Ohjelmistoprojektityöskentely Sitowisella

Kalle Tolonen

Haaga-Helia ammattikorkeakoulu

It-tradenomin tutkinto

Amk-opinnäytetyö

2023

Tiivistelmä

Tekijä(t)
Kalle Tolonen

Tutkinto
Tradenomi

Raportin/Opinnäytetyön nimi
Ohjelmistoprojektityöskentely Sitowisella

Sivu- ja liitesivumäärä
42 + 3

Päiväkirjamuotoisessa työssä seurataan uuden ohjelmistokehittäjän matkaa hieman kokeneemmaksi ohjelmistoalan ammattilaiseksi. Kirjoittajan työpaikkana on Sitowise Oy, jossa työskentelee muutama sata IT-ammattilaista isomman konsultointiyrityksen osana. Päiväkirjan ensimmäiset merkinnät on tehty kirjoittajan perehdyttämisjakson viimeisellä kolmanneksella ja se päättyy koeajan puolivälin jälkeen. Oppimista kuvataan arkipäivien merkintöjen kautta ja jokaista viikkoa seuraa analyysi kyseisen viikon tapahtumiin.

Opinnäytetyön johdanto-osuudessa kuvaillaan opinnäytetyön tavoitetta, keskeisiä ammatillisia käsitteitä, opinnäytetyön tekijän senhetkistä osaamista, asiakasprojektista löytyviä sidosryhmiä ja työtehtävissä tapahtuvia vuorovaikutustilanteita.

Päiväkirjaosuudessa kuvaillaan ohjelmistokehittäjän arkea Sitowisella ja toimimista ohjelmistoprojektissa. Siinä kuvaillaan päivittäiseen tekemiseen kuuluvia työkaluja niiden käytön kautta ja käydään läpi sitä sisäistä keskustelua, jota on käyty eteen tulevien ongelmien ratkaisuja etsiessä.

Tämän opinnäytetyön pohdintaosuudessa reflektoidaan alkutilannetta siihen, mitä kehitystä on tapahtunut kahdeksan viikon päiväkirjaopinnäytetyöprosessin aikana. Siinä käydään läpi alussa määriteltyjä tavoitteita ja sitä, miten niissä onnistuttiin. Kirjoittaja käsittelee myös sitä, mitä työn tekemisen analysointi on tuonut mukanaan ja mitkä ovat hänen näkemyksensä omasta tulevaisuudestaan nyt tehdyn perusteella.

Asiasanat
Ohjelmistokehitys, ketterät menetelmät, tietotekniikka, Java

Sisällys

Johdanto

1.1 Työympäristön, toimintatapojen, osaamisvaatimusten ja työtehtävien kuvaus

Opinnäytetyö tehdään Sitowise Oy:lle asiakasprojektissa ohjelmistokehittäjänä työskennellen. Projekti on asiakkaalle tuotettava ohjelmistoprojekti, joka liittyy omaisuudenhallintaan.

Sitowise on älykkäisiin kaupunkiratkaisuihin, liikkumisen sujuvuuteen ja elämisen tiloihin suuntautunut suomalainen pörssiyritys (Sitowise s.a. a). Opinnäytetyön tekijä työskentelee Sitowisella Digi -liiketoiminta-alueella ohjelmistokehittäjänä. Ohjelmistokehittäjän työ vaatii ohjelmointiosaamista, ryhmässä työskentelyä, oman työn itsenäistä johtamista, projektityön osaamista ja substanssiosaamista laajasta työkalukirjosta. Oleellisia työkaluja ja tekniikoita ohjelmistokehittäjälle tässä asiakasprojektissa ovat: olio-ohjelmointi, Jira, Confluence, Git, Github, Linux, Java & Spring Boot, Swagger, Angular & TypeScript ja ketterät kehitysmenetelmät.

Digitaaliset palvelut (Digi -liiketoiminta-alue) kehittää asiakaslähtöisesti digitaalisia ratkaisuja, jotka tuovat datan näkyväksi, edistävät resurssiviisasta toimintaa ja lisäävät ymmärrystä tulevaisuudesta ja sen tuomista muutoksista. Digin palveluihin kuuluvat asiakaslähtöinen tietojärjestelmäkehitys, omat tuotteet, analytiikka ja tiedonhallinta, sekä asiantuntija- ja konsultointipalvelut. (Sitowise s.a. b)

1.2 Ammatillisen kehittymisen tavoitteet, aikataulu ja rajaukset

Työn keskeisinä tavoitteina on kehittyä ohjelmoijana, ohjelmistokehityksen ammattilaisena (ketterät kehitysmenetelmät), oppia käyttämään projektin kannalta keskeisiä työkaluja (Java, Angular, Git, Github, Swagger, PostgreSQL, Spring Boot, Jira, Confluence) ja hyödyntää näissä parhaita menettelytapoja. Ammatillisen kasvun lähtökohtana on uran ensimmäinen työpaikka, sekä työkokemus ohjelmistokehittäjänä.

1.3 Keskeiset ammattikäsitteet ja lähteet

Käsite	Määrittely
Ohjelmisto	Ohjeet, jotka kertovat tietokoneelle mitä sen tulee tehdä
Ohjelmistokehittäjä	Henkilö, joka kehittää ohjelmistoja
Git	Versionhallintajärjestelmä

Github	Microsoftin omistama palvelu, joka toimii versionhallinnan palvelimena
Ketterät kehitysmenetelmät	Kehitysmenetelmiä, jotka tuottavat asiakkaalle arvoa siten, että tuotteesta tehdään nopeaan tahtiin julkaisuja
Java	Ohjelmointikieli, jolla projektin backend on ohjelmoitu
Angular	Ohjelmointikehikko, jonka varaan projektin frontend rakentuu
Swagger	Rajapintojen kuvauksen ja rakentamisen työkalu
PostgreSQL	Tietokantaohjelmisto
Spring Boot	Javan lisäosa
Määrittely	Reunaehdot ja vaatimukset toiminnallisuudelle
Dokumentaatio	Projektia koskeva, erikseen tallennettu informaatio
Confluence	Dokumentaation tallennusohjelmisto
Jira	Projektinhallinnan työkalu
Tiketti	Työtehtävän seurattava tunniste Jirassa
Sprintti	Kolmiviikkoinen kehitysjakso, jonka jälkeen tehdään julkaisu
Inkrementti	Kolmen sprintin kokonaisuus
Inkrementtisuunnittelu	Inkrementtiä edeltävä suunnittelutapahtuma kaikkien tiimien ja asiakkaan edustajien kanssa
Sprinttisuunnittelu	Tiimin sisäinen suunnittelu ennen sprintin käynnistämistä
Demo	Uusien toiminnallisuuksien esittelytilaisuus
Scrum master	Projektipäällikkö, joka vastaa tiimin toiminnan sujuvoittamisesta
DoD	Definition of Done – tikettiä koskeva valmiin määritelmä, jonka jälkeen tiketti voidaan sulauttaa kehityshaaraan gitissä

JSON	Standardimuotoinen merkkijono-olio, joka toimii tiedonsiirron välineenä
Kanban	Projektinhallinnan työkalu, jossa tehtävät jaotellaan niiden statuksen mukaan
YAML	JSON-merkintätavan kaltainen tapa merkitä asioita – Yet Another Markup Language
Acceptance Criteria	Hyväksymiskriteerit – näiden perusteella tiketti voidaan hyväksyä, käytetään usein bugikorjauksien yhteydessä
Testaus	Ohjelmistokehityksen vaihe, jossa katsotaan, että tuotos tekee sen mitä siltä odotetaan
Palvelin	Backendin ohjelmistoa ajava tietokone
Käyttöliittymä	Käyttäjälle esitettävä edustakoodin tuotos
Avoin lähdekoodi	Ohjelmisto, jonka käyttöä ei ole rajoitettu

Taulukko 1. Peittomatriisi päiväkirjaopinnäytetyön käsitteistä

Ensimmäisinä viikkoina työ keskittyy ohjelmistokehittäjän päivittäisen tekemisen kannalta keskeisimpään asiaan – eli ohjelmointikielien ja niiden rakenteiden käyttämiseen työssä. Työn edetessä siirrytään kohti vaikeampia asioita, jotka vaativat kokonaisuuden ymmärtämistä. Viimeisellä seurantajaksolla keskitytään alustavan suunnitelman mukaan tukijärjestelmien ymmärryksen kehittämiseen.

1.4 Työn ulkopuolelle rajatut kokonaisuudet

Työssä ei käsitellä ohjelmistokehitystä muissa alan yrityksissä. Työssä käsitellään vain yhdessä asiakasprojektissa työskentelemistä.

Lähtötilanteen kuvaus

Työn tekeminen tapahtuu fyysisesti Sitowisen Turun toimistolla ja satunnaisesti myös etäyhteyksien varassa. Ohjelmoinnin työympäristönä toimii full stack-ohjelmistoprojekti, jossa backend on toteutettu Javalla ja Spring Bootilla. Backendin tietokantana toimii Hibernate/PostgreSQL-kanta, jota ajetaan konteissa Dockerissa. Tietoa serverin ja käyttöliittymän välillä vaihdetaan Swaggerilla generoitujen rajapintojen välityksellä. Projektissa on myös aikaisemmassa kehitysprojektissa tehtyjen ratkaisujen takia toinen backend, jonka kautta osa käyttöliittymän tarvitsemasta tiedosta liikkuu. IDE-ohjelmistoina projektissa käytetään IntelliJ Ideaa ja Visual Studio Codea, joissa on asennettuna yhdessä sovittuja lisäosia koodin ulkoasun ja toiminnallisuuden tarkastamiseksi.

1.5 Oman nykyisen työn analysointi

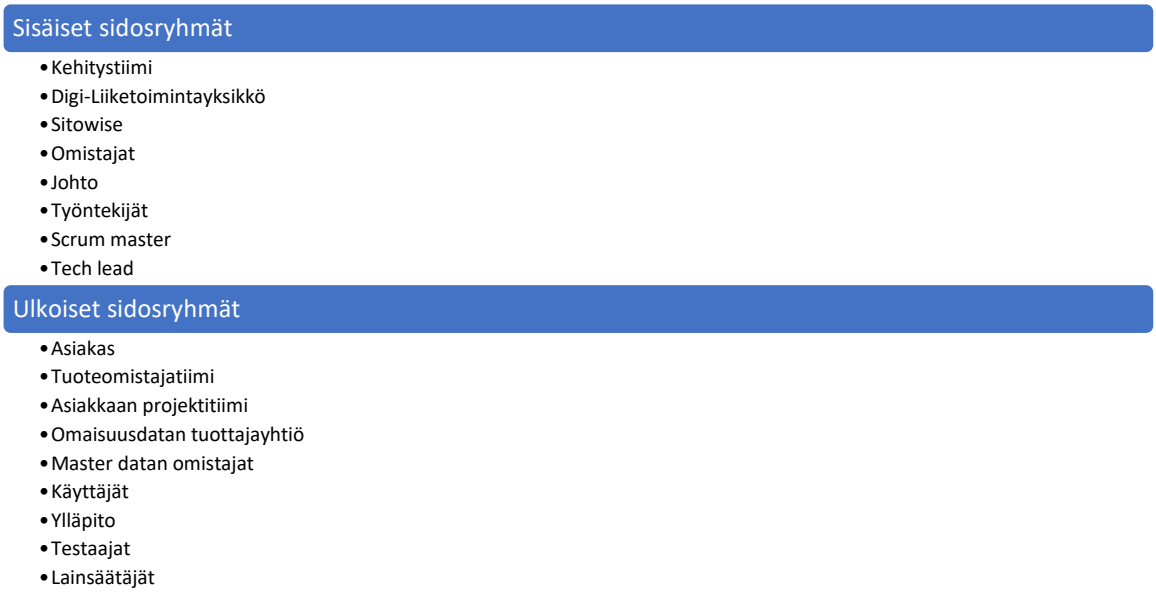
Nykyisessä työssäni ohjelmoin serverillä ajettavaa koodia Javalla ja käyttäjän koneella suoritettavaa koodia Angularilla (TypeScript/JS). Tämän lisäksi työhön kuuluu asioiden selvittelemistä tietokannasta (PostgreSQL), rajapintojen rakentamista (Swagger) ja käyttämistä.

Työ vaatii perusteellista ymmärrystä ohjelmointikielien loogisista rakenteista, tietorakenteista, tietokannoista, rajapinnoista, ohjelmointialan käytännöistä, projektityöstä ja ketteristä kehitysmenetelmistä. Nämä ovat kaikki osa-alueita, joihin minun tulee panostaa, jotta kehityn ammattilaiseksi. Opittavia asioita on valtavasti ja työt aloittaessani koin projektin laajuuden lievästi järkyttävänä, kun olin koulussa ja omassa harrastelussani tottunut siihen, että koko projekti ja koodi on helposti yhden ihmisen ymmärrettävissä. Asia on aivan toinen, kun kyseessä on vuodesta 2017 asti käynnissä ollut tuotos, jonka parissa työskentelee satoja ihmisiä eri yrityksistä ja sidosryhmistä.

Tähän mennessä olen hankkinut Java-osaamista koulusta, jonka lisäksi olen korjannut projektissa olevia bugeja, sekä rakentanut kaksi uutta rajapintaa. JavaScript-osaamiseni on vajavaista, enkä omaa juuri ”hello worldia” suurempaa kokemusta Angularista. Tietorakenteissa ja tietokannoissa minulla on koulun kautta opitut valmiudet, eli osaan tehdä erilaisia tietokantakyselyitä ja ymmärrän tietokantojen perusasiat, kuten välitaulut, taulujen avaimet ja SQL:n syntaksin aloittelijan tasolla. Ammatillisesti olen vasta-alkaja ja työkokemusta on tähän mennessä kertynyt kaksi kuukautta. Osaamistasoni on työtehtäviin nähden alhainen – olen aloitteleva toimija. Kehitystä ja oppimista tapahtuu jokaisena päivänä, eli koen olevani hyvässä asemassa, kun minulla on mahdollista kartuttaa taitojani ja saada siitä samalla palkkaa.

1.6 Sidosryhmien esittely

Tärkeimmät sisäiset sidosryhmät ovat: projektin kehitystiimi, Digi-liiketoimintayksikkö, Sitowise, omistajat, johto, työntekijät ja Scrum Master Ville. Ulkoisia sidosryhmiä ovat: Asiakas, Tuoteomistajatiimi, omaisuusdatan tuottajayhtiö, master datan omistajat, palvelun käyttäjät, ylläpito, testaajat ja lainsäätäjät.



Kuva 1. Sidosryhmät

Kehitystiimin intresseissä on tuottaa asiakkaalle ja käyttäjille arvoa, sekä huolehtia siitä, että kaikki projektiin tehtävästä työstä on laskutettavaa työtä. Scrum master ohjaa työtä ja vastaa sen sujuvuudesta, eli käytännössä poistaa esteitä kehitystiimin työltä ja organisoi sitä. Digi-liiketoimintayksikön intresseissä on kasvattaa olemassa olevia asiakkuuksia, sekä hankkia uusia projekteja taloon. Sitowisen tulee tuottaa omistajilleen arvoa, jota mitataan pörssiyhtiön tapauksessa markkina-arvolla, arvostuskertoimilla ja osingoilla. Johto pyrkii siihen, että omistajien tahtotila toteutuu päivittäisessä toiminnassa ja Sitowise tuottaa omistajilleen arvoa. Työntekijät haluavat pitää työympäristönsä miellyttävänä, kuormansa sopivana ja korvauksensa työstä hyvässä suhteessa tuotettuun työpanokseen. Tech lead pyrkii kehittämään projektia ja ohjaamaan siinä tehtäviä isoja päätöksiä sellaiseen suuntaan, jossa ominaisuuksien kehittäminen, bugien korjaaminen ja ylläpito ovat helppoja, laadukkaita ja nopeita toteuttaa.

Asiakas haluaa itselleen tuotteen, joka on heidän toiveidensa mukainen ja täyttää sille asetetut vaatimukset – tämän tulee tapahtua sovituissa aika- ja kustannusraameissa. Tuoteomistajatiimi kantaa asiakkaan ääntä päivittäisessä toiminnassa ja ohjaa toimintaa eri tiimien välillä. Asiakkaan

projektitiimin intresseissä on huolehtia siitä, että toivottuja ominaisuuksia varten tuotetaan määrittelytietoa ja toimitaan linkkinä käyttäjien ja tuotekehityksen välissä. Omaisuusdatan tuottajayhtiö vastaa aineiston tuottamisesta järjestelmään ja sen käsittelystä. Master datan omistaja toimii master datana useille sovelluksen kohteiden tiedoille. Käyttäjien intresseissä on saada käyttöönsä helppokäyttöinen ohjelmisto, joka täyttää heidän tarpeensa ja mihin tehdään käyttäjälähtöisesti parannuksia. Ylläpidon kannalta ohjelmistotuotteen on oltava vakaa, mahdollisimman bugiton, testattu ja turvallinen – on myös syytä huomioida se, että tuotteen suorituskyky on oleellinen osa ylläpidon elämää. Testaajia kiinnostaa se, että testattavat kokonaisuudet ovat järkevän kokoisia ja että niihin on helppo kehittää automaatti- tai manuaalitestejä. Lainsäätäjien intresseistä oleellisinta on se, että projektissa noudatetaan lakia ja säädöksiä.

