

Päiväkirja DevOps- ja testausautomaation asiantuntijan arjesta

Sergei Pavlov



Tekijä(t) Sergei Pavlov	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko Päiväkirja DevOps- ja testausautomaation asiantuntijan arjesta	Sivumäärä 58
Opinnäytetyön otsikko englanniksi A diary thesis about QA consultants work.	
<p>Tässä opinnäytetyössä seurataan DevOps- ja testausautomaation asiantuntijan työtä yrityksessä Comiq 10 viikon ajan 18.2 – 28.4.2019. Opinnäytetyö on tehty päiväkirjamenetelmällä ja siinä seurataan päivittäisiä tavoitteita ja tuloksia.</p> <p>Comiq on keski-suuri it-alan konsulttiyritys, joka työllistää pääsääntöisesti DevOps- ja testausautomaation asiantuntijoita. Yrityksen pääasiallinen liiketoiminta on konsulttipalveluiden tarjoaminen asiakkaille, joten suurin osa yrityksen työntekijöistä työskentelevät asiakkaan tiloissa. Testaus on osa laadunvarmistusta, jonka avulla saadaan kehittäjille tietoa tuotteen ongelmista ja varmistetaan sen toimivuus. Paljastuvat virheet voivat olla pieniä, jotka eivät käytössä aiheuta ongelmia tai voivat olla myös isoja ja johtaa esimerkiksi tietovuotoon.</p> <p>Päiväkirjan tavoitteena on asettaa työlleen tavoitteita ja opinnäytetyön päätteeksi tutkia omaa kehittymistä 10 viikon ajalta. Samalla päiväkirja auttaa tuomaan DevOps- ja testausautomaation ammattilaisten työtehtäviä esille ja tutustuttaa opinnäytetyötä lukevia ihmisiä niihin.</p>	
Asiasanat Projekti, konsultti, asiantuntija, projekti, testausautomaatio, asiakas.	

Sisällys

1	Johdanto	1
2	Lähtötilanteen kuvaus	4
2.1	Oman nykyisen työn analyysi	4
2.2	Sidosryhmät työpaikalla	5
2.3	Vuorovaikutustaidot työpaikalla	6
3	Päiväkirjaraportointi	8
3.1	Seurantaviikko 1	8
3.2	Seurantaviikko 2	13
3.3	Seurantaviikko 3	18
3.4	Seurantaviikko 4	23
3.5	Seurantaviikko 5	29
3.6	Seurantaviikko 6	34
3.7	Seurantaviikko 7	37
3.8	Seurantaviikko 8	42
3.9	Seurantaviikko 9	46
3.10	Seurantaviikko 10	50
4	Pohdinta ja päätelmät	54
	Lähteet	56

1 Johdanto

Kyseinen opinnäytetyö kirjoitetaan päiväkirjamuodossa ja ensimmäinen raportoitava päivä on 18.02.2019. Opinnäytetyötä tehdään tasan 10 viikkoa ja viimeinen seurattava päivä on 28.04.2019. Päiväkirjassa työlle asetetaan päivittäin tavoitteita, jotta päivän päätteeksi voidaan tarkastella saavutuksia ja viikoittain tehdään viikkoanalyysi.

Opinnäytetyössä seurataan DevOps- ja testausautomaation asiantuntijan arkea Comiq nimisessä yrityksessä ja mitä kaikkea työ sisältää. Testaus IT-alalla on osa laadunvarmistusta, mikä on todella tarpeellista tehdä, jotta ulos saadaan käyttäjälle turvallinen ja käytännöllinen tuote.

Comiq on laadunvarmistukseen erikoitunut konsulttiyritys, joka tarjoaa testausautomaatiota ja DevOpsia asiakkaiden tarpeisiin. Yrityksessä on reilut 70 työntekijää, joista suurin osa testausasiantuntijoita. Itse aloitin kyseisessä työtehtävässä marraskuussa 2018 noin kolmen kuukauden mittaisella perehdytysjaksolla, jonka aikana opetettiin kaikki mahdollinen työssä tarvittava taito. Nyt kun perehdytys on loppunut, meidän ryhmää työstää sisäistä projektia samalla kun meille haetaan asiakasprojekteja, joissa olisi laadunvarmistukselle tarvetta. Meidän toimistomme tiloissa on tällä hetkellä kolme työskentelytilaa: avoin toimisto, hiljainen huone ja neuvotteluhuone. Suurimman osan ajasta meidän ryhmämme on työskennellyt neuvotteluhuoneessa, jotta me emme häiritse muiden työntekoa ja meitä ei häiritse ulkopuolinen hälinä. Nyt kun meistä tuli täyspäiväisiä konsultteja, me työskentelemme avoimessa tilassa.

Kyseinen työpaikka on minulle ensimmäinen alan työpaikka, joten osaaminen oli aika perustasolla, pohjalla pelkästään käytyt kurssit. Perehdytysjakso kuitenkin auttoi todella paljon pääsemään mukaan asiaan ja nyt kykenen työskentelemään itsenäisesti. Testausautomaatio on tiettyssä mielessä hankala aihe koska uusia asioita tulee jatkuvasti enemmän ja meillä jokainen uusi ryhmä opiskelee asioita yhä pidemmälle ja pidemmälle. Monet asiakkaat toisaalta käyttävät vanhoja työkaluja mikä johtaa siihen, että usein konsultit opiskelevat vaikeita uusia asioita, joita ei kuitenkaan voi käyttää asiakkaalla. Kuitenkin asiakkaan käyttämiä työkaluja ei ole järkevää opiskella, koska ne eivät ole ajankohtaisia. Tosi moni meillä käytetty työkalu on sen verran uusi, että niistä ei varsinaisesti ole muuta kirjallisuutta kuin esimerkiksi internetartikkelit. Testausautomaatiossa käytetty kieli on Robot Framework (RF) joka on suomalaisen Pekka Klärkin kehittämä ja hyödyllisin info mikä asiasta löytyy, on Pekan itse kirjoittamat ohjeet RF internetsivuilla. DevOpsissa taas virtualisointi ja niin sanottujen konttien käyttö on asian ydin ja esimerkiksi Docker -työkalun sivuilla löytyvät parhaan asiaan liittyvät tekstit, vaikka esimerkiksi James Turnbullin kirjoit-

tama The Docker Book: Containerization is the new virtualization (James Turnbull, 2014) on myös mainio kirja aiheesta. Kirjan saa ostettua sähköisenä osoitteesta docker-book.com ja sitä päivitetään jatkuvasti.

Yllä mainittujen asioiden lisäksi työssä pitää osata ohjelmointikieliä kuten Python, Java ja C#. Keskeinen taito on osata virtualisoida kone ja siellä automaatiolla testata jokin ohjelma. Perehdytyksen aikana me opiskelimme myös paljon eri taitoja, esimerkiksi html-ohjelmointia, tietokantoja ja palvelinohjelmia.

Alla on listattu keskeiset käsitteet:

DevOps: DevOps on sähköisten palveluiden toimintamalli, joka pyrkii automatisoimaan kehityksen, testauksen ja ylläpidon sekä muut tähän liittyvät toiminnot.

Testausautomaatio: Testausmenetelmä, jossa testit suoritetaan automaattisesti, mikä säästää reilusti aikaa verrattuna manuaaliseen testaukseen.

Robot Framework (RF): Robot Framework on suomalaisen Pekka Klärckin kehittämä testiautomaatio-framework hyväksymistestaukseen ja hyväksymistestausvetoiseen ohjelmistokehitykseen.

Konsultti: Yrityksen ulkopuolinen neuvova asiantuntija.

Kontti: Tietosäiliö missä jokin järjestelmän osa pyörii, esim. Tietokanta voi pyöriä yhdessä kontissa, itse sivu pyörii toisessa kontissa.

Scrum: On projektinhallinnan viitekehys, jota käytetään yleisesti ketterässä ohjelmistokehityksessä. Vaikka Scrum on kehitetty erityisesti ohjelmistoprojektien hallintaan, sitä voidaan soveltaa myös yleisesti projektinhallinnassa.

Sprint: On scrumin ydin, enintään kuukauden pituinen tai sitä lyhyempi aikaraja, jonka sisällä tuotetaan "valmiin" määritelmän täyttävä, käyttökelpoinen ja potentiaalisesti julkaisukelpoinen tuoteversio. Sprinteillä on sama pituus koko kehityksen ajan. Uusi sprintti alkaa välittömästi edellisen päätyttyä.

Daily Scrum (Suom. päiväpalaveri) : On enintään 15 minuutin mittainen aikarajattu palaveri, jossa kehitystiimi tahdistaa keskinäiset työnsä ja luo suunnitelman seuraavil-

le 24 tunnille. Tämä tapahtuu tarkastelemalla edellisen päiväpalaverin jälkeen tehtyä työtä ja ennustamalla, mitä voidaan toteuttaa ennen seuraavaa päiväpalaveria.

2 Lähtötilanteen kuvaus

2.1 Oman nykyisen työn analyysi

Tällä hetkellä työnimikkeeni on DevOps- ja testausautomaation asiantuntija ja tehtävänä on varmistaa asiakkaan tuotteen laatu yksinkertaisesti sanottuna. Käsitteet eivät kuitenkaan ole kaikille tuttuja ja ovat paljon vaikeampia selittää kuin yhdellä rivillä.

DevOps tulee sanoista Software Development eli Ohjelmistokehitys ja IT operations eli IT toiminta. Tämä sähköisten palveluiden toimintamalli pyrkii automatisoimaan kehityksen, testauksen ja ylläpidon sekä muut tähän liittyvät toiminnot. Minkä takia tarvitaan DevOps -mallia tuotannossa? Nykyaikaisen työkalujen ansiosta, on mahdollista viedä asioita pidemmälle ja automaatioista myös ohjelmistokehitys ja sen ylläpito sekä testaus, jotta työskentely on mahdollisimman tehokasta. Jos esimerkiksi kymmenen vuotta sitten ohjelmisto testattiin painamalla fyysisesti nappeja, niin nykyään sen hoitaa robotti, vaikka tuhat kertaa samaan aikaan.

Asiantuntijana ja konsulttina tehtävänäni on varmistaa, että asiakkaan tuote on mahdollisimman laadukas ja käyttäjäystävällinen. Tämä luo konsulteille myös isot paineet, koska usein ihmiset ajattelevat, että jos testeissä ei ilmennyt virheitä, tuote toimii täydellisesti ja siinä ei ilmene ongelmia. Kun taas käy toisin, asiasta helposti syytetään testaajia, vaikka faktaa on se, että vaikka testit menivät täydellisesti läpi, se ei tarkoita, että ohjelmassa ei ole virheitä, koska jopa nykyään automaatioaikaan on varsin mahdotonta testata täysin kaikkea ajallisesti. Kuitenkin testaajat parhaimman mukaan yrittävät varmistaa, että ohjelmassa on mahdollisimman vähän virheitä ja ongelmia. Kun aloitin kyseisessä työssä, minulla ei ollut muuta kokemusta kuin opiskelustani ja jouduinkin opettelemaan hirveästi uusia asioita. Hakiessani en edes tiennyt mitä on DevOps ja automaatiotestaus. Perehdytysjakson aikana opin paljon erilaisia taitoja, joita voin hyödyntää asiakkaalla. Ymmärrän esimerkiksi paljon enemmän virtualisoinnista ja mihin sitä tarvitaan, sekä miten se tehdään. Osaan kirjoittaa toimivia ja laajoja testejä käyttäen Robot Framework -kieltä, josta en aikaisemmin edes tiennyt.

