjaimage olisi siinä kunnossa, että voisi ajaa gradle buildin suoraan. Vaatii jonkunlaista säätämistä exportattaviin ympäristömuuttujiin ja uudelleenkäynnistelyä.

Prodista prodfixiin merge ei ilmeisesti ollut ihan niin suoraviivainen juttu kuin olisi voinut kuvitella. Aluksi virhe oli kantapakettien suorittamisesta, jonka jälkeen muka Javatestit olisivat hajonneet. Ajoin buildin useamman kerran varmistaakseni sen, ettei kyse ollut mistään ympäristöongelmista. Vastaavanlaista on joskus ennenkin ollut esimerkiksi siksi, että tietokantaan ei saa yhteyttä, tai että buildikoneen muisti on loppunut. Jätin buildit pyörimään päivän päätteeksi ja jatkan tästä aamulla.

Tiistai 03.09.2019

Toinen buildi ei mennyt läpi eri testin takia kuin aikaisemmin. Kuulemma tässä olisi saatanut olla kyse jostain testidatan kovakoodauksista. Toinen buildi ei taas ollut edes ajanut koska Bamboo Agent oli yön aikana kaatunut. Näitä olisi nyt tänään tarkoitus yrittellä selvittää toisten projektien palaverien ohessa. Palavereissa katsotaan arkkitehtuuriasioita, sekä myöhemmin iltapäivällä tehtävien siirtoprojektin aloituskokous.

On lähes varmaa, että prodfixin testit hajosivat siksi, että palvelimella oli muisti lopussa. Tietokantaosaajien kanssa katsottiin proseduureja käsin kutsuttaessa tulevan virheen olevan tähän liittyvää. Tämä ei taas enää kuulu omalle vastuualueelleni, joten testit odottelevat nyt tätä. Jonkun muun olisi kuitenkin vielä tarkoitus katsoa, että mergeissä ei ole

jäänyt puuttumaan mitään. Pääsen siis itse kuitenkin irti näistä mergetöistä ainakin hetkeksi.

Arkkitehtuuripalaverissa käytiin läpi viimeisen kuukauden aikana suoritettuja hommia, ja sitä mitä olisi järkevä tehdä seuraavaksi. SVN -> Git siirtymä olisi hyvä aloittaa nyt. Siihen liittyen varmaan huomisessa tuotehallintapalaverissa katsotaan tarkemmin mitä lähdetään aluksi tekemään. Tarkoitus on kuitenkin ensiksi siirtää Core sovelluksen kehitys Gittiin, jonka jälkeen päästään siirtämään kaikki muut. Tämä edellyttää ainakin sitä, että Core olisi saatavilla riippuvuutena jostain.

Tehtävien siirtoprojektin palaverissa käytiin hieman läpi sitä, mitä projekti pitää sisällään ja mitä projektin suhteen on jo mietitty aikataulujen ja tehtävien puolesta. Vielä on hieman epäselvää mikä oma lopullinen roolini tulee tässä olemaan, mutta todennäköisimmin se kuitenkin liittyy rajapintojen kehitykseen.

Keskiviikko 04.09.2019

Päivä näyttää taas melko raskaalta palaverien suhteen. Aamupäivästä sidosryhmien kanssa palaveri tehtävien siirrosta, ja iltapäivällä uusi tuotehallintapalaveri sekä versioitoimitusten retrospektiivi yksikölle.

Ajattelin työstää robottitestejä siltä osin kuin palaverien välissä ehdin. Tarkoitus olisi niihin saada ainakin kirjoitettua omaa logiikkaa henkilötunnuksen luontiin, jotta voidaan luopua kolmannen osapuolen toteutuksesta. Vastaavanlaisen koodipätkän olen kuitenkin jo aikaisemmin kirjoittanut pariinkin kertaan, eikä validin henkilötunnuksen generointi vaadi juuri-kaan mitään ihmeitä. Testit pitäisi saada näin korjattua loppuun, joten pääsisin tekemään uusia testejä. Uusien testien osalta olisi varmaan järkevää pitää palaveri testaajien kanssa, jotta saisin paremman käsityksen siitä, minkälaisia testejä tarvitaan. Wikissä on kuvattu sanallisesti nämä tapaukset, mutta olisi hyvä nähdä ihan oikeasti mistä nämä toiminnallisuudet ovat saatavilla.

Onnistuin päivän aikana hajottamaan Bamboossa pyörivät regressiotestit. Ilmeisesti Seleniumia ja Robot Frameworkia ei saanutkaan päivittää liikaa Java 7 takia. Kaikki ongelmat tuntuvat aina loppujen lopuksi johtavan näihin samoihin päätelmiin. Kummastakaan ei saanut olla versio suurempi kuin 3.0.

Versioitoimituksen retrossa saatiin ihan hyvin ongelmakohdista keskusteltua. Esille nousi se, miten joidenkin palvelinten yliheittoja on jätetty täysin kesken, ja se että ne pitäisi hoi-

taa loppuun. Lisäksi ongelmakohtana pidetään sitä, että uusien versioiden smoke-testaamiseen menee kohtuuttoman paljon aikaa. Tähän pohdittiin ratkaisuksi testiautomaatiota. Kuulemma sitäkin on joskus yritetty, mutta luovuttu siksi että testit olleet liian epävarmoja. Tässä vaiheessa oli hyvä tuoda esille se, että teen omassa tiimissäni pientä demoa näistä testeistä. Näitä olisi kai tarkoitus saada oikeasti käyttöön, jos saan nämä testit järkevään kuntoon. Pitää siis yrittää panostaa tähän, että nämä nyt oikeasti saataisiin.

Torstai 05.09.2019

Pääsen tänään jatkamaan testien tekemistä. Tämän hetkinen tilanne on mielestäni hyvä siksi, että saan aika pitkälti itse päättää mitä teen. Tänään listalla on siis jostakin sovelluksesta nappien XPath-selectorien kasaamista Robot Frameworkin resursseiksi, ja mahdollisesti ihan perustestien tekeminen. Varaan ehkä iltapäivälle tai huomiseksi testaajan kanssa palaverin, jossa katsottaisiin mitä testejä oikeastaan pitää tehdä, ja mitkä olisivat mahdollisesti hyviä tapoja tehdä testejä niin, että ei välttämättä aina vaatisi isompaa osaamista.

Testeissä ongelmaksi osoittautui se, että elementit eivät välttämättä ole näkyvissä sillä hetkellä, kun niitä yritetään klikata. Ratkaisuksi kelpaa aika hyvin Seleniumin "Wait Until Element Is Visible" metodi. Ongelmana on silti se, että vaikka elementit ovat näkyvissä, niiden sisältö ei välttämättä ole täysin ladannut. Tämä johtaa siihen, että asioita yritetään silti tehdä liian aikaisin. Ei ole missään määrin hyvä käytäntö viljellä koodiin sleeppiä, joka pidentäisi testien ajoaikaa. Vaatii varmaan vaan lisää tarkistuksia, jotta tiedetään milloin elementtejä saa oikeasti klikata.

Perjantai 06.09.2019

Perjantai jatkuu eilisten testiongelmien selvittelyllä. Lisäksi olisi myös mahdollista katsoa vähän rikkinäistä buildia, joka on toisen kehittäjän jäljiltä. Sähköpostissa oli odottamassa viestin, jossa luki "Rikoin buildin. Saat katsoa, jos halut/ehdit". Tämä nyt ei välttämättä ole kiireellinen, ja ei tunnu kovin järkevältä lähteä korjaamaan toisen aiheuttamia ongelmia, josta ei tiedä mitään. Kyseinen henkilö on itse tänään vapaalla, eli katson tätä iltapäivästä, jos aikaa riittää. Pidän tällä hetkellä näitä testejä paljon tärkeämpänä. Myös aikaisemmin mainittua Git siirtoa pitäisi jo pikkuhiljaa aloitella, mutta se varmasti menee ensi viikolle.

Ehdin päivän aikana saamaan regressiotestit menemään vihreälle. En mene ihan takuuseen siitä, että ne menisivät läpi ihan joka kerralla, mutta tästä on ainakin hyvä jatkaa

eteenpäin. Loppupäivästä käytin aikaa hieman kyselyyn siitä, mitä steppejä SVN Git siirtymä edellyttää. Pääsin kokeilemaan Git-komennon ajoa, jonka pitäisi kloonata SVN repository Git repositoryksi. Näytti päällisin puolin toimivan ihan hyvin. Reilun 40 000 revision prosessointi kesti aika tarkalleen tunnin, jonka jälkeen pääsin selvittämään lisää ongelmia. Idea on kuitenkin se, että isommasta master projektista revitään projektit omiin Git repositoryihin. Master projektin build.gradlessa on erittäin paljon toiminnan kannalta merkittäviä taskeja, jotka tulisi nyt yksitellen siirtää irrotetulle projektille. Tämä koituu varmasti haasteelliseksi, kun toimenpide pitää suorittaa kaikille projekteille. Tästä jatkuu ensi viikolla

Viikkoanalyysi

Tämä viikko tuntui yleisesti ottaen älyttömästi paremmalta aikaisempiin viikkoihin verrattuna. Tuntui kun olisi päässyt tekemään työtä, jolla on tarkoitus. Mieleen painuvin tehtävä oli perjantai iltapäivästä SVN Git siirtymän aloittaminen. Tätä olen ainaki itse odottanut kauan, ja nyt pääsen sitä itse tekemään. Tällä viikolla muutenkin selvisi aika hyvin kaikki tulevat työtehtävät ja se, mitä ne pitävät sisällään.

Perjantaina vähän harmitti lähteä kotiin, kun pääsin vihdoinkin tekemään kauan odotettua Git siirtymää. Ehdin sen verran kokeilla veden pintaa, että siirsin yhden osan master projektista Gittiin. Tässä vaiheessa kuitenkin huomasin, että gradle buildi ei mitenkään mene läpi, ja sen että tämä tulee vaatimaan paljon työtä ja lisäopiskelua. Gradle ei ole itselleni kovin tuttu, muuten kuin vain riippuvuuksien hallinnan työkaluna. Saan tähän tietenkin tarvittaessa apua, mutta haluan yrittää selvittää alkuun itsenäisesti. Liian pitkäksi aikaa kuitenkin en aio jäädä miettimään. Tulin myös siihen tulokseen, että kaikki tähän liittyvä tulee dokumentoida erityisen tarkkaan. On tärkeää, että vastaavanlaiset toimenpiteet voidaan tulevaisuudessakin suorittaa hyvien ohjeiden kanssa, eikä kaikkea tarvitse keksiä uudelleen.

SVN repositoryn muuttaminen Git repositoryyn kävi helposti ajamalla komento "git svn clone *repositoryn osoite*", jolloin Git alkoi käymään svn repositoryn historiaa läpi alusta asti. Näin isolle projektille se oli tunnin homma tehdä muutos. Historiatiedoissa kuitenkin on paljon kaikkea, mikä ei edes repositoryyn päädy. Ensimmäisenä siirrettävä projekti on kuitenkin erittäin pieni osa tätä kokonaisuutta.

Tehtävien siirtoprojektin sisältöön sain arvokasta lisätietoa, vaikka arkkitehtimme sanoikin tiistaina, ettei minun välttämättä kannata osallistua palaveriin, koska niiden sisältö ei yleensä ole ollut kovin tärkeää. Onneksi kuitenkin ehdin tähän osallistua, koska heti pala-

verin jälkeen oli tämän suhteen paljon selkeämpää mistä koko hommassa on edes kyse. Tässä oli uusina teknologioina RabbitMQ, joka on avoimen lähdekoodin viestijonopalvelu. Vastaavanlaisista järjestelmistä olen aikaisemmin käyttänyt MQTT protokollaa julkisen liikenteen seuraamiseen.

RabbitMQ on avoimen lähdekoodin viestinvälittäjä-sovellus, joka hyödyntää Advanced Message Queuing protokollaa. CloudAMPQ:n Lovisa Johansson tiivistää hyvin sen, milloin viestinvälittäjiä on hyödyllistä käyttää. Viestinvälittäjää on hyödyllistä käyttää silloin, kun viestejä vastaavat operaatiot saattavat kestää hieman. Näin saadaan käyttäjälle suoraan vastaus ilman että vaaditaan odottelua. Tietenkin lopputulosta joudutaan silti odottamaan, mutta parhaimman käyttökokemuksen kannalta nopeat vastaukset ovat tärkeitä. (Lovisa J. 2015. Viitattu 15.09.2019)

Ymmärsin että RabbitMQ tarkoitus olisi toimittaa viestit siitä, jos jotakin tietoja muuttuu. Varsinaiset tiedot tulee sitten hakea rajapinnasta erikseen. Erikoisena ratkaisuna tähän oli se, että viestijono ei käytä samoja yhteyksiä kuin varsinainen rajapinta. Tähän tulisi siis rakentaa omat yhteydet ympärille.

Kokonaisuudessaan viikosta jäi erittäin positiivinen mielikuva. Seuraavaa viikkoa ainakin jaksaa odottaa innolla. Tuntuu että olemme yksikössä saaneet nyt järkevämmiin ihmisiä työllistettyä osaamisen ja toiveiden mukaan.

3.4 Seurantaviikko 4

Maanantai 09.09.2019

Viikonlopun jälkeen löydän edestäni lisää SVN Git siirtymän tehtäviä. Tänään olisi tarkoitus korjata gradle buildi, puskea perjantaina muutettu repository Gittiin ja alkaa GitLab CI työstäminen. Nähtäväksi jää kuinka suuren työmäärän tämä aiheuttaa. Gradle ja GitLab CI eivät kumpikaan entuudestaan ole kovin tuttuja, mutta ymmärrän kuitenkin mihin näitä käytetään ja mikä niiden tarkoitus on. Ei siis pitäisi olla mikään ongelma ottaa selvää. Yksiköstämme ei taida kovin montaa osaajaa löytyä näistä, vaikka aikaisemmin onkin jotain tehty.

Koko päivä osoittautui erittäin onnistuneeksi. Sain gradle buildin menemään läpi niin sanotusti leikkaamalla ja liimaamalla. Sen sijaan että olisin lähtenyt kasaamaan uutta build tiedostoa, sain tämän toimimaan hyvin tuomalla vanhat build-tiedostot ja poistin niistä viitaukset aikaisempiin aliprojekteihin.

GitLab CI kanssa pääsin myös hyvin vauhtiin. Jouduin tosin väliaikaisesti pystyttämään GitLab Runnerin lokaaliin virtuaalikoneeseen. Tähän pitää saada jostain joku pysyvämpi ratkaisu. Varmaan tällaisen tilaaminen IT-tuesta. Varsinaiset CI ohjeet olivat toistaiseksi yllättävän helpot. GitLab CI tukee kolmea vaihetta: build, test ja deploy. Deploy vaiheen jätin toistaiseksi tekemättä, koska mielestäni siihen kohtaan tulisi artefaktien lähettäminen Maven repositoryyn. Tämä ei kuitenkaan vielä ole mahdollista. Pitää myös tarkistaa projekti eri ympäristöjen kannalta. Toin vanhasta projektista vain lokaaliin kehitykseen tarvittavat propertyt, mutta näitä on ilmeisesti jotain muitakin.