1.7 Työpaikan vuorovaikutustilanteet

Päivittäisessä työssäni tapahtuu vuorovaikutusta tiimiläisten kanssa jatkuvasti. Käyn töissä fyysisesti paikan päällä, jotta kynnys asioiden kysymiselle kollegoilta on mahdollisimman alhainen. Koko tiimin kanssa kohtaamme päivittäin daily-tapaamisissa, joka on osa ketterien kehitysmenetelmien käytäntöjä (Scrum.org s.a.). Näihin tapaamisiin osallistuvat kehittäjien lisäksi tuoteomistaja, scrum master ja testaajat. Tapaamisten keskeisenä sisältönä on työn edistymisestä kertominen ja mahdollisten esteiden poistaminen kehitystyön tieltä – ne pitävät tiimin myös omalla tavallaan tilivelvollisena, sillä joka päivä tulee kerrottua siitä mitä on saanut aikaan ja mitä aikoo seuraavaksi tehdä.

Viikottaisena rutiinina on myös kaikkien projektiin liittyvien kehittäjien ja ylläpidon tapaaminen viikoittaisessa kehittäjätapaamisessa. Tämän kohtaamisen kantavana ajatuksena on tuottaa synergiaa eri tiimien välille, sekä keskustella ajankohtaisista asioista, sekä siitä miten muutokset eri järjestelmissä vaikuttavat toisiin järjestelmiin. Näissä tapaamisissa keskustellaan asioista hyvin käytännönläheisesti.

Sprinttien (eli kolmen viikon kehitysjaksojen) päätteeksi pidetään demotilaisuus sisäisesti ja ulkoisesti sprintin aikaansaannoksista. Pääsääntöisesti uuden ominaisuuden tai demottavan asian demonstroi sen kehittäjä. Sprinttiin liittyviä vuorovaikutustilanteita ovat lisäksi retrot ja suunnittelut. Retroissa katsotaan menneeseen ja peilataan siinä tehtyjä ratkaisuja ja toimintamalleja, sekä keskustellaan siitä, miten toimintaa on mahdollista parantaa jatkossa. Sprinttisuunnittelussa suunnitellaan tulevan sprintin tehtävien toteutusta.

Kaikista harvinaisimmat vuorovaikutustilanteet ovat inkrementtiin liittyviä. Inkrementti on kolmen sprintin mittainen kehityspaketti. Inkrementtidemossa demotaan kaikki kyseisellä inkrementillä teh-

dyt uudet ominaisuudet asiakkaalle ja käydään laajemmalla joukolla samankaltainen retro- & suunnittelurupeama kuin sprinttien tapauksessa. Ratkaisevana erona on se, että inkrementtisuunnittelu tapahtuu tässä projektissa fyysisenä kahden päivän rupeamana, jossa on runsaasti myös asiakkaan edustajia paikalla.

Kaikissa näissä vuorovaikutustilanteissa on samankaltaiset haasteet osaamiselleni – eli teknisen tietämyksen taso, sekä ymmärrys kehitysprosessista kokonaisuutena. En vielä täysin ymmärrä kaikkia teknisiä yksityiskohtia. Kehitysprosessin kokonaiskuva on alkanut hahmottua ja sen kanssa ei enää ole suuria haasteita. Sosiaalisesti olen kykeneväinen ja entinen ammattini on ollut ihmisten johtamisessa tapahtuma-alalla, joten pystyn toimimaan hyvin erilaisten ihmisten kanssa.

Seurantajakson raportointi viikkoanalyyseineen

Seurantajakson pitää sisällään päivittäisiä työtehtäviä ja tapahtumia päiväkirjamuotoisena tekstinä, sekä viikon päätteeksi tapahtuvan viikkoanalyysin. Analyysiosiossa perehdytään käsitteisiin ja teoriaan viikon tapahtumien taustalla.

1.8 Seurantaviikko 1

Maanantai 21.11.2022

Maanantai alkoi osaltani siten, että jatkoin edellisellä viikolla miltei valmiiksi saattamani tiketin tekemistä. Tässä tiketissä oli kysymys ohjelmistokirjaston päivittämisestä toiseen, sillä aikamääreisiin erikoistuneesta moment.js-kirjastosta on tuki ja jatkokehitys loppunut (Moment.js s.a.). Tiimi on valinnut tilalle date-fns-kirjaston, joka on hyvin suosittu kirjasto – ainakin Githubin 1.7 miljoonan sitä hyödyntävät projektin mittarilla (Github 2022), ja siten sillä on todennäköisesti pitkä tuki myös tulevaisuudessa. Sain tiketin tehtyä miltei valmiiksi, mutta päädyin jakamaan siihen liittyvänä tehtävänä toiseen tikettiin deprekoituneiden testien osalta kirjaston poistamisen, jotta yksittäisessä tiketissä ei tehtäisi suuria muutoksia, koska ne hidastavat katselmointia ja testaamista. Tein tämän lisäksi vielä kolmannen tiketin, jossa poistettaisiin kaikki deprekoituneet testit repositoriosta, jotta ne eivät ole siellä tyhjän panttina paisuttamassa lähdekoodin kokoa ja luomassa turhia riippuvuuksia.

Tein katselmointia kollegani tekemään ”git hooks”-tikettiin, jossa projektiin luodaan ensimmäisiä Gitin tukemia automaatiotoiminnallisuuksia. Tiketeissä on kyse committien viestien standardoinnista hakujen helpottamiseksi ja lisäksi toisessa tiketissä on tekeillä pull requesteihin malli, joka placeholderin avulla ohjaa kehittäjiä linkittämään varmasti pull requestin ja Jiran tiketin toisiinsa.

Näiden töiden lisäksi osallistuin Dailyyn, joka on päivittäin järjestettävä scrum masterin vetämä kokous, jossa tuoteomistaja, kehittäjät ja testaajat kohtaavat pikaisesti ja keskustelevat tämänhetkisistä tehtävistään ja niitä mahdollisesti koskevista esteistä.

Toinen päivän kokous liittyi saavutettavuuteen, joka on hankkeessamme nouseva teema. Kaikki kehittäjät osallistuivat saavutettavuuskoulutukseen, jossa kävimme läpi sitä mitä saavutettavuus on ja mitä se tarkoittaa ohjelmistoprojektissa. Kehittäjien mielestä saavutettavuuden ja käytettävyyden ero oli usein häilyvä ja saavutettavat sivut eivät välttämättä ole käytettäviä. Onkin tärkeää huomioida rajoitteiset käyttäjät siten, että käytettävyys säilyy ja tässä ovat suurena apuna neljä saavutettavuuden peruspilaria: havaittavuus, hallittavuus, ymmärrettävyys ja lujatekoisuus (WCAG 2022).

Havaittavuus tarkoittaa sitä, että käyttäjällä tulee olla kyky havainnoida tietoa, eikä se esimerkiksi saa olla hänen aisteilleen näkymätöntä. Hallittavuus tarkoittaa sitä, että ohjelmiston elementit ovat

saavutettavia ja niihin pääsee käsiksi myös apuvälineitä käyttämällä. Ymmärrettävyys tarkoittaa sitä, että sisältö on selkeää ja yksiselitteistä. Lujatekoisuus tarkoittaa sitä, että teknologian kehitys ei estä sisällön saatavuutta tulevaisuudessa. (WCAG 2022)

Tiistai 22.11.2022

Tiistaina en ehtinyt ohjelmoimaan kovin paljoa, sillä suurin osa ajasta meni kokouksissa. Tämä tuntui, ja tuntuu edelleen, hurjalta ajankäytöltä, kun on vuosikymmeniä tehnyt töitä siten, että palavereissa vietetty aika on minimaalista.

Päivä lähti käyntiin Sprint Reviewillä ja sen yhteydessä meillä oli sisäinen demo sprintillä valmistuneista tiketeistä niiltä osin, kuin mitä niistä pystyisi asiakkaalle demoamaan. Tällä sprintillä valmistui todella paljon bugikorjauksia ja serveripuolen koodia, joka ei sellaisenaan näy loppukäyttäjälle kuin korkeintaan välillisesti parantuneena käyttökokemuksena. Näiden takahuonekorjausten lisäksi pääsin demoamaan tekemiäni CSV-parannuksia, jotka tulevat näkymään asiakkaan arjessa huomattavasti parantuneena tuonti- ja vientikäyttäytymisenä. Tällä päättyneellä sprintillä korjasin tiedoston nimen generointia, skandinaavisten merkkien toimivuutta ja päivämäärien muotoilua määrittelyä vastaavaksi.

Toisena kokouksena oli päättyneen sprintin retrospektiivi, jossa kävimme läpi hyvin tehtyjä asioita, sekä sellaisia, missä meillä on vielä tekemistä. Näiden kohtaamisten tarkoituksena on parantaa prosessia. Tärkeimpänä asiana tästä kokouksesta jäi mieleeni se, että Git Hooksit eivät vielä ole valmiita ja se, että ne ovat kyllä toivottu ominaisuus.

Päivän päätteeksi järjestimme kaikkien hankkeen kehittäjien ja asiakkaan edustajien kanssa sprinttidemon, jossa esittelimme sanallisesti bugikorjausten ja muiden näkymättömien parannusten sisältöä ja sen jälkeen demosin asiakkaalle CSV-käsittelyyn tulleita muutoksia. Tämä oli etukäteen aika jännittävää, sillä en ollut koskaan aikaisemmin demonnut mitään ja valmistauduin tilaisuuteen huolellisesti, jossa tuo sisäinen demo myös varmasti auttoi. Muut tiimit demosivat myös tekemäänsä työtä ja lopuksi pidimme lyhyen keskustelun retroiluhengessä tiimien kesken.

Keskiviikko 23.11.2022

Tänään oli ensimmäinen päiväni päivystäjänä – meillä on tiimisssämme sellainen käytäntö, että jokainen vuorollaan päivystää. Päivystäminen tässä projektissa tarkoittaa sitä, että päivystäjä huolehtii, että tiimillä on edustus kaikissa tärkeissä kokouksissa ja pitää näiden sisällöstä myös päiväkirjaa, joka on kaikkien nähtävillä. Päivystäjä toimii myös scrum masterin sijaisena, jos tämä on estynyt. Mielestäni kaikista kriittisin päivystäjän tehtävistä on kuitenkin se, että päivystäjä pitää huolta

siitä, että eri kanavista tulevat viestit tiimille saavat niihin reagointia, eli siis usuttaa oikean tiimin jäsenen kulloisenkin kysymyksen äärelle, jos ei siihen itse osaa vastata.

Dailyssä ei tapahtunut mitään ihmeellistä, kävimme tikettien työtilannetta läpi ja sanoin että maanantaina tekemäni poistotiketti on nyt testausta vaille valmis mergettäväksi, kun se meni katselmoinnista läpi. Tämä olisi mielestäni hyvä testata pian, sillä se estää toisen tiketin etenemistä.

Iltapäivällä osallistuin saavutettavuuskoulutuksen jälkimmäiseen osuuteen, jossa käytiin läpi saavutettavuutta teknisemmistä lähtökohdista, eli esimerkiksi siitä, että näytönlukijan kannalta Aria-label-määritteet ovat hyvin tärkeitä. Lisäksi mieleeni jäi myös, että linkit tulee tyylitellä linkeiksi ja painikkeina tulee käyttää button-elementtiä. Opimme myös erilaisista työmenetelmistä saavutettavuuden parantamiseksi ja yksi näistä oli mielestäni erityisen tärkeä, eli sivujen testaaminen näytönlukijan avulla. Vain testaamalla toimintaa konkreettisesti voi varmistua sen toimimisesta. Olin ennen koulutusta ollut jollain tasolla tietoinen siitä, että näytönlukijoita löytyy ja siitä, että erilaisia avustimia on, mutta oli mukava nähdä niiden käyttöä todellisuudessa. Koulutus pidettiin virtuaalisesti Teamsissa ja siinä hyödynnettiin Miro-alustaa, jolla vuorovaikutus esityksen kanssa on mahdollista. Tämä voitti perinteiset kalvosulkeiset helposti, sillä tässä yleisöä osallistettiin moneen otteeseen ja meille syntyi hyvää keskustelua.

Päivän viimeisessä palaverissa tuurasin scrum masteria Vikaraadissa. Vikaraati on kokous, jossa käydään viimeisimmän viikon kaikki tuotantokäytöstä löytyneet bugit läpi ja määritellään niille sopiva prioriteetti. Lisäksi käydään myös muualta kuin tuotantokäytöstä löytyneitä bugeja läpi, jos niissä on high- tai critical-tasoinen prioriteetti kirjattuna.

Kehitystyön puolella aloitin uutta tikettiä sillä välin, kun odotin Moment.js-poistoa käsittelevän tikettini esteenä olevan tiketin valmistumista testausputkesta. Uusi tehtäväni on CSV-käsittelyn korjaaminen muutamien sarakkeiden osalta, jotka generoidaan tällä hetkellä poikkeuksellisesti johtuen historiallisesta painolastista ja tämän objektityypin erityisominaisuuksista.

Torstai 24.11.2022

Torstai alkoi SoS-kokouksella, eli Scrum of Scrumsilla, jossa kaikkien hankkeessa olevien tiimien tämänhetkisestä tilanteesta pidetään Dailyä vastaava tapaaminen, mutta vain viikoittain. Tällä tavalla jokainen projekti pysyy paremmin kartalla siitä, mitä muissa hankkeeseen kuuluvissa projekteissa tapahtuu tällä hetkellä. Kirjasin päivystäjänä tässä kokouksessa ilmi tulleet asiat ja jaoin ne tiimille, sekä tallensin ne ajankohtaisiin asioihimme. Tästä jatketiin luontevasti tiimin keskeiseen Dailyyn, jossa nostin esille sen, että toisen tiimin osalta oli tullut meidän omistajuudessamme olevaan repositorioon pull request ja että se tulisi käsitellä. Tästä syntyi keskustelua ja päädyimme

kutsumaan kokoon palaverin siitä, miten nämä tiimien keskeiset asiat olisi helpointa ratkaista prosessiteknisesti ja mahdollisimman vähällä byrokratialla.

Seuraavaksi kohtaisimme lounaan jälkeen tämän toisen tiimin kanssa ja sovimme yhteiseksi käytännöksi Rocket.Chat-ohjelmaan (jossa on hankkeen laajuinen viestintä jo muutenkin) projektimme kanavalle perustettavaksi uuden keskustelun pieniä kysymyksiä varten, erityisesti tätä toisen tiimin kehittämää ja meidän omistamaamme repositoriota koskien. Lisäksi sovittiin siitä, että kaikki pull requestit, jotka tarvitsevat katselmointia tulee saada meille Jiraan, josta ne sitten hyväksymme ne normaalin prosessimme mukaisesti.

Iltapäivällä oli kehittäjätapaaminen, mikä on viikoittainen yhteistyöpalaveri kaikkien hankkeen kehittäjien ja ylläpidon kanssa. Tein virheen ja en puhunut mitään silloin kun meidän tiimimme vuoro oli, sillä paikalla oli tiimistämme kokeneempia kehittäjiä ja myös lead developer. Tämä paljastui myöhemmin virheeksi ja seuraavalla kerralla pidän sitten puhetta.

Kehitystyönä tein töitä CSV-tikettini kanssa ja minulla oli todella suuria vaikeuksia RxJS:n kanssa – en vain tunnu käsittävän sitä, miten sen subscribe- ja pipe-metodit toimivat. En ole kovin kokenut edustakoodin kehittäjä ja huomaan että se on selkeästi heikoin osaamisalueeni.

Perjantai 25.11.2022

Perjantaina nostin päivystäjänä Dailyssä asiakkaalta tullutta tuontihuolta ja siitä kirjattiin myöhemmin tuoteomistajan toimesta kriittinen bugi. Asiakkaan projektipäällikkö myös esitti vahvan toiveensa siitä, että tämän ongelman ratkaiseminen olisi ensiarvoisen tärkeää. Kollegani kysyi minulta apua ongelman ratkaisemiseksi ja yhdessä löysimme sen juurisyyn, joita oli kaksi. Asiakas oli tehnyt tietojen muotoilussa virheen ja tuontitoiminnallisuudesta ilmaantui jo aiemmin korjaamaani bugiin liittyvä virhe virheilmoitusten generoinnissa. Aiemmassa bugikirjauksessani totesin, että virheet generoidaan väärin tietyillä tiedoilla ja tässä asiakkaan tapauksessa asiakkaalle ei edes näytetä mitään virhettä, vaikka asiakkaan tekemä tiedon muotoiluvirhe oli kehittäjille CSV-tiedostoa katsomalla miltei välittömästi ilmeinen.

Muu päivä meni suurelta osin CSV-tikettini kanssa – minulla oli edelleen suuria ymmärrysvaikeuksia asynkronisten toimintojen toiminnan kanssa. Kysyin kollegaltani neuvoa ja se auttoi hieman eteenpäin. Tämän päivän jälkeen luulen, että tilanteeni on se, että onnistun hakemaan toivomani JSON-objektin rajapinnasta ja seuraavan viikon asioiksi jää sitten sen jatkokäsittely, jotta siitä saadaan asiakkaan toivomat tiedot irti. Kysyin myös kollegaltani siitä, että miten saan työhaaraani develop-haaraan kehitysaikana tapahtuneet muutokset mukaan ja hän valaisi, että ne voi mergetä siihen. Sovin myös toisen kollegani kanssa alkuviikolle tapaamisen, jossa hän avaa minulla rautalangasta vääntämällä RxJS:n ja Angularin toimintaa.