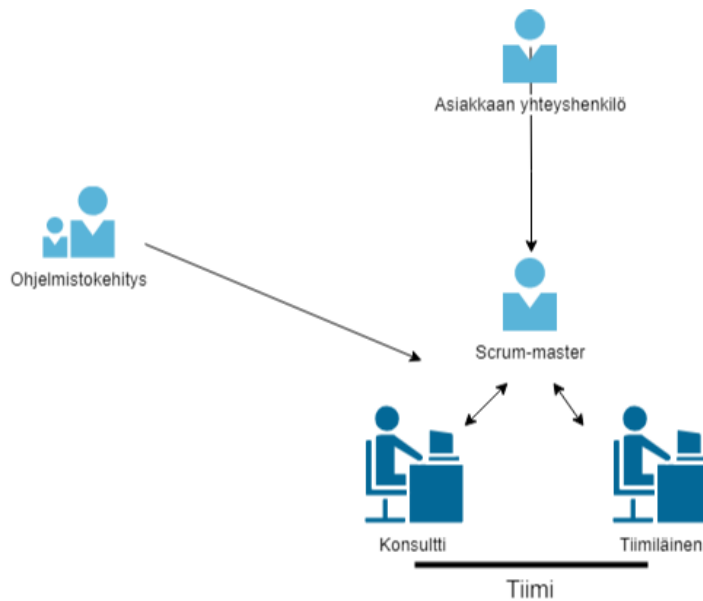
Tärkeintä työssäni on olla avoin uusille haasteille ja olla valmis oppimaan vaikeatkin työkalut. IT-alalla asiat muuttuvat todella nopeasti, eikä ole varaa jäädä vanhan osaamisen varaan. Pitää olla työssään tarkka ja vaativa itseään kohtaan, laadunvarmistuksessa ei ole varaa huolimattomuuteen.

Tällä hetkellä työssäni olen aloitteleva toimija. Olen ollut alalla töissä vasta neljä kuukautta ja edessä on vielä hirveästi opiskeltavaa. Pitkä perehdytysjakso kuitenkin auttoi todella paljon ja antoi hyvät pohjat siirtyä varsinaiseen työelämään. Olen nyt neljän kuukauden jälkeen paljon varmempi kuin aloittaessani työt, vaikka olikin alan opiskelua takana vuosia. En kuitenkaan koe, että tämä häiritsee itseäni tällä hetkellä hirveästi, koska IT-alalla pitää kehittyä ja oppia koko ajan uutta, minkä takia moni asia mitä me ollaan opeteltu ei ole vanhemmille työntekijöille tuttu. Täten työnteko on jatkuvaa opiskelua myös vuosia yrityksessä olleille tekijöille. Yrityksessämme on kuitenkin hyvä yhteishenki ja muiden kunnioittaminen, joten apua saa aina joltakin, eikä tarvitse yksin asian kanssa olla.

Ammatillisessa kehittämisessäni olen vasta alkuvaiheessa, mutta kuitenkin hyvän määrän osaamista omaava asiantuntija. Kehittymisalueita on hurjasti, koska tällä hetkellä osaan kaikesta perusteet, jotta voin työskennellä, mutta en kuitenkaan voi sanoa osaavani jotain erikoisen hyvin. Päivä päivältä kuitenkin kehitän taitoni parhaani mukaan, jotta voisin kehittyä sellaiseksi ammattilaiseksi, jonka asiakkaamme tarvitsevat. Tähän mennessä olen kuitenkin työskennellyt ainoastaan sisäisissä projekteissa, eikä ole vielä päässyt tekemään työtä asiakkaan tiloihin. Nykyisessä projektissa toimin kuitenkin jo itsenäisesti ja voin pitää etäpäiviä ilman että tarvitsen jatkuvaa tukea. Jatkuva kehittyminen on kuitenkin osaamisen ydin ja yritän pitää huolta siitä, että taitoni ovat ajan tasalla jatkuvasti.

2.2 Sidosryhmät työpaikalla

Talon sisäisessä projektissa sidosryhminä ovat oma mahdollinen tiimi sekä esimies. Asiakkaalla toimiessa kuitenkin voi olla erilaisia sidosryhmiä riippuen asiakkaasta ja projektista. Alla on Kuvassa 1 on esitetty mahdolliset sidosryhmät asiakasprojektissa.



Kuva 1. Sidosryhmät projektissa

Kuvassa 1 on esitetty sidosryhmät asiakasprojektissa. Lähtökohtaisesti työhön liittyvät kontaktit sisältävät omat tiimiläiset ja scrum-masterin, jotka voivat olla muita konsultteja, talon omia testaajia tai vaikka ohjelmoijia, asiakkaan yhteyshenkilöön ja ohjelmistokehitysryhmään, jonne ongelmakohtat raportoidaan.

2.3 Vuorovaikutustaidot työpaikalla

Työskentely tähän mennessä on edennyt tiimin jäsenenä, joten suurin osa työstä valmistuu jonkun kanssa yhteistyössä, välillä jopa koko tiimin yhteisvoimin. Näissä tilanteissa on tärkeää löytää yhteinen sävel, koska jokaisella on oma työskentelytapa ja jokainen haluaa luoda asioita oma tavalla. Meillä se ei onneksi kuitenkaan aiheuta haasteita, koska kaikki ovat avoinmielisiä ja mielellään antavat toisen luoda, kehittymisen ja projektin parhaaksi. Tämän takia 70% viime projektista loimme tiimin kanssa yhteistoimin ryhmäkoodaus menetelmällä. Kyseinen tapa ei kuulosta helpolta tai tehokkaalta, mutta viisi päätä yhdessä saavat paljon enemmän aikaan kuin yksi, jos kaikki ovat samalla aaltopituudella. Meillä kun kaikki nauttivat työstään ja osaavat ilmaista mielipiteen, saimme projektin päätteeksi esimieheltä kehuja parhaimmasta lopputuloksesta tähän mennessä.

Ongelmat saattavat kuitenkin tulla eteen, kun joutuu siirtymään asiakasprojektiin ja siellä on sitten ihmisiä eri kokemuksella, jotka eivät välttämättä ajattele samalla tavalla. Kuitenkin uskon, että tulen hyvin pärjäämään erilaisten ihmisten kanssa.

3 Päiväkirjaraportointi

3.1 Seurantaviikko 1

Maanantai 18.02.2019

Uusi viikko tuo tullessaan uuden projektin. Meidän tiimimme tavoitteena on jatkossa tutkia Magento -nimistä verkkokauppa-alustaa ja testata sitä, sekä raportoida automaatiotestauksen hyödyt ja alusta hyödyllisyys perehdytyksissä jatkossa. Itse en ole aikaisemmin kuullut kyseisestä alustasta, joten päivän tavoitteena on saada pohjajymmärrys Magentosta ja miten sitä voi käyttää. Uuden työkalun käyttöön ottaminen on aina haastavaa, etenkin jos se ei ennestään ole ollut käytössä yrityksessä.

Magento on todella monipuolinen verkkokauppa-alusta, jota käytetään ympäri maailmaa verkkokauppojen rakentamiseen. Kuten Teemu Malinen kirjoittaa blogissaan ”Magento verkkokauppa – Mitä se maksaa?” (2014), yhä useampi ensimmäistä verkkokauppaa perustava tai vanhaa alustaa uusimassa oleva kauppias on kiinnostunut avoimen lähdekoodin vaihtoehtoista, joista Magento on markkinaykkönen. On järkevää ottaa juuri tällainen järjestelmä testaukseen, näin työtä voi hyödyntää mahdollisen asiakkaan verkkokauppaan testaukseen, jos Magentoa käytetään siinä. Koko päivä meni Magenton tutustumiseen ja sen päätteeksi saimme järjestelmän käyttöön.

Tiistai 19.02.2019

Viikon agenda on saada Magento -alusta käyttöön ja siihen koedata testejä varten. Eilen saimme järjestelmän toimimaan, joten nyt toisena päivänä on tarkoitus saada siihen koeverkkokauppa, jotta myöhemmin sitä voisi testata. Tarkoitus olisi myös tehdä tästä kaikesta github -säiliö, jotta tulevat juniorit voivat käyttää tätä säiliötä heidän perehdytyksessä ja demoa voi myös näyttää esimerkiksi potentiaalisille asiakkaille.

Itsessään ensimmäistä kertaa asennettuna Magento ei sisällä yhtään mitään testattavaa, joten siihen on saata demodata. Magenton nettisivuilla on valmis paketti dataa ja ohjeet miten kyseinen data saadaan käyttöön. Valitettavasti virtualisointiympäristö on sen verran epävaka, joten helposti vierähtää kokonainen päivä saada toimimaan asia, johon on olemassa ohjeet. Virheiden kautta asia kuitenkin saatiin hoidettua ja meillä olikin päivän päätteeksi toimiva verkkokauppa, jossa oli iso tietokansa ja paljon

tuotteita. Kahden päivän aikana huomasin kuinka monta vaihdetta järjestelmän asennuksessa oli, vaikka asia näytti alussa paljon helpommalta. Virtualisoinnissa koskaan ei voi olettaa, että asiat onnistuvat helposti, koska kaikilla on eri toimintaympäristö ja se voi aiheuttaa ongelmia asennuksessa ja käytössä.

Keskiviikko 20.02.2019

Tämän päivän tehtävänä on saada Magento-pohjaiseen demo-verkkokauppaan maksuominaisuus, jotta järjestelmälle voidaan suorittaa End-to-end testi, mikä tarkoittaa, että testataan koko prosessi alusta loppuun, eli kirjautumisesta verkkokauppaan tilauksen tekoon ja maksamiseen. Paytrail tarjoaa maksuttoman maksuominaisuuden verkkokauppaan ja siihen demotunnukset, joten päivän agenda on saada se toimimaan.

Maksuominaisuuden asentaminen järjestelmään on vähän hankalampi tehtävä, vaikka ohjeet löytyvätkin. Maksuominaisuus on asennettava samaan konttiin missä Magento -järjestelmä pyörii ja sitten admin -sivun kautta konfiguroida asetukset. Asentaminen on kuitenkin astetta hankalampi, koska ympäristö toimii vielä niin epävakaisesti ja ensimmäiset asennuskerrat johtivat johonkin virheeseen ja järjestelmän rikkoutumiseen. Tässä vaiheessa tulee esille tärkein etu ryhmätyöskentelyssä. Vaikka meidän tiimissämme kaikki omaa saman koulutustausta, jokaisella on alue, jota osaa paremmin ja joka kiinnostaa enemmän. Yhteisellä panostuksella sitten saimme kaiken toimimaan, jottei tarvitse jättää asioita kesken ja lähteä kotiin.

Torstai 21.02.2019

Tänään on tarkoitus saada järjestelmästä toimiva docker -paketti, jonka voisi vaan kloonata meidän gitlab -säiliöstä ja se toimisi suoraan. Se vaatii sen, että järjestelmästä on täysin toimiva versio, josta sitten voidaan luoda paketti, joka viedään säiliöön gitlabin, josta sitten kuka vaan voi asentaa sen omalle laitteelle toimivana ja testausvalmiina. Tässä tapauksessa kuitenkin luodaan tutun git -paketin sijaan docker -paketti, joka on virtualisointipaketti. Tämän paketin kloonaukseen tarvitaankin sitten Docker -sovellus tietokoneelle asennettuna.