Tiistai 10.09.2019

Eilisen jäljiltä on hyvä päästä jatkamaan Git siirtymää. Minulle oli kuitenkin laitettu joku tehtävä uuden rajapinnan luomiseen, joka kutsuisi päivitettyjä tietokantaproseduureja. Luulen että tämä pitäisi saada pois alta ensin. Tarkoitus on luoda rajapinta, joka vain ottaa vastaan enemmän rajauseräparametreja. Tämä taitaa olla asia, jonka kanssa on taisteltu ennenkin. Tarpeet muuttuvat sen jälkeen kuin jotain on jo tehty.

GitLabin osalta joudun käydä arkkitehdin kanssa keskusteluja siitä, miten ja milloin saadaan artefaktit Maven repositoryyn, sekä mistä saataisiin palvelin pyörittämään GitLab Runneria. Myös eri ympäristöjen buildpropertyt tulisi vielä tarkistaa.

Arkkitehdin kanssa käydyt keskustelut osoittautuivat turhiksi. En saanut haluamiani vastauksia mihinkään. Vielä on epäselvää mihin GitLab Runner voidaan asentaa, ja millä tunnuksilla ja mihin repositoryyn projektin artefaktit voidaan lähettää. Ajattelin jos testimielessä lähettäisin ne lokaaliin Maven repositoryyn. Näin pääsisin tämän suhteen edes hie- man eteenpäin.

Osa iltapäivästä meni yksikön kuukausikokouksessa, sekä tehtävien siirtoprojektin viikoit- taisessa palaverissa. Kuukausikokouksessa käytiin läpi, mikä asia on kenenkin vastuu- alueella, mihin on tulossa lisää työntekijöitä, lähtevät työntekijät sekä avoinna olevat rekry- toinnit. Tehtävien siirtoprojektin viikkopalaverissa otettiin nopea yhteenveto viime viikolla olleesta teknisestä palaverista, ja pohdittiin siellä ilmenneitä asioita. Kummalliseksi todet- tiin se, että herätteet eivät käytä palveluväylää, vaan niitä varten tulee rakentaa omat yh- teydet.

Keskiviikko 11.09.2019

Git siirtymän edistäminen ei luultavasti tänään edisty, sillä arkkitehdit ovat poissa toimistolta. Onneksi eilen en saanut jäsenrekisterin viestinlähetyksrajapintaa muutettua, koska generoidut Java-luokat eivät olleet oikein. Tässä on ainakin vähän tekemistä tälle päivälle. Yritän luultavasti vielä kokeilla sitä, miten nykyiset projektit toimisivat, jos poistan Gittiin siirretyn projektin SVN repositorysta lokaalisti. Tämä ei sinänsä vielä ole ajankohtaista, mutta helpottaa ainakin tulevaisuudessa, jos on jo kokeiltu, aiheuttaako tämä muita muutoksia.

Sain päivän päätteeksi kysyttyä arkkitehdiltä, onko asiat Nexus repositoryn suhteen edenneet. Kuulemma näistä vastaava henkilö IT servicestä on tämän viikon lomalla, joten jouduu odottaa ensi viikkoon.

Ehdin vähän kokeilla käyttää irrotettua projektia riippuvuutena muissa projekteissa. Ilmeisesti ei onnistu ihan niin helposti, koska eihän kirjaston riippuvuuksia voi käyttää sen ulkopuolella. Esimerkiksi springit ja muut pitäisi erikseen listata riippuvuuksina, tai jotenkin saada jar-tiedostot niistä. Nykyisellään kirjasto, joka tuo kaikki riippuvuudet, puskee nämä lokaaliin ja tomcatin lib kansioon.

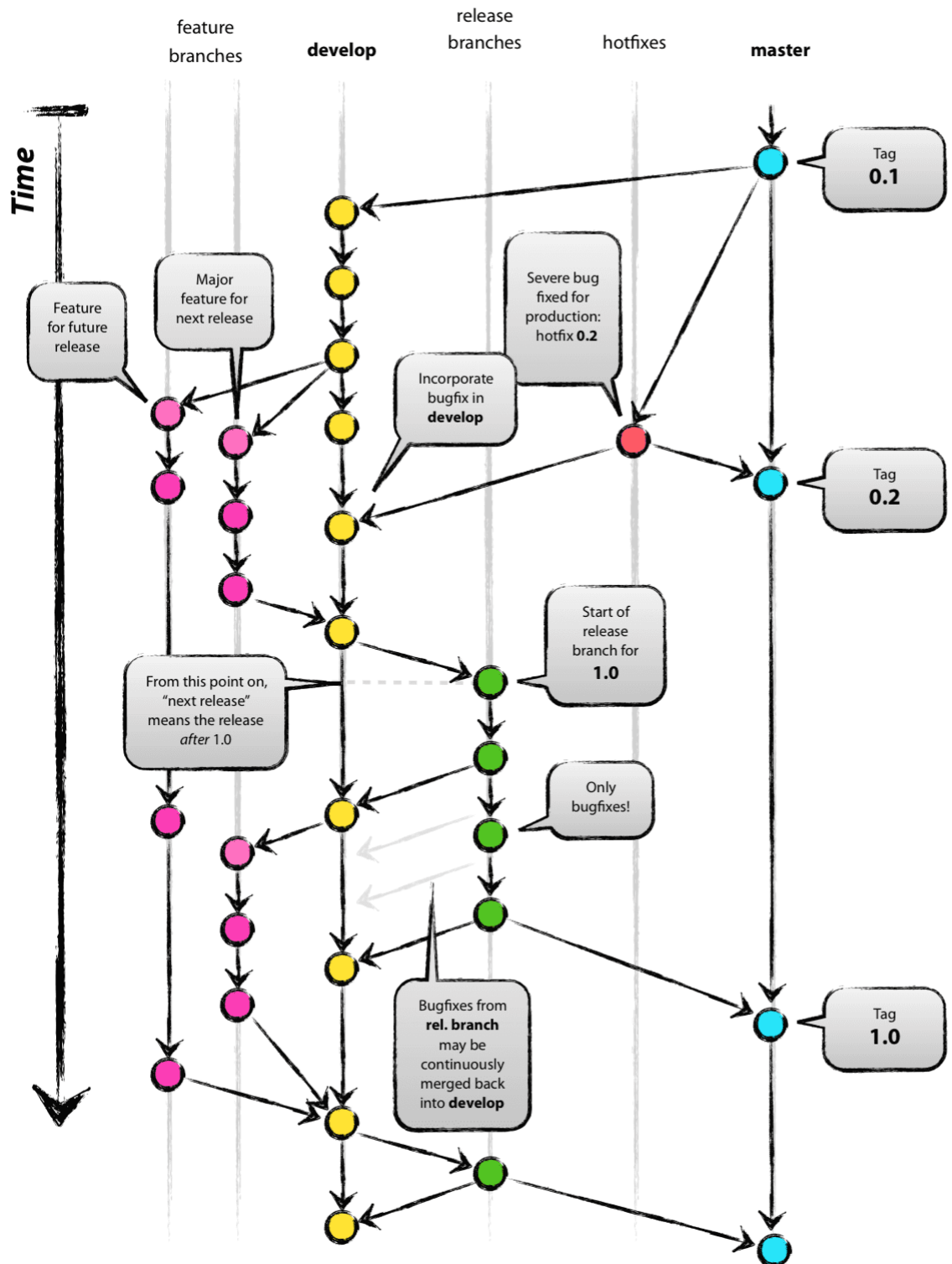
Jäsenrekisterin viestinlähetyksrajapintaan muutosten tekeminen sujui lähes vaivatta. Tähän tarvittiin kuitenkin pientä säätöä, koska generoidut luokat eivät olleet aluksi käytettävissä saman proseduurin nimen takia. Tämän muututtua sain paketin käyttööni, ja vaihdoin objektityyppiä rajapintaan.

Torstai 12.09.2019

Harmittaa vähän, että tänään ei pääse varsinaisesti työstämään Git siirtymää. Arkkitehdin kalenterin ollessa liian täynnä, suunnittelupalaveri tähän on varattu vasta perjantaksi. Heti aamusta oli kuitenkin robottitestit hajonneet, joten siitä löytyy hetkeksi tekemistä. Edelleenkin testit ovat siis sellaisia, etteivät hajoa ihan joka kerta. Pitäisi keksiä joku järkevämpi tapa käsitellä käyttöliittymää niin, ettei testeissä tarvitse loputtomasti kutsua Sleep funktiota.

Iltapäivällä ajattelin hieman valmistautua perjantaita varten. Tarkoitus olisi siis suunnitella ja dokumentoida Git-käytäntöjä, miettiä miten kokonaisuus saadaan toimimaan, ja listata erinäisiä haasteita ja kehityskohtia. Käytäntöjen osalta haluaisin hieman suunnitella minikälaiset kehityshaarat meillä tulee olemaan. Nykyisin kehittäjät eivät ole järin tyytyväisiä siihen, että SVN on oma kehityshaarat kummallekin tiimille. Haluaisivat kuulemma, että

kehitys tapahtuu yhdessä haarassa. Aion kyllä itse vahvasti suositella yleisesti hyvänä pidettyä Git workflowta.



(Vincent D. 2010. Viitattu 15.09.2019)

Se nyt on vielä epäselvää, halutaanko kummallekin tiimille oma develop haara, ja sitten olisi jonkunlainen master develop.

Iltapäivästä sain kirjoitettua ylös muutaman asian, jotka pitäisi Gittiin siirryttäessä hyvä ottaa huomioon. Omat ajatukset kehityshaaroista on aika pitkälle sama kuin yllä olevassa kuvassa. Sen mukaan olen aikaisemmin työskennellyt, ja havainnut erittäin hyväksi. Kirjasin myös ylös hyödyllisiä Git komentoja siltä varalta, että joku aikoo käyttää Gittiä komentorivin kautta. Tai jos joskus tarvitsee. Uskoisin vahvasti, että täällä halutaan käyttää jotain graafista työkalua kuten SVN kanssa. Tällä hetkellä yksikön ohjeissa on mainittu TorstoiseGit, johon on varmasti helppo siirtyä TorstoiseSVN maailmasta. Listasin myös muutama kysymyksen, joihin olisi hyvä miettiä huomenna vastauksia. Tarvitaanko tiimeille omat develop haarat, miten riippuvuudet hoidetaan, miten GitLab CI:ssä huomioidaan se, että projektien pitää toimia yhteen, minkälaiset edellytykset on sille, että projektin irrotus masterista on onnistunut, sekä milloin loput projektit on tarkoitus siirtää Gittiin. Loppujen siirto ei kuitenkaan ole mikään ihan pieni juttu, ja se olisi hyvä tehdä niin, että siitä aiheutuu mahdollisimman vähän katkoja työskentelyyn.

Tein myös jonkin asteisen katselmoinnin autentikointisovelluksellemme. Olin muiden kanssa samaa mieltä siitä, että käyttäjien luominen nykyisellään on tehty huonosti. Tunnuksia ladataan muistiin sovelluksen koodista. Tämä vaatii siis aina muutoksia uusien asiakkaiden myötä. Pidin myös hieman ongelmallisena sitä, että autentikointipalvelussa on pelkästään Robot Framework -testejä. Oauth-sovellusta on ehkä vähän hankala lähteä yksikkötestaamaan, tai muutenkaan testaamaan tuon enempää. Pitäisin silti järkevänä sisällyttää ainakin muutaman Java-testin tälle palvelulle.

Perjantai 13.09.2019

Tänään olisi tarkoitus miettiä Git-käytäntöjä arkkitehdin kanssa. Toivottavasti saadaan jotain järkevää keskustelua, niin pääsen jatkamaan näiden hommien parissa. Tämän päivän työtehtävät tulevat luultavammin selviämään aamun palaverin jälkeen.

Iltapäivästä on sprintin katselmointi, jossa taas esitetään aikaansaannoksia asiakkaille. Olen itse ollut tämän sprintin sen verran projektista irti, etten ole täysin varma mitä kaikkea on tehty tässä sprintissä. Oma työpanokseni projektin suhteen on ollut lähinnä bugikorjauksia ja automaatiotestien virittelyä.

Saimme vähän käytyä läpi meille sopivia haaroituskäytäntöjä, sekä mahdollisuuksia GitLab CI hyödyntämiseen. Isoiksi esteiksi kuitenkin osoittautuu se, että osa tarvittavista toiminnoista vaatisi premium GitLabin. Tulin kuitenkin itse päivän aikana siihen lopputulokseen, että tämä koko homma pitäisi miettiä jotenkin uusiksi. Yritin kollegalle selittää tätä koko kuviota, ja tajusin etten pystynyt edes järkevästi pukemaan sanoiksi sitä, mitä en-

simmäisessä vaiheessa tulee tehdä. Olisi mielestäni paljon järkevämpää yrittää saada asiat maaliin yksi kerrallaan. Pienemmät kokonaisuudet pysyvät kasassa paljon helpommin.

Gittiin siirtyminen jo itsessään vaatii yhdeltä ihmiseltä useamman viikon työtunteja. Tätä tukee LiveRamp Engineeringin blogi vastaavasta aiheesta. Heidän tapauksessaan piti tehdä vielä hieman ylimääräistä, koska heillä repositoryssa oli SVN ulkoisia riippuvuuksia. (LiveRamp. 2012. Viitattu 16.09.2019)

Pitäisi perehtyä mahdollisiin menetelmiin, sekä mitä kaikkea siirtymän eteen tarvitsee tehdä. Näin ison repositoryn kloonaminen on itsessään jo aikaa vievä prosessi. Varsinkin silloin, kun mukaan halutaan ottaa haaroja ja tageja. Ehkä ensi viikko selkeyttää miten tulemme tämän kanssa toimimaan.

Viikkoanalyysi

Perjantaita lukuun ottamatta työviikosta jäi mieleen positiivisia ajatuksia. Perjantaikin muuten, mutta tuntuuhan se vähän ikävältä tajuta yrittäneensä tehdä liikaa. Ensi viikolla varmasti tuon tämän asian edelle, ja vaadin että tätä suunnitellaan edes vähän uusiksi. Jos Tarkoitus olisi eritellä projekteja omiksi kokonaisuuksiin, se vaatisi myös koko firman tasolla GitLabin lisenssin päivitystä premium versioon. Uskon vahvasti, että tähän ei ole minkäänlaisia mahdollisuuksia ainakaan vaatimallamme aikataululla. Pitäisi myös saada lisää vastauksia Nexus repositoryn käyttöön. Jos vaikka joskus irrotettaisiin projekteja master projektista. En ole varma onko kaiken irrottaminen tarpeellista, mutta jonkunlainen uusiminen rakenteeseen voisi olla hyvä. Ei ole mitään järkeä, että kaikki sovellukset joutuvat clean buildissa kääntämään kaikki muutkin sovellukset.