Ilmoittauduin myös tammikuussa järjestettävälle ja työnantajani tarjoamalle DevSecOps-kurssille, jossa tullaan käsittelemään todella kiinnostavia aiheita ja päästään syventymään minulle pintapuolisesti tuttuun aiheeseen laajemmin käytännön harjoitusten kautta.

Viikkoanalyysi 1

Tämän viikon tavoitteina oli oppia ohjelmistokehityksen prosesseista tässä projektissa ja kehittää ymmärrystäni edustan koodista. Osa käyttöliittymän generoivasta koodista on minulle jo ennestään tuttua, sillä olen tehnyt siihen pieniä bugikorjauksia. Myös ohjelmistokehityksen prosessit ovat osittain tuttuja, sillä olen työskennellyt tässä projektissa nyt syyskuusta lähtien, eli kohta kolme kuukautta. Tämä sprintti on silti ensimmäinen, jossa olen päivystäjänä ja käytän siten enemmän puheenvuoroja ryhmän puolesta kuin aiemmin.

Tärkeimpänä asiana kehitystyössä on oppia lisää Angularista, Typescriptistä ja RxJS:stä, sillä ne tekniikat ovat käytössä projektin käyttöliittymässä. Serveripuolen sovellukset ja rajapinnat on rakennettu Javalla ja se on minulle tuttu ja turvallinen ympäristö koulusta. Joudun tehtäväni toteutuksessani hakemaan tietoa rajapinnasta ja käsittelemään tätä vastaanottamaani tietoa siten, että siitä jalostuu asiakkaalle hyödyllinen ominaisuus tuotteeseen, eli uusia rivisisältöjä CSV-vientiin. Minulla on hyvin vajavaiset tiedot siitä, mitä esimerkiksi Subscription-metodi tekee. Dokumentaatiossa sen kerrotaan olevan objekti, joka edustaa kertakäyttöistä resurssia (RxJS s.a.). Käytännössä tuntuu siltä, että se on objektiin kuuluva metodi, jota kutsutaan, jos halutaan iteroida jotain minkä saapumista joudutaan odottamaan. Viikolla minulle tuotti vaikeuksia se, että debuggerissa näkyi, kun sain objektin otettua vastaan, mutta en pystynyt kutsumaan sitä funktioni ulkopuolelta. Kollegani valaisi minua ja kertoi, että suoritus tapahtuu sitten kun se tapahtuu ja tieto ei ehdi minun kutsuuni mukaan. Sain tehtyä koodinpätkän, joka hoitaa noutamisen siten, että minulla on tarvitsemani tieto. Koodini ratkaisu ei ole mitenkään optimaalinen tai kaunis tällä hetkellä, mutta se on hyvä lähtöpiste.

Koen päässeeni tavoitteisiini, sillä opin paljon toimiessani päivystäjänä ja nostin sellaisia asioita pöydälle, jotka tarvitsivat tiimin huomiota. Kehittämisen puolella opin myös paljon, sillä pääsin minulle haastavassa tehtävässä alkuun ja nyt minun on mahdollista jatkaa työtä eteenpäin käsittelemällä noudettua tietoa. Seuraavalla viikolla otan tavoitteekseni kehittyä edustan koodin kanssa.

1.9 Seurantaviikko 2

Maanantai 28.11.2022

Viikko alkoi sillä, että epäonnistuin develop-haaran mergeämisessä työhaaraani, jossa ei kyllä onneksi ollut mitään kovin tärkeää. Täytyy varmasti harjoitella myös Gitin käyttöä enemmän, jotta se on vastaisuudessa itsevarmempaa ja tuo toivottuja tuloksia.

Päätin aloittaa kehitystyön alusta CSV-ongelmassani, sillä sulauttaminen osaksi jo olemassa olevaa vientirutiinia on osoittautunut todella vaikeaksi, kun siinä on todella paljon minulle uusia asioita, jotka liittyvät Observableen ja Subscribeen. Olen saanut tukea osaamattomuuteeni kollegoilta. Lisäksi apua on antanut toimiston, itseni jälkeen, junioreimman kehittäjän kommentti siitä, että hänellä kesti vuoden ennen kuin Angularin ja RxJS:n asiat oli kunnolla sisäistetty. Päivän lopputuloksena oli se, että onnistuin listaamaan objektieni oidit, eli yksilölliset tunnisteet (Alvestrand 1997).

Tiistai 29.11.2022

Dailyssä ei tullut eteen mitään sellaista, joka olisi estänyt kenenkään työtä. Oma Moment-kirjaston päivittämiseen liittyvä tehtävän ositustikettini oli edelleen testaussarakkeessa, joten en päässyt edistämään sitä asiaa.

CSV-tiketin kanssa vaihdoin lähestymistapaa ja sain haettua kannasta käyttöliittymään generoidut kenttien nimet, joiden perusteella minun olisi mahdollista generoida otsikkokentät lopulliseen tuotokseen. Myös kenttien käännökset pitäisi saada poimittua jostain, mutta se jäi tässä vaiheessa vielä hämärän peittoon.

Osallistuin asiakkaan, tuotepäällikön ja hankkeen tuoteomistajien kanssa palaveriin, jossa käsiteltiin hyvin teknisellä tasolla fyysisten ja loogisten objektien tilanhallintaa. Fyysiset objektit ovat tässä tapauksessa konkreettisesti maailmassa olevaa asiakkaan omaisuutta, eli jos asiakas olisi sähköyhtiö, niin ne voisivat olla, vaikka muuntamoita tai sähköjohtoja. Loogiset objektit taas tarkoittavat metakäsitteitä, jotka liittyvät fyysisten objektien hallinnointiin – esimerkiksi tässä kuvitteellisessa sähköyhtiössä voisi olla käytössä muuntajien osoitteisto, jossa muuntaja sijaitsisi kantaverkon solmukohdassa xyz. Meidän hankkeessamme tietoa tuodaan järjestelmiin monista lähteistä, joissa tiedon master datan, eli lähtötiedon, omistaja vaihtelee. Tiedon lähtöjärjestelmien ja meidän järjestelmiemme on syytä noudattaa yhteisiä käytäntöjä tai muuten kehitystyöstä tulee mahdotonta. Todettiin että on tarpeellista kehittää työkalut eri tilaisen (käytössä, ei käytössä, suunnittelu, jne.) tiedon hallinnointiin. Kokonaiskuva eri hankkeiden välillä on tällä hetkellä sumea, sillä ei ole mitään sateenvarjo-organisaatiota, jonka alle kaikki hankkeet olisi kerätty, vaan ne toimivat omina kokonaisuuksinaan. Tästä tulee kovasti mieleen vanha työni tapahtumatuotannossa, jossa on hyvin samankaltaisia haasteita tiedonkulussa ja kokonaiskuvan hahmottamisessa.

Keskiviikko 30.11.2022

Dailyssä ei tullut esiin mitään työtä hankaloittavia asioita. Päivän toisena kokouksena oli Vikaraati, johon osallistuin päivystäjänä. Tällä viikolla ei ollut löytynyt uusia meidän projektiamme koskevia tuotantobugeja, mutta muista hankkeen projekteista niitä oli löytynyt muutamia. Keskustelu eksyi vikojen lisäksi kehitteillä olevaan massamuutostyökaluun, jolla tullaan tulevaisuudessa hallinnoimaan kohteiden tiloja. Myös tässä palaverissa tuli ilmi se, että projektien välillä ei ole tarpeeksi tiedonvaihtoa. Se tuntuu olevan ongelmana tiimien sisällä, tiimien välissä, projektien välillä, hankkeiden välillä ja toimijoiden välillä – suomeksi sanottuna ongelma on siis kulttuurillinen ja todella syvällä toimintatavoissa, eikä sitä ole helppoa muuttaa. Tietoa on myös aivan järkyttävä määrä – pelkästään meidän hankkeemme Confluensen dokumentaatio on tuhansia sivuja pitkä, keskenään ristiriitainen ja päivittämätön.

CSV-tiketissäni vaihdoin taas lähestymistapaa ja sain parempia tuloksia, kun onnistuin noutamaan tarvitsemani otsikkokentät kannasta jo olemassa olevilla metodeilla. Sisällön noutamisen kanssa oli kamppailua ja huomasin Visual Studio Codesta erinomaisen ominaisuuden, jolla tiedon polun voi kopioida suoraan objektista ja sitä ei tarvitse kirjoittaa käsin. Copy as Expression -toiminto helpottaa kehitystyötä ja käytännössä poistaa objektirakenteen tulkitsemisesta johtuvat virheet kehitystyössä!

Torstai 01.12.2022

Torstai on työpäivänä hyvin palaveripainotteinen, ainakin näin päivystäjänä. Pidän silti tätä käytäntöä todella hyvänä, sillä se lisää nopeasti kokonaiskuvan ymmärrystä ja tietoa tiimin sisällä. Lisäksi päivystäjänä tulen pääsemään asentamaan sovelluksia palvelimelle sprintin loppupäässä ja käyttämään IaC-työkaluja käytännössä. IaC-työkaluja voi kuvailla parhaiten infran ja konfiguraation automatisoimisella manuaalisten prosessien sijaan – niillä voi esimerkiksi varmistaa ympäristön samat asetukset kaikilla kohdekoneilla (Red Hat 2022).

Scrum of Scrums -kokouksessa todettiin, että inkrementin tavoitteet etenevät hyvin ja suunniteltuja ominaisuuksia ollaan todennäköisesti saamassa valmiiksi. Dokumentaation siirtoon vanhan hankkeen Confluencesta uuteen Confluenseen päästään mahdollisesti jo kuluvan sprintin aikana. Keskustelua käytiin myös siitä, että onko inkrementtisuunnittelun fyysinen kohtaaminen oikeasti tarpeellista ja yritettiin miettiä keinoja siihen, että miten siitä saataisiin synergiahyötyä enemmän irti osallistujille. En ollut yllättynyt, että kohtaamiseen suhtauduttiin penseästi pitkän matkan päästä osallistujien parissa ja suopeasti lähempää matkaavien kanssa. Seuraavan version julkaisu vuodenvaihteen ympärillä todettiin yhteisellä konsensuksella haasteelliseksi, kun siinä on tiimeillä ja ylläpidolla paljon poissaoloa ja pyhäpäivien pilkkomaa aikaa. Asiakas kertoi, että meillä on varattuna alustavasti toiseksi seuraavan inkrementtisuunnittelutapaamisen paikaksi Itä-Suomesta tilat, joissa pääsemme tutustumaan fyysisiin kohteisiin, joita hankkeeseen tallennetaan.

Dailyssä käytiin läpi sitä, että uuden massamuutostyökalun käyttöliittymää varten tulee tehdä kunnollinen käyttöliittymäsuunnitelma. Suunnitelmassa tulee huomioida se, että toteutus tehdään prosessin perusteella, eikä siten että työkalu tulee vahingossa muotoilleeksi prosessista hankalamman.

Kehittäjätapaamisessa oli puhetta Moment-kirjaston päivityksestä ja paljastui, että tiimeillä oli hieman erilaisia lähestymistapoja asiaan – toinen tiimi oli poistanut Moment-riippuvuuksia aina sitä myötä kuin se oli käsitellyt sellaisia komponentteja missä se oli käytössä ja me olimme päättäneet poistaa koko kirjaston ja kaikki sen komponenttiriippuvuudet kerralla. Tuotannossa oli havaittu liikaa hälytyksiä ja niiden määrän vähentämiseksi tulee tehdä toimenpiteitä.

CSV:n parissa pääsin hieman eteenpäin ja kenttien avain-arvo-parit tuottivat harmaita hiuksia – täytyi vain todeta, että asia ei tulisi ratkeamaan tänään.

Perjantai 02.12.2022

Dailyssä nostin esiin sen, että olin liittänyt yhteen tikettiin havaitsemani bugin ja toivoin että katselmoija ja testaajat katsoisivat myös sen, jotta nähtäisiin, korjaantuuko kyseisen bugin ongelma tiketissä olevalla korjauksella. Moment-kirjastoon liittyvä erillistikettini päätyi tänään testiin.

Arkkitehtuuripalaverissa esillä olivat päivämääräasiat ja todettiin, että kaikissa hankkeen sovelluksissa on käytettävä standardimallista päiväystä, eli ISO8601-standardia. Standardissa päivämäärä muotoillaan muotoon ”YYYY-MM-DDThh:mm:ssTZD”:

- YYYY = vuosi
- MM = kuukausi
- DD = päivä
- hh = tunti
- mm = minuutti
- ss = sekunti
- TDZ = aikavyöhyke

(Wolf, M., & Wicksteed, C. 1998)

Sovellusohjeiden asennusohjeisiin päätettiin hyödyntää olemassa olevia ja aktiivisesti päivittyviä readme-tiedostoja, sekä IaC-työkalujen konfiguraatiotiedostoja, sillä niiden avulla asennukset tehdään tällä hetkellä joka tapauksessa ja ne ovat siten ajantasaisia.

Digin projekti-infossa hiljattain ostettu yritys esitteli karttateknologiaan perustuvaa sovellustaan, jossa asiakkaat voivat täyttää karttapohjaisia hakemuksia ja lähettää niitä hyväksyttäviksi, sekä

seurata hakemuksien edistymistä. Kiinnostavinta tässä oli se, että puhuja esitteli yksityiskohtaisesti tuotteen kehityskaarta. Tuotekehitys alkoi työryhmätyöllä, jossa ryhmissä oli maksimissaan neljä osallistujaa, jotta saataisiin ideointivaiheessa enemmän ideoita. Tämän jälkeen tehtiin suunnittelua ja siitä opin, että johdonmukaisuus on tärkeää. Suunnittelun avulla yhteiset käytännöt parantavat kehittäjien ja suunnittelijoiden yhteistyötä ja rajoitteet nopeuttavat kehitystyötä. Käytettävyystestauksella validoitiin suunnitteluvaiheen tulokset ja lopulta siirryttiin tekemään itse sovellusta, jossa hyödynnettiin asiakkailta saatua palautetta.

CSV:n kanssa koin voimakasta turhautumista ja pääsin hieman eteenpäin, kun pakotin funktion odottamaan lähtötietoja combineLatest-metodin avulla. Minulla on edelleen hahmotusongelma, sillä nyt uusi rutiinini palauttaa täsmälleen samanlaisen objektin kuin edellinen toteutus, mutta en saa työtä jatkettua siitä eteenpäin. Sovin kokeneemman kehittäjän kanssa palaverin sitten kun hän on seuraavan kerran toimistolla ja hän lupasi auttaa minua ongelmani kanssa. Päivän päätteeksi mergesin Moment-haaraani testistä valmistuneen osatoteukseni ja siinä oli ratkaisemattomia konflikteja ja olin myös unohtanut konkreettisesti poistaa Momentin asennuksen, joten tämä tehtävä saatetaan loppuun sitten maanantaina.

Viikkoanalyysi 2

Toisen viikon tavoitteina oli kehittyä edustan koodin kanssa. Koen päässeeni tavoitteeseeni, sillä ymmärrykseni Angularista, projektista ja RxJS:stä on kasvanut. Samalla on myös tullut selväksi, että matkaa front end-kehityksen ammattilaiseksi on vielä runsaasti. Huomaan myös puutteita Git-osaamisessani, mikä tekee haarojen yhdistämisestä epävarmaa. Minusta tuntuu jotenkin hölmöltä, että kehittäjän tulee käsin määrittää niin paljon edustassa koodin käsittelytavasta – eli sitä että haetaanko joku tieto asynkronisesti. En vajavaiselta kokemukseltani keksi tapausta, jossa haluaisin, että koodi jatkaa eteenpäin ilman että sillä on vielä noudettuna tarvittavia tietoja. Django ei esimerkiksi ohjelmistokehikkona vaadi näiden asioiden määrittelemistä käsin vaan hoitaa ne oman kokemukseni mukaan ns. pellin alla. Kyseessä on asia, joka on seurausta siitä, että JavaScriptiä ajetaan funktio kerrallaan ja tietoa ei ehditä hakea automaattisesti mukaan, kun serveriltä odotetaan vastauksia (Adhikary 2021). Tästä herää väistämättä kysymys, että miksi tätä odotusta ei tarvitse spesifisti sitten määritellä myös serverillä ajettaviin tietokantahakuihin, kun ei tieto silloin ole varsinaisesti keskusmuistissa nopeasti saatavilla?

Tiedän että tulen voittamaan nämä haasteet ja pääsemään kipuiluni yli. En tiedä meneekö siihen viikko vai vuosi, mutta se on varmaa, että tulen siinä onnistumaan. Projektin kokonaiskuva osana hankkeiden yhteenliittymää on tällä viikolla kirkastunut. Olen myös löytänyt lisää yhtäläisyyksiä tapahtumatuotannon ja ohjelmistotuotannon välillä ja ne kaikki liittyvät projektityöhön. Kehittäjä on

eriytetty asiakkaasta ja tuoteomistajatiimi edustaa asiakasta kehittäjätiimille. Kollegani valaisi minua, että tämä nykyinen, ketterän kehityksen, työmalli on kuitenkin huomattavasti lähempänä asiakasta kuin vanhemmat toimintamallit. Nyt palautteesta tulee konkreettisesti asioita kehitystyöhön ja nopean julkaisusyklin myötä asiakkaalle tuotetaan arvoa (Forbes 2016). Jotenkin silti tuntuu siltä, että asiakkaan äänen parempi kuuluvuus olisi hyväksi tällä alalla ja mielellään näkisin kokeilun, jossa tuoteomistajan lisäksi kokouksiin osallistuisi loppukäyttäjä, joka oikeasti käyttää tuotetta arjessaan.