Docker-paketin luominen ei ollut vaikeampaa kuin git-paketin luominen, joten me saimme sen hyvin vietyä meidän gitlab-säiliöömme. Kuitenkin kun paketin yritti kloonata omalle tietokoneelle, maksuominaisuus ei toiminut, vaikka kaikki näytti olevan kunnossa. Pitkän pohdiskelun jälkeen päädyimme siihen tulokseen, että sitä ei saa toimimaan meidän kokonaisuudessamme ja meidän pitää rakentaa parempi paketti. Internetistä kuitenkin onneksi löytyi vapaassa käytössä oleva meille sopiva paketti, jonka otimme työn alle. Tätä uutta pakettia muokkaamalla saimmekin luotua meille oman toimivan docker-paketin, jossa maksuominaisuus pysyi toimivana.

Perjantai 22.02.2019

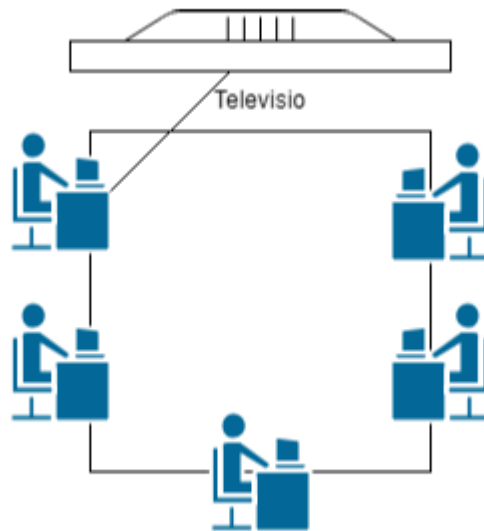
Perjantai on meidän yrityksessämme yleensä rennompia ja monet asiakkaalla olevat konsultit pitävät etäpäivän. Tänään hiemme viikon aikana tehdyksi saatuja asioita paremmiksi ja varmistamme että kaikki toimii, jotta voimme rauhassa siirtyä viikonlopun viettoon ja jatkaa seuraavalla viikolla projektia. Ensi viikolla tulemme tekemään paljon testejä Magentoon, joten olisi hienoa, jos kaikki olisi valmiina heti maanantaina.

Kuten yleensä, perjantai on astetta rennompia päiviä, vaikka tietenkin teemme kaikki päivän työt valmiiksi ennen kotiin lähtöä. Alkupäivästä testasimme meidän docker-pakettiamme eri koneilla, koska halusimme olla varmoja siitä, että kaikki asentuu oikein ei laitteilla. Kun kaikki varmasti toimii oikein, teemme siihen lyhyen pienen testin, missä katsomme, että testi menee läpi eikä kaadu mihinkään. Nopeaan testaukseen riittää testi missä avataan selain, mennään johonkin sivulle ja suljetaan selain. Näin saadaan nopeasti ja helposti testattua saako kontit toisiinsa yhteyden ja onko kaikki reitit kirjoitettu oikein. Jos testi menee läpi, tiedetään että järjestelmä tekee sen minkä sen pitäisikin ja yhteys on olemassa. Kun taas testit eivät mene läpi, tarkastellaan virheilmoitus ja lokit, mistä sitten selviää epäonnistumisen syy. Se voi olla kiinni yhteydestä tai ihan pienestä kirjoitusvirheestä. Kun kaikki toimii, päivitämme viimeisimmän toimivan version gitlabin ja siirrymme viikonlopun viettoon.

Viikkoanalyysi

Tämän viikon tärkein uusi asia oli Magento-järjestelmän käyttöönotto. Itselläni ei aikaisemmin ole Magentosta ollut mitään kokemusta, edes nimeä en tiennyt. Näin parin kuukauden työskentelyn jälkeen uudet järjestelmät eivät kuitenkaan aiheuta pelkoa ja paniikkia niin kuin ne aiheuttivat alkuvaiheessa. Silloin jokainen uusi asia ahdisti, koska kaikki vaikutti todella vaikealta ja monimutkaiselta. Neljän kuukauden perehdytys kuitenkin auttoi paljon ja nyt uusi asia ei aiheuttanut samanlaisia tunteita. Opeteltavaa oli kuitenkin todella paljon kuten aina uuden asian työstämisessä. Näin viikon jälkeen löytyy vahva ymmärrys siitä, kuinka tämän tyyppinen järjestelmä laitetaan pystyyn ja miten sitä voi käyttää testaamiseen. Ohjelma on todella käytännöllinen ja laaja, joten sen käyttöönotto oli fiksua.

Työssä meidän ryhmämme käyttää Extreme Programming -menetelmää, joka on ketterän kehityksen metodologia. Kuten Agile Alliance kertoo sivuillaan (2019), Extreme Programming menetelmä tähtää tuottamaan korkeamman laatuista ohjelmaa sekä parempia elämäntapoja tekijöille. Don Wells kuvailee hyvin tekstissään Extreme Programming: A gentle introduction (2013) missä vaiheessa kyseistä menetelmää voi käyttää. Hän kirjoittaa, että menetelmä sopii hyvin, jos ohjelmiston vaatimukset muuttuvat dynaamisesti, työssä on aikataulun aiheuttamat riskit, ohjelmistoa tekee pieni tiimi ja jos työkalut mahdollistavat automatisoinnin. Meillä ei aikataulut aiheuta mitään riskejä ja vaatimukset ovat tällä hetkellä aika suuntaa antavia, mutta meillä on kyllä pieni tiimi ja automatisointia tukevat työkalut, joten kyseinen menetelmä sopii meille vallan mainiosti. Koko perehdytyksen ajan olemme työstäneet paria eri projektia ja ensimmäinen aloitettiin niin, että päivän alussa pidimme pienen kokouksen, jonka jälkeen jokainen työsti omaa osiota itse. Tämä tapa oli hankalaa, kun kenelläkään ei ollut oikein osaamista käytössä olleista työkaluista. Jossain vaiheessa siirryimme pariohjelmoinnin kautta ryhmäohjelmointiin ja asiat alkoivat sujumaan paljon helpommin. Alussa pelkäsimme, että viisi ihmistä saman asian äärellä on turhan paljon ja se hidastaa etenemistä, mutta tosin kävi. Me valitsimme sellaisen tavan, jossa yhden tiimiläisen tietokone on niin sanottu pääkone ja on kiinni televisiossa/videoprojektorissa. Näin muut neljä seuraavat vierestä, sekä antavat vinkkejä ja ideoita. Kuvassa 2 näkee kuinka meidän paikkamme ovat jaettu, jotta saamme parhaimman tuloksen.



Kuva 2. Extreme Programming

Monesti projektien aikana meillä on ollut tilanteita, kun olemme keksinyt ratkaisut viiden ihmisen yhteisvoimin. Viidestä esimerkiksi yksi voi jatkuvasti etsiä tarvittavia komentoja internetistä, kun pääkoneen käyttäjä kirjoittaa ohjelmaa. Näin luominen ei pysähdy usein ja luominen on jatkuvaa.

Extreme Programming (XP) menetelmästä saa hyvää lisätietoa chromatic:n kirjasta Extreme Programming Pocket Guide (2009), jonka saa niin fyysisenä kuin sähköisenä. Siinä kirjailija auttaa lukijaa ymmärtämään XP menetelmän hyödyt ja milloin sitä voi käyttää. Koska kyseinen kirja on taskuversio, se on kompakti mutta hyödyllinen ohjelmoijan kaveri.

Viikon isoin asia oli uuden työkalun käyttöönotto, mikä on aina työläs ja pitkäkestoinen prosessi, etenkin jos työkalu on sellainen, mitä kukaan ei ole sen enempää käyttänyt. Tämä viikko on mennyt täysin sen tutkimiseen ja pystyttämiseen, seuraava viikko taas sisältää järjestelmän testejä. Suurempia ongelmia ei tällä kertaa tullut eteen, enemmänkin meillä oli tietämättömyyden virheitä, joita me loppujen lopuksi korjattiin yhteistyöllä.

3.2 Seurantaviikko 2

Maanantai 25.02.2019

Tällä viikolla meidän tiimin tarkoitus on jatkaa testien parissa. Itse pidän todella paljon Robot Frameworkin käytöstä ja käytän sitä mielellään, joten mielellään otin omaksi tehtäväksi testien teon. Nyt me jaettiin vähän tehtävät ja jokaisella oli oma tehtävä. RF on todella kätevä framework, koska se on kirjoitettu selkokielellä ja siinä on helppo päästä alkuun, mutta se muuttuu helposti vaikeaksi. Tämän päivän aikana tarkoituksena on saada pari pienempää testiä, jotta saamme sitten myöhemmin tehtyä laajoja end-to-end testejä.

Heti alkuun testasin, että kaikki vieläkin toimii ja testejä on mahdollista kirjoittaa. Ympäristö helposti saattaa korruptoitua, jos esimerkiksi tietokone suljetaan viikonlopun ajaksi. Järjestelmä kuitenkin toimii toivotulla tavalla ja testien kirjoittaminen oli mahdollista. Aluksi kirjoitin pienen testin, joka testasi admin-sivun kirjautumisen. Siinä testi avaa admin sivun, täyttää tarvittavat kirjautumistiedot ja yrittää kirjautua. Jos tiedot ovat oikein, järjestelmä antaa kirjautua, jos tiedot ovat väärin, testi ei mene läpi. Kun tämä testi on valmis, tein asiakaskirjautumistestin. Testi toimii samalla periaatteella kuin admin-sivun kirjautumistesti. Testissä avataan Magenton etusivu ja yritetään kirjautua asiakastiedoilla. Näillä testeillä testataan, että sivun perustoiminnot toimivat, jotta isompia testejä voidaan kirjoittaa.

Tiistai 26.02.2019

Tämän päivän tehtävänä on kirjoittaa lisää pieniä testejä ja vähän myös suunnitella laajemmat testit, jotka testaisivat koko järjestelmää. Isoimmista testeistä on tärkeätä, että on suunnitelma, muuten niiden kirjoittaminen on turhan hankalaa. Pienemmät testit toimivat hyvin isompien pohjana, koska niitä voi sitten hyödyntää isommissa testeissä. Nyt on kuitenkin tavoitteena saada vielä testattua pienillä testeillä järjestelmää ja suunnitella huomisen testiä.

Ensimmäinen pieni testi minkä kirjoitin testaa sivun hakutoimintoa. Siinä testi avaa Magento-sivun, kirjoittaa etsittävän tuotteen nimen hakukenttään ja sivun pitäisi näyttää haettava tuote. Toinen lyhyt testi testasi tuotteen lisäystä ostoskoriin tuotesivulta. Tämän testin aikana selvisi, että sivu toimii välillä sen verran hitaasti, että testi kaatuu

siihen, kun painettavia nappeja ei vielä sivulla ole. Tämä ongelma on yllättävän yleinen, kun sivu on iso laajalla tietokannalla ja testit pyörivät liian nopeasti suhteessa sivun latausaikaan. Tämä kuitenkin tarkoittaa sitä, että laajasta testistä tulee pitkä, kun sivu ei aina ehdi ladata ajoissa.

Keskiviikko 27.02.2019

Tänään on tehtävänä kirjoittaa laaja end-to-end testi. End-to-end testi tarkoittaa selaista testiä, joka testaa järjestelmää päästä päähän. Siinä testitapaus alkaa selaimen etusivun avaamisella ja loppuu tilauksen olemassaolon tarkistukseen admin-sivulla. Sellaisella testillä pyritään jäljentämään asiakkaan koko polku, jotta nähdään ovatko osat toisistaan riippuvaisia vai ei. Laaja testi on kuitenkin haastava, koska sen suorittaminen kestää kauan ja sen takia virheiden todentaminen, sekä niiden korjaus on pitkäkestoinen prosessi.