Ammattiliittojen jäsenrekisterin uudistaminen ei ole tällä viikolla tarkoituksella työllistänyt minua kauheasti. Projektipäällikkö ymmärtää, että Git siirtymä pitää tehdä nyt, ja resurssini ovat käytössä vain, jos täysin välttämätöntä.

Syvensin myös hieman omaa osaamista OAuth2 osalta, kun minua pyydettiin tekemään katselmointia aikaisemmin tehtyyn ratkaisuun. Ainoa ongelma, johon törmäsin, oli jo tiedostettu ongelma siitä, että käytössä olevat tunnukset löytyvät koodista ladattuina muistiin. Tähän on varmasti olemassa järkevä tapa saada rekisteröinti dynaamiseksi. Helpotaisi jos uusien asiakkaiden saaminen ei aiheuta muutoksia koodiin.

Pienen tutkinnan jälkeen löysin tästä aiheesta blogipostin, jossa käytiin läpi dynaaminen tunnusten rekisteröinti. Dynaaminen rekisteröinti toteutuu erittäin pienellä vaivalla, ainakin

jos se on alusta asti tehty näin. Toteutettua ratkaisua on ehkä vähän ikävämpi lähteä muuttamaan. Käytännössä siis samanlainen koodipohja, mutta tarvitsee määrittellä tietolähde erikseen, josta tunnukset haetaan. (Eugen P. 2018. Viitattu 15.09.2019)

Blogin ohjeita seuraamalla tarvittaisiin kantaan taulu, johon tunnukset tallennetaan, sekä jokin tapa viedä tunnuksia kantaan. Olisi tietenkin järkevä jotenkin rajoittaa tunnusten tekemistä, vaikka jonkun admin tunnuksen taakse.

3.5 Seurantaviikko 5

Maanantai 16.09.2019

Päivän tavoite on yrittää vakuuttaa arkkitehdit siitä, että Git -siirtymässä ei kannata lähteä tekemään kaikkea heti. Olen itse sitä mieltä, että olisi paljon tärkeämpi saada asiat maaliin yksi kerrallaan, kuin kaikki heti. Siirtymä olisi kuitenkin tarkoitus tehdä nopealla aikataululla, mikä ei todellakaan ole nykyisen suunnittelun pohjalta mahdollista. Ei ole mitään järkeä alkaa miettimään koko projektin rakennetta uudelleen samalla kun pitäisi yrittää siirtää kehitystä Gittiin samaan aikaan aiheuttamatta häiriöitä kehitykseen. CI:n siirtäminen Bamboosta GitLabiin on tietenkin tärkeä tehdä, mutta ei ole mitään syytä tehdä tätä heti.

Tällä hetkellä Gittiin siirtyminen edellyttäisi yksittäisen projektin irrottamista SVN:stä, gradle buildin korjaamisen, lähdekoodin siirtämisen GitLabiin, CI putken rakentamisen, irrotetun projektin julkaiseminen Digian sisäiseen Nexus repositoryyn, sekä irrotetun osuuden poistaminen nykyisestä repositorystä ja vaihtaa muut projektit hakemaan tämä riippuvuutena. Kaikille projekteille pitäisi tehdä paketin julkaisua lukuun ottamatta nämä kaikki vaiheet. Sen lisäksi tässä pitäisi miettiä CI pipeline järjestäminen niin, että irrallaan olevat projektit saadaan toimimaan keskenään. Tätä myös rajoitti se, että osa tarvittavista toiminnallisuuksista on GitLabin premium toimintoja. Myös nykyinen installer ratkaisu, jolla asiakkaiden sovellukset päivitetään, vaatisi isoja muutoksia tämän myötä.

Mielestäni olisi järkevämpää siirtää ensiksi koko projekti Gittiin, ja laittaa Bamboo hakemaan lähdekoodit Gitistä, tai vaihtoehtoisesti luoda kopiot Bamboon buildeista GitLabiin. Näiden jälkeen voidaan erikseen lähteä miettimään, mitä steppejä rakenteen parantamiseksi voisi tehdä. Yritin etsiä tapauksia, joissa siirtymää on tehty, ihan vain verratakseni työvaiheita. Ainoa maininta riippuvuuksien korjauksesta, jonka löysin, koski pelkästään SVN externals tyyppisiä riippuvuuksia. Käsittääkseni projektissamme mitään ei ole hoidettu näin, eli tätäkään tapausta ei voi suoraan verrata siihen, mitä haluamme.

Kaikissa löytämissäni tapauksissa siirto oli kuitenkin useamman viikon homma. Syinä oli SVN repositoryjen määrä, tai muutokset joita repositoryyn piti tehdä ennen siirtoa.

(Ethan G. 2017. Viitattu 16.09.2019)

(Bala V. 2018. Viitattu 16.09.2019)

Arkkitehdit eivät paljoa suostuttelua vaatineet. Ilmoitin vain tekemäni migraation niin kuin itse koen parhaaksi. Vastaukseksi tähän riitti vain simppele "Ok". Aloitin kloonamaan koko repositorya, jossa tulee todennäköisesti kestämään hetki. 40 000 revision läpikäynti koko projektille on näköjään paljon isompi homma kuin sama määrä osalle projektista. Tässä taidetaan puhua useammasta työpäivästä. Repositoryn muunnon ohessa tarvitsee kuitenkin katsoa mitä kaikkea Bamboo vaatii, jotta saadaan CI buildi pyörimään. Tähän listaan kuuluu ainakin GitLab tunnukset, jotta voidaan hakea koodit sieltä Subversionin sijasta. Ehdin hieman tutkia mitä kaikkea Bamboo tekee, ja ainakin Subversionin revision number pitää korvata jotenkin. Vastaava koodi Gitissä olisi commitin hash, mutta se ei varmaan aja ihan samaa asiaa, sillä tätä olisi kai tarkoitus näyttää käyttöliittymässä jossain kohdassa.

Sain arkkitehdilta lisää oikeuksia Bamboo buildien muokkaamiseen. Helpottaa ainakin omia testailuja. Ehdin katsoa hieman mitä kaikkea Bamboossa tapahtuu, buildin rakenne näytti helposti ymmärrettävältä. Ainoa iso miinus siitä, että osa syötetyistä muuttujista on deprekoituneita. Jotenkin ei jaksa enää yllättää, että prosesseja ei ole kauheasti jaksuttu uusia. Muuttujista löytyi ainakin muuttuja bamboo.custom.svn.revision.number, joka on Atlassianin mukaan deprekoitunut. Tähän listaan kuuluu myös useita muita SVN liittyviä muuttujia. (Atlassian. 2019. Viitattu 16.09.2019)

Tiistai 17.09.2019

Olin jättänyt töistä lähtiessä koneen päälle ajamaan repositoryn muunnosta. Tässä ei tosin oltu pääsy kovin pitkälle, sillä yhteys oli jossain välissä katkennut. Päivän tavoitteena on kuitenkin testata miten buildit saadaan Bambooseen. Gradle buildille ainakin syötetään parametreina SVN revision number, ja se, hakeeko buildi revision numberin itse. Tähän voisi helposti muokata käyttöön Gitin commit hashin hakemisen ainakin toistaiseksi. Buildin muokkausta pystyy kuitenkin testaamaan käyttämällä aikaisemmin irrotetun projektin koodeja. Tarvitsee vain kysellä GitLab tunnuksien perään.

Yritin päivän aikana selvittää, miksi git svn clone menee niin hitaasti. Stackoverflow käyttäjän manojlds ja harmv mukaan kyse on git svn Windows portista, ja joistakin Perl-kirjastoista. Esimerkissä Windowsilla 75 revision repositoryn kloonamisessa on kestänyt

25 minuuttia, kun taas Linuxilla vastaava on 45 sekuntia. (Käyttäjät manojlds ja harmv. 2010, 2014. Viitattu 17.09.2019)

Laitoin kloonauksen pyörimään virtuaalikoneelle, ja nopeus on selvästi huomattava. Windowsilla läpikäytyjen revisioiden määrä päivässä meni Linuxilla noin tunnissa. Kloonasin tällä kertaa koko repositoryn, jotta saadaan historia kunnolla talteen.

Esimieheni oli varannut pienen tilannekatsauksen. Tällä kertaa agendalla oli se, että eihän minulla varmasti ole nyt liikaa töitä. Kuulemma nimeni on noussut esiin sen verran monessa projektissa, että esimies halusi varmistaa, että tekeminen on mielekästä eikä sitä ole liikaa.

Iltapäivästä oli tehtävien siirtoprojektin viikkopalaveri, jossa mietittiin henkilöiden tulevia rooleja, sekä mitä mahdollisesti tarvitaan kehityksen edetessä. Projektin suhteen ei ole vielä tehty tarkempia teknologiavalintoja, sillä on epäselvää, mitä kaikkea tarvitsee tehdä.

Keskiviikko 18.09.2019

Keskiviikko onkin varattu täyteen palavereja. Puolet päivästä menee tehtävien siirtoprojektin palavereissa, joiden tarkoitus on käydä läpi mahdollisia riskejä, teknisiä ratkaisuja asiakkaiden kanssa sekä riippuvuuksien arkkitehtuuria. Omana tavoitteena näiden osalta on yrittää saada parempi ymmärrys siitä, mitä tullaan tekemään, ja miettiä hieman mahdollisia ratkaisuja.

Iltapäivällä on jäsenrekisteriprojektin groomaus, jossa tavalliseen tapaan tarkennetaan joitain tulevia tehtäviä. Groomaukseen on toimitettu käsiteltävät tehtävät listana etukäteen, jotta niihin voisi halutessaan hieman perehtyä. Omalle tekemiselle ei tästä palaverista ole suuret odotukset, sillä muut projektit työllistävät tällä hetkellä ihan hyvin. Ja meillä on kuitenkin toinen kehittäjä, joka tarvitsee töitä. Tämän jälkeen on kassapuolen sähköisen asioinnin hakemusuudistuksen läpikäynti. Palaveriin on kutsuttu vastaavanlaisen toteutuksen tekijöitä, jos tätä olisi mahdollista hyödyntää uudistuksessa. Minut taas on kutsuttu paikalle todennäköisesti siksi että minun olisi tarkoitus tätä lähteä tekemään. Tavoitteena siis ymmärtää jälleen mistä uudistuksessa oikeastaan on kyse.

Päivän palaverien jälkeen hieman viisaampana useamman projektin suhteen. Mietimme tehtävien siirtoprojektin riskejä, tietokantaratkaisua sekä seuraavia vaiheita. Sähköisen asioinnin osalta oli tarkoitus miettiä lomakepalvelun hyödyntämistä hakemusuudistuksessa. Tätä varten palaverissa oli muutama lomakepalvelun kehittäjä selventämässä mistä on kyse. Lomakepalvelu on tehty sen verran generiseksi, että uusien lomakepohjien te-

keminen onnistuisi tähän hyvin, mutta kävi ilmi, että se ei välttämättä taivu ihan kaikkiin vaatimuksiin ainakaan suoraan. Esimerkiksi hakemuksen yhteenveto ei välttämättä onnistuisi suoraan.

Torstai 19.09.2019

Olin yöksi jättänyt koko svn repositoryn muunnoksen päälle komennolla, joka suorittaa kloonausta niin kauan, kunnes palautuskoodi on 0. Tällä pääsin eroon siitä ongelmasta, että yhteys aikakatkeaa välillä. Aamulla repository oli valmis käyttöön, ja nyt tälle päivälle olisi tarkoitus katsoa, että buildi edelleen menee läpi, siirtää koodit Gittiin ja testata laittaa Bamboo hakemaan koodit Gitistä. Joitakin ongelmia varmaan ilmenee, esimerkiksi taskeissa, jossa haetaan SVN versioita.

Testasin gradle buildin lokaalisti toimivaksi ja puskin koodit luomaani GitLab repositoryyn. Bamboon kanssa lähdekoodien hakeminen osoittautui ongelmalliseksi. Jostain syystä buildi kaatui heti koodien hakemiseen, ja syyksi ilmoitettiin `java.io.FileNotFoundException: /tmp/com.atlassian.bamboo.utils.i18n.I18nBeanFactoryImpl.data (Permission denied)`. Tälle virheelle ei internetistä juurikaan löytynyt selityksiä. Ainoassa vastaavassa tapauksessa yritettiin ajaa montaa Bamboo Agenttia samalla koneella, ja ehdotettu ratkaisu oli agentille temp-kansion määrittäminen käsin. Meillä pyörii palvelimella kaksi Bamboo Agenttia, eikä näitä buildeja käyttävällä ollut oikeuksia mainittuun tiedostoon. Lisättyäni oikeudet, buildi lähti etenemään normaalisti, ja oli vihreänä myöhemmin. Seuraavaksi pyysin Service Centeristä nightly buildin kloonaa, jota voisin siirtää osittain Gittiin. Näitä buildeja tulee vielä jonkun verran mietitä, koska properties-tiedostoja ladataan SVN haarojen nimien mukaan. Gittiin laitoin toistaiseksi masterin lisäksi trunk nimisen haaran, jotta propertyt saadaan ladattua. Master haaran buildi ei varmaan tästä syystä mene edes läpi. Olen itse sitä mieltä, että nämä buildien propertyt pitäisi syöttää CI:ssä, eikä lähdekoodista.

Siirtymän suhteen olisi järkevää mahdollisimman pian saada kaikki kehittäjät commitoimaan viimeiset koodit SVN, tekemään merge trunkkiin ja siirtämällä kerralla koko kehitys Gittiin. SVN muutoksien hakeminen git reponon pitäisi onnistua ajamalla Gitin svn rebase komento. Tästä syystä en vielä lähde muokkaamaan Git repon.

Perjantai 20.09.2019

En päivän päätteeksi saanut vielä kloonaa nightlyyn build planista, joten aamu menee ihmetellessä nykyistä nightlyä ja sen konfiguraatioita. Lisäksi pitäisi käydä asiakkaan testiym-

päristöön laittamassa autentikoinnin päälle. Tämä on aikaisemmin ollut odottamassa sitä, että ympäristöön saadaan toimivat tunnukset.

Mainitsin toiselle arkkitehdilla Git repositoryn tilanteesta, ja kuulemma hyvältä vaikutti. Varsinainen siirtymä olisi tarkoitus tehdä sprintin päätteeksi, eli todennäköisimmin ensi viikon jälkeen. Ehdin myös trunkkiin tulleen pienen mergen avulla testata svn rebase toiminnon toimivuutta uusien muutoksien hakemisesta repon.