Seuraavan viikon tavoitteenani on kehittyä Gitin ja RxJS:n kanssa, sillä nämä asiat ovat juuri nyt akuutisti työpöydälläni.

1.10 Seurantaviikko 3

Maanantai 05.12.2022

Dailyssä oli hiljaista itsenäisyyspäivän aattona. Työyhteisöstä oli paikalla alle puolet normaalista vahvuudesta ja ymmärtäähän sen, kun yhden arkipäivän vapaalla on mahdollista jatkaa viikonloppu neljäpäiväiseksi. Tiimin työn edessä ei ollut esteitä poistettavaksi, joten kokous oli ohi nopeasti.

Sain tehtyä Moment-kirjaston päivitystikettini loppuun, kun poistin sen käytöstä koko projektista ajamalla ”npm uninstall”-komennon src-kansiossa. Vastasin tähän tikettiin liittyen kollegani katselmointikysymyksiin, jossa hän kysyi aiemmin mainitusta alitehtävästä, että oliko tarkoituksena poistaa tiedostot kokonaan. Vastasin myöntävästi ja kerroin myös, että olin varmistanut asian tech leadilta.

CSV-tiketissä sain läpimurron, kun kollegan kautta ymmärsin, että map- ja mergeMap-funktioilla on eronsa. Parwinderin (2020) mukaan oleellisena erona on se, että map ottaa jokaisen arvon Observablelta, tekee sille operaation ja palauttaa Observablen ja mergeMap voi generoida useita Observableja. Ihan vielä asia ei minulta selkärangasta lähde, mutta nyt kun tiedän että niillä on eroa niin yrityksen ja erehdyksen kautta on mahdollista edetä. Sain haettua serveriltä haluamani objektit ja iteroitua niitä siten, että tulosteena syntyi haluamani kaltainen tiedosto.

Tiistai 06.12.2022

Itsenäisyyspäivä, eli en ollut töissä.

Keskiviikko 07.12.2022

Dailyssä oli pitkä keskustelu sellaisen järjestelmän edustajan kanssa, joka toimii lähtötietona (master data) meidän järjestelmäämme. Keskustelimme massamuutostyökalusta, joka on siis hiljattain saatu valmiiksi, ja sen mahdollisesta käyttöliittymästä. Todettiin että tarvitaan laajempi palaveri ja toinen formaatti kokoukselle, jotta voidaan päättää asioiden toimintalogiikasta.

Keskiviikkona oli myös 3 kk kestäneen perehdytysjaksoni päätöspalaveri esihenkilöni kanssa. Totesimme, että kehitys on oikealla raiteella ja keskusteltiin myös siitä, että kompensaation tarkastelu voidaan pitää koeajan jälkeen, eli puolen vuoden kohdalla. Mitään konkreettisia välitavoitteita ei tähän hetkeen asetettu, joten kehityksen jatkuminen on selkeä tavoite itsessään – täytyy siis jatkaa itsensä haastamista ja vaikeampien kokonaisuuksien haltuunottoa.

Moment-tiketti tuli takaisin katselmoinnista, sillä siinä oli pieniä lintterivirheitä. Lintteri on työkalu yleisimpien muotoiluvirheiden ja esimerkiksi tyypitysten korjaamiseen (Testim 2021). Tämä oli yllätyksellistä, sillä minulla piti olla projektiin kuuluva linter-työkalu käytössäni, mutta asiaan perehdyttyäni huomasin, että toinen niistä puuttui, joten asensin sen. Tiketissä oli myös virheitä durationin laskemisen suhteen, ja se alkoi mystisesti heittämään kaksi päivää pidemmillä aikajaksoilla. Tein siihen korjauksen lisäämällä alkuun manuaalisesti kaksi päivää.

Minulla oli haasteita projektin toisen tietokannan kanssa, koska siihen tarvittavia tunnuksia ei ole Confluence-työkalussa dokumentoituna. Löysin nämä sitten lopulta kollegan vinkin kautta tämän sovelluksen repositorion automaatiotyökalun konfiguraatiotiedostosta.

CSV-tiketin kanssa totesin, että haku kestää kauan ja että siitä löytyy myös muita optimointitarpeita. On aivan turhaa lähettää serverille jokaisesta objektista erikseen pyyntö, kun ne voi yhdistää myös yhdeksi pyynnöksi. Siirsin vanhasta metodista löytyvät toiminnallisuudet uuteen metodiin ja totesin että minun tulee vielä rakentaa tätä varten virheenkäsittelytoiminnallisuus.

Vikaraadissa todettiin, että tuotannosta ei ole löytynyt uusia vikoja. Lisäksi keskustelimme siitä, että seuraavan julkaisun ajankohta on haasteellinen, koska sen kanssa osuvat lomat päällekkäin.

Torstai 08.12.2022

Kehittäjäpalaverissa oli puhetta, että meidän projektimme sovellusten osalta tehtäisiin normaalien haarojen lisäksi myös hotfix-haara, joka lisäisi myöhemmin yhden kriittisen bugin korjauksen. Hotfixilla tarkoitetaan päivitystä, joka ajetaan tuotantokäytössä olevaan järjestelmään sisään (Kidd 2022).

Tein tuotantoon vientiin tarvittavat haarat eri sovelluksista Gitillä ja myös aiemmin mainittu kriittinen bugi saatiin testattua ajoissa, joten myöhäisemmälle hotfix-haaralle ei ollut tarvetta. Käytännössä

se tapahtui sulauttamalla viimeisin develop-haaraan mergetty commit uuteen release-xx.xx-haaraan. Ymmärrykseni mukaan näitä haaroja käytetään sitten automaatiotyökalujen toimesta sovellusten tuotantoasennukseen. Lisäksi kopioin tietokantamigraatioskriptit tuotantoon viennin ohjeistukseen, jotta ne ajetaan prosessin yhteydessä.

Palasin korjaamaan Moment-tikettiäni, sillä tajusin että en ollut huomioinut korjauksessani lyhyen aikavälin tapahtumia. Ongelma paljastui odotettua vaikeammaksi, sillä en voinut olla varma siitä oliko vanha metodi laskenut kestoja oikein. Lähdin ratkaisemaan ongelmaa Excelin NETWORKDAYS-funktion ja manuaalisen tarkistuksen kautta. NETWORKDAYS palauttaa viikonpäivät ja erikseen määritellyt lomakaudet (Microsoft s.a.), joten se oli erinomainen tarkistuskeino tarkoitusperiini.

Scrum of Scrums-kokouksessa keskusteltiin myös nykyisen tuotantoon viennin lisäksi siitä, että seuraava tuotantoon vienti tulisi siirtää. Dailyssä ei tullut esiin mitään huomionarvoista.

Perjantai 09.12.2022

Dailyssä todettiin, että massamuutostyökalun käyttöliittymä tullaan tekemään suoran lähtötietoa tarjoavaan järjestelmään. Todettiin myös, että kolmansien järjestelmien osalta tarvitaan tarkastelua siitä, mitä tapahtuu, kun lähtödata muuttuu.

Moment-tiketissäni päätin tehdä erillisen kontrollirakenteen ja ratkaisun sen pohjalta, onko duraationa yli vai alle vuorokauden mittainen tapahtuma. Ratkaisin tämän tekemällä overloadin yhteen funktioon valinnaisilla parametreilla, jolloin pystyin rakentamaan siihen vaihtoehtoista toiminnallisuutta. Rakensin toiminnallisuuden myös siten, että se vastaa Excelin NETWORKDAYS-funktion tuottamaa dataa. Alkuperäinen metodi ei tuottanut vastaavaa dataa, joten katsotaan, mikä on katselmoijan ja tuoteomistajan mielipide ratkaisustani.

CSV-tiketissäni pääsin eteenpäin ja sain luotua metodin, joka hakee yhdellä pyynnöllä kaikki objektit tuhansien pyyntöjen sijaan. Seuraavaksi haasteeksi jäi sitten näiden pyyntöjen ajaminen erissä, jotta serveri ei herjaisi liian suuresta pyynnöstä. Tulen paneutumaan tähän katsomalla edustan koodista vastaavan kaltaisia toteutuksia ja niitä peilaamalla.

Viikkoanalyysi 3

Tavoitteenani oli kehittyä Gitin ja RxJS:n kanssa ja molemmissa pääsin eteenpäin. Gitissä käytin aktiivisesti stash-parametriä, jolla voin tallentaa paikallisen haaran muutokset paikallisesti ja siten

helposti hyppiä eri haarojen välillä (Git s.a. a). RxJS:n parissa koen myös edistyneeni, sillä sain rakennettua uusia toiminnallisuuksia siihen pisteeseen, että käyttäjän syötteestä muodostettu kysely tuottaa halutun kaltaisen tiedoston, jonka sisältö generoidaan dynaamisesti.

Tämä viikko osoitti jälleen sen, että koodin katselmointi on tärkeässä osassa laadun parantamisessa – olin testannut duraation laskemista, mutta liian lyhyillä ajanjaksoilla, jolloin sain valheellisesti hyväksyttäviä tuloksia, mutta kollegani sai toiset tulokset, kun testattava ajanjakso oli riittävän pitkä.

Vahvistin myös käsitystäni siitä, että dokumentointi on hankalaa ja silti todella tärkeää. Tämä tuntuu olevan universaali ongelma elämässä, sillä hiljaista tietoa tuntuu olevan aina enemmän kuin formaalisti tallennettua. Koodissa on mielestäni tärkeää dokumentoida se mitä koodi tekee ja miksi – osa tästä tapahtuu nimeämällä metodit ja muuttujat hyvin, mutta miksi-kysymyksen vastaus jää usein vaillinaiseksi. Jälkikäteen on todella vaikea arvailla syitä siihen, että miksi joku ratkaisu on tehty ja voisiko sen tehdä jollain toisella tavalla. En pitäisi pahana, jos esimerkiksi määrittelyyn viitattaisiin koodissa aina niissä paikoissa, kun määrittely määrittää asioita.

Turhautumisen tunteet ovat helpottaneet, kun olen saavuttanut pieniä voittoja. On tärkeää pitää myös mielessään se tosiasia, että tilaaja päättää ja määrittelee, ja koodari toteuttaa – sama sääntö pätee muissakin projekteissa, eli sitä tuotetaan mitä tilaaja haluaa, eikä jäädä hieromaan liikaa asioita, joilla ei ole suurta merkitystä (Burke 2022).

Seuraavaa viikkoa varten tavoitteinani on suorittaa onnistunut vahdinvaihto päivystäjänä, oppia hakemaan tietoa serveriltä järkevän kokoisissa paloissa ja oppia lisää Gitistä.

1.11 Seurantaviikko 4

Maanantai 12.12.2022

Tänään käytin Gitin työkaluja runsaasti ja kiinnitin tietoisesti huomiota siihen, mitä olen oikeastaan tekemässä. Käytin esimerkiksi haarojen välillä hyppiessäni edelleen stash-parametriä ja palasin takaisin tiettyyn committiin revert-parametrilla. Git Revert -komento palaa tietyn commitin hash-tunnisteen perusteella tunnistettuun committiin ja sillä voi siten palata takaisin menneisyyteen (Git s.a. b). Luonnollisesti syynä näiden komentojen käyttämiseen olivat tekemäni virheet, mutta sen takia ne ovat olemassa – jotta asioilla on historia, jäljitettävyys ja jotta niihin voidaan palata.

Dailyssä sain lisää ruutuaikaa, sillä scrum master ja tuoteomistaja olivat kumpikin poissa, joten päivystäjänä vedin sen kokouksen. Tuotantoon viennissä oli ilmennyt jonkun verran ongelmia, joita sitten lead developer oli korjannut ja asiat oli saatu järjestettyä.

CSV-tikettini kanssa minulla oli vaikeuksia pyyntöjen erittämisessä, kun en halunnut lähettää serverille tuhansia pyyntöjä. Minulla oli kaksi vaihtoehtoa toiminnallisuuden toteuttamiseksi:

1. Headerien pilkkominen järkevän kokoisiksi kimpaleiksi
2. Uuden rajapinnan kirjoittaminen

Päädyin työskentelemään näistä ensimmäisen parissa ja kysyin apua myös Open AI:n ChatGPT:ltä, jolla on erinomainen kyky luoda ohjelmointirakenteita selkokielisistä lauseista. ChatGPT pystyy esimerkiksi selventämään JavaScriptin toimintoja (OpenAI s.a.). Sain tehtyä vaihtoehtoon 1 perustuvan ratkaisun kollegan avustuksella toimivaksi.

Tiistai 13.12.2022

Aamu alkoi toimitusjohtajan kahvitilaisuudella. Keskustelua käytiin innovoinnista ja yrityksen nykytilanteesta. Yrityksen tasolla Sitowisella on kustannuspaineita, mutta meidän liiketoiminta-alueemme asiat ovat hyvin.

Dailyssä ei ollut ihmeellisiä tapahtumia. Sprinttisuunnittelussa todettiin, että tulevalle sprintille priorisoidaan bugeja sitten kun kaistaa vapautuu meneillään olevilta tehtäviltä. Backlogilla oli runsaasti bugeja ja etsin Jirasta raporttia siitä, että kuinka paljon uusia bugeja kirjataan ja kuinka paljon niitä korjataan. Minulla on mielikuva siitä, että bugeja kirjataan nopeammin kuin niitä korjataan, jolloin tuloksena on luonnollisesti se, että backlogin määrä kasvaa. Tämä on varmaan ihan normaali tilanne kettärän kehityksen ohjelmistossa, jossa se etenee sellaista tahtia, että osa bugeista jää epärelevanteiksi luonnostaan ja niitä ei siten tule korjattua.

Sprinttidemossa tiimillämme ei ollut varsinaisesti demottavaa, eli sprintiltä ei valmistunut sellaisia tehtäviä, jotka näkyisivät käyttäjälle suoraan. Välillisesti ne toki näkyvät, eli luotettavuus ja sovellusten nopeus tulevat kasvamaan sprintillä tehtyjen tehtävien ja bugikorjausten myötä. Sprintillä on tehty töitä esimerkiksi seuraavien asioiden parissa:

- Dokumentaatio
- Koordinaatiston visualisointi
- Robottitestien päivityksiä
- Massamuutostyökalua varten rakennettu rajapinta
- Refaktoroitu koodia
- Tehty ikoneita
- Korjattu UI-bugeja
- Korjattu integraatioita
- Korjattu käännöksien toiminnallisuutta

CSV-tiketin kanssa tein erien koon optimointia, mutta päädyin lopulta käyttämään metodina mahdollisimman pientä pyyntöjen määrää serverille, vaikka lokaalisti havaitsin kaikista nopeimmaksi vaihtoehdoksi 20–50 pyyntöä kaikkien objektien noutamiseksi. Netin yli haettaessa tilanne olisi tietenkin erilainen, sillä silloin serverin ja käyttöliittymän suorituskykyviiveiden lisäksi tulisi muuttujaksi verkon latenssi. Paloittelin toiminnallisuutta pienempiin funktioihin ja tein tyypitystä varten interfaceja, joiden tarkoituksena on kertoa minkä tyyppistä tietoa metodeissa liikkuu (Typescriptlang 2022).

Keskiviikko 14.12.2022

Tein Gitissä uudet haarat seuraavaa versiota varten, sekä lisäsin kehityshaaroihin seuraavat versionumerot. Tämän toiminnan tarkoituksena oli se, että tämän jälkeen automatiikka huomaa tarkistuskierroksellaan, että tietyn nimeämiskäytännön mukaisesti on tullut uusi versio ja sovelluksista asennetaan automaattisesti palvelimille eri versioita yksinkertaisen logiikan perusteella: kehittäjien ja testaajien käytössä on serverillä pyörimässä uusinta uutta, hyväksymistestauksessa on tulevan julkaisun tavara ja tuotannossa & koulutusympäristöissä pyörii loppukäyttäjillä edellinen julkaistu versio.

Dailyssä kysyin tuoteomistajalta Moment-tikettiin kehittämäni poikkeavaan duraation keston muotoiluun kommenttia, ja hän ei osannut asiaa siltä istumalta kommentoida. Päivystysvuoro vaihtui onnistuneesti, kerroin seuraavalle päivystäjälle kaikki keskeneräiset asiat ja hän otti soihdun kan-

taakseen. Päivitin vielä viimeisenä työnäni päivystäjänä Confuenceen tuoreimmat sovellusten versionumerot. Tätä varten jouduin perehtymään Linuxin komentoihin, tarkemmin ottaen Findin RegEx-merkintätapaan, jonka avulla onnistuin etsimään kaikista alikansioista tietyn nimistä kohdetta komennolla: ”find . -name ”nimi*” (Stackoverflow 2011). Palautin tiketin katselmointiin päivän päätteeksi.