Tänään kirjoitettu osa sisälsi testin etusivulta asiakkaan tilausvahvistukseen. Laajoja testejä kirjoittaessa on tärkeää, että testi voidaan aina suorittaa loppuun onnistuneesti, vaikka se on vielä kesken. Näin sitä voidaan aina jatkaa pidemmälle myöhemmin, jos sellaiselle on tarvetta. Yhden testin kirjoittaminen ei välttämättä kuullosta isolta tehtävältä, mutta jos sivu on kirjoitettu vähänkin epäloogisesti, testin kirjoittaminen voi viedä aika paljon aikaa. Magento-sivu on kaikin puolin hyvä järjestelmä, mutta siinäkin on tiettyjä epäloogisia asioita, minkä takia testin kirjoittaminen vie aikaa. Tämän kaiken lisäksi sivun hitaus vaikuttaa todella paljon työn nopeuteen. Magento on ollut hidas jopa lyhyessä testissä, laajassa sitten testit epäonnistuivat puolet ajasta hitauden takia ja koska jokainen muutos testissä on testattava, siihen meni paljon aikaa. Päivän päätteeksi oli kuitenkin käsillä valmis laaja testi, joka alkoi siitä, että asiakas avaa etusivun ja loppui siihen, että asiakas saa tilausvahvistuksen järjestelmästä. Tämä sisälsi tuotteen etsimisen, kirjautumisen järjestelmään ja maksamisen, asiat mitkä testattiin pienemmillä testeillä aikaisemmin viikolla. Tässä projektissa sovelletaan pitkälti aikaisempi osaaminen, joten tällä viikolla ei mitään uutta asiaa tule.

Torstai 28.02.2019

Tämän päivän agendana on kirjoittaa laaja testi loppuun. Eilisen päivän tuloksena testi testaa tilauksen teon alusta loppuun, mutta tänään on tarkoitus saada varmistus admin-sivun kautta, että kyseinen tilaus löytyy sieltä, mutta kuitenkin ilman tietokanto-

jen käyttöä. Yleensä tietokantatarkistus on helpoin, mutta aina ei ole pääsyä siihen, joten pitää keksiä muita ratkaisuja. Jonkin sortin tarkastus kuitenkin pitää aina olla, minkä takia meidän testissä tarkastus tapahtuu admin-sivulla.

Päivä meni pitkälti kokonaisuudessaan testin kirjoittamiseen. Tässä tilanteessa taas tuli eteen se, että helpolta kuulostava tehtävä vei käytännössä kokonaisen työpäivän. Suurin hidastaja tässä vaiheessa oli se, että jokaisen muutoksen jälkeen testit joutuivat ajamaan, mihin meni reilu minuutti yhtä ajoa kohtaan. Siinä vaiheessa, kun testi on laaja ja muutoksia on paljon, tulee testejä ajettua helposti monta kymmentä kertaa. Näin kävikin, kun ajoin testejä päivän aika melkein 100 kertaa. Päivän päätteeksi testi oli kuitenkin valmis alusta loppuun, missä asiakasvahvistussivun jälkeen testi siirtyi admin-sivulle tarkastamaan, että tilaus löytyy siellä. Yksi haastava osa tässä oli se, että tilausnumero generoidaan testin aikana, eikä sitä voinut kirjoittaa mihinkään vaan piti kopioida testin sisällä. Se kuitenkin loppujen lopuksi onnistui ja testistä tuli siisti ja toimiva.

Perjantai 01.03.2019

Perjantai on taas klassisesti rennompia päiviä ja tarkoituksena olisi siistiä testejä. Testien siisteys on tärkeää, jotta ne olisivat helppolukuisia ja selkeitä muillekin kuin tekijälle. Suurin osa tehtävistä hoidettiin viikon aikana, jotta perjantai olisi helpompi päivä ja viikonlopun viettoon voisi siirtyä ilman suurempia stressejä.

Päivän aikana rauhassa siistin viikon aikana tekemiäni testejä. Robot Framework kirjoitetaan käyttäen keywordeja eli avainsanoja, jotka joko löytyvät valmiiksi kirjoitetuista kirjastoista tai kirjoitetaan itse. Helposti monessa eri testissä käytetään samoja itse tehtyjä avainsanoja, joten on turhaa, jos ne on avattu monessa tiedostossa, vaan ne voisi kirjoittaa yhteen tiedostoon ja muut linkittävät siihen. Hyvä rakenne on esimerkiksi yleisten avainsanojen kirjoittaminen omiin tiedostoihin ja testikohtaiset testitiedostoihin. Näin on sitten helpompaa löytää tarvittava avainsana.

Viikkoanalyysi

Tämä viikko sisälsi pitkälti juuri sitä testausautomaatiota, josta opinnäytetyön otsikossa puhutaan. Meidän projektissa testauspuoli on kirjoitettu Robot Frameworkin avulla, käyttäen Selenium kirjastoja. RF käyttää avainsanapohjaista rakennetta testien kuvaamiseen eli Keyword-driven testing, KDT. Siinä testitapausten askeleet kuvataan avainsanoilla, jotka on kirjoitettu taulukkomaisesti allekkain. Kuvassa 3. on esimerkki RF testistä.

Testitapaus	Toiminta	Argumentti
Demo	Log	Hello world

Kirjoitettuna taulukkoon:

```
*** Test Cases ***
Demo
  Log Hello world
```

Kuva 3. RF testi (kuvakaappaus sivulta https://fi.wikipedia.org/wiki/Robot_Framework)

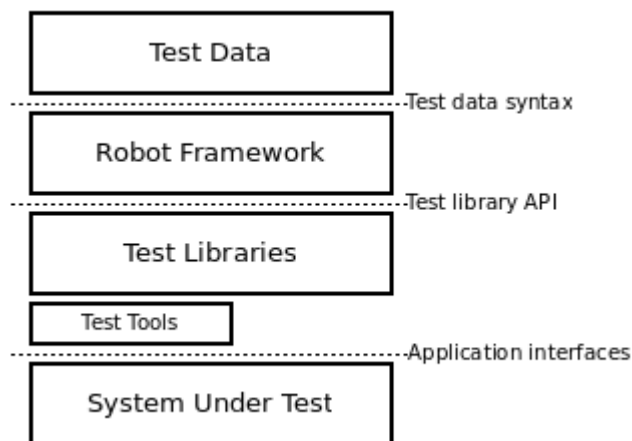
Yllä olevassa esimerkissä Log on sisäänrakennettu avainsana ja se kirjoittaa sille annetun parametrin tekstilokiin. Käyttämällä hyödyksi erilaisia RF varten tehtyjä kirjastoja, kuten esimerkiksi meidän käyttämä SeleniumLibrary, testien kirjoittaminen on yksinkertaista. Esimerkki näkyy kuvassa 4.

Testitapaus	Toiminto	Argumentti	Argument
Demo	Open Browser	https://www.google.com	ie
	Input Text	id=lst-ib	Hollywood Celebrities
	Click Button	Google Search	

Kuva 4. Yksinkertainen web-testi (kuvakaappaus sivulta https://fi.wikipedia.org/wiki/Robot_Framework)

Kuvassa 4 avataan selaimessa sivu www.google.com ja etsitään hakusanalla Hollywood Celebrities. RF testit ovat helposti kirjoitettavia ja luettavia, minkä takia niitä on helppo käyttää, sekä oppia kirjoittamaan.

Kuten Dzmitry Ihnatseyu mainitsee tekstissään Robot Framework: The Ultimate Guide to Running Your Tests (2018), RF on hyvä ratkaisu myös hyväksymistestauksen ja hyväksymisvetoiseen ohjelmistokehitykseen (Acceptance test-driven development, ATDD). Hyväksymistestaus arvioi vastaako kehitetty tuote asiakasvaatimuksia. Ajamalla hyväksymistestit, varmistamme että järjestelmä toimii kokonaisuutena ja voi olla asiakkaiden käytössä. Tähän RF on mainio työkalu, sillä avainsanoja voi käyttää läpi kaikkien testien, vaikka se ei rajoitu siihen. RF:llä on laajat valmiit kirjastot täynnä hyödyllisiä avainsanoja, mutta niitä voi kirjoittaa myös itse. Testien kirjoittamiseen voi myös hyödyntää esimerkiksi Javaa ja Pythonia.



Kuva 5. RF arkkitehtuuri (Stresnjak, 2011)

Kuvassa 5 on esitetty RF arkkitehtuuri, kuvakaappaus sivulta www.robotframework.org. Kun RF käynnistetään, se prosessoi testi dataa, käy läpi testitapaukset ja luo lokit, sekä raportit. Framework ei tiedä mitään järjestelmän sisällystä (System Under Test) ja kaikki vastoinkäymiset hoidetaan testikirjastojen avulla.

Itse pidän kovasti Robot Frameworkin käytöstä ja testausautomaatiosta. Haluaisin tehdä asiakkaalla juuri testausautomaatiota, kun lähtö asiakasprojektiin on ajankohdainen. Pidän siitä kuinka selkeä ja helppolukuinen tämä framework on, ja sillä on helppo kirjoittaa automaatiotestejä. Tällä viikolla ei tullut uusia asioita opiskeltavaksi, mutta RF on niin laaja alue, että sitä voi opiskella aika pitkään ja hartaasti. Olen viikon päätteeksi tyytyväinen omaan panokseen ja viikko loppuu hyvin mielin.

3.3 Seurantaviikko 3

Maanantai 04.03.2019

Kolmas viikko kyseistä projektia edessä ja tämän viikon agendana muuttaa projekti käyttämään Kubernetes-tekniologiaa. Kubernetes ohjaa automaattisesti konttien luomista ja käynnistää ne ongelman tullen uudestaan. Viikko alkaa vähän flunssaisena, joten tämä päivä menee etänä. Meidän yrityksessä on mahdollisuus pitää etäpäiviä, mikä auttaa tosi paljon, jos on heikompi tai väsyneempi olo kuin normaalista. Pitämällä etäpäivät voi välttää työmatkat ja samalla työskennellä omalta sohvalta. Pidän kuitenkin yhteyttä koko päivän ryhmäläisiini, jotta pysyn perillä siitä, mitä he tekevät toimistolla.

Itselläni päivä meni suurimmalta osin opiskelun parissa. Viikon tavoitteena Kubernetes-tekniologian integroinnin lisäksi viedä Magento-järjestelmä Jenkins-palvelimelle automatisointia varten. Jenkins on avoimen lähdekoodin palvelin, joka on kirjoitettu Javalla. Jenkins auttaa automatisoimaan ohjelmointiin liittyvät prosessit, jotka eivät vaadi jatkuvaa ihmisen osallistumista. Siinä esimerkiksi muutosten jälkeen koko ohjelma voidaan pystyttää automaattisesti alusta loppuun, jopa ajaa ohjelman testit samalla. Se auttaa automaattisesti testaamaan ohjelman toiminnallisuudet muutosten ja erilaisten korjausten jälkeen.