Päivän tavoitteena on saada Git repon perustuva nightly deploy menemään läpi. Kuvittelin tämän olevan melko helppoa, mutta ei sitä koskaan tiedä.

Nightly buildi meni lähes ok. Varsinainen versio meni palvelimelle asti, mutta buildin artefaktien tallennus Bambooseen epäonnistui. Tähän syytä pelkästään se, että en ollut tajunnut muuttaa konfiguraatioista kaikkia muuttujia pois. Näissä yritettiin siis vielä käyttää vanhoja SVN revision numeroita.

Kyselin arkkitehdilta, milloin siirto olisi hyvä tehdä. Tässä yhteydessä kävi ilmi, että toisen tiimin olisi tarkoitus toimittaa väliversio lokakuussa. Väliversion takia kehityshaaraa ei voida mergetä trunkkiin. Joudun todennäköisesti yrittää vielä ottaa SVN repositorysta sellaisen version, jossa on kehityshaarat mukana. Jos koko kehitys saataisiin näin Gittiin, ilman että joudutaan tekemään mitään suurempia järjestelyjä.

Viikkoanalyysi

Koen että viikon aikana Git siirtymää saatiin edistettyä hyvin. Sain paremman käsityksen siitä, miten tämä nyt tulisi saada maaliin. Jouduin jättämään viikonlopuksi uuden repositoryn muunnoksen pyörimään, jotta saataisiin haarat mukaan. Maanantaina voidaan katsoa miten tämän kanssa uskaltaa edetä. Versiotoimituskuviota pitää kyllä jonkun verran selvittää, sillä mielestäni ei kuulosta lainkaan järkevältä toimittaa versioita kehityshaaroista ja olla siksi mergeämättä trunkkiin.

Uskalsin viikon aikana ottaa enemmän vastuuta tästä ja tehdä itsenäisiä päätöksiä. Tällä hetkellä olen kuitenkin yksikössämme se henkilö, jolla on näistä asioista eniten kokemusta ja tietämystä. Onhan se tietenkin hyvä, että muut yrittävät suunnitella miten olisi järkevää, mutta tutkimisten perusteella lähestymistapaa kannatti kuitenkin vaihtaa. Aikaisemmalla suunnittelulla toisen arkkitehdin mielestä siirtymä olisi valmis vasta alkuvuodesta 2020. On siis ihan hyvä, että minulla on uskallusta ja valtaa tämän suhteen, että voidaan saada tämä aikaisemmin. Subversionin käytöstä on tässä vaiheessa enemmän haittaa kuin hyö-

tyä. Tiimien palavereissa nousee jatkuvasti esille se, kuinka paljon merget työllistävät kehittäjiä, ja kuinka olisi järkevämpää siirtää kehitys pois useammasta haarasta mergejen määrällisen vähenemisen takia.

Esimiehen kanssa käydyssä keskustelussa viikolla nousi esille, että minussa olisi potentiaalia tulevaisuudessa nousta jonkinlaiseen arkkitehdin rooliin. Myönsin olevani miettinyt tätä aikaisemmin, ja olimme yhtä mieltä siitä, että hieman isommalla kokemuksella tämä olisi järkevä ja luonnollinen ratkaisu. Tätä varten on siis ihan hyvä, että nyt toistaiseksi yritän olla mukana useammassa projektissa laajentamassa omaa osaamista. Pelkän kehitystyön ohessa on ihan hyvä saada kokemusta taustajärjestelmistä, ja yrittää hieman ymmärtää sovelluksien kokonaisuutta. Tällä hetkellä muut vastaavia töitä tekevät henkilöt ovat olleet talossa pitkään, ja mielipiteet ja näkemykset saattavat jo tästä syystä olla hieman vanhentuneita. Esimiehen mukaan olen siis mukana tuomassa uutta näkemystä. Tulevalla viikolla on useampaan projektiin liittyen palavereja, joissa varmasti keskustellaan käytettävistä teknologioista erilaisten uudistusten takia. Haluaisin päästä vaikuttamaan käytettäviin teknologioihin ja menetelmiin. Haluan myös saada Git siirtymän ainakin lähes valmiiksi. Jos viikonloppuna repositoryn haarojen kloonaus on onnistunut, pitää vielä katsoa miten saan SVN haarat muutettua Gitin haaroiksi. Tällöin kehitys ei keskeytyisi kummassakaan tiimissä, vaikka mergeä ei heti saisikaan tehdä.

3.6 Seurantaviikko 6

Maanantai 23.09.2019

Tänään suunnitelmissa miettiä vähän sitä, kuinka Git siirtymän kanssa tulisi edetä. Viime viikolla ilmenneet rajoittavat tekijät saattavat olla ongelmallisia, jos repositorya ei saada kloonattua kehityshaarojen kanssa. Aikatauluja siirtymän suhteen on silti hyvä alkaa miettiä jo nyt.

Iltapäivä menee kokonaan tehtävien siirtoprojektin palavereissa asiakkaiden kanssa. Tarkoituksena on määritellä kokonaisprosessia. Ehkä tässä pikkuhiljaa päästään oikeasti tekemään jo yhteyksien testaamista ja jopa toteuttamaan jotain.

Aamupäivä meni Git-käytäntöjä suunnitellessa. Tulimme siihen tulokseen, että olisi järkevää noudattaa aikaisemmin esille nousutta mallia kehityksen suhteen. Sillä poikkeuksella kuitenkin, että aloitetaan kolmella develop haaralla. Näistä haaroista kaksi on tiimien omat develop haarat, ja kolmas on varsinainen develop missä on kaikki kehityksessä olevat toiminnallisuudet. Tiimien omat kehityshaarat on kuitenkin tarkoitus tappaa pois heti kun

mahdollista. Kassatiimin kehityshaara mahdollisesti heti, ja liittotiimin kehityshaara väliversion toimituksen jälkeen. Myös rebase komennon hyödyntäminen nousi esiin. Rebasen tarkoitus on pitää Gitin historia siistinä. Rebase ei kuitenkaan ole komento, joka olisi kovin käyttäjäystävällinen, jos käyttäjä työskentelee ensimmäistä kertaa Gitin parissa. Atlasianin Git ohjeissa sanotaan, että rebase ei myöskään sovellu käytettäväksi, jos useampi henkilö työskentelee saman kehityshaaran parissa. (Atlassian. Viitattu: 23.09.2019). Meidän tapauksessamme samaan kehityshaaraan saattaisi eri kerrosten takia koodata ainakin kolme henkilöä, joten rebase tuskin tulee kysymykseen.

Päätimme yhdessä, että rebasea ei tulla käyttämään. Juuri siksi, että aloittelevalla käyttäjälle se voi olla hankalaa, ja koska nykyisen suunnittelun pohjalta yhteen feature haaraan koskee samanaikaisesti enintään kolme kehittäjää. Kehityksen siirtäminen olisi näillä näkymin tarkoitus tehdä kolmen viikon päästä.

Iltapäivän tehtävien siirtoprojektin asiakaspalaveri valaisi enimmäkseen siitä, mistä tässä on kyse. Projektin nimen mukaisesti, kyseessä on siis TE-toimiston tehtävien siirtämistä työttömyyskassoille ja maakunnille. Yleisen työttömyyskassan sivuilta löytyvän tiedon mukaan alustava aikataulu tähän oli 2020, mutta muutosta on siirretty vuodella. (Peter E. 2018. Muutoksia aktiivimalliin ja muita viilauksia. Viitattu 23.09.2019)

Tiistai 24.09.2019

Eilisen Git-suunnittelun pohjalta aloin kloonamaan repositorya uusiksi niin, että saadaan myös kehityshaarat mukaan. Kloonaminen ei kuitenkaan onnistunut ihan noin vain, ja tuntui että näin ison ja "sotkuisen" repositoryn kloonaminen koituisi lähes mahdottomaksi. Eilen arkkitehti oli sitä mieltä, että ei välttämättä tarvitse koko revision historiaa. Otin siis kokeilumielessä kloonin repositorysta niin, että historia säilyy vain revisiosta 30 000 asti. Eli neljäsosa koko repositorysta olisi mukana. Tämä päivä menee tätä viilatessa. Jos näin olisi mahdollista siirtää kehitys ennen väliversion toimitusta.

Osittaisen historian sisältävä repository ei ollut niin toimiva, kun olisin halunnut. Kehityshaaroja ei voinut mergetä keskenään, koska olin ottanut ne liian myöhäisestä vaiheesta. Haaroilla ei siis ollut yhteistä historiaa, ja Git ei tietenkään suostu yhdistämään toisiinsa liittymättömiä historioita. Jätin päivän päätteeksi koko repositoryn kloonautumaan komennolla "git svn clone -s *repositoryn osoite*". Vipu -s tarkoittaa sitä, että SVN repository noudattaa niin sanottua oletusulkoasua. Oletuksena repositoryn juuressa on kansiot trunk, branches ja tags. SVN ei kuitenkaan pakota tätä ulkoasua, eli jos ulkoasu olisi ollut erilai-

nen kuin oletus, olisin joutunut erikseen kertomaan missä poluissa trunk, haarat ja tagit sijaitsevat.

Päivän palavereissa groomattiin hieman lisää Java-työtä, joka ei todennäköisesti tällä hetkellä koske minua. Olin joka tapauksessa mukana antamassa tehtäviin työmääräarvioita.

Tehtävien siirron osalta mietittiin kansallisen palveluväylän hyödyntämistä. Ulkoiset rajapinnat, joita projektissa käytetään, tulevat palveluväylän kautta. Palveluväylä taas vaatii oman liityntäpalvelimen, joka Digialla onneksi on jo toteutettu. Palveluväylä on suomi.fi tuottama palvelu, jonka tarkoitus on tarjota käyttäjälle tietoa erilaisista järjestelmistä ja rekistereistä. Palveluväylää käyttämällä parannetaan palvelujen laatua, läpinäkyvyyttä, tietoturvaa ja tietosuojaa. (Suomi.fi. Viitattu 24.09.2019)

Sähköisen asiointin teknologiauudistuksen palaverissa tarkasteltiin Digian Jyväskylän kehittäjien aikaansaannoksia ja ideoita jatkon suhteen. He kertoivat, mitä on mahdollista tehdä ja minkälaisessa ajassa. Kevyempi uudistus kattaisi vain sen, että hakemuksien lähettäminen uudistetaan. Isompi uudistus vaatisi sen, että koko sähköinen asiointi tehtäisiin paremmin ja järkevämmäksi. Isomman uudistuksen edellytyksenä on kuitenkin se, että hakemusten lähettäminen on ensin tehty uusiksi. Siitä on hyvä lähteä jatkamaan eteenpäin.

Keskiviikko 25.09.2019

Töihin saapuessani repositoryn kloonaukseen oli vieläkin pyörimässä. Tässä vaiheessa kloonaukseen on pyörinyt lähes 20 tuntia, ja jäljellä olisi vielä pari tuhatta revisiota. Uskon vahvasti, että tällä kertaa kloonaukseen menee läpi. Luulisin myös, että tällä kertaa haarat ovat mergetävissä keskenään. Eilinen "unrelated histories" ongelma tuntui olevan sellainen, mitä kehenkään ei ole ollut. Pääsen oikeasti nyt testaamaan, jos tämä versio olisi se, jota siirtymässä voidaan käyttää. Pitää ottaa varmuuskopio muutetusta repositorysta, ettei tätä tarvitse tehdä uusiksi.

Repository näyttäisi nyt olevan kunnossa siirtymää varten. Seuraavana täytyy vain tuoda muutoksia SVN:stä niin kauan, että siirtymä ollaan valmis tekemään. Bamboo buildit on tehty lähes valmiiksi uutta ympäristöä varten. Tiimeille täytyy myös asiasta hieman tiedottaa, ja todennäköisesti järjestää jotakin koulutusta. Toinen arkkitehteistä on luvannut parin viikon päästä järjestettävillä suunnittelupäivillä kertoa hieman uusista käytännöistä. Sen verran olen itse sitä esitystä nähnyt, että siinä on kuvattu SVN ongelmia, ja miten

Gitin odotetaan pelastavan tilanne. Perus workflow on myös kuvattu. Käytämme aluksi tiimeille omia develop haaroja featureita varten. Nämä develop haarat mergetään sitten yhteiseen develop haaraan. Tiimikohtaisista haaroista on kuitenkin tarkoitus luopua heti, kun se on mahdollista.

Torstai 26.09.2019

Tarkoitus olisi tiedottaa tiimejä siitä, että Gittiin siirtyminen tulee tapahtumaan lähiaikoina. Ajattelin, että voisin kirjoittaa yksikö Teams-kanavalle vähän aikatauluista tähän liittyen, sekä päivittää wikiin Git-ohjeistukseen perus workflowin ainakin komentorivin osalta. Graafista käyttöliittymää en mielelläni itse Gitin kanssa käytä, mutta siihen olisi varmaan myös hyvä olla jotkut ohjeet. Tämän voi tietenkin delegoida sellaiselle, joka haluaa työkaluja käyttää.

Tiimeille kirjoitettu teamsiin viesti, jossa kerron Git siirtymän nykytilanteesta ja annan kaikki tarvittavat linkit repositoryihin ja käyttöohjeisiin. Pyysin samalla tiimiläisiä lukemaan ja seuraamaan ohjeet läpi, jos niissä olisi jotain parannettavaa tai epäselvyyksiä. Päivän aikana tiedotukseen tuli kuitenkin positiivisia reaktioita. Nyt odotan jonkun Gitistä kokemattoman ehtivän käyvän ohjeet läpi.

Kassapuolen tulorekisteri integraatiosta sain iltapäivällä lisätöitä. Yleinen työttömyyskassa tarvitsee jotakin rajapintoja tulorekisterin raporttien hakuun. Näihin ehdin aikaisemmin tehdä jo jotakin pohjia, mutta jäin odottamaan tietokantatoteutuksia. Nyt näitä pääsee työstämään vähän pidemmälle, mutta nähtäväksi jää vielä se, miten nämä oikeasti pitäisi toteuttaa. Joudun kysymään hieman siitä, miten data on tarkoitus luoda, kun tietokannasta palautuu vain XML-stringi. Vastaavanlaisia toteutuksia ei muualla näyttänyt olevan, eikä tähän taida olla mitään testidataakaan, joka hankaloittaa omia kokeiluja.

Perjantai 27.09.2019

Vielä ei eilisen jälkeen kukaan ollut ehtinyt kommentoida sen kummemmin Git asiaa, joten kehityksen siirtämisen ohjeistus on nyt näiltä osin jäissä. Tämän osalta tarvitsee enää ajoittain hakea uusimmat muutoksen SVN:stä.