Torstai 15.12.2022

Ylläpidosta tuli kysymys siitä, että mikä haara heidän tulisi asentaa erääseen ympäristöön. En tiennyt asiasta sen enempää kuin hekään, mutta selvitin sitä ”git diff”-komennolla, joka kertoo kahden haaran välisistä eroista (Git s.a. c). Haaroilla ei ollut muuta eroa kuin nimi, joten kehotin asentamaan sen, jossa oli informatiivisempi nimi.

Dailyssä oli puhetta määrittelyistä ja CSV-tikettini siirtyi holdiin, sillä siihen tarvitaan asiakkaalta kommentointia toivottujen kenttien suhteen. Tuoteomistaja lupasi selvittää asiaa. Toteutus on kuitenkin nyt korjauksia vailla valmis ja sain kollegalta positiivista palautetta koodini laadusta.

Moment-tiketissä päätin yrittää toteutusta edellisen kirjaston toiminnallisuutta matkien. Minusta tuntuu silti siltä, että parempi lopputulos käyttäjän kannalta olisi tulostaa duraatioksi yksiselitteinen päivien määrä ja antaa käyttäjän muotoilla jatkokäyttötarpeissaan tietoa miten parhaaksi katsovatkaan. On tietysti mahdollista, että toiminnallisuutta varten on rakennettu loppukäyttäjille joku todella monimutkainen Excel-funktio, joka osaa ottaa syötteen muotovaihtelut huomioon ja tuottaa siten lajiteltavaa tietoa käyttäjille.

Perjantai 16.12.2022

Käytin koko päivän yrittäen selvittää miten saisin Date-fns:n ja Momentin olemaan yhtä mieltä siitä, että kuinka monta päivää ja kuukautta kussakin duraatiossa on. Tehtävä ei ollut helppo, sillä Momentilla on tapana olla sitä mieltä, että välillä duraatio on muotoa: ”2 kuukautta 30 päivää” ja välillä ”3 kuukautta 0 päivää”. Tämä tarkoittaa aikamoista testirupeamaa, sillä en löytänyt logiikkaa tai ennalta-arvattavuutta siihen, että milloin tulostetaan ”n kuukautta 30 päivää” tai ”n+1 kuukautta, 0 päivää”. Palautin tiketin kommentoitavaksi ja sain siihen palautetta, jossa kommentoitiin yksikkötestien epäonnistumista. Yritin tämän jälkeen saada tulosteita muutettua samankaltaisiksi, mutta aina jostain löytyi uusi reunatapaus, joka sotki suunnitelmani.

Päädyin korjaamaan yhden yksikkötestin uuteen uskoon, sillä se vaati sellaista tulostetta, joka oli kieliopillisesti väärin. Kerroin tästä toki myös muille, mutta mielestäni on turha vaatia ohjelmaa tekemään yksiselitteistä virhettä.

Viikkoanalyysi 4

Tällä viikolla työskentelin hieman vielä päivystyksen parissa ja sprintin vaihtuminen jännitti kovasti, kun en ollut koskaan aiemmin tehnyt ympäristöjä varten erillisiä haaroja. Paljastui kuitenkin, että se oli yhtä yksinkertainen toimenpide kuin mitä on haarojen tekeminen kaikkiin muihinkin tarkoituksiin. Haarat ovat äärimmäisen käteviä, kun ne tarjoavat jäljitettävyyttä ja tuottavat monien kehittäjien työstä koherentin kokonaisuuden. Koin onnistumisen tunteita, kun kommentoin ylläpidolle asioita ja toimiessani scrum masterin sijaisena Dailyssä.

Kohtasin haasteita määrittelyjen muodossa, sillä CSV-tikettiäni ja Moment-tikettiäni yhdistää se, että kummastakaan ei ole olemassa varsinaista määrittelyasiakirjaa ja kehittäjät ovat ymmärrettävän arkoja muuttamaan asioita, jotka ovat tuotannossa käytössä. On mahdotonta tietää miten käyttäjä käyttää tuotetta ja mikä on käyttäjälle tärkeää, jos sitä ei käyttäjältä ensin selvitä. Vaatimusten määrittäminen on vaikeaa ja ratkaistavan ongelman tarkka määrittäminen vaatii asiantuntemusta (Paakki, 2011, s. 6–8).

Koin ChatGPT:n hyödylliseksi lisätyökaluksi, joka nopeuttaa työntekoa. Siitä on todella paljon apua, kun voi kysyä yksinkertaisia syntaksiin tai logiikkaan liittyviä kysymyksiä, joihin saa välittömästi toimivan ratkaisun tai ainakin sen esiasteen. Ohjelmointi on mielestäni helppoa siinä vaiheessa, kun osaa muotoilla asiansa oikein ja kysyä oikeita kysymyksiä, joten tekoälyn käyttäminen ei sellaisenaan ole edes ”huijausta” – siinähän vain käytetään konetta apuna omien ajatusten tulkitsemiseksi koneelle ymmärrettävään muotoon, mitä ohjelmointi muutenkin on.

Löysin erilaisia muutoksia testaillessani Visual Studio Codesta äärimmäisen kätevän toiminnallisuuden (ctrl + shift + L), jolla on mahdollista valita kaikki samannimiset muuttujat kerralla ja kirjoittaa ne uusiksi (Stackoverflow 2017).

Sprinttidemossa minua mietitytti asiakkaan näkökulma asioihin, sillä kehitystyö ei ole halpaa ja se voi varmasti tuntua helposti siltä, että mikään ei edisty, jos käyttäjälle ei näy konkreettisia muutoksia käyttöliittymässä. Scrum masterina saattaisin pyrkiä ohjaamaan tiimiä siihen, että jokaisessa sprintissä valmistuisi aina myös jotain sellaista, mikä näkyisi loppukäyttäjälle, jotta käyttäjille tulisi myös selkeämpi kuva siitä, että ohjelmistoa kehitetään aktiivisesti ja se muuttuu jatkuvasti paremmaksi.

Seuraavan viikon tavoitteenani on keskittyä serveriohjelmointiin ja tietokantoihin.

1.12 Seurantaviikko 5

Maanantai 19.12.2022

Moment-päivitystiketissäni laskin manuaalisesti auki millisekuntitasolla sen tuloksen, joka oli toivottava lopputulos. Se erosi kuitenkin Moment-kirjaston tuottamasta datasta. Laitoin tiketin takaisin katselmointiin, kun korjasin siinä olleet puutteet negatiivisten duraatioiden laskemisessa. En halunnut kovakoodata ratkaisuun päivämäärien korjauksia enempää, sillä niitä oli nyt jo muutama ja pelkäsin tulevien päivitysten sitten rikkovan ne. En myöskään voinut olla varma siitä, kykenisin löytämään rajattomasta päivämäärien avaruudesta jokaisen eroavaisuuden Momentin ja Date-fns:n välillä, jotta voisin tehdä niihin käsikorjaukset.

Rajapintatiketissä palasin kehittämään vaadittuja muutoksia. Tiketin perimmäisenä tarkoituksena on mahdollistaa toisessa sovelluksessa rajapinnan kautta tapahtuva metatiedon muokkaaminen. Tästä päivästä iso osa meni rajapinnan toiminnan hahmottamiseen ja sen tekemien tietokantamuutosten ymmärtämiseen. Rajapinnan kautta tehtävät muutokset tulee suhteuttaa siihen, mikä on tietokannan eheyden kannalta mahdollista ja mitä tiketissä on pyydetty.

Tiistai 20.12.2022

Dailyssä ei tullut erityisiä asioita eteen – tiimin työskentely sujuu hyvin ja kaikki pääsivät eteenpäin tehtäviensä parissa.

Uutena tehtävänä aloittamani rajapintatiketti eteni osaltani hyvin, sillä löysin täsmälliset paikat koodissa, joissa pääsin muokkaamaan tarvittavia tietoja. Aloitin muokkaamalla yksinkertaisinta mahdollista tietoa, eli yhtä boolean-tyyppistä arvoa. Ensin onnistuin vaihtamaan sen yhden kerran, ja sen jälkeen korjasin koodia siten, että vaihtaminen on mahdollista tehdä niin monta kertaa kuin käyttäjä haluaa. Lopulta sain kaikkia metatietokenttiä muutettua tietokantaan. Aloitin tämän jälkeen validoinnin rakentamisen käyttäjän tuottamille syötteille. Käytin pitkän aikaa miettiessäni sitä, että miksi rajapinta ei reagoi muutoksiin, kunnes huomasin, että olin korjannut väärää paikkaa. Tämä tarjosi oppimismahdollisuuden siitä totuudesta, että yksinkertaisin selitys on hyvä aloituspaikka selvitystyölle – mikä on merkityksellisesti sama kuin Occamin partaveitsi (Duignan 2022).

Opin myös tietoja käsitellessäni, että Javassa null-arvoa ja falsea voidaan käsitellä kahdella eri tavalla, kun kyseessä on primitiivityypin boolean-tietotyyppi tai Boolean-tietotyyppi. Boolean voi olla arvoiltaan ”true, false tai null” ja boolean puolestaan ”true tai false” (Stackoverflow 2009). Null muuttuu siis false-arvoksi, jos boolean-tyyppiseen muuttujaan syöttää arvoksi ”null”.

Kyselytunnilla oli puhetta Swaggerin erilaisilla tavoilla generoiduista rajapinnoista ja niiden dokumentoinnista. Asiakkaan VPN:stä oli myös keskustelua, sillä minulla ja toisella kehittäjällä ei ole vielä pääsyä suoraan asiakkaan servereille, jotta pääsisimme tutustumaan tarkemmin Jenkinsiin ja Ansibleen, sekä kehittämään niihin liittyvää toiminnallisuutta projektissa.

Keskiviikko 21.12.2022

Dailyssä oli vajaa osallistujamäärä, sillä tuoteomistaja ja scrum master olivat molemmat muissa kokouksissa. Kävimme tehtävien tilanteet läpi ja palasimme päivän toimintojen pariin.

Moment-tiketistä tuli katselmoinnista kommenttia, että muokkaa testiä uuden kirjaston mukaiseksi. Olin ilahtunut kommentista, sillä olin itse asiasta samaa mieltä – on turha lähteä käsin korjaamaan sellaista, mistä ei ole mahdollista saada kaikkea mahdollista tietoa käsiinsä.

Rajapintatiketissä sain tehtyä validoinnin siten, että validoitavat kohteet haetaan tietokannasta ja syötettä verrataan niihin. Tein muutoksia poikkeuksiin, koska halusin muuttuneen rajapinnan myötä muuttaa myös sen tarjoamia poikkeuksia, koska entiset eivät enää kuvanneet tilannetta tarkasti. Tämä aiheutti bugeja muualle järjestelmään, sillä testit hajosivat, kun ne luonnollisesti olettivat saavansa entisen rajapintaratkaisun mukaisia poikkeuksia käsittelyynsä, mutta ne saivatkin täysin uudenlaisia syötteitä.

Torstai 22.12.2022

Tänään opin uutta Gitistä, sillä kirjoitin vahingossa vääränlaisen commit-viestin committiini. Commit-viestin korjaus onnistui kätevästi komennolla: ”git commit –amend -m ”viesti”. Tällä komennolla on mahdollista muuttaa commit-viestiä ennen kuin se on pushattu repositorioon (Atlassian s.a.).

Rajapintatiketissä korjasin testien odotettua tulostetta vastaamaan sitä, mitä rajapinta tuotti. Minulla oli suurta kipuilua Timestamp-tietotyyppien validoinnin kanssa. Syyksi paljastui se, että projektissa oli käytössä validaattori, joka väitti validoivansa aikaleimojen validointisääntöjä, mutta todellisuudessa se validoikin aikaleimoja itsessään.

Dailyssä käytiin läpi dokumentaation siirtoa uuteen työtilaan, todettiin että Jenkins-palvelimelle ei ole kaikilla edelleenkään pääsyä ja VPN on siihen syynä. Yhdessä tiketissä oli tehty mielenkiintoista optimointia Postgre-tietokantaan, jossa karttageometrian hakufunktiota oli optimoitu käyttämällä yksinkertaista &&-operaattoria, eli ehtolausetta ”ja”. Käyttöliittymään on tulossa suurempi refaktorointityö linkkausten osalta, joka alkaa suunnittelun myötä ensi vuonna.

Muokkasin Moment-tikettiäni katselmointikommenttien mukaiseksi ja palautin sen katselmointiin.

Kehittäjäpalaverissa oli keskustelua sovelluksien karttakuvien lähteistä ja todettiin, että ui-common-kirjastoon, jota siis eri sovellukset käyttävät kartan osalta, olisi hyvä saada layers.json-tiedosto, jonka avulla voitaisiin määrittää karttatasot erikseen kehitysympäristöissä. Tämän muutoksen takana on se, että asiakkaan sisäverkossa olevat kartat eivät luonnollisesti lataudu, jos sovellus ei

pääse käsiksi asiakkaan sisäverkkoihin. Keskustelimme Proof of Consept -tiketistä, jossa buildattaisiin eri ympäristöihin sovellukset yhdestä artifaktista. Tämä uusi toimintatapa ja sen myötä luodut konfiguraatiotiedostot olisi syytä saattaa versiohallinnan piiriin, jotta niitä on helppo hallinnoida.

Perjantai 23.12.2022

Dailyssä oli puhetta siitä, että jatkossa on syytä siirtää testausta enemmän taustamekanismien testaukseen kuin yksittäisten käyttöliittymäasioiden testaamiseen, sillä käyttöliittymät muuttuvat jatkuvasti ja taustalla olevan mekanismin testaaminen koetaan asiakkaan tiimoilta tärkeämmäksi.

Rajapintatiketissä sain korjattua aikaleimojen validointia ja testaamista siten, että tein erillisen kontrollirakenteen molempia käyttötarkoituksia varten, eli sitä kun testataan validaatiosääntöä tai itse aikaleimaa. Seuraavaksi korjasin testiä, joka toimi väärällä olettamalla. Testi oletti, että sen saamassa objektissa ei olisi kenttä x muuttunut, mutta rajapintaamuutoksen jälkeen kenttää x pystyi muokkaamaan, joten testi epäonnistui. Korjasin vielä aikaleimojen validointia toimivammaksi ja ajoin kaikki projektin testit läpi.

Tulin myös kehittäneeksi vikoja etsiessäni uuden toimintatavan vastaisuuden varalla Occamin partaveitsen innoittamana:

1. Aloita vian selvitys debug-tilassa
2. Selvitä onko vika syötteessä
3. Selvitä onko vika siinä, mitä syötteelle tehdään
4. Selvitä onko vika siinä, mitä tuloksia odotetaan
5. Selvitä onko odotettu tulos relevantti testattavan asian kannalta vai tuleeko sitä muuttaa

Viikkoanalyysi

Näin taaksepäin katsoessani uusi toimintatapani vaikuttaa järkevältä – ehkä siitä täytyy vielä siirtää viimeinen kohta ensimmäiseksi, sillä paras muutos on sellainen, jossa muutosta, uutta osaa tai toiminnallisuutta ei tarvitse tehdä, kuten Elon Musk on sanonut (Neut, M. 2021) . Hyvin suuri osa ongelmistani johtuu siitä, että pyrin selittämään asioita liian monimutkaisesti ja etsimään vikaa sellaisesta paikasta, josta se ei voisi edes löytyä. Täytyy siis pitää kirkkaana mielessä ohjenuorana se, että vain on syytä tarkistaa aina ensin, onko muutoksessa järkeä ja sitten varmistaa muokkaavansa oikeaa kohtaa koodissa.

Onnistuin pääsemään eteenpäin ohjelmointitaitojeni kehittämisessä ja ymmärtämään enemmän tietokannoista. Tietokantojen eheyttä koskeva havaintoni oli merkityksellinen, sillä päättelin tiketin tehneen tahon tehneen virheen tehtävän määrittelyssä ja pyytäneen mahdotonta – tai no, tietokoneiden maailmassa kaikki on lähtökohtaisesti mahdollista, eli olisi varmasti mahdollista rakentaa

ratkaisu, jossa kenttien tietotyyppiä muutetaan käyttäjän toimesta, mutta silloin myös jo kannassa oleva dataan tulisi suorittaa migraatio ja korjata se vastaamaan uutta tietotyypppiä. Tietotyypin vaihtaminen ei tietenkään tulisi kyseeseen, jos tietokannassa olisi tallennettuna tietoa muotoon desimaaliluku ja se tulisi muuttaa, vaikka muotoon aikaleima. Olisi myös työlästä kirjoittaa kaikki datatyyppimuunnokset huomioiva muutostyökalu.

Olen myös tyytyväinen tekemään ratkaisuun Moment-kirjaston vaihdossa, jossa pyritään korjaamaan asioita mahdollisimman vähän kovakoodauksen kautta, sillä kovakoodaus aiheuttaa muutostarvetta tulevaisuuteen (SImmons 2018)

Seuraavan viikon tavoitteenani on ymmärtää enemmän Javan testeistä ja parantaa ymmärrystäni TypeScriptistä.

1.13 Seurantaviikko 6

Maanantai 26.12.2022

Tämä oli arkipyhäpäivä, joten en ollut töissä.