Tiistai 05.03.2019

Päivän aikana on tarkoitus viedä Magento-järjestelmä Jenkins-palvelimelle, jotta testien ajo olisi automatisoitua. Jenkinsiä meidän tiimi on käyttänyt aikaisemmassa projektissa, joten periaate on tuttu kaikille. Tavoitteena on saada mahdollisuus automatisoidusti ajaa projekti niin, että ohjelma itse automaattisesti laittaisi järjestelmän pystyyn. Tämä auttaa testaamaan järjestelmää, kun siihen on tehty jotkut muutokset ja saamaan laajat lokit, joita Jenkins tarjoaa. Tällä tavalla Jenkinsin avulla on helpompaa huomata virheet ja korjata ne.

Jenkinsin pystyttäminen ei itsessään aiheuttanut suurempia ongelmia, mutta Magenton pystyttäminen ja testien suorittaminen Jenkinsissä vei turhan pitkään. Kuvake minkä me saimme luotua aikaisemmin, on liian iso ja hidas kehitysprosessia varten, minkä takia järjestelmän pystyttäminen Jenkinsissä vie reilut 30 minuuttia, vaikka jär-

jestelmä ei ole erikoisen raskas. Samalla testeissä menee reilusti enemmän aikaa kuin ilman Jenkinsiä, mikä on ongelma parilla testillä ja tulisi olemaan sietämätöntä isommalla määrällä testejä. Vaikka päivän tavoite onkin saavutettu, aikaongelma on ratkaiseva ja jatkuva automaattinen pystyttäminen on turhaa, pitkän pystytysajan takia.

Keskiviikko 06.03.2019

Edellisenä päivänä käyttöön otettu kuvake on liian raskas Jenkinsiä varten ja testeissä menee liian pitkään, joten päivän aikana on tarkoitus saada parempi kuvake Magentosta. Kehitysympäristössä järjestelmän pystyttämisen pitäisi olla nopea asia, puolen tunnin pystyttäminen paria testiä varten ei ole ollenkaan toimiva vaihtoehto. Jos saamme päivän aikana kehitettyä parempi kuvake, testien ajaminen nopeutuu reilusti ja kehittäminen etenee paremmin.

Päivän aikana saimme luotua uuden kuvakkeen Magentosta, joka auttoi pudottamaan ajon keston puolesta tunnista noin viiteen minuuttiin. Tämä mahdollisti Jenkinsin tehokkaamman käytön, koska nyt kaikki muutokset pystyimme tarkistamaan nopeasti ja helposti. Tämä myös teki Jenkinsistä taas hyödyllisen, koska puolen tunnin ajojen aika ei oikein voinut tehdä mitään muuta, joten työ pysähtyi siksi aikaa. Nyt kun ajoaika on viisi minuuttia, tauko on todella pieni, eikä työ pysähdy.

Torstai 07.03.2019

Tänään yritämme integroida vihdoinkin koko projektin Kubernetesiin. Tiimiläinen on työstänyt Kubernetes-konfiguraatiota viikon alusta ja päivän tehtävänä on saada se toimimaan osana kokonaisuutta. Kubernetes-työkalun avulla konteinereiden kanssa työskentely ja sen automatisointi on helpompaa. Kubernetes on hyvä virtualisointia varten ja siinä jokainen palvelu pyörii omassa kontissa. Meidän projektissa esimerkiksi Magento-järjestelmä pyörii yhdessä kontissa, testit toisessa ja videopalvelu kolmannessa. Näin esimerkiksi, jos yksi konteista sulkeutuu, se käynnistää itsensä uudestaan ja järjestelmän toiminta ei lakkaa. Saman Kubernetes tekee, jos tietokone esimerkiksi sammutetaan ja käynnistetään uudestaan.

Tiimiläinen on tehnyt hyvää työtä viikon aikana ja konfiguraatio on hyvä, joten projektin vieminen Kubernetesiin ei tuota ongelmia. Olemme käyttäneet Kubernetesia ai-

kaisemmassa projektissa, joten tietämys asiasta löytyi kaikilta tiimin jäseniltä. Työkälun avulla voimme nyt jatkossa ajaa testejä samasta järjestelmästä monta päällekkäin samaan aikaan. Kubernetesille on myös luotu työpöytä, jonka kautta pääsee käsiksi kaikkiin konteinereihin ja voi tarkastella esimerkiksi testien kulkua.

Perjantai 08.03.2019

Perjantai on taas viimeistelyyn suunniteltu päivä, mutta tällä kertaa meillä on myös demo. Yrityksessä perjantaisin on demohetki, jolloin tiimit voivat demota mitä ovat tehneet. Demojen avulla harjoitellaan projektin esittämistä yleisölle ja samalla voi saada tärkeätä palautetta tai vinkkejä työkavereilta. Meidän tiimi ei ole tähän mennessä pitänyt kuin yhden demon, silloin kun saimme tehtyä edellisen projektin valmiiksi, mutta nyt eteneminen on ollut nopeampaa, joten on jotain näytettävää viikon välein.

Aamupäivän aikana viimeistelimme projektiamme, jotta demon aikana kaikki toimii niin kuin pitää. Demo on yleensä vapaamuotoinen ja sen pitää joko koko tiimi tai yksi tiimiläinen. Meidän tiimi ei ole pitänyt yhtäkään demoa vielä tästä Magento-projektista, joten meillä oli hyvin näytettävää. Lounaan jälkeen meidän tiimin lisäksi pari muuta tiimiä pitivät demonsa ja loppupäivä meni kaikilla korjauksien ja viimeistelyjen merkeissä.

Viikkoanalyysi

Viikon isompana teemana on ollut DevOpsin käyttö jatkuvassa kehittämisessä. DevOps on toimintamalli, jossa ohjelmistojen kehitys, testaus ja julkaisu tapahtuvat nopeasti ja automatisoidusti. Tässä projektissa meidän tiimi käyttää jatkuvaa integraatiota, sekä testauksen automatisointia ja automatisoitua konfigurointia. Jatkuva integraatio ja testauksen automatisointi antaa mahdollisuuden testata järjestelmää jatkuvasti pienenkin muutoksen jälkeen. Kuten Ilkka Toivanen mainitsee blogissaan Devops jatkuvan kehittämisen tukena (2017), tavoitteena on rakentaa toimintamalli, missä kehitettävä ohjelmisto on jatkuvasti mahdollisimman kattavasti testattu ja sen konfiguraatiot hallinnassa siten, että ohjelmisto on milloin tahansa lähes reaaliaikaisesti julkaitavissa. Koko prosessi tapahtuu ketterästi ja on automatisoitu.

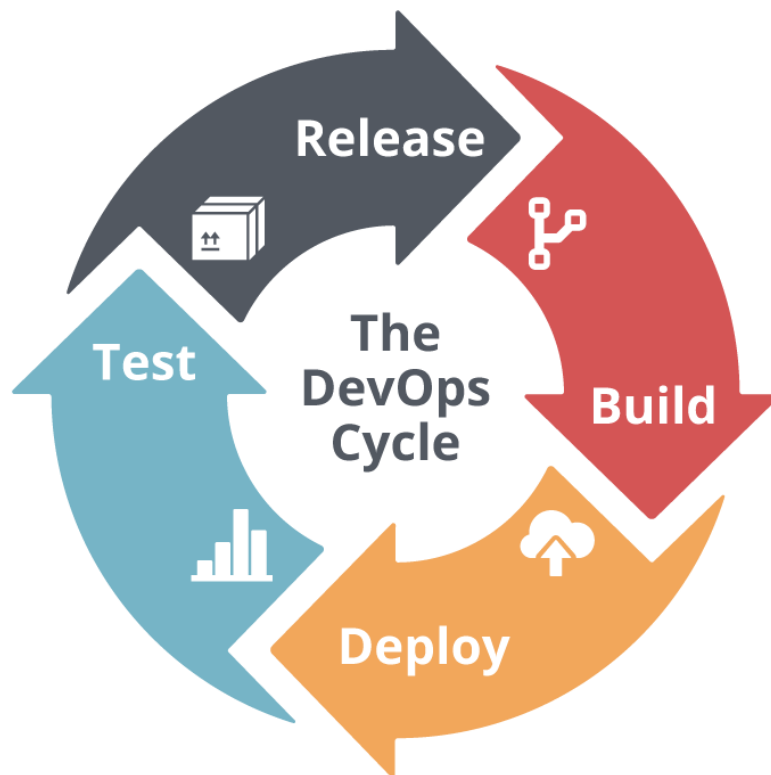
Jenkinsin avulla meidän projektissa me olemme automatisoineet testauksen. Jenkins konfiguraatio näyttää suurin piirtein samalta kuin kuvassa 6.



Kuva 6. Jenkins (kuvakaappaus: <https://sdk.cn>)

Kuvassa 6 näkyy esimerkki siitä miltä Jenkins-konfiguraatio näyttää. Vaakatasossa olevat erilaiset build-vaiheet, missä yksi build-vaihe on järjestelmän pystyttäminen alusta loppuun. Pystytasossa ovat yhden build-vaiheen eri alavaiheet, esimerkiksi pystyttäminen, testaus ja järjestelmän putsaus. Vihreällä värillä järjestelmä merkitsee onnistuneet vaiheet ja punaisella värillä merkitsee epäonnistuneet. Jokaisessa vaiheessa näkee kuinka kauan vaiheen suorittaminen kesti ja lokeista voi lukea, mitä vaiheessa tapahtui. Tämän avulla järjestelmän pystyttämistä voi seurata vaihe vaiheelta ja heti nähdä, mikä vaihe epäonnistuu ja miksi.

Automatisointi ja virtualisointi vähentävät manuaalista, puuduttavaa ja toistuvaa työtä, sekä pienentävät rutiinityön aiheuttamien virheiden määrän. DevOps tehostaa ja nopeuttaa uusien versioiden julkaisua ja pienentää kehitys- ja ylläpitokustannuksia. Meidän työssä DevOps on oleellinen osa, koska tarkoitus on automatisoida työtä, helpottaa asiakkaan arkea ja vähentää kustannuksia. Kuvassa 7 näkee kuinka DevOpsia voi hyödyntää projektin eri vaiheilla, kuten testauksessa tai julkaisussa.



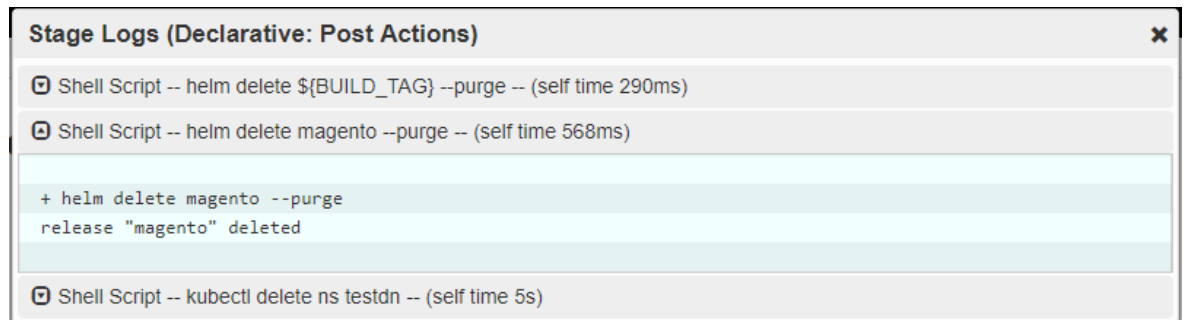
Kuva 7. DevOps -malli (Bocetta, 2019)

DevOps sisältää monia toimintamalleja, jotka sopivat kaikkiin kehityshankkeisiin ja se on riskeistä huolimatta suositeltava, sekä toimiva toimintamalli moderniin ohjelmisto-kehitykseen (Toivanen, 2017).