Ajattelin jatkaa tulorekisteri integraation rajapintojen työstöllä. Tämä kuitenkin vaatii sen, että näistä enemmän tietävät saapuvat töihin. Tässä on kuitenkin hieman opittavaa, koska kyseessä on SOAP-rajapinta. Aikaisemmin olen ollut tekemisissä vain REST-rajapintojen kanssa. Tietämys näistä kuitenkin siirtyy aika helposti. Suurimpana erona on kuitenkin

mielestäni se, että SOAP antaa ulos XML-dataa, kun taas REST antaa yleensä Json-dataa. Swati Dhingra FirstRain Software Centeristä sanoo, SOAP-rajapinnat yleisesti vaativat vähemmän niin sanottua "boilerplate"-koodia, sillä tietoturva, transaktiot ja oikeudet eivät tarvitse välttämättä omaa toteutusta. (Swati D. 2016. Viitattu 26.09.2019)

Git siirtymän tilanteesta tiedottaminen on ainakin yksikön teamsissa kerännyt ison kasan tykkäyksiä, mutta vielä ei ole kenelläkään ollut suurempaa kysyttävää. Arkkitehti haluaisi kokeilla tehdä julkaisun Gitissä olevista lähdekoodeista kokeilumielessä. En pitänyt tätä kovin hyvänä ideana, koska siirtymä on vielä kesken. Julkaisu kuitenkin tehtäisiin omasta haarasta, jota ei olisi tarkoitus yhdistää mihinkään. Tämä siis luultavasti ei haittaa siirtoa. Tulorekisteri-integraation SOAP-rajapinnat tulivat mielestäni valmiiksi. Rajapintojen testit menivät lokaalisti läpi. Rajapinnoissa ainoa ongelma on se, että näihin ei ole valmiina järkevää testidataa. Data pitäisi erikseen käydä hakemassa tulorekisteristä. Testidatan puuttuessa XML-datan muuttaminen Java-objektiksi on vielä epävarmaa, koska tätä en ole voinut testata. Lähes kaikki muut toteutetut rajapinnat hakevat tietokannasta suoraan objekteja, mutta jostain syystä tulorekisteridatasta on tietokannassa vain raporttien tunniste ja sisältö XML-stringinä. Mielestäni hieman kyseenalainen ratkaisu. Tätä kuitenkin testataan paremmin ensi viikolla.

Viikkoanalyysi

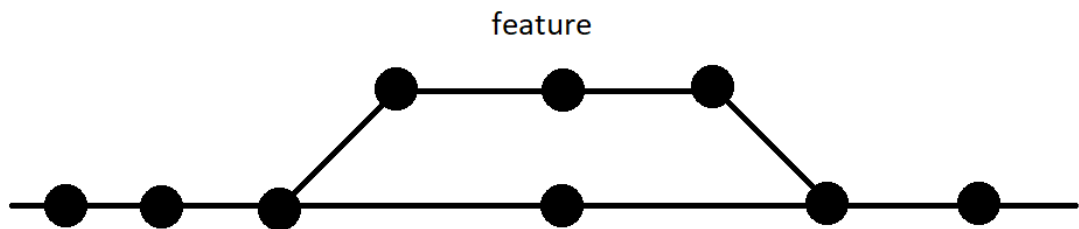
Viikon aikana Git siirtymän työstäminen tuntui välillä raskaalta ja toivottomalta. Ihan kuin missään ei päästäisi eteenpäin, kun vaatimukset haaroille ja aikatauluille muuttuivat. Repository on nyt kuitenkin kloonattu, ja sitä tulee vielä parin viikon ajan päivittää, ennen kuin varsinainen siirto tapahtuu. Kummassakin tiimissä on pari ihmistä, jotka ovat aikaisemmin olleet tekemisissä Gitin kanssa, joten siirtymän tapahduttua joudun varmasti tukemaan tiimejä jonkin verran. Tärkeää on kuitenkin se, että päästään kohti enemmän käytettyjä teknologioita, ja että työnkulku helpottuu jatkossa.

Viikolla uutena tuttavuutena tuli SOAP-rajapinnat. Yritin viikon aikana ymmärtää, mitkä ovat SOAPin hyötyjä RESTin ylitse. Samaa voi varmaan sanoa monessakin asiassa, eli teknologia kannattaa valita käyttötarpeiden mukaan. Stackify blogissa on aikaisemman tiedon kanssa yhteensopivaa tietoa siitä, että SOAP tarjoaa valmiiksi paremman tietoturvan pienemmällä vaivalla, ja pyyntöjen suorittamista on epäonnistumistilanteessa mahdollista yrittää uudelleen, toisin kuin RESTissä, jossa lähettäjäsovelluksen täytyy toteuttaa virhetilanteiden käsittely itse. (Stackify. 2017. Viitattu 29.09.2019)

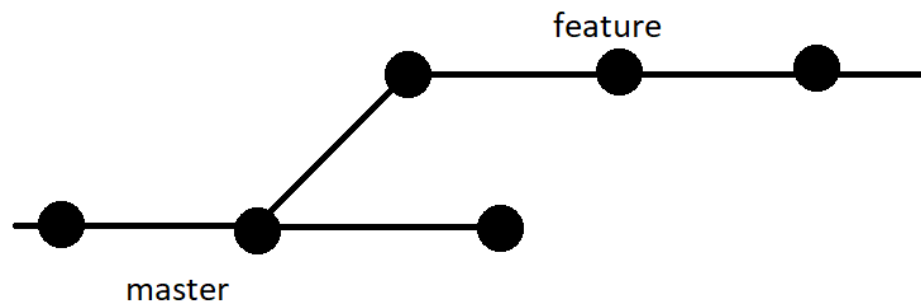
Viikon työtehtävistä on mielestäni selvitty paremmin kuin hyvin. Git siirtymä pysyy näillä näkymin aikataulussa, eikä vielä ole sen suhteen enempää haasteita tiedossa. Aika näyttää, miten todellisuudessa käy.

Vaikka SOAP-rajapinnat olivat minulle uutta, sain vaaditut tehtävät silti paljon työmääräarvioita nopeammin valmiiksi. Rajapinnat olivat myös valmiita projektin aikataulujen mukaan, vaikka jouduin tekemään nämä aika lyhyellä varoitusaajalla. Työtehtäväni ovat kuitenkin siinä tilanteessa, että pystyn irrota tarpeen vaatiessa muiden projektien käytettäviin.

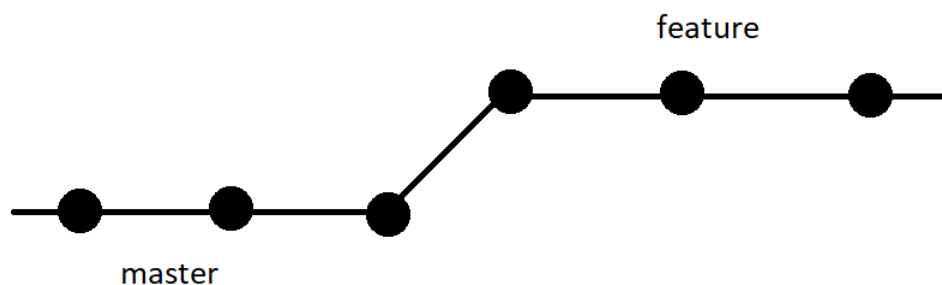
Git Rebasen käyttö nousi myös viikolla esille. Sen idea olisi pitää Gitin historia lineaarisena. Komento siirtäisi feature haaran alkamaan masterin viimeisimmästä commitista, jolloin historia olisi siistimmän näköinen



Kuva 1. Feature haara pelkällä mergellä



Kuva 2. Feature haara ennen rebasea



Kuva 3. Feature haara rebasen jälkeen

Kuvien tilanteet itsessään näyttävät siisteiltä, mutta kuvan 1 tilanne alkaa nopeasti paisumaan, jos feature haaroja on useita, ja niitä joudutaan mergeämään paljon. Masterin viides commit tässä tapauksessa on merge commit, jossa muutokset tuodaan masteriin.

Kuvassa 2 feature haara on aloitettu masterin toisesta commitista, ja kuvassa 3 se on rebasella siirretty masterin kärkeen. Näin selvittää ylimääräisiltä sotkuilta repositoryn historiassa.

3.7 Seurantaviikko 7

Maanantai 30.09.2019

Päivää alkaa hakemalla SVN tehdyt muutokset Git repositoryyn. Kyse ei kuitenkaan ole pitkästä operaatiosta, jos tätä tekee tarpeeksi usein. Komennoilla git svn fetch, ja gitissä oleviin haaroihin git svn rebase, saadaan kehityshaarat ajan tasalle. Git push --all niin saadaan muutokset GitLabiin. Ainoa vaiva tässä on se, että olen itse estänyt kehityshaaroihin muutoksien tekemisen GitLabissa siksi aikaa, että siirtymä tehdään. Tämän takia pitää väliaikaisesti käydä sallimassa uuden koodin puskeminen.

Perjantain SOAP-rajapintoja olisi tänään tarkoitus testata oikean datan kanssa. En itse tiedä mistä tämä data pitäisi hakea, joten se jää odottamaan siksi aikaa, että osaavat henkilöt saapuvat töihin.

Iltapäivällä on liittotiimin sprintin retrospektiivi ja suunnittelu. Uskoisin voivani tähän sprinttiin ottaa jo liittotiimin töitä, sillä kaikki muu tekeminen alkaa olla hieman vähissä.

Sprintin retrospektiivissä käytiin läpi ajatuksia sprintistä. Omalta osalta ajatukset jäivät vähiin, kun lähes koko sprintti meni muissa töissä. Git siirtymän työstö ei kuitenkaan kuulu tähän. Suunnittelua varten ilmoitin olevani nyt paremmin tiimin käytettävissä, sillä Gittiin siirtymisessä ei ole enää juurikaan tekemistä. Jos ohjeiden kanssa tulee joillain ongelmia, niin niitä sitten. Sprinttiin tuli iso kasa aikaisempien REST-rajapintojen bugeja ja puutteellisuuksia, jotka pitäisi korjata. Näissä riittää tekemistä ainakin hetkeksi. Rajapinnoissa puuttui lähinnä kenttiä, tai haettiin väärää arvoa, ehkäkin heikon määrittelyn seurauksena. SOAP-rajapintojen testaamiseen en saanut arkkitehtiä kiinni koko päivänä, joten siirsin tiketit hänelle ja ilmoitin asiasta.

Tiistai 01.10.2019

Aamu alkaa heti SVN muutosten hakemisella Git repon ja sovellusten päivittämisellä lokaaliin ympäristöön, jotta voin testata rajapintojen puutteellisuuksia omalla koneella debuggerin kanssa. En usko, että bugitickettien kanssa kestää kovin kauaa, sillä nopealla

silmäilyllä vaikuttivat helpoilta korjattavilta. tietenkin ne on syytä testata nopeasti läpi jälkeenpäin.

Iltapäivällä on taas backlogin groomaus, joten lisää töitäkin on ehkä tulossa. Ehdin tällä kertaa varmaan käydä läpi etukäteen lähetetyn listan käsiteltävistä tiketeistä.

Rajapintojen korjauksiin ei mennyt kauheasti aikaa. Kyseessä tosiaan oli vain päivämäärien formaatteja, väärän datan hakua indeksistä ja joitain puuttuvia tietoja. Yhdessä tapauksessa haluttiin näyttää sellaista dataa, jota tietokannasta ei haettu. Tämä siis siirtyy jollekin toiselle työstettäväksi.

Päiväpalaverin yhteydessä sain tehtäväksi tutkia, miksi asiakkaan ympäristössä connection poolin refresh ei toimi aina. Sain alkuun aika heikot tiedot tämän selvittämiseen, kun missään ei kuulemma ilmaannu mitään virhettä. Kuitenkin asiakkaan ympäristössä kokeilun jälkeen tuli jotakin Redis virhettä. Lokaalissa ympäristössä sain täysin samaa, joka korjaantui käynnistämällä lokaali Redis uudelleen. Asiakkaan viimeisimmistä lokeista ei löytynyt mitään connection poolin refreshiin viittaavaa, joten tämän suhteen nyt ehkä hieman jumissa. Täytyisi päästä enemmän sisällä admin työkalujen toteutukseen, jotta ymmärtäisi paremmin, miksi tämä ei aina toimi.

Keskiviikko 02.10.2019

Tänään tarkoitus olisi yrittää saada enemmän selkoa connection poolin ongelmaan, käydä läpi yksikön arkkitehtuuri- ja alusta-asioita, sekä tehtävien siirtoprojektin viikkopalaveri. Työmäärä alkaa taas tuntua hieman vähäiseltä, vaikka eilisessä groomauksessa olikin hieman puhetta parista tulevasta hommasta.

Kokonaisuudessaan tänään ei ehtinyt oikein tapahtua mitään. Käytiin arkkitehtuuri- ja alustapalaverissa läpi kuukauden aikana tehtyjä töitä näihin liittyen, ja katsottiin arkkitehdin suunnittelupäiville valmistelemaa esitystä Gitistä. Ensi viikolla olisi vielä tarkoitus avustaa sen valmisteluissa ennen suunnittelupäivää. Ehdin esityksen aikana huomata, että ainakin muutama termi ja kuvausnuoli oli hieman väärin.

Tehtävien siirron viikkopalaverissa mietittiin käyttäjätarinoita. Ei siis omalta osalta vielä kään mitään liian oleellista, koska ollaan vielä niin ylätasolla.

Torstai 03.10.2019

Koko päivä on vielä vähän auki. Todennäköisesti katsotaan toisen kehittäjän kanssa lomakepalvelun toimintaa. Lomakepalvelussa on kuitenkin aika monta osaa, joista osassa esimerkiksi Node.js käytössä. Tämä ei ole itselle liian tuttua, mutta nopealla silmäilyllä ymmärsin kyllä mistä on kyse. Perustietämys kuitenkin aika hyvin siirtyy kieleltä toiselle.

Päivä meni pitkälti lomakepalveluun perehtymisessä. Sain mielestäni ihan hyvän kuvan siitä, miten palvelu on rakennettu. Käytetyt Node.js frameworkit eivät tuntuneet ihan niin ylitsepääsemättömiltä mitä olisin luullut.

Ehdin myös hieman päivittää yksikön Git-ohjeita ongelmatilanteiden pohjalta. Sain sähköpostin, jossa sanottiin, että GitLab ilmoittaa access denied repositorya kloonatessa, ja että toisella henkilöllä tämä kuitenkin toimii, vaikka kummatkin seurasivat ohjeita vaihe vaiheelta. Ratkaisu tähän oli poistaa tai muokata Windowsin ohjauspaneelista tallennetut GitLab tunnukset. Johtui siis siitä, että aikaisemmin oli kloonattu HTTPS yli jotakin, ja muistissa on vanhat tunnukset. Tämä on nyt dokumentoitu wikin ohjeisiin FAQ kohtaan.