Tiistai 27.12.2022

Olen ollut lähiaikoina sen suhteellisen paljon etätöissä ja siten sammuttanut koneeni usein. Tämä on johtanut siihen, että kehitysympäristön pystyttämisestä on tullut vaivalloista, joten päädyin kehittämään itselleni sopivat skriptit Dockerin käynnistämiseen ja konttien pystyttämiseen, sekä projektin kahden eri backendin ajamiseksi terminaaleissa. Näistä tuli pienellä hiomisella kelpo työkalut ja seuraavaksi kehitysaskeleeksi jäi vielä mahdollisuus suorittaa esimerkiksi Dockerin pystytysskripti samalla kun avaa Windows Subsystem for Linuxin. Toistaiseksi olen tyytyväinen elämääni näiden skriptien myötä ja tämä askel jää tulevaisuuteen.

Dailyssä mainitsin, että rajapintatikettini on valmis katselmoitavaksi ja otin itse vastaavasti yhden tiketin katselmointiin.

Katselmoitava tiketti koski projektissa käytössä olevaa ui-common-koodikirjastoa, josta löytyy monien hankkeen sovellusten käyttämiä yhteisiä komponentteja. Kyseessä oli hyvin yksinkertainen bugikirjaus, jossa pudotusvalikon vaihtoehdot eivät olleet aakkosjärjestyksessä ja tiketin ratkaisu korjasi tämän luomalla uuden lajittelutavan vaihtoehdoille. Katselmoinnissa vaikeutta aiheutti se, että ui-common on projektissa mukana Gitin Subtreen kautta ja kollegat eivät osanneet auttaa sen suhteen, että miten pääsisin käsiksi vain tähän yhteen haaraan ja siten tarkastelemaan muutosta.

Toteutin katselmoinnin tekemällä koodimuutokset paikalliseen haaraani käsin ja päätin palata aiheeseen sitten kun seuraavan kerran katselmoin jotain subtree-rakenteen alaista muutosta, missä tehdään suurempia muutoksia koodiin ja käsikopiointi on liian vaivalloista.

Rajapintatikettini tuli takaisin katselmoinnista ja sitä koskevat muutospyynnöt koskivat Boolean-tietotyyppisiä rakenteita, jotka olivat koodissa jo ennestään. Tein niihin pyydetyt muutokset ja muutin samalla muutamien metodien nimiä kuvaavimmiksi, sekä poistin asioita, jotka jäivät tarpeettomiksi tietotyypin muututtua Booleanista booleaniksi. Moment-tikettini eteni katselmoinnista testiin.

Gitissä totesin IDE:n tarjoaman Git Diff-työkalun olevan melkoisen raskassoutuinen, ja aloin käyttämään git diff-komentoa suoraan komentoriviltä. Komennolla ”git diff tiedosto” on mahdollista vertailla eroja tiedosto kerrallaan (Git s.a. a).

Keskiviikko 28.12.2022

Tein katselmointia dokumentaatioon ja totesin että siellä on todella paljon sellaista tietoa, jonka ylläpitäminen on manuaalisesti työlästä. Dokumentaatiosta löytyy esimerkiksi luokkakuvauksia, rajapintakuvauksia ja json-esimerkkimerkkijonona, jotka täytyy muistaa käydä muuttamassa aina kun rajapintoihin tehdään muutoksia. Toisaalta tämä on jo vanha ja kypsä projekti monilta osiltaan, joten rajapintoihin tehdään varsin maltillisesti muutoksia, ja ne ovat aina taaksepäin yhteensopivia muutoksia, niin siten voidaan sanoa, että tehtävä on hallittavissa.

Korjasin CSV-tikettiäni kommenttien mukaan sillä välin kuin odotamme määrittelyä – tein korjauksia vain sellaisiin kohtiin, jotka eivät ole riippuvaisia määrittelyn tuloksesta, jotta en myöhemmin olisi tilanteessa, jossa päätyisin muokkaamaan samaa koodia toiseen kertaan.

Otin katselmointiin käyttöliittymätiketin, jossa korjataan haun tuloksia oikeiksi. Pyysin koodin toteuttajaa tekemään samalla toiseen tikettiin liittyvän pienen kosmeettisen korjauksen, joka oli minun kirjaamani bugi. Tuoteomistaja oli aikoinaan todennut kirjaamastani bugista, että korjataan se sitten samalla kun tulee jotain muuta korjattavaa kyseiseen templaattiin ja tämä oli sopiva paikka toteuttaa korjaus.

Tämän jälkeen tein katselmointia serveripuolen tikettiin ja ihmettelin hetken, että miksi muutoksia ei tapahdu develop-haaran ja tiketin haaran välillä. Muutoksia ei tietenkään tapahtunut, kun muutoksena oli tietokantamigraatio ja normaali käynnistysrutiini ei aja tietokantamigraatioita, joten olin taas tilanteessa, jossa en etsinyt vastausta yksinkertaisimmasta mahdollisesta virheestä, eli siitä että suoritetaanko muokkaamaani koodia? Sain tiketin katselmoitua, kun virheeni selvisi ja laitoin sen testiin.

Loppupäivä meni kehitysympäristön debuggaamisen parissa, sillä oletin (virheellisesti), että tietokantamigraatiot olisivat saaneet sen solmuun. Pitkällisen selvitystyön päätteeksi paljastui, että minulla oli väärät käynnistysasetukset toisen backendin kanssa ja siksi se ei toiminut. Karttakomponentti jäi edelleen puuttumaan sovelluksesta ja en saanut selvittyä syytä siihen.

Torstai 29.12.2022

Kehitysympäristöni kartta koki ihmeparantumisen Windowsin uudelleenkäynnistymisen myötä. Rikoin jälleen ensimmäistä sääntöäni, eli sitä että tarkista yksinkertaisin vika ensin. On yleisesti tiedetty fakta, että uudelleenkäynnistäminen ratkaisee suurimman osan selittämättömistä vioista (Cox 2018).

Aloitin uuden tiketin parissa, joka ensin alkuun vaikutti olevan käyttöliittymän hakutoiminnon suodatusongelma, mutta lähemmässä tarkastelussa se paljastui datavirheeksi tietokannassa. Käyttöliittymä palautti haussa objekteja, joilla piti olla sijainti sijaintiluokassa x, mutta fyysisesti kartalta katsottaessa objektit sijaitsivat selvästi sijaintiluokan x ulkopuolella. Käyttöliittymän kautta on estetty sijainnin määrittäminen siten, että sijaintiluokka x on mahdotonta sijoittaa kartalle muualle kuin sijantiluokka x:n alaiselle alueelle. Tuotantotestauksen tietokannasta paljastui datavirheitä, joissa oikeasti sijaintiluokka y:ssä sijaitsevien kohteiden sijaintiluokka oli säädetty käsin tietokannasta tai aikaisemmalla sovellusversiolla (jossa siirtäminen on ehkä ollut mahdollista) sijaintiluokkaan x. Onnistuin myös toistamaan ongelman lokaalisti säätämällä sijaintiluokaksi objekteille sijaintiluokan x, vaikka ne oikeasti koordinaattiensa puolesta kuuluivatkin sijaintiluokkaan y. Tech lead oli myös sitä mieltä, että virhe on datassa, ei ohjelman toiminnallisuudessa. Voisin myös olla sitä mieltä, että tietokannassa tulisi olla rajoitus, joka estäisi sijaintiluokan määrittämisen kulloiseenkin sijaintiluokkaan, jos objektin sijaintitiedot eivät täyttäisi sijaintiluokan reunaehtoja.

Seuraavana tikettinä oli käännöstiketti, jossa korjasin serverin virheilmoituksen näyttämän ilmoituksen. Ilmoituksen tiedot näytettiin suomeksi ja englanniksi, vaikka asiakas halusi virheet pelkästään suomeksi, joten poistin englanninkielisen version, vaikka se tarjosi hieman enemmän informaatiota käyttäjälle.

Torstain tekniikkapalaverissa oli puhetta kaikkien sovellusten buildaamisesta yhdestä artifaktista. Pidemmän ajan tavoitteena on selkeä CI-pipeline, eli jatkuvan toimittamisen toimintamalli. Uusi meidän sovellukseemme kehitetty ratkaisu voidaan ehkä miltei sellaisenaan siirtää myös hankkeen muihin sovelluksiin. Ansiblen playbookit, eli konfiguraatiot, tulee saattaa versionhallinnan alaisiksi ja tätä varten on asiakkaalta saatava repositorio Gittiin. Renovate Bot:ia suositeltiin otettavaksi käyttöön meidän sovelluksiimme, sillä siitä on saatu hyviä kokemuksia hankkeen muissa sovelluksissa.

Huomasin backlogilta töitä etsiessäni, että yhden tiketin ongelma korjaantuu develop-haaraan mergetyssä tiketissä, joten tein tarvittavat hallinnolliset liikkeet ja merkkasin bugin korjatuksi siten, että se korjaantuu toisessa tiketissä.

Perjantai 30.12.2022

Dailyssä oli keskustelua obsoleteksi kirjaamastani bugista, jossa siis todettiin, että kyseessä on datavirhe. Testaajat ja tuoteomistaja nostivat aiheellisesti huolta siitä, että virheellinen data tulisi poistaa järjestelmistä, sillä se vaikeuttaa testaamista ja aiheuttaa hämmennystä – kuten oli käynyt juuri tässä tiketin bugissa, joka oli alun perin löytynyt hyväksymistestauksessa, kun testattiin aivan muita asioita. Totesin laittavani tästä tietoa ylläpidolle ja myöhemmin en enää pystynyt toistamaan ongelmaa siellä ympäristössä, missä se oli alun perin havaittu. Tech lead oli tänään poissa, joten en pääse varmistamaan häneltä asiaa ennen kuin hän palaa töihin ja kirjoitin tikettiin hänelle kysymyksen, jossa kysyn, että onko hän jo tehnyt migraatioita virheellisen datan poistamiseksi tai ajanut ympäristöön uuden tietokantadumpin tuotannosta.

Aloitin työt uuden käyttöliittymätiketin parissa. Paljastui kuitenkin, että kyseessä on serverikoodia koskeva virhe. Tiketissä on kyse aiheettomasta virheilmoituksesta, jossa objekti perii sijainnin yläkohteelta ja sitä ei voi muokata, sillä kohteen yksi arvo on liian pieni, vaikka tähän muokattavaan alakohteeseen ei kosketa tuon arvon kohdalta. Edustan koodissa ei tapahdu mitään tähän liittyviä tarkistustoimenpiteitä, joten virheen on siten oltava jossain muualla.

Minulla oli taas ongelmia kehitysympäristöni kanssa ja en saanut toista hankkeen backendeistä toimimaan IDE:ssä mitenkään. Otin konsultointisession kokeneemman ohjelmistokehittäjän kanssa ja emme yhdessäkään saaneet ongelmia ratkaistua. Pääsin kuitenkin hyvin vauhtiin tiketin ongelman ratkaisun kanssa.

Viikkoanalyysi 6

Tämän viikon lähtötavoitteena oli kehittää ymmärrystä käyttöliittymän koodista ja ymmärtää enemmän Javan testeistä. Opin ymmärtämään molempia paremmin, joten tavoite tuli sinänsä täytettyä.

Huomaan toistavani edelleen samaa perusvirhettä, jossa en aloita vianselvitystä yksinkertaisimmasta mahdollisesta selityksestä, vaan hukkaan aikaa ja resursseja tutkimalla ensin paljon monimutkaisempia selityksiä. Occamin partaveitsi on kuitenkin riippuvainen kontekstista ja täytyy ymmärtää se, että yksinkertaistaminen on tarpeellista, mutta sitä tulee tehdä vain siihen pisteeseen saakka kuin siitä on hyötyä (Muguda 2021). Ohjelmistokehityksen keskeisenä periaatteena on se, että abstraktio säilyttää tiedosta vain oleellisen osan (Monteiro 2022). Mielestäni tästä voi sanoa, että vikaa etsiessään on hyvä toimia siten, että keskittyy oleelliseen ja kokeilee helpointa ratkaisua

ensin. Abstraktion myötä on tosin välillä hieman hankalaa hahmottaa sitä, että mikä on se tosiasiallinen ongelma, jota yrittää ratkaista. Esimerkiksi vaikeasta selvitystyöstä kuluneelta viikolta tulee mieleen ympäristöni kartan hajoaminen: pystytin karttakomponentin aivan täsmälleen samalla tavalla kuin aina ennenkin ja silti se ei toiminut, joten metsästin virhettä tekemisestäni, enkä fyysisessä laitteesta (eli ympäristöstä, jossa sovellusta ajetaan).

Seuraavan viikon tavoitteenani on oppia enemmän ohjelmistokehityksen hallinnollisesta puolesta ja keskittyä ratkaisemaan eteeni tulevia haasteita siten, että mietin ensin erilaiset vaihtoehdot vian lähteeksi valmiiksi ja vasta sitten alan ratkaisemaan vikaa.

1.14 Seurantaviikko 7

Maanantai 2.1.2023

Jatkoin työtä tikettini parissa, josta oli viime viikolla löytynyt serveripuolen virhe.

Dailyssä tech lead sanoi, että hän ei ollut poistanut kannasta dataa, mutta epäili, että ylläpito on saattanut ajaa testipalvelimelle uuden tietokantadumpin. Myöhemmin paljastui, että olin itse katsonut väärää ympäristöä tehdessäni havainnon. Yksinkertaisten virheiden tekeminen jatkuu siis edelleen. Laitoin ylläpidolle viestiä ja sovin heidän kanssaan, että he ajavat hyväksymistestauksen valmistuttua uuden tietokantadumpin serverille ja siten selviää asioista taas enemmän.

Otin yhden tietokantamigraatioita koskevan tiketin katselmointiin – minulla oli hieman haasteita saada muutoksia näkyviin, mutta kyseessä oli taas perusvirhe, eli se että en ollut ajanut Mavenin ”clean install”-sykliä läpi haarassa, jolloin migraatiot jäävät tekemättä. Tämän tehtyäni pääsin katselmoimaan muutoksia. Tiketissä poistettiin turhia metatietokategorioita ja orvoksi jääneitä käännöksiä – tiketti teki sen mitä lupasikin ja en löytänyt koodista huomautettavaa, joten laitoin sen eteenpäin testaukseen.

Inkrementtidemoa varten etsin Jirasta kätevintä tapaa katsoa valmistuneita ominaisuuksia ja demottavia bugeja. Päädyin katsomaan Reports-näkymästä ”Sprint report”-raportteja ja sieltä löytyi sprinttikohtaisesti valmistuneet tehtävät. Minulla oli demottavana ensimmäisessä sprintissä valmistunut CSV-tikettini, sekä siihen liittyvät muutaman bugikorjaukset. Autoin myös uutta kollegaani hahmottamaan sitä, missä menee demottavien asioiden raja – demotaan siis pääsääntöisesti uusia, käyttöliittymässä näkyviä ominaisuuksia, mutta myös jotain käyttökokemuksen kannalta merkittäviä bugikorjauksia voi demota.

Tiistai 3.1.2023

Aamupalaverissa kävimme läpi tulevan inkrementtisuunnittelun ominaisuuslistaa. Tämän toiminnan kantavana ajatuksena on se, että ominaisuudet eivät tule kehitystiimille yllätyksenä inkrementtisuunnittelussa ja mahdollisiin epäkohtiin tai toteutuksen haasteisiin päästäisiin kiinni mahdollisimman aikaisessa vaiheessa. Seuraavalle inkrementille on tulossa paljon omaisuuslajimuutoksia, jotka tarkoittavat päivityksiä metamalleihin ja backendin koodiin. Näitä metamallimuutoksia tehdään toistaiseksi käsin, mutta tulevaisuudessa on tahtotilana se, että käyttäjä voi tehdä niitä käyttöliittymän kautta. Se on todella isotöinen lisäomaisuus, kun silloin täytyy tehdä mekanismit jo olemassa olevan datan konvertoimiseksi käyttäjän valitsemalla tavalla. Todettiin että testaajilla on tällä hetkellä kädet täynnä töitä.

Dailyssä scrum master painotti meneillään olevien tehtävien saattamista valmiiksi, sekä dokumentaation saattamista valmiiksi.

Jäin dailyn jälkeen keskustelemaan rajapintatikettiäni testaavan testaajan kanssa. Testaaminen todettiin vaikeaksi ja päädyin muokkaamaan koodia siten, että nyt validointi timestamp-tietotyypille toimii samalla tavalla kuin edellisessäkin versiossa, mutta tarkistaa ennen validointia onko syöte aikaleima vai aikaleiman validointisääntö. Selvitin myös yhtä toista validointiin liittyvää ongelmaa, joka paljastui sellaiseksi, että develop-haarassa validaatiosäännöksi käy ””, eli tyhjä merkkijono ja omassani ei, joten muokkasin omani entisen kaltaiseksi.

Toinen tikettini, jossa korjattiin englanninkielisiä virheilmoituksia, meni läpi katselmoinnista ja siirtyi testausvaiheeseen.

Jatkoin työtäni seuraavan tiketin kanssa (serverivirhetiketti), jossa selvisi minulle uusi tapa vertailla numeerista järjestystä suoraan merkkijonosta käyttämällä compareTo-metodia, joka palauttaa tuloksen unicode-järjestyksen perusteella (W3Schools s.a.)**.**

Keskiviikko 4.1.2023

Työn alla olevan serverivirhetikettini kanssa tuli seuraava yllätys vastaan. Hämmennyin kovasti, kun siinä objektilla oli kolme toistensa kanssa ristiriitaista sijaintia: alkusijainti, loppusijainti ja sijaintinauha, jossa on sijaintipisteet nauhana. Backendin Java-koodin validointi pisteelle ei kuitenkaan osaa ottaa huomioon kuin nauhamaisen sijainnin validoinnin tämän tyyppiselle objektille. Kysyin aiheesta tech leadilta ja hän oli kanssani samaa mieltä siitä, että tosiasiallinen ongelma tässä on virheellinen sijainnin generointi, ei sijainti itsessään.