3.4 Seurantaviikko 4

Maanantai 11.03.2019

Testeissä tärkein ominaisuus on lokitietojen saaminen, jotta virheiden analysointi olisi helpompaa. Tällä viikolla me yritämme saada lokitiedostot Jenkins-järjestelmästä kopiaitua Minio-järjestelmään. Minio on startup teknologia ja sopii hyvin esimerkiksi juuri lokitiedostojen säilöntään. Me haluamme saada lokitiedostot ulos, jotta virheiden tul- len, voimme lähettää tarkemmat tiedot ongelmista kehittäjille korjausta varten. Lokit näkyvät myös Jenkinsissä, mutta Minion tapaisessa pilvipalvelussa ne säilyvät pa- remmin ja on helpompaa tallentaa tai jakaa.



The screenshot shows a Jenkins 'Stage Logs (Declarative: Post Actions)' window. It contains three entries, each with a status icon (a square with a checkmark) and a self-time measurement. The second entry is expanded to show the command output.

```
Stage Logs (Declarative: Post Actions) x
[+] Shell Script -- helm delete ${BUILD_TAG} --purge -- (self time 290ms)
[+] Shell Script -- helm delete magento --purge -- (self time 568ms)
+ helm delete magento --purge
  release "magento" deleted
[+] Shell Script -- kubectl delete ns testdn -- (self time 5s)
```

Kuva 8. Jenkins-lokit

Jenkins-järjestelmästä saa myös lokit ulos, mutta niissä pitää tarkastella vaihe kerrallaan, eivätkä ne ole helppoja ulkopuoliselle luettavaksi. Kuvassa 8 näkee esimerkin Jenkins-lokista, testin viimeisestä vaiheesta, missä testissä tehdyt muutokset poistetaan. Tämä ei kuitenkaan ole ulkopuoliselle niin selkeälukuista, eikä esimerkiksi testeistä saa ulos muuta kuin sen, että testit ovat suoritettu onnistuneesti tai epäonnistuneesti. Tämän takia yritämme saada Minion toimimaan. Päivä meni tietojen etsimiseen ja aiheen tutkimiseen, koska meillä ei varsinaisesti ole edellistä kokemusta Miniota.

Tiistai 12.03.2019

Tämän päivän aikana yritämme saada testien lokit menemään Minio-järjestelmään. Testien lokit ovat korjauksia varten tärkeitä, kuitenkin niin kauan kuin testit ovat kirjoitettu selkeästi ja hyvin.

TEST Find Product Test

Full Name: Magento Tests.Tests.03 Find Product Test.Find Product Test
Start / End / Elapsed: 20190326 14:33:40.374 / 20190326 14:34:03.002 / 00:00:22.628
Status: **PASS** (critical)

- + **SETUP** zalenium_resource . Setup Test Rig
- + **KEYWORD** resource . Find Product
- + **KEYWORD** Insert Product Data
- + **TEARDOWN** zalenium_resource . Teardown Test Rig

Kuva 9. Onnistuneen testin lokit

[-] **TEST** Buy Product

Full Name: Magento Tests.Tests.04 Buy Product Test.Buy Product
Start / End / Elapsed: 20190326 11:52:35.054 / 20190326 12:56:04.753 / 01:03:29.699
Status: **FAIL** (critical)
Message: Setup failed:
WebDriverException: Message: <html>
<head><title>504 Gateway Time-out</title></head>
<body bgcolor="white">
<center><h1>504 Gateway Time-out</h1></center>
<hr><center>nginx/1.10.3 (Ubuntu)</center>
</body>
</html>

[-] **SETUP** zalenium_resource . Setup Test Rig

Start / End / Elapsed: 20190326 11:52:35.055 / 20190326 12:56:03.731 / 01:03:28.676

- + **KEYWORD** SeleniumLibrary . Set Selenium Timeout 20 seconds
- + **KEYWORD** zalenium_resource . Add Test Name To Zalenium Video Name
- [-] **KEYWORD** SeleniumLibrary . Open Browser \${url}, \${browser}, None, \${gridurl}, desired_capabilities=\${captureName}

Documentation: Opens a new browser instance to the given url.
Start / End / Elapsed: 20190326 11:52:35.065 / 20190326 12:56:03.693 / 01:03:28.628

- + **KEYWORD** SeleniumLibrary . Capture Page Screenshot

11:52:35.066 **INFO** Opening browser 'chrome' to base url 'http://magento2.testdn/' through remote server at 'http://zalenium:4444/wd/hub'.
12:56:03.692 **FAIL** WebDriverException: Message: <html>
<head><title>504 Gateway Time-out</title></head>
<body bgcolor="white">
<center><h1>504 Gateway Time-out</h1></center>
<hr><center>nginx/1.10.3 (Ubuntu)</center>
</body>
</html>

- + **TEARDOWN** zalenium_resource . Teardown Test Rig

Kuva 10. Epäonnistuneen testin lokit

Kuvassa 9 näkyy esimerkki testilokista, jossa testi etsii Magento-sivulla tiettyä tuotetta. Vihreä väri lokissa merkitsee, että testi on onnistunut ja kaikki vaiheet menivät läpi. Tästä sitten jopa ulkopuolinen voi seurata vaihe vaiheelta mitä testissä tapahtuu. Jos testin jokin osa epäonnistuu, väri muuttuu punaiseksi kuten kuvassa 10. Siinä vaiheessa sitten on tärkeää, että lokit ovat saatavilla ja niistä helposti näkee mitä on tapahtunut. Järjestelmä ottaa myös screenshotin virheestä, jotta sitä katsomalla tietää missä vaiheessa testi meni rikki.

Päivän aikana saimme Minion toimimaan ja kaikkien testien lokitiedostot sinne tallentamaan. Minio on kätevä järjestelmä, jossa eri testiajojen lokit tallentuvat eri omiin

kansioihin, niin kuin myös kuvakaappaukset virheistä. Nämä tiedoston on sitten helppoa ladata pakettina omalle tietokoneelle ja lähettää eteenpäin tai tallentaa.

Keskiviikko 13.03.2019

Sen jälkeen, kun Minio on saatu toimivaksi, on aika siirtyä rajapintatesteihin. Rajapintatestaus on osa integraatiotestausta, jossa testataan useiden komponenttien yhteistoimintaa ja löytää virheitä, jotka eivät tulleet esiin yksikkötestissä. Siinä pyritään todentamaan rajapintoja komponenttien ja ohjelmistosuunnittelun välillä. Tätä varten seuraavien päivien aikana yritämme saada luotua rajapinta- eli API-testit luotua Magentolle. Minun on tarkoitus tehdä GET-lausekkeen avulla Flask-työkalua käyttäen yksinkertainen sivu, jossa näkee kaikki Magentossa olevat asiakastilaukset.

Itselleni Flask ei ole erityisen tuttu järjestelmä, olen käyttänyt sitä vähän vain yhdessä projektissa aikaisemmin. Flask-työkalun avulla voi luoda ja pystyttää helposti web-ohjelmia ja se käyttää html-pohjaisia tiedostoja. Päivän aikana oli tärkeätä saada Flask ylipäättään toimimaan, jotta sivua pääsee tarkastelemaan myös toisen tiimiläisen tietokoneelta. Flask onneksi sisältää yksinkertaisen palvelimen ja avoimen sivun teko ei ole erikoisen monimutkainen.

Torstai 14.03.2019

Tänään on tarkoitus jatkaa eilen kesken jäänyttä Flask-aplikaatiota käyttäen GET-funktiota. Sivun tarkoitus on tulostaa kaikki tilaukset, jotka on Magenton kautta tehty rajapintojen avulla, jotta testi voi sieltä tarkistaa, onko tilaus mennyt läpi ja löytyykö se tietokannasta. Sivuston ulkonäkö on mahdollisimman yksinkertainen, koska tässä tapauksessa se on kehittäjien sisäiseen käyttöön ja sillä ei ole varsinaisesti väliä, miltä sivusto näyttää, kunhan se tekee sen minkä pitää.

Päivän päätteeksi sain tehtyä toimivan Flask-sivuston, johon jopa sai yhteyden toisesta tietokoneesta, joten sitä pystyi käyttämään testauksessa. Sivusto sisältää tilauksien perustiedot kuten tilausnumeron, tilaajan tiedot ja tilauksen status. Otin tämän tehtävän itselleni koska meidän tiimistä ymmärrän web-ohjelmoinnista vähän muita enemmän ja pidän siitä, vaikka en esimerkiksi Flaskia ole erikoisemmin käyttänyt.

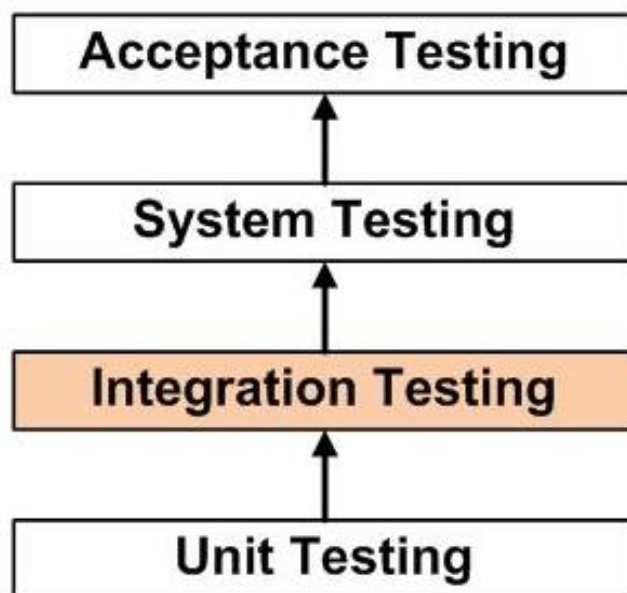
Perjantai 15.03.2019

Perjantai on klassisesti demopäivä ja tällä kertaa meillä on myös demo. Saimme tämän viikon aikana riittävästi valmiiksi, jotta voimme jakaa muille opitut asiat. Nykyään meillä on demoja joka viikko, kun on monta tiimiä toimiston sisäisissä projekteissa. Demon lisäksi siistimme myös tehdyt testit ja applikaatit, jotta meillä on taas käyttövalmis kokonaisuus.

Päivä oli taas vähän rennompi ja lyhyempi. Tämän viikon demon veti toinen tiimiläinen, tarkoitus on saada jokainen vetämään ainakin yhden, jotta kaikki saamme edes vähän kokemusta asiasta. Tein myös viimeistelyjä Flask-applikaatioon pienien HTML- ja css-muutoksien myötä.

Viikkoanalyysi.

Tämä viikko vahvasti painottui rajapintatestaukseen, joka on osa integraatiotestausta. Integraatiotestaus yksi testauksen tasoista, jossa testataan kuinka yksittäiset osat toimivat keskenään. Usein osat voivat toimia hyvin yksinään ja yksikkötestit menevät läpi, mutta integraatiotestit sitten tuovat virheet esille.