Perjantai 04.10.2019

Muiden sairastelujen ja hoitovapaiden takia lomakepalveluun perehtymisestä on tuskin liikaa hyötyä. Olen ehtinyt jo katsoa suurin piirtein sen toimintaa, ja mitä uusien lomakkeiden lisääminen vaatii mihinkin Java sovellukseen. Jos mitään kiireellisempää ei tule, käyn ajatuksella läpi omat Git ohjeet ja parantelen niitä.

Git ohjeisiin lisätty ohjeistusta committien perumiseen. Ohjeistuksessa mainitaan myös, ettei tämä ole hyvä idea, jos commit on ehditty puskea remote repositoryyn.

Esimiehen kanssa oli puhetta projektista, jonka tarkoitus olisi vähentää projektimme teknisistä velkaa. Teknisellä velalla tarkoitetaan aikaa, joka menee uusiin toiminnallisuuksiin, korjauksiin siitä syystä, että aikaisemmissa vaiheissa on valittu helpot ja nopeat ratkaisut. Tämän aikatauluista ei ole vielä mitään sen tarkempaa, mutta saan tähän työpariksi firman sisältä jonkun.

Viikkoanalyysi

Git siirtymä ei enää työllistä paljoa, ja viikko muutenkin meni pitkälti kaikissa pienissä korjauksissa, suunnitteluissa ja muiden auttamisessa. On varmaan ihan hyödyllistä, että minulla on nyt aikaa perehtyä enemmän muidenkin ongelmiin. Saa olla hyödyksi myös sellaisten asioiden selvittämisessä, jotka eivät ole itselle välttämättä niin tuttuja. Pidän tätä kuitenkin merkittävänä tilaisuutena kehittää omaa osaamista, jotta voin kehittyä haluamal-

lani tavalla. Edelleen mietityttää parin viikon takainen keskustelu esimieheni kanssa, jossa oli puhetta arkkitehdin roolista tulevaisuudessa. Tämän takia olisi erittäin hyödyllistä saada parempi ymmärrys kokonaiskuvasta. Koen kuitenkin, että olen tähän mennessä viihtynyt Digialla ihan hyvin, vaikka välillä on työtehtävien suhteen ollutkin puutetta. Tuntuu pitkältä aikaa siltä, että on mahdollista oikeasti edetä uralla.

Gitin osalta aikaisemmin oli puhetta, että joku kehittäjä voisi käydä läpi tekemäni ohjeistuksen. Tähän tarpeeseen sellainen kehittäjä olisi paras, jolla on mahdollisimman vähän kokemusta Gitistä. Nyt tässä viikon aikana jotkut ovat ehdineet käydä ohjetta läpi, ja sain lisättyä muutamaan ongelmatilanteeseen ratkaisun. Yhdessä päiväpalaverissa puheeksi tuli, että tämä olisi hyvä siksi, ettei varsinaisen siirtymän tapahduttua kaikki ole kaikille epäselvää. Kommentoin tätä sanomalla, että todennäköisesti kaikki tulee silti olemaan epäselvää. Viikolla Git ongelmia selvitellessä mielipide tästä vahvistui, ja mietin, missä välissä olisi sopivaa järjestää jonkunlaista Git koulutusta. Tähän olisi myös hyvä miettiä riittääkö yksikön tasolla se, että minä järjestän tähän jotakin koulutusta, vai olisiko oikeasti hyvä hankkia talon ulkopuolelta jotain. Tai miksei sisäpuolelta, jos sellaista olisi tarjolla. Koen kuitenkin, että siitä olisi taas minulle henkilökohtaista hyötyä pitää tällainen.

3.8 Seurantaviikko 8

Maanantai 07.10.2019

Uusi viikko on varmaan hyvä aloittaa lomakepalvelun tutkimisella. Pitäisi katsoa, miten viestijonosta haetaan viestit consumer-applikaatiossa. Sovelluskokonaisuudessa on kolme sovellusta; käyttöliittymä, RabbitMQ jonoon hakemusdatan lähettävä sovellus, ja jonoa lukeva sovellus, joka myös lähettää hakemusdatan jäsenrekisteriin. Näiden lisäksi on myös erillinen sovellus, jonka tarkoitus on luoda täytetystä hakemuksesta pdf-tiedosto. Aikaisempien tutkimisten perusteella tässä vaiheessa tarvitsisi koskea vain consumer-applikaatioon, jossa data muutetaan jäsenpalvelun lukemaan muotoon.

Consumer applikaatio vaikuttikin paljon selkeämmältä, kun vähän tarkemmin pääsi tutustumaan. Nämä sovellukset on toteutettu Digian toisessa yksikössä, joten mitään sen suu-rempia esittelyjä ei tällä aikataululla kannattanut toivoa. Luulin kuitenkin saaneeni ihan hyvän käsityksen itse tutkimalla. Sovellukset vaikuttivat selkeiltä kokonaisuuksilta, vaikka näitäkään ei erikoisemmin ole testattu.

Consumer applikaatio tarvitsi jokaista lomaketta kohti oman ApplicationSender-luokan, sekä MessageTransformer-luokan. Senderin tehtävä on valita käytettävä transformer, ja

transformeriin tulee koodata kaikki erikoiskäsittelyt, esimerkiksi päivämäärien muodot. Erikoiskäsittelyt jäivät vielä kesken, koska en ole ihan varma tässä vaiheessa, mitä tällaisia tapauksia on. Lisäksi käyttöliittymästä tulee kentät vielä väärillä nimillä, joten tämä odottaa käyttöliittymäkäyttäjien työtä. Uskon kuitenkin, ettei näiltä osin enää tarvita mitään ihmeellisempää.

Tiistai 08.10.2019

Tarkoitus olisi jatkaa consumer applikaation asiakaskohtaisia erikoiskäsittelyjä. Eiliset ongelmakohdat olivat ainakin se, miten lomakkeelta tulee puhelinnumerot ja näiden suuntanumerot. Suuntanumerot olivat omassa listassaan, joka vaikutti ehkä vähän oudolta. Tätä myös selvitettävä käyttöliittymäkehittäjän kanssa.

Päivän aikana tuli selvitettyä ties minkälaisia Git ongelmia, käytyä kirjoitettua ohjetta läpi henkilöiden kanssa, joilla Gitistä täysin nollakokemus. Ohjeissa oli ehkä pieniä epäselvyyksiä, johtuen siitä, että itselle on tietenkin hieman hankalaa kirjoittaa vaaditusta näkökulmasta. Saatiin kuitenkin järkeviä muutoksia aikaan, ja ohjeet ovat nyt hieman selkeämmät. Odotan kuitenkin tähän vielä lisää palautetta. Käyttöliittymäpuolella oli taas Git saatu sellaiseen tilaan, että repositoryyn puskeminen ei onnistunut. Tämä johtui siitä, että conflicteja oli selvitetty GitLabin käyttöliittymästä, ja lokaali haara oli tehty asettamatta upstreamia. Push onnistui, koska se tehtiin komennolla `git push origin haaran_nimi`. Tästä kuitenkin selvittiin helpolla asettamalla haaran upstreami oikeaksi.

Lomakepalvelun osalta pääsin melko pitkälle. Sovittiin käyttöliittymäpuolen kanssa, että suuntanumerokentät nimetään niitä vastaavan puhelinnumerokentän mukaan. Näin Java-kerrokseen ei tule suuntanumerolistaa, vaan voidaan sitten yhdistää oikeat numerot toisiinsa. Consumer applikaation toteutuksia on tehty vielä vähän geneerisimmiksi, jotta saatiin pois duplikaattikoodia. Odottelen tähän nyt käyttöliittymäpuolelle kenttien nimien korjauksia. Huomiselle todennäköisesti jää myös jonkunlainen testien kirjoittaminen. En ihan luota siihen, etteikö pienet muutokset koodiin olisi saattanut hieman muuttaa sitä, mitä jäsenpalvelulle lähetetään.

Keskiviikko 09.10.2019

Tänään olisi tarkoitus saada lomakepalvelun uudet lomakkeet toimimaan. Käyttöliittymätyön jäljiltä tässä pitäisi olla oikean nimiset kentät, jotta Java-kerros osaa ne nyt käsitellä. Myös testit MessageTransformereille pitää tehdä. Varmaan tässä vaiheessa yksi testi

transformeria kohti riittää. Ajattelin, jos syötän sisään lomakkeen JSON-datan ja tarkistan, että lopputulos vastaa haluttua.

Käyttöliittymäkehittäjä olikin tänään sairaana, joten toisesta lomakkeesta jäi kenttien nimien päivitykset puuttumaan. Olisin varmaan osannut tämän tehdä itsekin, mutta ei ehkä sopivaa astua toisten tontille. Luottamustehtävälomake kuitenkin saatiin menemään jäsenpalveluun oikeanlaisena, ja sitä voidaan perjantain asiakasdemossa esitellä.

MessageTransformerien testien kanssa törmäsin pieniin ongelmiin. Jouduin selvittämään, minkä riippuvuuden takia JSONObject luokka käyttäytyi testejä ajaessa eri tavalla kuin normaalisti. JSONObject(String data) konstruktorilla kenttä, jonka arvo on null, alustui testien ajossa merkkijonoksi, joka oli arvoltaan null. Normaalissa ajossa tämä kuitenkin alustui oikein null arvoksi. Eliminointiprosessilla selvitin, että spring-boot-starter-test paketin mukana tuleva Skyscreamerin JsonAssert aiheutti sen jostain syystä.

En onnistunut selvittämään syytä ongelmalle, eikä vastaavaa ongelmaa tuntunut edes esiintyvän missään muualla. Ratkaisuksi päädyin laittamaan pom.xml tiedostoon, että tätä riippuvuutta ei tarvita Spring Bootin testipaketista. JsonAssert ei kuitenkaan meillä ollut käytössä missään. Sain kuitenkin transformereille kirjoitettua testit, joten nyt voin olla varma, etten muutoksillani ole onnistunut hajottamaan mitään.

Torstai 10.10.2019

Koko päivä meni yksikön suunnittelupäivässä Vuosaassa. Päivän ohjelmaan kuului nykytilanteiden katsaus liikevaihtoihin, asiakkaisiin ja asiakastyytyvääisyyteen, sekä tulevaisuuden suunnitelmiin. Tulevaisuuden suunnitelmissa käytiin läpi myös Gittiin siirtymistä arkkitehdin toimesta. Esityksessä tuotiin esille se, mitä haasteita nykyisessä SVN-mallissa on ollut, ja miten uusi Git-malli tulee toimimaan. Loppujen lopuksi kehitysmalli näytti hyvinkin samanlaiselta. Erona tietenkin se, että jatkossa kaikki kehittäjät tulevat tekemään mergejä.

Suunnittelupäivän iltapäivän osuudessa jakauduimme tiimeihin, ja saimme tehtäväksi pohdita tiimin sisällä toimivia ja toimimattomia prosesseja, työnkuvia ja työtapoja. Etsittiin erilaisia kipukohtia, joihin pyrittiin myös löytämään jonkunlaisia ratkaisuja. Päivän lopuksi havainnot käytiin läpi keskustellen muiden tiimien kanssa.

Perjantai 11.10.2019

Perjantain suunnitelmat ovat vielä hieman auki, sillä lomakepalvelun edistäminen edellyttäisi sitä, että käyttöliittymäkehittäjä saisi tehtyä loput muutokset. Tänään siis todennäköisimmin valmistaudun iltapäivällä pidettävään asiakasdemoon ja päivitän Git-ohjeesta löytyneitä puutteita. Puutteet ovat lähinnä ongelmatilanteita, joissa push epäonnistuu, pitäisi ratkaista conflicteja, tai pull epäonnistuu. Kaikkiin ongelmiin on kuitenkin helpot ratkaisut, sillä Git komento kertoo nämä jo siinä vaiheessa, kun jotain menee pieleen. Pitäisi myös saada kehittäjät lukemaan virheilmoitukset, eikä heti nostamaan kädet ilmaan.

Päädyin käyttämään päivän CI buildien siirtämiseen Bamboosta GitLabiin. Bamboossa ajettavat komennot siirtyivät aika pienellä vaivalla gitlab-ci.yml tiedostoon. Pelkän buildin pipelineissa ei ollut mitään ihmeitä, kun tästä ei oteta edes talteen mitään artefakteja.

Nightly buildissa talteen kerättävien artefaktien toiminnasta en ole ihan varma, sainko nämä vielä toimimaan. Bamboossa näille oli oma steppi, jossa ne siirrettiin luettavaan kansioon. GitLab CI:ssä pitää vain kertoa, missä tiedostot sijaitsevat. En päässyt testaamaan buildeja kunnolla, sillä aluksi oli palvelimen levytila vähissä, ja jouduin olla service centeriin yhteydessä saadakseni sitä järjestettyä. Seuraavaksi buildin ajo keskeytyi, koska palvelimelta loppui RAM. Jätin tästä pyynnön service centerille, ja pääsen ehkä ensi viikolla jatkamaan tätä.

Iltapäivän asiakasdemoon en joutunut juurikaan valmistautumaan. Piti vain näyttää lomakkeen lähetys, ja se että lomake täytetty lomake ilmestyy jäsenpalveluun. Aamusta jäi mietityttämään vähän, kun lomakkeen lähetys ei muilla tuntunut toimivan. Testasin kuitenkin oikeassa ympäristössä lomaketta, eli mistään paikallisen koneen asetuksista ei voinut olla kyse.

Viikkoanalyysi

Viikot tuntuvat vierähtävän kauhean nopeasti. Tästä viikosta meni yksi työpäiväkin muissa kuin normaaleissa työtehtävissä. Viikosta on kuitenkin jäänyt taas erittäin onnistunut olo. Vaikka lomakepalvelun jatkokehitys olikin ehkä tarkoitus olla enemmän toisen kehittäjän vastuulla. Sanotaan nyt kuitenkin, että päädyimme tekemään sen osittain yhdessä. Oma tarkoitus oli ennemminkin tutkia, miten lomakepalvelua on tehty. Päädyin sitten kuitenkin aloittamaan toteutuksen, kun kaikki tuntui jotenkin niin selvältä. Lomakepalvelussa on vielä vähän auki se, kuinka helposti pystyy lisäämään toiminnot uusille asiakkaille. Nykyinen kehitys koski kuitenkin vain olemassa olevan asiakkaan toimintojen lisäystä.

Suunnittelupäivien jälkeen oli puhetta siitä, että pitäisi kerätä kaikki oleelliset henkilöt kaasaan ja lyödä joku päivä lukkoon, milloin tuotekehitys siirretään kokonaan Gittiin. Mielestäni tässä on sen verran lyöty jo jarrua, että joku päivämäärä tosiaan olisi hyvä. Ehdin mahdollisesti sitä päivää odotellessa siirtää CI:n GitLabiin, niin seuraavat versiot voidaan toimittaa sitä kautta.