Ilmoitustiketilleni löytyi lisää testissä englanninkielisiä virhetilanteita, joten otin sen takaisin työn alle.

Inkrementin retrospektiivissa kokeiltiin poikkeuksellisesti uutta formaattia, eli Spotifyn tunnetuksi tekemää Squad Health Check-metodia, joka tekee näkyväksi ne kohdat, joihin tulee keskittää huomiota (Kniberg & Lindwall 2014). Kipukohdiksi koettiin speed, releases ja teamwork. Näiden kohtien tilanteesta käytiin keskustelua ja esitettiin mielipiteitä & parannusehdotuksia.

Torstai 5.1.2023

Hyväksymistestaus valmistui ja sain ylläpidolta viestin, että uusi tietokantadumppi oli ajettu palvelimelle. Kävin tarkastamassa datavirhetiketin tilanteen ja ongelma oli edelleen olemassa. Virheellisiä ilmoituksia oli noin 20 kappaletta. Kysyin tech leadin mielipidettä asiaan ja hän oli sitä mieltä, että niiden kanssa voi elää. Otin asian puheeksi myös dailyssä ja siellä käytiin laajemmalla joukolla keskustelua, jossa päädyttiin samankaltaiseen konsensukseen, eli että ei lähdetä laittamaan työpanosta näiden korjaamiseen migraatiolla, sillä kyseessä on vain väärälle alueelle osuvista ilmoituksista, ei itse objekteista. Asian juurisyy jäi kuitenkin hämärän peittoon, vaikka epäily on olemassa siitä, että käyttäjä on tehnyt nämä virheelliset ilmoitukset vanhalla massamuutostyökalulla, sillä käyttäjä, joka ne teki, työskenteli hankkeen aineiston parissa.

Korjasin ilmoitustikettini pikaisesti – tein vain tätä spesifiä virheilmoitusta koskevan käännöksen parhaan kykyni mukaan. Käännöksen toiminnallisuudesta tuli hieman erilainen kuin muista samantyyppisistä käännöksistä, sillä backend tuottaa kyseisen poikkeuksen erilaisella tavalla. Täsmälleen samanlaisen ratkaisun tuottaminen olisi siis vaatinut suurempia muutoksia useampaan ohjelman komponenttiin ja koin tämän ratkaisun tässä tapauksessa paremmaksi.

Metadatatiketti palasi katselmoinnista, sillä siihen tuli kollegalta erinomainen huomautus tietoturvariskistä, kun käyttäjän katseluoikeuksia ei tarkisteta siinä vaiheessa, kun metadatakäyttöliittymän kautta muokataan katseluoikeuskenttää. Tämä voi johtaa potentiaalisesti siihen, että henkilö pääsee oikeuksia muokkaamalla tilanteeseen, jossa hän pääsee katselemaan sellaista tietoa, johon hänellä ei muuten olisi katseluoikeuksia.

Katselmoin kollegan tekemää, yhteiseen käyttöliittymäkirjastoon kohdistuvaa, tikettiä. Siinä karttakomponenttiin tehtiin muutoksena yhden popupin korjaus. En saanut vikaa toistettua omalla koneellani, joten pidimme asian tiimoilta Teams-puhelun ja sen jälkeen onnistuin vian toistamisessa. Huomioni oli sellainen, että tämä kollegan tiketti ei näy tuotantokäytössä ollenkaan, kun tiketin näkymän sai käyttöönsä vain ajamalla karttasovellusta erilaisilla käynnistysparametreilla.

Inkrementtidemossa demosin ensimmäisessä sprintissä valmistuneita CSV-muutoksiani. Demon jälkeen olleessa, hankkeen tasoisessa, retrospektiivissä oli puhetta siitä, että suunnittelun muotona on rolling window, jossa tavoitteena on ennustaa lähiaikojen tapahtumia (Patil 2021). Näkymän tavoitteena on saada näkyvyyttä hankkeeseen 10 viikkoa kerrallaan ja tilanne on nyt sellainen, että

osa ominaisuuksista valmistuu ajoissa, osa ennen aikojaan ja osa valmistuu myöhässä. Tuntemattomat tuntemattomat ovat haasteellisia suunnittelun kannalta. Tuntemattomia tuntemattomia ovat sellaiset asiat, joista projektin vetäjä ole tietoinen ja joilla voi olla yllättäviä seuraamuksia (Ramasesh & Browning 2014, s. 3). Tämän parantamista varten voidaan suunnitella spike-tyyppisiä tehtäviä, joiden tarkoituksena on saada paremmin hahmotettu kuva ongelman laajuudesta ja vaikeustasosta (Drewniak 2018).

Perjantai 6.1.2023

Tämä päivä oli arkipyhä, joten en ollut töissä.

Viikkoanalyysi 7

Tämän viikon tavoitteinani oli oppia enemmän ohjelmistokehityksen hallinnollisesta puolesta ja keskittyä siihen, että löytäisin järjestelmällisellä työllä ratkaisun nopeammin ongelmiin. Ensimmäinen tavoite onnistui hyvin, sillä perehdyin hyvin Jiraan ja sen tarjoamiin näkymiin. Onnistuin etsimään Jirasta edellisen inkrementin aikana valmistuneet tehtävät ja katsomaan omista tehtävistäni, että missä niistä on käyttäjälle näkyviä, tarpeeksi suuria muutoksia, joita tulee inkrementtidemossa demota. Toisen tavoitteeni kanssa tulokset olivat ristiriitaisia – toisaalta onnistuin selvittämään haasteellisen datavirhetiketin, mutta sitten taas minulla oli haasteita toisen tiketin virheen toistamisessa, joten prosessi jatkuu vielä ennen kuin muutos on pysyvää.

Squad Health Check-metodi oli minusta perinteistä retroilua parempi tapa tarkastella kuluneen ajanjakson elämää. Syy tähän on selvä, sillä perinteissä retrossa voi olla korkea kynnys puhumiseen, kun ei ole käsitystä ryhmän sisäisestä tunteesta eri osa-alueiden suhteen. Uudella metodilla käydyssä keskustelussa tuli anonyymilla kirjanpidolla selkeästi ilmi, että on olemassa haasteellisia asioita, jonka jälkeen näistä pystyttiin tarkemmin keskustelemaan ja käymään läpi sitä, mitä asian edistämiseksi voisi tehdä. Osa asioista on tietenkin sellaisia, joihin meillä ohjelmistokehitystiiminä ei ole vaikutusmahdollisuuksia kuin vähän. Näitä haasteita ovat esimerkiksi asiakkaan massan hitaudesta johtuvat tai ulkopuolisten tahojen toimintatapojen tuottamat toimintahaasteet.

Minulle on muodostumassa kuva siitä, että kuinka tärkeää koodin laadun kannalta sen testaaminen ja vertaisarviointi on. Se on äärimmäisen tärkeää, sillä omille virheilleen on sokea, eikä yhden ihmisen ole mahdollista huomioida kaikkia näkökantoja. Tarkoitan erityisesti tämän viikon tarkastelujaksolla rajapintatikettiäni, jossa ensin päädyin muokkaamaan koodia sellaiseksi, jossa sen toiminnallisuus vastasi edellistä enemmän ja muutos vei minulta pari tuntia – toisena vaihtoehtona olisi ollut muokata robottitestejä monen päivän ajan. Tämän muokkauksen seurauksena laitoin tiketin uudestaan katselmointiin ja sen katselmoi eri kehittäjä kuin aiemmin, joten hän kiinnitti huomiota päivän-

selvään tietoturvariskiin, joka ei ollut tullut minulle, testaajalle tai aikaisemmalle katselmoijalle mieleen. Koodin laatu todella paranee sen myötä, mitä kriittisemmin sitä tarkkailee. Sitten syntyy linkki toiseen teemaan, joka on se, että milloin jokin on tarpeeksi hyvää. Kaikessa työssä täytyy päättää, milloin työ on valmis, ja milloin se kaipaa lisätyötä. Välillä tämän päätöksen tekeminen on helppoa, eli siitä löytyy DoD (definition of done), joka kertoo kehittäjälle mitkä ovat vaatimuksen valmiille ratkaisulle ja välillä asiassa täytyy käyttää kaupunkilaisjärkeä. Sattumoisin myös retrossa keskusteltiin tästä ”mikä on riittävän hyvä”-teemasta ja todettiin, että asiakas on tietysti se taho, joka viime kädessä päättää, että nyt on hyvä.

Ohjelmistoalalla toteutuu maailman peruslaki, joka on tuttu myös tapahtumatuotannosta: poppari tilaa, poppari saa. Kaikissa töissä myös mielestäni toteutuu se, että asiakkaan kanssa käydään myös neuvonpitoa siitä, mikä on mahdollista ja missä aikataulussa. Ohjelmistokehitys on perusluonteeltaan tuotekehitystä, joten tarkan päivämäärän antaminen valmistumiselle on mahdotonta, mutta järjestelmällisellä toiminnalla on mahdollista tarjota projektiin parempaa näkyvyyttä. Ongelmien pilkkomisella tarpeeksi pieniksi ja analysoimalla niitä ennen toteutusta voidaan parantaa ennusteen tarkkuutta ja tuotepäällikön sanoin: ”erottaa ammattimainen tuotekehitys harrastustoiminnasta”.

Seuraavan viikon tavoitteenani on tuottaa tiimille ja asiakkaalle arvoa inkrementtisuunnittelussa ja saada opittua lisää tietokannoista & Javasta.

1.15 Seurantaviikko 8

Maanantai 09.01.2023

Tänään oli ensimmäinen päivä inkrementtisuunnittelussa Pasilassa asiakkaan tiloissa. Tein junamatkalla töitä ja aloin diagnosoimaan loppukäyttäjältä tullutta CSV-tuonnin ongelmaa. Ongelmassa oli kyse virheilmoituksesta, jonka tuonti antoi. Tuonti kertoi käyttäjälle, että hänellä oli objekti, jonka sijainti poikkesi sen OID-tunnuksen perusteella validoidusta sijainnista. Minulla ei ollut käytettävissäni loppukäyttäjän käyttämää CSV-tiedostoa, joten testailin asiaa parhaan kykyni mukaan ilman sitä ja pyysin ongelman raportoineen yhteyshenkilön toimittamaan meidän käyttöömme sen tiedoston, jolla ongelma on saatu aikaan. Samalla löysin myös uuden bugin järjestelmästä, kun junassa oli heikot tiedonsiirtoyhteydet. Heikon yhteyden varassa CSV-tuonti meni läpi, mutta yhtään tuotua riviä ei hyväksymisnäkymään päätynyt. Front endin koodissa on siis todennäköisesti jätetty huomioimatta sellainen mahdollisuus, että pyydettyä tietoa ei koskaan saavu käyttäjältä ja tilanne käsitellään tyhjänä tiedostona. Kirjasin havaintoni bugitiketiksi.

Inkrementtisuunnittelu alkoi kaikkien hankkeen sovellusten yhteisellä osuudella, jossa käytiin yleisiä asioita läpi. Tästä haluan nostaa esiin sen, että puhuimme WIP-rajoitteista, joilla voitaisiin konkreettisesti rajoittaa sitä työn määrää, joka on kulloinkin tehtävänä. Hitain vaihe on se, joka määrittää tuotteen ja ominaisuuden valmistumista. Meidän projektissamme se on testaus, joka johtuu mielestäni siitä, että testataan ehkä enemmän yksittäisiä asioita kuin mekanismeja ja testaajilla ei kaikilla ole riittävästi osaamista robottitestien rakentamiseksi. Testattavia asioita on myös paljon, mikä johtuu projektin luonteesta.

Scrum master kysyi, että olisiko joku kehittäjistä vapaaehtoinen ottamaan vähän myös testaajan hattua itselleen ja ilmoittauduin vapaaehtoiseksi. Olen opiskellut Robot Frameworkia, jota käytetään projektissa automaatiotestauksen työvälineenä ja minusta tuntui hyvin mielenkiintoiselta päästä kokeilemaan myös tätä osaa ohjelmistokehityksen prosessista. Keskustelin asiasta testaajien kanssa ja sovimme, että alan ottamaan työn alle (nykyisten tikettieni valmistuttua) robottitestien tekemistä sellaisiin tiketteihin, joita en ole itse tehnyt tai katselmoinut.

Kävimme keskustelua projektiin tälle inkrementille tulevasta saavutettavuusanalyysistä. Todettiin että tehdään analysointia automaattisilla työkaluilla ja manuaalisesti. Saavutettavuusanalyysin tuotoksena on kasa tehtäviä, joiden perusteella voidaan sitten konkreettisesti parantaa sovelluksen saavutettavuutta.

Inkrementtisuunnittelun varsinaisena ohjelmana oli käydä inkrementille tarjolle olevia ominaisuuksia läpi, sekä suunnitella niiden toteuttamisen vaatimia tehtäviä konkreettisesti. Loppupäivä meni näiden asioiden parissa, mutta harhauduimme hieman liian detaljitasoiseen ja teknisesti korkealentoiseen keskusteluun yksittäisistä tehtävistä.

Tiistai 10.01.2023

Inkrementtisuunnittelun toisena päivänä veimme keskustelua eteenpäin yksittäisistä tehtävistä, sekä pyysimme asiakkaan ja tuoteomistajien kautta apua muilta tiimeiltä ja substanssiosaajilta niissä kohdin, kun määrittely tai ominaisuudessa toivottu lopputulos olivat liian epäselviä ominaisuuden toteuttamiseksi.

Keskustelimme yksityiskohtaisesti esimerkiksi objektin sijaintiarvon ylimäärittämisestä käsin tietyille ominaisuuksille, sekä siitä, mitä tämän muutoksen toteuttaminen tarkoittaisi tietokannan tauluille. Tämän testaaminen todettiin testaajien toimesta haastavaksi, koska vaikka mekanismille on suhteellisen suoraviivaista rakentaa testi, niin silti täytyy vähintään manuaalisesti testata, että muutos näkyy myös käyttöliittymässä. Onnistuin tekemään uusia ominaisuuksia toteuttavien tikettien sisältöjä, eikä niistä tullut tiimiltä huomautettavaa.

Sain haltuuni loppukäyttäjältä tulleen CSV-tuonnin virheen aiheuttaneen tiedoston ja pääsin testaamaan sen toistamista siinä vaiheessa, kun suunnittelussa ei ollut enää akuuttia tekemistä tarjolla. Totesin, että järjestelmän virheilmoitus oli täysin aiheellinen, mutta ei tosiasiallisesti käyttäjää auttava. Käyttäjä oli ottanut järjestelmästä objekteja koskevan CSV-viennin ja sen OID-tunnuksia, ja objektien sijainteja koskevat tiedot olivat siten lähtökohtaisesti oikein, koska ne generoidaan dynaamisesti järjestelmän tietokannasta. CSV-tiedoston avaaminen Excelissä rikkoi kuitenkin yhden sijaintitiedon validointimekanismeista, jossa tuotavana tietona voi olla esimerkiksi merkkijono ”001”, jonka Excel muuttaa oletusasetuksillaan sujuvasti muotoon ”1”. Sitten kun tätä tietoa tuodaan järjestelmään, järjestelmä toteaa, että OID-tunnus ja tämä ”1”-tunnistetieto eivät löydy samalta objektilta ja tuotu rivi on siten virheellinen. Ohjeistin käyttäjää vaihtamaan Excelissä kyseisen sarakkeen kategoriaksi ”Teksti”, jonka jälkeen Excel sallii etunollien olemassaolon sarakkeessa. Suljin avaamani tiketin ja loin sen pohjalta ”User Story”-tyyppisen tehtävän, jossa tämän virheilmoituksen yhteydessä käyttäjälle näytettäisiin opasteteksti, jossa kehotettaisiin tarkistamaan Excelistä sarakkeen muotoilu.

Keskiviikko 11.01.2023

Aloin tekemään muutoksia rajapintatikettiini. Työskentelin järjestelmällisesti, eli kun jokin asia ei toiminut, niin selvitin ensimmäisenä sen mitä tosiasiallisesti olen kysynyt tai mitä arvoja olen vertailemassa toisiinsa. Sain rakennettua käyttäjäoikeuksien validoinnin, mutta ongelmaksi jäi vielä havainto validointisäännöstä, jossa tietokannasta löytyy null-arvoisia validointisääntöjä, mutta uusien ominaisuuksien luominen vaatii arvon.

Osallistuin tiedonvaihtopalaveriin, jossa keskusteltiin tiedon formatoinnista ja erilaisista käyttötavoista hankkeen eri sovellusten välillä. Palaverista jäi konkreettisesti käteen paikkatiedon määritelmä, jonka yksi osallistuja tiivisti muotoon: ”paikkatieto on tietoa, jossa sijaintiin yhdistetään ominaisuustieto”. Ennen tätä kokousta paikkatiedon käsite oli minulle huomattavasti epämääräisempi, joten hyvä näin.