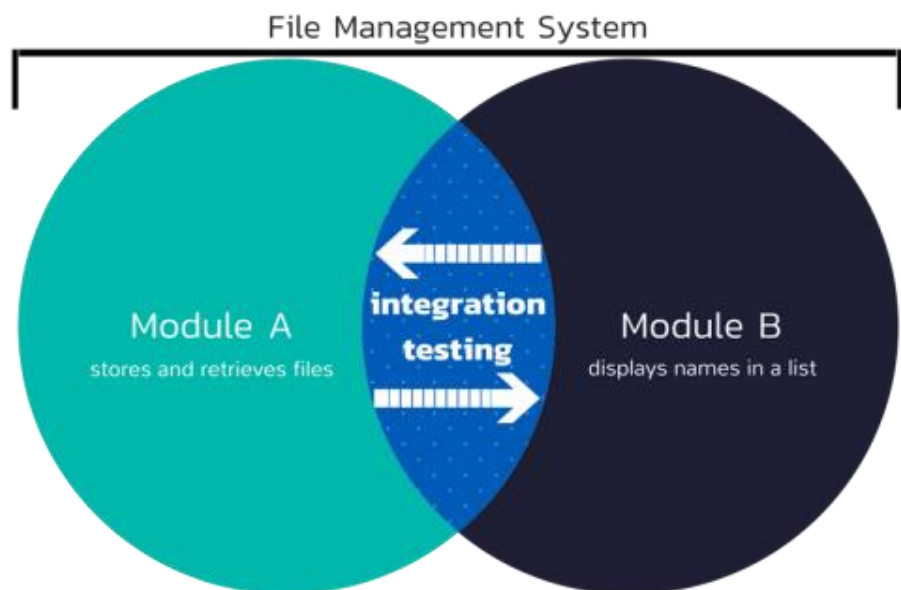
Kuva 11. Testauksen tasot (Bui, 2017)

Kuten kuvassa 11 näkyy, ensimmäiseksi testauksessa suoritetaan yksikkötestit. Niissä testataan, kuinka yksittäiset osat toimivat keskenään. Tämän jälkeen siirrytään in-

tegraatiotesteihin, joissa on tarkoitus testata kuinka osat käyttäytyvät keskenään. Software Testing Fundamentals -sivusto käyttää esimerkkinä seuraavaa mielenkiintoista tilannetta (2018). Kuulakärkikynässä kaikki osat tuotetaan ja testataan erikseen, kuten esimerkiksi kärki, runko tai korkki. Kun kaksi osaa ovat valmiita ja yksikkötestattuja, niihin suoritetaan integraatiotestit, esimerkiksi sopiiko korkki runkoon.

Integraatiotestit ovat tärkeä osa testikokonaisuutta, koska koskaan ei voi olla varma kuinka kaksi eri osaa toimivat yhdessä, vaikka niitä olisikin testattu erikseen. Guru99 kertoo sivullaan miksi kannattaa käyttää testauksessa integraatiotestausta (2019). Usein ohjelmiston eri osia luovat eri kehittäjät, joiden ajattelutapa voi olla täysin erilaiset. Tämän takia on pakko varmistaa, että kahdella eri logiikalla tuotetut osat toimivat keskenään. Tämän takia meidänkin tiimi käyttää integraatiotestausta yksikkötestauksen lisäksi.

Meidän tilanteessa, Magenton on luonut joku meille tuntematon ulkopuolinen henkilö. Me testaajina emme voi tietää, miten asioita on testattu kehitysvaiheessa ja kuka on kehittänyt minkäkin osan. Tämän takia alussa me suoritimme yksikkötestausta, jossa esimerkiksi sivustolta etsittiin tuote. Kun taas yksikkötestejä tuli tarpeeksi, me siirryimme integraatiotestaukseen. Yksi testiesimerkki on viikolla tehty Flask-sivu, jossa nähdään, ilmestyykö sinne eri kautta tehty tilaus. API-testit ovat kuitenkin vain yksi esimerkki integraatiotesteistä, vaikka se onkin todella hyvä tällaisen ohjelmiston testaukseen. Toinen tapa on aikaisemmin tehty End-to-end-testi, jossa testataan koko prosessi alusta loppuun.



Kuva 12. Moduulien välinen interaktio (www.imagenesmi.com, 2019)

Meidän Flask-sivuston Get-testi toimii kuvassa 12 esitetyllä periaatteella. Sivu, jota kuvassa esittää moduuli B, tulostaa käyttäjälle tiedot, jotka ovat tallennettu tietokantaan, joka on esitetty moduulilla A. Me taas suoritamme sinisellä alueella olevan integraatiotestin, jotta saamme selville, tulostaako sivu oikeat tiedot. Näin integraatiotesti auttaa meitä testaamaan ohjelmaa paljon tarkemmin ja syvemmin.

3.5 Seurantaviikko 5

Maanantai 18.03.2019

Uusi viikko jatkuu ensimmäiset pari päivää rajapintatesteillä. Ajatuksen mukaan rajapintatestit suoritetaan yhtenä kokonaisuutena yhdessä rf-tiedostossa. Näitä testejä varten meidän pitää luoda Python-kirjastoja, joiden avulla testit suoritetaan. Näiden rajapintatestien avulla testataan samat asiat kuin aikaisemmissa testeissä, mutta ilman käyttöliittymän käyttöä.

Testien kirjoittaminen tapahtuu yleensä vaiheittain, joten niihin saattaa mennä montaakin päivää, kunhan testit välissä toimivat toivotulla tavalla. Esimerkiksi jos testin tarkoitus on testata tilauksen teko alusta loppuun, testin voi jättää kesken kohtaan, jossa tuote lisätään ostoskoriin, kunhan testit menevät läpi. Tietenkin välillä käy myös niin että testit jäävät keskeneräisiksi ja niiden korjaaminen vie monta päivää. Tämän päivän aikana saimme aloitettua rajapintatestien teko ja luotiin osan python-kirjastoista.

Tiistai 19.03.2019

Tämän päivän tarkoitus on jatkaa eilen aloitettuja testejä. Testien tarkoitus on testata koko ostoprosessi alusta loppuun kuten end-to-end testissä. Näiden avuksi luodut python-kirjastot ovat erillisessä kansiossa ja niitä voikin hyödyntää moneen eri testiin. Rajapinta-testeissä haasteena on kirjoittaa python-kirjastot käytettäviksi, tavallisten Robot Framework -testien lisäksi. Tämän takia asia vie enemmän aikaa kuin edelliset tekemämme testit.

Tekemämme testit ovat yksi laaja API-testi, joka testaa Magento-järjestelmä rajapintoja. Tähän varsinaisesti loppuu meidän testien tekeminen Magentolle, koska puolet meidän tiimistä on siirtynyt jo asiakasprojekteihin ja meidän pieni joukko yrittää vaan viimeistellä Magentoa. Päivän päätteeksi ajettiin vielä valmiit testit kaikkien koneilla, jotta voimme olla varmoja, että testit toimivat.

Keskiviikko 20.03.2019

Nyt kun rajapintatestit ovat valmiita, alan työstämään jenkins-integraatiota kuntoon lisäämällä siihen post-commit hookin, joka toimii commitin jälkeisenä tehtävänä. Tämän tyyppisten laukaisijoiden tarkoitus on automatisoida asioita kehityksessä. Esimerkiksi jos on ohjelma, jota yksi tiimi työstää, muutosten puskeminen github -alustalle laukaisee testien ajon. Näin saadaan testit ajettua automaattisesti ilman että testaaja ajaa niitä. Tämä auttaa etenkin silloin, kun testit ovat laajat ja niiden ajamisessa kestää pitkään, niin ne voivat automaattisesti ajaa itseään yöllä. Meidän tilanteessa me käytämme post-commit hookia, mikä on commitin jälkeen tapahtuva laukaisu eli kun yksi käyttäjä vie tavaroita gitlab -alustalle, jenkins ajaa testit itse.

Lähtökohtaisesti git-commit hookia käytetään ilmoituskäytössä, koska se ei voi vaikuttaa commit-toiminnan lopputulokseen (Atlassian Bitbucket, 2019). Meidän tapauksessa me haluamme saada siitä irti enemmän kuin pelkän ilmoituksen. Tähän on onneksi olemassa Jenkinsissä asetus, jossa ajo laukaistaan post-commit hookin avulla. Suurin osa päivästä meni asiaa tutkiessa ja testaillessa, kuitenkin toimivaa versiota en ole saanut valmiiksi päivän aikana, vaan työstäminen loppuun siirtyy seuraavalle päivälle.

Torstai 21.03.2019

Päivän tarkoitus on saada hook toimimaan odotetulla tavalla. Lähtökohtaisesti kukaan tiimissä ei tiennyt, että sellainen asia on olemassa ja miten sitä käytetään, joten pohjalla on laaja selvitystyö. Kuitenkin kun asian selvittää, käytön ei pitäisi olla erikoisen vaikeata, mutta siitä saa kuitenkin ison hyödyn irti.

jenkins-test-pipeline > .git > hooks				
Name	Date modified	Type	Size	
applypatch-msg	22/01/2019 14.36	SAMPLE File	1 KB	
commit-msg	22/01/2019 14.36	SAMPLE File	1 KB	
fsmonitor-watchman	22/01/2019 14.36	SAMPLE File	4 KB	
post-commit	22/03/2019 10.02	File	1 KB	
post-update	22/01/2019 14.36	SAMPLE File	1 KB	
pre-applypatch	22/01/2019 14.36	SAMPLE File	1 KB	
pre-commit	22/01/2019 14.36	SAMPLE File	2 KB	
prepare-commit-msg	22/01/2019 14.36	SAMPLE File	2 KB	
pre-push	22/01/2019 14.36	SAMPLE File	2 KB	
pre-rebase	22/01/2019 14.36	SAMPLE File	5 KB	
pre-receive	22/01/2019 14.36	SAMPLE File	1 KB	
update	22/01/2019 14.36	SAMPLE File	4 KB	

Kuva 13. Git hooks

Jokaisessa git-kansiossa on piilotettu .git -kansio, jonka sisällä on hooks-kansio. Kansion sisällön näkee kuvassa 13, lisätty tiedosto on nimeltä post-commit. Lähtökohtaisesti kansion sisällä on vain sample-tiedostoja kaikista hookeista, jotka voi sitten muuttaa käytettäviksi tiedostoiksi. Tämän tiedoston luonnin lisäksi piti tehdä muutoksen Jenkins-ajon, jotta kyseinen hook laukaisisi ajon. Päivän päätteeksi meillä oli toimiva hooks, jonka toiminta oli kuitenkin vajanainen, koska vain minun omat muutokset laukaisivat testien ajot, eikä kaikkien tiimiläisten.

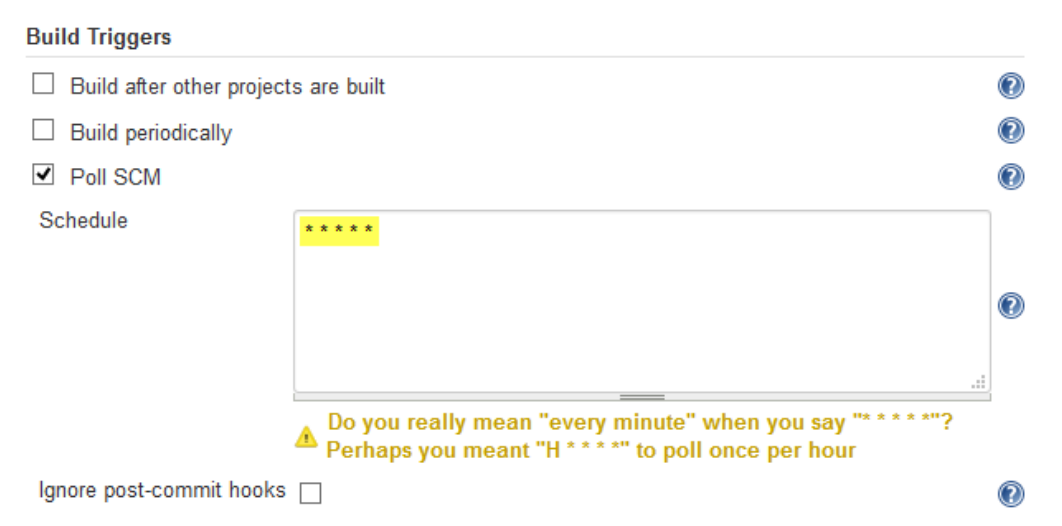
Perjantai 22.03.2019

Klassisesti perjantai on demopäivä ja meidän on tarkoitus näyttää hookien käytön, koska kukaan muuta sisäisessä projektissa oleva ei sellaisia käytä. Demojen tärkein tarkoitus onkin kertoa muille käytetyistä teknologioista ja oppia samalla uusia työkaluja. Meidän pitää nopeasti vielä testilla meidän hook, jotta se laukaisee myös muiden muutosten jälkeen testien ajot.