Välillä on vähän harmittanut se, että on aika paljon joutunut olla tekemisissä legacy-koodin kanssa. Nyt ollaan kuitenkin menossa oikeaan suuntaan, ja olen saanut tehdä tärkeitä asioita koko yksikön eteen.

Koeaikani loppuu ensi viikolla, ja esimiehen kanssa pitäisi puhua jatkosta. Tämän suhteen nyt ei kuitenkaan ole mitään epäselvää.

3.9 Seurantaviikko 9

Maanantai 14.10.2019

Tarkoitus olisi työstää lisää GitLab CI pipelinea. Nykyisestä ratkaisusta olisi myös hyvä erotella esimerkiksi build ja test omiksi vaiheikseen. Olisi helpompi huomata nopealla katsomisella, missä on mennyt vikaan. Myös asiaankuuluvat artefaktit, eli esimerkiksi testien tulokset tai sovelluksen asennuspaketit olisivat oikeissa paikoissa.

Iltapäivällä on edellisen kehityssprintin retrospektiivi ja seuraavan suunnittelu. Tässä on varmaan odotettavissa jotain työtehtäviä.

Sain aamulla GitLab Runnerin palvelimelle lisää muistia, joten pystyin kokeilla buildien ajoa uudestaan. Tällä kertaa buildi kaatui siihen, että tietokantaskriptit eivät menneet läpi. Kuvittelin tähän syyksi sen, että yritetään liian vanhan version päälle viedä liian uutta. En ehtinyt tätä selvittää enempää, mutta pitäisi selvittää nopealla kysymisellä joltakin tietokanta-asiantuntijalta.

Buildi on nyt kuitenkin eroteltu kolmeen vaiheeseen; build, test ja deploy. Nightly ympäristöille on näistä vielä omat versiot, koska näissä suoritetaan hieman enemmän toimenpiteitä. Olen määritellyt konfiguraatioihin niin, että nightly buildin asetukset ajetaan vain ajastetuissa buildeissa.

Sprintin suunnittelussa mietittiin taas sitä, kuinka paljon aikaani voin käyttää liittotiimien tekemiseen. Varattiin 10 päivästä puolet, ja se tulee menemään kokonaan testiau-

tomaation kehittämiseen. Tällä kertaa on vähän selvemmat määrittelyt siitä, mitä kaikkea tarvitaan testata. Hiljattain jäsenyyden päättäminen oli hajonnut, ja se todellakin olisi ollut helposti huomattavissa testien avulla. Toive olisi saada testattua kaikkien eri tietojen muokkaamiset. En näihin menevän kokonaiset viisi päivää, mutta testien vaiheet olisi hyvä tehdä niin, että ne ovat mahdollisimman helposti uudelleenkäytettävissä. Testattavia toimintoja kuitenkin näytti olevan paljon, joten saatan hyvinkin olla väärässä. Kaikenlaisen hienosäätöön saa kyllä kulumaan aikaa tarpeen vaatiessa.

Tiistai 15.10.2019

Heti aamusta robottitestit näyttivät punaista, ja ehdin jo luulla testien hajonneen sattumasta. Tällä kertaa kyseessä oli kuitenkin aiheellinen hajoaminen, eli testeistä oli todetusti hyötyä. Testejä ei siis sen enempää tarvinnut selvittää, ja voin loput päivästä käyttää CI buildien kehittämiseen. Ilmeisesti kaatuvat constraint violationiin siksi, koska tietokantatesteissä on enemmän dataa kuin ikivanhan palvelimen kannassa. Yritetään siis päivittää sellaisia rivejä, joita ei ole olemassa. Olisi varmaan järkevää tähän hätään ottaa varmuuskopio kannasta, jossa on oikea data ja yrittää uudestaan.

Buildeissa käytettiin tiedostoja, joiden nimi tulee target systemin mukaan. Tämän takia osa aikaisemmin tekemistäni muutoksista johti siihen, että oletuksena käytettiin lokaaleja propertyja. Olen siis yrittänyt saada kaikki maininnat Bamboosta pois ja vaihtanut ne ci nimisiksi. Päivän lopuksi buildi edelleen kaatui siihen, että joissain testeissä yritettiin ladata väärä propertyjä. Pitää vielä selvittää, missä tämä määritellään.

Keskiviikko 16.10.2019

Eilisen päätteeksi päälle jätetty buildi oli epäonnistunut samasta syystä kuin aikaisemmat, vaikka kuvittelin saaneeni viimeiset maininnat Bamboosta pois. Tänäpäin pitää varmaan yrittää vertailla toimivaa paikallista buildia vasten eroavaisuuksia. Tässä on kuitenkin kyse pelkästään testien käynnistämisestä, eli ainakin iso osa buildin siirrosta on jo oikein.

Sain päivän lopuksi testit käynnistymään. Seuraava ongelma on se, että ilmeisesti testidata viedään väärään kantaan, jonka takia muutama testi kaatuu. Pitäisi siis yrittää selvittää missä kohdassa näitä oikein määritellään. Lokien nopealla vertauksella Bamboossa ja GitLabissa pyörivä buildi näytti käyttävän oikeita tietokantoja, mutta vaatii lisää selvityä.

Torstai 17.10.2019

Yritän vielä tänään saada GitLabin buildia läpi. Olisi sitten helpompi siirtää kehitys Gittiin, kun kaikki ympäristöt olisivat valmiina. Päätimme kuitenkin, että siirtyminen tehdään vasta väliversion toimituksen jälkeen. Tähän on siis vielä pari viikkoa aikaa. Pitää varmaan kuitenkin jossain välissä saada ajatukset hetkeksi muualle, ja alkaa tekemään robottitestejä jäsenrekisterille.

Koko päivä tosiaan meni yrittäessä saada buildia läpi, eikä se päivän päätteeksi onnistunut. Mietin että tämä voisi olla ympäristöongelma, eli jos huomenna kokeilisin pyörittää GitLab-runneria samalla palvelimella samaa kantaa vasten kuin Bamboo.

Perjantai 18.10.2019

Päivän tavoitteisiin kuuluu yrittää hetki buildin saamista läpi eri palvelimella. Tämän luovutan hetkeksi, ja yritän keskittyä liittotiimin robottitesteihin. Näihin on kuitenkin varattu aikaa, ja buildien säätämiseen alkaa pikkuhiljaa olla käytetty tähän sprinttiin tarpeeksi aikaa. Jos buildi ei nyt aamusta etene, kyselen apua jostain muualta, sillä omat ideat alkavat olla lopussa. Bamboossa meillä on kuitenkin build planit Gittiä varten, että tämä ei ole mahdotoman kriittistä. Kuitenkin buildien siirto on tarkoitus tehdä heti seuraavaksi Git siirtymän jälkeen.

Sain buildin menemään läpi, kun vaihdoin sen ajamaan toisella palvelimella. Ilmeisesti kyseessä oli siis oikeastikin ympäristöongelma. Pitää myöhemmin vielä ajaa nightly buildi läpi ja varmistaa, että se toimii oikein.

Robottitestejä ehdin vähän aloittaa. Uusia testejä en tehnyt, vaan kirjoitin testien logiikkaa uusiksi niin, että jatkossa uusien testien tekeminen olisi helpompaa. Ei ole mielestäni kovin käytännöllistä joka kerta lisätä iso kasa uusia XPath-muuttujia resurssitiedostoihin. Järkeilin niin, että input kenttiin voitaisiin syöttää dataa niihin kuuluvan otsikon perusteella. Näin saataisiin testien avainsanoja hieman geneerisemmiksi.

Viikkoanalyysi

Kuluvalla viikolla piti olla ties mitä palavereja, jotka syystä tai toisesta peruuntuivat. Tämän takia ehdin kuitenkin paremmin keskittyä konkreettiseen työhön. GitLabin CI-pipelinet ovat lähes siinä kunnossa, että niiden käyttöön voitaisiin siirtyä muun siirtymän ohessa. Pitää vielä selvittää, miksi artefaktit eivät tule näkyviin, vaikka buildissa lukee, että niiden lähettäminen on onnistunut.

Pidän edelleen Gittiin siirtymisessä isona haasteena sitä, miten tiimi oppii itse tekemään merget, ja onko code review oikeasti hyödyllistä. Ainakin sen pitäisi olla. Nykyisellään kuka tahansa on voinut puskea versionhallintaan mitä tahansa, eli ainakin tämä tulee muuttamaan.

Tällä viikolla olisi myös pitänyt osittain alkaa jonkinlainen teknisen velan maksamisen projekti, jossa olisi osallisina minä ja toisesta yksiköstä joku minulle ennalta tuntematon henkilö. Kuitenkin kaikki tästä vastuussa olevat henkilöt olivat tämän viikon syyslomalla, enkä osaa sanoa onko tässä nyt tapahtunut mitään. Toisen yksikön henkilö piti kuitenkin suht nopealla aikataululla saada näihin hommiin, ja hänen perehdyttäjänsä oli kuitenkin nyt lomalla. Lupasin itse omalle esimiehelle, että pystyn kertomaan jotakin perusjuttuja, jos tarve vaatii, mutta ilmeisesti tältä osin ollaan nyt vähän jäissä.

3.10 Seurantaviikko 10

Maanantai 21.10.2019

Tarkoitus olisi nyt saada työstettyä liittotiimin robottitestejä. Vaikka mielelläni näitä teen, on mielestäni silti erikoista, että näiden tekeminen ei varsinaisesti kuulu tiimimme testiajan vastuulle. Tahdon kuitenkin kehittää tiimimme prosesseja, ja automaatiotestit ovat välttämättömiä.

Tänään on myös yksikkömme arkkitehtien viikkopalaveri, jossa tarkoitus on käydä läpi erilaisia projektikohtaisia työtehtäviä. Viikkopalaverin tarkoitus on jakaa tietoa, että kaikki pysyvät ajan tasalla siitä, mitä ollaan tekemässä.

Toin arkkitehtien viikkopalaverissa esille muutamia ongelmakohtia GitLabin CI putkien kanssa. Ongelmat olivat palvelimen TLS sertifikaattien vanhentuminen, joka ilmeisesti on alustatiimin toimesta korjattu, sekä GitLabin artefaktien kokorajoitus. Tähän piti service centeristä pyytää lisää tilaa, sillä asennuspakettimme ovat tällä hetkellä lähes 350mb. Uusia testejä en päässyt vielä kirjoittamaan. Tämä päivä meni lähinnä avainsanojen muuttamisessa geneerisimmiksi. Robot Frameworkin resurssitiedostoissa oli iso kasa muuttujia, joiden arvot olivat täsmälleen samat. Poistin myös useamman avainsanan kokonaan, koska ne eivät olleet käytössä missään, tai niissä oli TODO-merkintä. Teen mieluummin uudelleen kuin yritän muuttaa vanhoja vastaamaan käyttötarpeita.

Tiistai 22.10.2019

Tänään on tehtävien siirtoprojektin asiakastilaisuus ja liittotiimin backlogin groomaus, joihin menee käytännössä koko päivä. Asiakastilaisuudessa olisi tarkoitus käydä asiakkaiden kanssa tarpeita ja priorisointeja läpi, sekä hieman hahmottaa heille sitä, minkälaisia työvaiheita projektiin sisältyy.

Asiakastilaisuus meni pitkälti odotetusti. Projektin tässä vaiheessa ei kuitenkaan ole mitään koodaustyötä, eli itse olen mukana lähinnä vain siksi, että pysyn kärryillä tarpeista. Backlogin groomauksessa oli tällä kertaa erittäin tärkeitä asioita. Asiakkaille pitäisi saada käyttöön eri tunnistautumisjärjestelmä, sillä nykyinen ratkaisu alkaa olemaan elinkaarensa päässä. Tähän mietittiin mahdollisia vaihtoehtoja toteutukseen. Toteutusvaihtoehdot olivat kutsua rajapintoja tietokantatriggeristä, kutsua rajapintoja meidän päivitysrajapintojen kutsumisten yhteydessä, toteuttaa päivityksen Spring Batch jobeina, tai laittaa käyttöliittymä lähettämään data MQ-jonoon, josta se sitten lähetettäisiin eteenpäin. Näistä vakavasti otettavat vaihtoehdot ovat kaksi viimeisintä. Jos joku olisi lähtenyt puskemaan rajapintojen kutsua tietokannasta, olisin kyllä pistänyt siinä vastaan. Edellisessä työpaikassa tehtiin tällainen nopea kokeilu, ja se oli täysin älytön viritys. Olisin itse ehkä tuon MQ-jonon kannalla, mutta se saattaa tietenkin olla pieni ylilyönti. Tärkeää olisi kuitenkin se, että tiedon päivitys ei ole riippuvainen kolmannen osapuolen järjestelmän ongelmista. Tieto ei kuitenkaan saa jäädä päivittämättä, jos sen lähetys kerran epäonnistuu.

Keskiviikko 23.10.2019

Tänään tavoite olisi saada robottitestejä tehtyä eteenpäin. Nämä nyt on kaikkien palaverien takia hieman siirtynyt, mutta uskon silti saavani valmista ennen sprintin loppumista. Jossain välissä pitäisi myös vaihtaa työpistettä ja neuvoa aikaisemmin mainittua ulkopuolista henkilöä projektiin sisään pääsemiseksi. En vielä tiedä tarkkaan, mikä tässä on tavoitteena, mutta sen selvittyä varmaan onnistuu "perehdyttäminen" paremmin.

Uudet robottitestit ainakin päällisin puolin ok. Niiden tekeminenkin onnistui vähän vauhdikkaammin mitä luulin, kun olin saanut avainsanat kirjoitettua. Pientä viilausta piti testien tekemisen ohessa vielä tehdä, mutta sain ne lokaalisti menemään läpi. Uuden henkilön perehdyttäminen jää huomiselle ja perjantaille, koska häneltä puuttui oikeuksia järjestelmiin, eikä esimerkiksi ilman lähdekoodeja pääse kovin pitkälle.

Torstai 24.10.2019

Päivä on hyvä aloittaa muuttamalla työpistettä, jotta pystyn olla enemmän läsnä uuden kehittäjän perehdytyksessä. Tänään olisi tarkoitus saada hänelle kehitysympäristö pystyyn, ja jos oikeudet versionhallintaan järjestyy, käydä ehkä läpi koodiakin.

Robottitestit olivat yöllä epäonnistuneet. Syynä tähän on se, että webdriver ei osaa scrollata tarpeeksi, ja klikattavat elementit jäävät fixed navigointipalkin alle. Varmaan olisi syytä päivittää palvelimen selain isompaan. Firefox 38 on kuitenkin yli 4 vuotta vanha, ja webdriveriin on varmaan ajan myötä tullut iso kasa korjauksia ja parannuksia.