Torstai 12.01.2023

Tein töitä huonojen yhteyksien päässä, sillä minulla oli pari matkustuspäivää edessä omien asioideni hoitamiseksi toisella puolen Suomea.

Osallistuin Dailyyn, jossa totesin, että minulla on haasteita yhden tiketin katselmoinnissa, sillä en ole saanut ympäristöä toimimaan, sillä tavalla, joka vaadittaisiin ongelman toistamiseen. Muokkasin rajapintatikettiäni koskevia testejä siten, että nyt ne testaavat sellaisia asioita, joiden testaamisesta on enemmän hyötyä.

Iltapäivällä oli edessä kehittäjätapaaminen, jossa keskusteltiin aluksi julkaisuaikataulusta ja ylläpito esitti vaatimuksen siitä, että heidän tulee saada seuraavan julkaisun paketit käyttöönsä haaroihin viikkoa ennen julkaisua, jotta asennukset ehditään varmasti tekemään ennen julkaisua edeltäviä testejä. Tämä tarkoittaa käytännössä sitä, että haarat tulee lukita sovelluksissa aikaisemmin kuin ennen. Buildin tekeminen yhdestä artefaktista on testissä meidän tiimillämme ja siitä saa todennäköisesti aika pienellä vaivalla tehtyä toteutukset myös toisille tiimeille. Tämä vähentää virheiden mahdollisuutta julkaisuvaiheessa, kun haaroja ei tarvitse tehdä jokaista ympäristöä varten enää käsin. Keskusteltiin myös siitä, että kehitystiimeillä on tarvetta eri tiimien sovellusten konttiversioihin, sillä välillä syntyy tarve testata omien muutosten vaikutusta eri järjestelmien välillä.

Asiakkaan suunnalta tuli hieman ristiriitaista viestintää, sillä painotettiin bugien korjaamisen tärkeyttä ja päiviteltiin suurta avoinna olevien bugien määrää, mutta samalla oli inkrementin sisällöksi haluttu suuri määrä uusia ominaisuuksia. Tuoteomistaja kommentoi tiimin puolesta asiaa siten, että korkean prioriteetin bugeja ei ole juuri avoinna, jos ei lasketa niitä, mihin odotetaan lisätietoa substanssiosaajilta tai loppukäyttäjiltä. Tiimi tietenkin tekee sitä mitä asiakas haluaa, sillä asiakas maksaa laskun, mutta on mielestäni epärealistista vaatia enemmän kuin se, mihin tiimillä on kaistaa.

Perjantai 13.01.2023

Perjantaina tein etätyöpäivän ystävän sohvalta, koska olin edelleen matkalla.

Dailyssä kysyin scrum masterilta mikä olisi hyvä homma ottaa seuraavaksi työn alle, kun näytti siltä, että rajapintatikettini näyttäisi valmistuvan tänään. Scrum master totesi, että hyviä tehtäviä seuraavaksi ovat katselmointi tai robottitestien rakentaminen.

Tein rajapintatikettini loppuun ja tech leadin konsultoinnin kautta todettiin, että validointisääntönä saa olla null, kun kerran kannassa niitä on olemassa muutenkin. Tein sitten päivitystä koskevan muutoksen koodiin, jossa tämä hyväksytään, sekä muokkasin testien ja Swaggerin kuvauksia sen mukaisesti, että niistä on toisille kehittäjille enemmän apua.

Valitsin minulle tarjotuista ensisijaisista työvaihtoehdoista kollegan tietokantamigraation ja edustan muutoksen yhdistävän tiketin katselmoinnin. En ole kirjoittanut kovin paljoa SQL-kyselyitä, jos koulussa tehtyjä harjoituksia ei oteta huomioon, joten tässä oli erinomainen paikka oppia lisää. Sain esimerkiksi selvittää, mitä tarkoittaa CURRVALL-funktio: se palauttaa kurantin arvon sekvenssistä (IBM 2017). Tiketissä oli myös mielestäni elegantti tapa käyttää kokonaislukumuuttujaa boolean-arvon tapaan:

DECLARE muuttuja int;

SELECT COUNT(*) FROM x WHERE y and z >= 0 INTO muuttuja;

Tätä tietoa käytettiin sitten arvona siihen, että ajetaanko seuraavaa funktiota, jos count palauttaa arvon, joka on positiivinen kokonaisluku.

En ollut aiemmin myöskään katselmoinut kovin paljon tietokantamigraatioita käsitteleviä tikettejä, joten sain ratkaista uusia ongelmia. Ongelmani oli virheilmoitus, joka kertoi, että migraatiota ei ole ajettu ja pääsin tästä lopulta eroon poistamalla tietokannasta asioita ”schema_version”-taulusta. Tein myös virheen, sillä tämä asia luki hyvin selkeästi myös tiketin kuvauksessa, eikä pyörää olisi siten tarvinnut kertoa uudelleen. Jäi siis opittavaa ja sisäistettävää myös tuleville viikoille.

Viikkoanalyysi 8

Ratkaisin hyvin dynaamisella otteella asiakkaan CSV-tuonnin ongelman, kun minulla oli junassa rajallinen määrä aikaa, enkä kokenut mielekkääksi siinä vaiheessa syventyä laajaan ongelmaan. Työskentely asiakkaan kokeman ongelman parissa toi myös ilmi toisen bugin, eli virheellisen tuonnin heikkojen verkkoyhteyksien yli. Aion tulevalla viikolla keskustella sovelluksen toiminnan testaamisesta hitaan nettiyhteyden varassa, jota voi simuloida selaimilla (Draniceanu 2021).

Inkrementtisuunnittelusta tarttui mukaan oppi, että työn suunnittelussa ja organisoinnissa auttaa Kanbanissa oleva mahdollisuus rajoittaa meneillään olevaa työtä sen perusteella, että onko seuraavissa työvaiheissa kapasiteettia jatkaa tiketin parissa työskentelyä. WIP-limiitit lisäävät työn ennustettavuutta. Ohjelmistokehityksessä tuotantolinjan tukkeutumista tai ruuhkautumista ei välttämättä huomaa niin helposti, koska tuotettava tuote ei ole fyysinen, kuten perinteisessä tehtaassa. (Kiiskinen 2022)

Tuleville viikoille ja vuosille jäi edelleen runsaasti opittavaa, sillä vaikka onnistuin osittain ratkaisemaan ongelmia ensimmäisistä periaatteista käsin, tein myös samoja virheitä, jolloin lähdin ratkaisemaan ongelmaa sattumanvaraisessa järjestyksessä järjestelmällisen prosessin sijaan.

Migraatioiden katselmointi opetti myös sen, että Mavenista ei tarvitse ajaa clean-install-sykliä, jotta migraatiot ajetaan, vaan se tapahtuu automaattisesti backendin buildauksen myötä.

Pohdinta

Työn keskeisinä tavoitteina oli kehittyä ohjelmoijana, jossa koen onnistuneeni. Toisena keskeisenä tavoitteena oli oppia ketteristä kehitysmenetelmistä ja niistä kokemusta karttui runsaasti, sillä ketterät kehitysmenetelmät ovat projektissa käytössä. Lisäksi oppia haluttiin keskeisistä työkaluista, joista Java, Angular, Git, Github, Swagger, PostgreSQL, Spring Boot, Jira & Confluence tulivat paljon alkupistettä tutummiksi.

Olen kehittynyt tasaisesti ja lisännyt ymmärrystäni kaikista työssäni käyttämistäni työkaluista ja työmenetelmistä. Kaikista suurin harppaus osaamisessa on tapahtunut TypeScriptin parissa, sillä osaamiseni sen kanssa oli aloittaessa hyvin lähellä nollaa, ja seurantajakson aikana onnistuin tekemään sitä käyttäen uusia ominaisuuksia ohjelmistoon – jos kohta ne odottavat edelleen tätä kirjoittaessa julkaisua, kun niistä puuttuu määrittelyjä. Vajaaksi oppiminen jäi tietokantatyön osalta, sillä sitä ei osunut niin paljon seurantajaksolle kuin mitä oletin. Ohjelmistokehityksen hallinnollisesta puolesta oppia kertyi paljon, sillä jaksolle osui päivystysvuoro, jolloin onnistuin kasvattamaan ymmärrystäni hankkeesta kokonaisuutena, muiden tiimien työpanoksesta, kokonaistoiminnan koordinoinnista, tuotehallinnan päivittäisestä työstä ja siitä miten isosta hankkeesta on kokonaisuudessaan kyse.

Päivittäiseen työhöni olen saanut seurantajakson aikana hieman lisää järjestelmällisyyttä, vaikka siinä on vielä matkaa valmiiksi. Vianmääritystaitoni ovat kasvaneet, sillä olen usein onnistunut kokeilemaan yksinkertaisinta mahdollista selitystä – ja usein se on ollut hyvä alku- tai loppupiste ongelmanratkaisussa. Joka tapauksessa järkeily perusasioista käsin on hyvä toimintamalli, jossa ennen asian optimointia täytyy käydä keskustelua siitä, että onko prosessi tarpeellinen ja oikea:

” The best part is no part the best process is no process.”

Samoin se on tärkeää, että ymmärtää olla itselleen armollinen ja sallia virheiden tekeminen. Hän joka ei tee virheitä, ei juuri mitään tee. On hyvä olla suunnitelma, mutta samalla on oltava valmis muuttamaan suunnitelmaa uusien tosiasioiden edessä. Ihminen tuskin tulee koskaan valmiiksi, mutta aina on tilaa kasvulle.

Päiväkirjamainen työskentely sopi hyvin minulle, sillä pystyin tekemään työpäivän päätteeksi muistiinpanot sinä päivänä tekemistäni asioista ja palaamaan niihin myöhemmin pidemmän formaatin reflektoinnin kautta. Tämä auttoi avaamaan erilaisia ovia, joista seurasi konkreettisia ahaa-elämyksiä. Huomasin työtä kirjoittaessani myös sen, että asioiden kirjoittaminen auttaa niiden muistamista. Elämänkokemuksesta ja laajasta projektityön osaamisesta on ollut eittämättä hyötyä työyh-

teisöön integroitumisessa ja arvon tuottamisessa asiakkaalle – päivystysvuoroni jälkeen sain runsaasti kiitosta aktiivisesta otteesta ja se sai myös minua seuranneet päivystäjät parantamaan juoksuaan.

Tulevaisuudessa aion kehittää osaamistani päivittäisen työn myötä. Tämän lisäksi suunnitelmissani on jatkaa päiväkirjan pitämistä – vähintään ranskalaisten viivojen tasolla, sillä se auttaa pääsemään eteenpäin ja tekemään tehtyä työtä näkyväksi, vaikka omaa työtä ei pääse aina muuten esittelemään tai se ei näy loppukäyttäjän arjessa. Suunnitelmissani on myös opiskella lisää – työnantaja tarjoaa tähän erinomaiset mahdollisuudet, sillä Sitowisella on mahdollista käyttää palkallista työaikaa oman osaamisensa parantamiseen. Keväällä järjestetään myös erilaisia kursseja, joihin voin työnantajan kustannuksella osallistua ja odotan erityisesti DevSecOps-koulutusta, sillä se on minulle melko vierasta maaperää. Näiden koulutusten lisäksi aion hakea maisteri- tai YAMK-ohjelmaan mahdollisimman pian, sillä koulutus ei koskaan sulje yhtään ovea vaan avaa niitä.

Lähteet

Sitowise. s.a. a. Luettavissa: https://www.sitowise.com/fi. Luettu: 01.11.2022.

Sitowise. s.a. b. Luettavissa: https://www.sitowise.com/fi/digitaaliset-palvelut. Luettu: 01.11.2022.

RxJS. s.a. Luettavissa: https://rxjs.dev/guide/subscription. Luettu: 26.11.2022.

Scrum.org. s.a. Luettavissa: https://www.scrum.org/resources/what-is-a-daily-scrum. Luettu: 27.11.2022.

Moment.js. s.a. Luettavissa: https://momentjs.com/docs/#/-project-status/. Luettu: 27.11.2022.

Github. 2022. Luettavissa: https://github.com/date-fns/date-fns. Luettu: 27.11.2022.

WCAG. 2022. Luettavissa: https://www.w3.org/WAI/WCAG21/Understanding/intro. Luettu: 27.11.2022.

Alvestrand, H. 1997. Luettavissa: https://www.alvestrand.no/objectid/. Luettu: 03.12.2022.

Red Hat. 2022. Luettavissa: https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac. Luettu: 03.12.2022.

Wolf, M., & Wicksteed, C. 1998. Date and time formats. W3C NOTE-datetime-19980827, August.

Forbes Technology Council. 2016. Luettavissa: https://www.forbes.com/sites/forbestechcouncil/2016/05/09/the-benefits-of-using-agile-software-development/?sh=6679f22ab0f8. Luettu: 04.12.2022.

Parwinder, B. 2020. Luettavissa: https://dev.to/bhagatparwinder/map-vs-mergemap-vs-switchmap-5gee. Luettu: 10.12.2022.

Testim. 2021. Luettavissa: https://www.testim.io/blog/what-is-a-linter-heres-a-definition-and-quick-start-guide/. Luettu: 10.12.2022.

Microsoft. s.a. Luettavissa: https://support.microsoft.com/en-us/office/networkdays-function-48e717bf-a7a3-495f-969e-5005e3eb18e7. Luettu: 10.12.2022.

Kidd, C. 2020. Luettavissa: https://www.bmc.com/blogs/patch-hotfix-coldfix-bugfix/. Luettu: 10.12.2022.

Git. s.a. a. Luettavissa: https://git-scm.com/docs/git-stash. Luettu: 10.12.2022.

Burke, S. 2022. Luettavissa: https://becomeaprojectmanager.com/gold-plating-in-project-management/. Luettu: 10.12.2022.

Git. s.a b. Luettavissa: https://git-scm.com/docs/git-revert. Luettu: 19.12.2022.

OpenAI. s.a. Luettavissa. https://beta.openai.com/examples/default-js-helper. Luettu: 19.12.2022.

Typescriptlang. 2022. Luettavissa: https://www.typescriptlang.org/docs/handbook/2/objects.html. Luettu: 19.12.2022.

Git. s.a. c. Luettavissa: https://git-scm.com/docs/git-diff. Luettu: 19.12.2022.

Paakki, J. 2011. Luettavissa: https://www.cs.helsinki.fi/u/paakki/Vaatimus-11-Luentokalvot-1.pdf. Luettu: 19.12.2022.

Stackoverflow. 2011. Luettavissa: https://stackoverflow.com/questions/5905054/how-can-i-recursively-find-all-files-in-current-and-subfolders-based-on-wildcard. Luettu: 19.12.2022.

Duignan, B. 2022. Luettavissa: https://www.britannica.com/topic/Occams-razor. Luettu: 23.12.2022.

Stackoverflow. 2009. Luettavissa: https://stackoverflow.com/questions/1295170/whats-the-difference-between-boolean-and-boolean-in-java. Luettu: 23.12.2022.

Atlassian. s.a. Luettavissa: https://www.atlassian.com/git/tutorials/rewriting-history. Luettu 23.12.2022.

Neut, M. 2021. Luettavissa: https://uxdesign.cc/design-like-elon-musk-using-6-fundamental-principles-4aaab08d5e41. Luettu 23.12.2022.

Simmons, J. 2018. Luettavissa: https://www.auraq.com/the-importance-of-avoiding-hardcoding/. Luettu: 25.12.2022.

Cox, G. 2018. Luettavissa: https://eu.thespectrum.com/story/news/2018/12/02/how-and-why-restarting-your-computer-can-fix-technical-issues/2164794002/. Luettu: 31.12.2022.

Muguda, N. 2021. Luettavissa: https://naveenkumarmuguda.medium.com/occams-razor-in-software-development-56ee3e8b8ce8. Luettu: 31.12.2022.

Monteiro, T. 2022. Luettavissa: https://www.freecodecamp.org/news/what-is-abstraction-in-programming/. Luettu: 31.12.2022.

W3Schools. s.a. Luettavissa: https://www.w3schools.com/java/ref_string_compareto.asp. Luettu: 07.01.2023.

Kniberg, H. & Lindwall, K. 2014. Luettavissa: https://engineering.atspotify.com/2014/09/squad-health-check-model/. Luettu: 07.01.2023.

Patil, V. 2021. Luettavissa: https://medium.com/analytics-vidhya/predictive-models-using-rolling-window-features-i-691172c19e95. Luettu: 08.01.2023

Ramasesh, R.V. & Browning, T.R., 2014. A conceptual framework for tackling knowable unknown unknowns in project management. *Journal of operations management*, *32*(4), s.190-204.

Drewniak, J. 2018. Luettavissa: https://confluence.technologynursery.org/display/TSSG/Spikes+in+Jira. Luettu: 08.01.2023.

IBM. 2017. Luettavissa: https://www.ibm.com/docs/en/informix-servers/12.10?topic=operators-using-currval. Luettu: 14.1.2023.

Kiiskinen, A. 2022. Luettavissa: https://www.eficode.com/blog/how-to-avoid-clogging-up-your-kanban-with-too-much-work-in-progress. Luettu: 14.1.2023

Draniceanu, A. 2021. Luettavissa: https://blog.testproject.io/2021/09/27/simulating-slow-network-connection-for-testing/. Luettu: 15.1.2023.