Jouduin vasten omia periaatteita tehdä testien suhteen hieman kyseenalaisen ratkaisun. Yritin päivitellä selainta ja muuta uusimpaan versioon, mutta oli jotenkin älyttömän hankalaa saada keskenään yhteensopivia versioita palvelimelle, joka on muutenkin jo elinkaarensa päässä. Saatavilla ei vain ollut tarpeeksi uusia versioita, enkä ole täysin varma siitä, olisiko jonkunlaisella kikkailulla saanut asennettua kaiken tarvittavan. Ongelma oli siis se, että webdriver ei osaa scrollata tarpeeksi sivulla. Ratkaisin ongelman yksinkertaisesti niin, että asetin palvelimen Xvfb displayn tarpeeksi korkeaksi. Display on aikaisemmin ollut 1600x1200. Omien muutosten jälkeen se on 1600x5120. Ratkaisu ei mitenkään ole se kaikista elegantin, mutta saa riittää toistaiseksi. Seuraavaksi pitää tarkistaa, että testit oikeasti menevät läpi.

Perjantai 25.10.2019

Jatkan tänään testien työstämisellä. Ilmeisesti webdriver ei vielä kukaan pysy kunnolla soveluksen perässä. Yritän saada ratkaistua tätä niin, että testeissä kutsutaan mahdollisimman paljon avainsanaa `Wait Until Element Is Visible`. Näin vältetään turhilta sleepeiltä, eikä testien ajo hidastu liikaa.

Olen myös tälle päivälle varannut esimieheni kanssa keskustelun, sillä koeaikani loppui jo viime viikolla. Sen suhteen ei tietenkään ollut mitään epäselvää. Tarkoitus olisi vain käydä läpi ihan yleisesti fiiliksiä, sekä mahdollisesti kysyä firman tarjoamista eduista.

Uskaltaisin melkein väittää, että robottitestit alkavat nyt olla suht hyvällä mallilla. Lisää testejä on kohtuullisen helppo tehdä, ja niiden pitäisi myös olla sen verran varmoja, että eivät satunnaisesti hajoa. Sain päivän lopuksi tehtyä pari uuttakin testiä, ja alkuun näytti ainakin hyvältä. Sain poistettua avainsanoista joitakin `Sleep` kutsuja, ja korvattua ne järkevämmillä `Wait Until` -avainsanoilla. Dynaamisen käyttöliittymän ohjaaminen automaattisesti on aina ollut vähän hankalaa. Onneksi sivujen toteutus on kuitenkin suhteellisen yhdenmukainen, mikä mahdollistaa samojen avainsanojen uudelleenkäytön. Käytin näissä myös aika paljon XPath-selectoreita, joissa valitsen elementtejä sen mukaan, mitä CSS-

luokkia niissä on. Tällä tavalla pystyin myös tarkistamaan sen, onko joku elementti vielä muokkaustilassa, vai onnistuiko tallentaminen.

Tähän tehtävään on ainakin hetkeksi käytetty tarpeeksi aikaa, eikä projektipäällikkö sanojensa mukaan viitsisi tästä kovin paljoa enempää laskuttaa tämän kuun aikana. Tämä on tietenkin mielestäni hieman typerä väite, sillä kattavan testauksen ei ikinä pitäisi olla huono asia. Kyseessä on kuitenkin asia, johon on hyvä käyttää aikaa niin paljon kuin mahdollista. Tietenkin eri asia siinä vaiheessa, jos uusien toiminnallisuuksia myötä menee enemmän aikaa vanhojen testien korjaamiseen, voisi olla pieni miettimisen paikka.

Viikkoanalyysi

Viikon suurimpia haasteita olivat käyttöliittymän dynaamisuus sekä vanhat palvelinympäristöt. Testien ajaminen on jotenkin työlästä, jos ne toimivat lokaalisti eri tavalla kuin varsinaisessa CI-ympäristössä.

Kaikesta huolimatta olen tyytyväinen aikaan saamiini testeihin, vaikka niiden ajamiseen jouduinkin tekemään hieman kyseenalaisia ratkaisuja. Jos vaihtoehtoina on palvelinympäristöjen päivitys tai jonkinlainen viritys, on varmaan järkevämpi toistaiseksi valita jonkinlainen viritys. Palvelinympäristöjen päivittäminen tietääkseni on jonkun työpöydällä. Eri asia sitten on taas se, milloin tämä on ajankohtaista. Luulen, että joudun varmaan itse alkaa ajamaan tätäkin asiaa eteenpäin.

Käyttöliittymän uudistusprojekti on tietääkseni aloitetut reilu neljä vuotta sitten, ja osa tehdyistä ratkaisuista tuntuu hieman erikoisilta. Tietenkin käyttökokemuksen kannalta uskaltaisain väittää, että ihan järkevän tuntuisia toteutuksia, mutta näiden testattavuus on erittäin hankalaa. Perustoimintojen testaaminen on silti mielestäni kannattavaa, jotta saadaan pienennettyä tiimimme testaajan työjonoa. Olisi varmaan kuitenkin erityisen järkevää tuoda käyttöliittymäpuolelle jonkunlaiset yksikkötestit. Backbonen näkymien testaaminen kuitenkin on mahdollista. Tämä on varmasti asia, jonka tuon päättyvän sprintin retrospektiivissä esille. Aluksi voisi olla järkevää kirjoittaa ainakin uusiin toiminnallisuuksiin testit, ja myöhemmin paremmalla ajalla lisätä vanhoihin.

Sain viikon aikana palautettua mieleeni XPath-syntaksia. Aikaisemmin korjaamissani robotitesteissä oli käytetty aika paljon CSS-selectoreita, mutta mielestäni XPath-selectoreilla saa varmemmin sen, mitä oikeasti halutaan. En ole ihan kaikkea kuitenkaan muuttanut, mutta sen verran kuitenkin, että saadaan avainsanoja paremmin uudelleenkäytettyä.

4 Pohdinta ja päätelmät

4.1 Kehittyminen työn aikana

Pystyn selvästi huomaamaan sen, miten olen ehtinyt kehittymään työssä päiväkirjan aikana. Olin ehtinyt olla Digialla ennen aloittamista töissä lähes neljä kuukautta, ja työn aikana olen saanut lisää vastuuta nykyisistä ja uusista projekteista. Osaan myös tuoda mielipiteeni esille, ja haastaa vähän kokeneempiakin työntekijöitä omilla näkökulmillani.

Olen laajentanut osaamistani erilaisista Java ohjelmistokehyksistä, palvelinpuolen työkaluista, jatkuvan integraation työkaluista ja menetelmistä. Lisää uutta tuntuu myös koko ajan olevan tulossa, ja vielä aihealueilta, jotka oikeasti kiinnostavat. Kehittymisen mahdollistuminen on siis ollut kiinni siitä, että ensin tietää itse paremmin mitä haluaa. Yrityksen sisällä on näin helpompi vetää sellaisista naruista, että pääsee työskentelemään mielekkäisiin tehtäviin.

Opin hyödyntämään myös aikaisempaa osaamista pienten manuaalisten tehtävien vähentämiseksi. Näitä automaatioita olisi varmasti hyvä miettiä laajemmin yksikön sisällä.

Päiväkirjatyön kirjoittamisen aikana opin paremmin lähestymään ongelmia useammasta eri näkökulmasta. Vaikka välillä deadlinet puskevat päälle, olisi silti hyvä pysähtyä miettimään muutakin kuin sitä, mikä tulee ensimmäisenä mieleen. Myös se, että mielipiteitään saa ehkä helpommin tuotua esille, jos niitä yrittää tukea jollain.

Huomasin päiväkirjatyön aikana, että ongelmanratkaisua helpotti niiden kirjaaminen ylös. Tästähän on terminäkin Rubber Duck Debugging, jonka ideana on käydä kirjoitettu koodi läpi kumiankalle. Ratkaisuja on tätä kautta helpompi saada aikaan. Tulevaisuudessa uskon, että tällaista lähestymistapaa voisi hyödyntää. Jos ei ehkä ihan kumiankalle, mutta ainakin jollekin toiselle kehittäjälle, tai vaikka itselle notepadiin.

Olen myös työn aikana saanut parempaa kuvaa siitä, mikä kiinnostaa ja mitä haluan tehdä tulevaisuudessa. Vaikka työssäni se ei vielä ole suuremmin näkynyt, tarkoitus olisi yrittää suuntautua sovellusarkkitehdiksi. Myös pilviteknologiat kiinnostavat, ja näitä aion jatkossa opetella lisää.

4.2 Jatkokehitys- ja tutkimusmahdollisuudet

Työni sisältää muutaman eri kokonaisuuden. Versionhallinnan vaihto, CI-järjestelmän vaihto, ja ihan vain kehittämistä muutamassa eri projektissa. Mielestäni versionhallinnan ja CI-järjestelmien vaihdosta olisi pystynyt kummastakin tehdä oman projektin.

Versionhallintaprojektiin voisi tarkemmin kuvata vanhaa ja uutta järjestelmää, listata niiden eroja, sekä hyviä ja huonoja puolia. Työhön myös sisältyisi henkilöstön opastus uuden järjestelmän käyttöönotossa ja ongelmatilanteissa avustaminen.

Nykyisten sovellusten CI-järjestelmän siirrosta saisi myös järkevän kokonaisuuden. Siinä voisi olla listattuna taas eroavaisuuksia, sekä hyvät ja huonot puolet. Uuteen järjestelmään olisi hyvä kuitenkin tehdä sen verran uusiksi asioita, että voitaisiin myös tehdä jatkuvaa toimitusta asiakkaille. Järkevillä toimintatavoilla voitaisiin korvata järkyttävä määrä manuaalista työtä.

Milecia McG dev.to blogissa listaa hyvän CI:n merkeiksi mainitsemani manuaalisen työn vähentämisen. Lisäksi hyvän CI-putken tulisi olla nopea, artefaktit olla riippumattomia kohdejärjestelmästä, ja koodista pitäisi pystyä milloin vaan toimittamaan mikä tahansa versio. (Milecia McG. Viitattu: 10.11.2019)

Tällä hetkellä yksikkömme CI-putki ajaa buildia ja testejä noin puolitoista tuntia, eikä sillä voi toimittaa versioita asiakkaille lainkaan. Nightly buildit kyllä päivittävät omat testiympäristömme. Tätä olisi luultavasti melko helppo muokata niin, että asiakkaalle toimitus saataisiin myös tätä kautta. Tämä tulisi olemaan manuaalinen vaihe, sillä asiakkaan ympäristön päivittäminen ei onnistu ihan mihin tahansa aikaan.

Koska suurin osa firman projekteista on asiakaslähtöisiä, ihan kaikkea haluttavaa ei ole mahdollista tehdä. Esimerkiksi juuri näiden julkaisujen parantaminen olisi hyvä tehdä josain välissä, mutta se kuitenkin vaatisi asiakkaan rahoitusta. Järkevää tietenkin olisi, että nämä toimenpiteet kuuluisivat asiakkaiden ylläpitosopimuksiin, jolloin niihin voisi käyttää enemmän aikaa.

4.3 Analysoinnin hyödyntäminen

Työn analysointi on ollut jossain määrin hyödyllistä. Näin olen saanut syvennettyä omaa osaamista mainittuihin teknologioihin, enkä vain oman työn ohessa raapaissut pintaa. Mielestäni on kuitenkin hyvä ymmärtää, mitä on tekemässä. Asiat jäävät näin varmasti paremmin päähän, eikä sitten tulevaisuudessa tarvitse miettiä, että mistähän tässä nyt taas olikaan kyse.

Lähteet

Digia. Viitattu 15.08.2019

<https://digia.com/yritys/historia-ja-yrityskaupat/>

Agile Alliance. Viitattu 15.09.2019

<https://www.agilealliance.org/glossary/backlog-grooming/>

Thorben J. Viitattu 15.09.2019

<https://thoughts-on-java.org/implementing-the-repository-pattern-with-jpa-and-hibernate/>

James S. 2006. Dependency-Injection-Demystified. Viitattu 15.09.2019

<https://www.jamesshore.com/Blog/Dependency-Injection-Demystified.html>

Microsoft. 2019. Docker for Visual Studio Code. Viitattu 15.09.2019

<https://github.com/microsoft/vscode-docker>

Käyttäjä Spoike. 2010. Viitattu 16.09.2019

<https://stackoverflow.com/questions/2471606/how-and-or-why-is-merging-in-git-better-than-in-svn/2472251#2472251>

Lovisa J. 2015. RabbitMQ for Beginners. Viitattu 15.09.2019

<https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html>

Vincent D. 2010. A successful Git branching model. Viitattu 15.09.2019

<https://nvie.com/posts/a-successful-git-branching-model/>

Eugen P. 2018. OAuth2.0 and Dynamic Client Registration. Viitattu 15.09.2019

<https://www.baeldung.com/spring-security-oauth-dynamic-client-registration>

Ethan G. 2017. Migrate Subversion Repositories to Git and Preserve Project History. Viitattu 16.09.2019

<https://medium.com/@edgincvg/migrating-subversion-repositories-to-git-and-preserve-project-history-c639cfc6f1f6>

LiveRamp. 2012. Migrating from SVN to Git. Viitattu 16.09.2019

<https://medium.com/liveramp-engineering/migrating-from-svn-to-git-94de78c30e4b>

Bala V. 2018. How to migrate SVN to GitLab with few commands. Viitattu 16.09.2019
<https://medium.com/@venkateshpnk22/how-to-migrate-svn-to-gitlab-1dfe8c9b2bb7>

Atlassian. 2019. Bamboo variables. Viitattu 16.09.2019
<https://confluence.atlassian.com/bamboo/bamboo-variables-289277087.html>

Käyttäjät manojlds ja harmv. 2010, 2014. Viitattu 17.09.2019
<https://stackoverflow.com/questions/7879099/why-is-git-svn-fetch-so-slow>

Atlassian. Git Rebase. Viitattu: 23.09.2019
<https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase>

Peter E. 2018. Muutoksia aktiivimalliin ja muita viilauksia. Viitattu 23.09.2019
<https://ytk.fi/pulssi/blogi/muutoksia-aktiivimalliin-ja-muita-viilauksia>

Suomi.fi. Suomi.fi-palveluväylä. Viitattu 24.09.2019
<https://esuomi.fi/palveluntarjoajille/palveluvayla/>

Swati D. 2016. REST vs. SOAP: Choosing the best web service. Viitattu 26.09.2019
<https://searcharchitecture.techtarget.com/tip/REST-vs-SOAP-Choosing-the-best-web-service>

Stackify. 2017. SOAP vs. REST: The Differences and Benefits Between the Two Widely Used Web Service Communication Protocols. Viitattu 29.09.2019
<https://stackify.com/soap-vs-rest/>

(Milecia McG. 4 Attributes of a Good CI/CD Pipeline. Viitattu: 10.11.2019)
<https://dev.to/flippedcoding/4-attributes-of-a-good-ci-cd-pipeline-2hf4>