Aamupäivä meni toiminnon testauksessa, jotta se toimisi toivotulla tavalla demossa. Pienten korjausten jälkeen, hook laukaisi ajon myös muiden muutoksista. Lounaan jälkeisen hetken käytin PowerPoint -esityksen tekoon, koska oli minun vuoro pitää demoesitys. Demoja oli todella paljon ja niihin me loppupäivä. Meidän demossa kaikki meni niin kuin piti ja hook toimii hyvin.

Viikkoanalyysi

Versiohallinnasta on tullut keskeinen vaatimus nykyaikaiselle ohjelmistokehitykselle. Se antaa projektille mahdollisuuden seurata muutoksia, sekä palauttaa aikaisemmat versiot ja tarkistaa yhtenäisyys (Ellingwood, 2014). Gitin paras ominaisuus kehitystyössä on sen joustavuus. Git antaa kehittäjille mahdollisuuden laajentaa sen toiminnallisuutta esimerkiksi kutsumalla Git hookit. Hookit ovat skriptejä, jotka ajavat itseään automaattisesti joka kerta kun tietty asia tapahtuu Git -kansiossa (Atlassian Bitbucket, 2019). Tämä antaa mahdollisuuden muokata Gitin sisäiset toiminnot ja laukaista muokatut toiminnot tiettyinä hetkinä kehityksessä. Vaikka suurimmaksi osaksi skriptien käyttötarkoitukset ovat viestintä ja ilmoitukset, ovat täysin muokattavissa. Meidän projektissa me halusimme saada hookit toimimaan, jotta automatisointi olisi viety mahdollisimman pitkälle. Magento-projektin tarkoitus on olla automaatiotestauksen demona niin, että sitä voi käyttää esimerkkinä esimerkiksi asiakkaille. Tämän takia halusimme tutkia hooks-scriptien käytön ja hyödyt projektissa.



Kuva 14. Build Triggers-asetukset (Andyfrench.info, 2015)

Jotta hook-skripti toimisi Jenkinsin kanssa, pitää testikokonaisuuden asetuksissa tehdä pieniä muutoksia. Kuten kuvassa 14 näkyy, siellä on olemassa asetuskohta nimeltä Build Triggers, jossa voi määrittellä erilaisia ajon laukaisimia. Kuvan esimerkissä muutosten kysely (Poll) tapahtuisi minuutin välein ja vaikka testejä ei välttämättä siinä ajeta, minuutti on liian pieni aika. Siellä voi määrittää myös automaattiajon, kun muut projektit ovat valmiita tai tietyin aikavälein, vaikka ei muutoksia ole tapahtunut. Meidän tapauksessa käytetään post-commit hook-skriptiä, jota laukaisee commit git-kansioon. Kun kysely on tehty, sivulle ilmestyy lokisivu, mistä näkee, mitkä olivat kyselyn tulokset.

Jenkins

Andy.French.Configuration.Service > Git Polling Log

ENABLE AUTO REFRESH

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Git Polling Log

Build History trend ==

Git Polling Log

```
Started on 15-Mar-2015 18:16:24
Using strategy: Default
[poll] Last Built Revision: Revision
7c962b6a0c7c5f1231d3a30774ddabe4407cb4cf
(refs/remotes/origin/master)
> git.exe --version # timeout=10
> git.exe -c core.askpass=true ls-remote -h
file:///c:/Source
/Andy.French.Configuration.Service # timeout=10
[poll] Latest remote head revision on
origin/master is:
7c962b6a0c7c5f1231d3a30774ddabe4407cb4cf -
already built by 11
Done. Took 0.18 sec
No changes
```

Kuva 15. Git kyselyn lokit (Andyfrench.info, 2015)

Kuvassa 15 on esimerkki lokisivusta. Siinä näkyy, milloin kysely on aloitettu, kauan se on kestänyt ja mikä oli lopputulos. Tässä esimerkkitapauksessa muutoksia ei löytynyt, joten testejä ei ajeta.

Automaatiotestauksessa tämän tyyppiset skriptit ovat tärkeä osa kehitystä. Automatisoinnissa on tärkeää saada asiat tapahtumaan itsestään, jotta yksi testaaja voi saada mahdollisimman paljon aikaa. Vaikka tämän tyyppisten työkalujen käyttöönotto vaatii opettelemista, kun asian osaa, se säästää paljon aikaa, mikä on automaatiotestauksen pohja.

3.6 Seurantaviikko 6

Maanantai 25.03.2019

Uuden viikon mukana meitä jää projektiin vain kaksi ihmistä alkuperäisestä kuudesta ja meidätkin molemmat ollaan jo myyty asiakasprojektiin. Tämä viikko menee pienien viimeistelyjen merkeissä ja samalla odotamme, että asiakkaan tunnukset tulisivat, jotta pääsemme tutkimaan materiaaleja. Tällainen väliaika on yleensä hiljainen ja jopa tylsä, mutta ajan voi hyödyntää oman osaamisen korottamiseen. Tänään tarkoitus on käydä tulevan asiakkaan yleiset tiedot läpi ennen huomista palaveria, jotta perustieto yrityksestä olisi hallussa.

Päivä meni pitkälti yrityksen tietojen tarkastellussa. Seuraavat pari viikkoa tulevat olemaan hiljaisia, koska ennen kuin voi aloittaa projektin, kaikki luvat on saatava kuntoon ja se vie oman aikansa. Usein konsultilla menee viikkoja ennen kuin varsinainen työ voi alkaa, koska lupien ja tunnusten hankkiminen vie paljon aikaa.

Tiistai 26.03.2019

Tänään meillä on kollegan kanssa palaveri asiakkaalla toimivien konsulttien kanssa. Palaverin tavoitteena on käydä läpi projekti missä me tulemme työskentelemään ja näyttää käytössä olevat työkalut. Aina ei tällaisia perehdytyspalavereja ole mahdollista pitää, mutta meillä onneksi on tilanne sen verran hyvä, että siihen on aikaa ja resursseja.

Aamupäivästä meillä oli palaveri projektissa pitempään olleiden asiantuntijoiden kanssa, joka kesti noin lounaaseen saakka. Siitä me kävimme läpi, millaisilla tekniikoilla projekti on toteutettu ja mitä työkaluja käyttäen. Kävimme myös läpi vähän asiakasyrityksen yleiset käytännöt ja tilasimme meille uudet tunnukset. Loppupäivänä toinen vanhemmista konsulteista näytti meille mitä hän työstää tällä hetkellä, jotta saisimme kuvan siitä mitä meitä odottaa.

Keskiviikko 27.03.2019/ Torstai 28.03.2019/ Perjantai 29.03.2019

Nyt kun meidän edellinen projekti loppui ja uutta emme vielä voi työstää, päätimme pitää loppuviikon etänä ja kehittää omaa osaamista. Asiakasprojektin ulkopuolella konsultilla voi helposti olla tällaisia viikkoja, kun ei oikein ole mitään tarkkaa työtä, se on yksi työn huonoista puolista. Tätä aikaa voi kuitenkin käyttää oman osaamisen kehittämiseen, mitä me teemmekin loppuviikon.

Kuten olikin suunnitelmissa, loppuviikko meni jokaisella omassa tahdissa opiskellessa.

Viikkoanalyysi.

Meidän yrityksemme on konsulttipalveluita tarjoava yritys, joten suurin osa sen työntekijöistä ovat it-konsultteja. Suomessa on jonkun laskelman mukaan yli 11.000 erilaista konsulttia, kouluttajaa, asiantuntijaa tai muuta konsultiksi laskettavaa ammattilaista (Tolvanen, 2013). Konsultin toimenkuva on tarjota yrityksille asiantuntijapalveluita, meidän tapauksessa laadunvarmistukseen liittyvää. Miksi sitten monet asiakasyritykset käyttävät konsultteja?

Konsultti on ulkopuolinen asiantuntija, jolla on valmiiksi koulutus työn suorittamiseen. Se antaa yritykselle mahdollisuuden säästää aikaa ja resursseja, jotka olisi pitänyt käyttää asiantuntija palkkaamiseen tai kouluttamiseen. Konsultti osaa ehdottaa asiakasta hyödyntäviä ratkaisuja esim. asiakkaan kanssa käytyjen keskustelujen perusteella tai perustuen omiin havaintoihinsa asiakkaan ympäristössä (Ekholm, 2017). Tämän lisäksi konsulttipalveluita tarjoavalla yrityksellä on useampi asiantuntija, joten jos projektissa työskentelevä henkilö on estynyt hoitamasta tehtäviään, korvaava asiantuntija voidaan saada tilalle. Näin työ ei pysähdy, jos konsultti joutuu esimerkiksi pitkälle sairauslomalle, vaan hänen tilalle saadaan korvaaja.

Kaikki eivät kuitenkaan haluavat käyttää konsulttipalveluita. Konsultointia kritisoivat eniten he, jotka eivät itse palveluita osta tai hyödynnä (Tolvanen, 2013). Monelle on kuitenkin jäänyt huono mielikuva konsulteista joko oman tai jonkun toisen kokemuk-

sen pohjalta. Riitta Lehtikoinen kertoo artikkelissaan ”Neljä syytä miksi yritys on tyytymättömän konsultointiin” (2017), miksi tällainen työskentelytapa ole kaikkien mieleen. Lehtikoisen mukaan isossakaan konsulttitalossa kaikki eivät ole parhaita osaajia. Tämä on iso osa asiakkaan pettymystä ja huonoa kokemusta, koska usein tarjottu konsultti ei vastaa vaadittua tasoa. Itselleni nykyinen työpaikka on ensimmäinen alan työpaikka, joten kokemusta on vasta kuukausia takana. Ensimmäisessä projektissa mihin minut tarjottiin, vaatimuksena oli kymmenen vuoden kokemus. Toinen syy on Lehtikoisen mukaan väärä osaamisprofiili. Isoissa konsulttitaloissa se on todennäköisesti ongelma, meillä siitä ollaan välttytty pitämällä tarkka profiili ja tekemällä vain yhtä asiaa. Kolmas syy on väärä tyyli. Sama konsultti voi tehdä todella hyvää ja todella huonoa jälkeä riippuen siitä, missä yrityksessä hän konsulttina on projektissa. Projektin tiimi, työskentelytavat ja työskentely-ympäristö vaikuttavat todella paljon konsultin työhön. Neljäs syy epäonnistumiseen on se, että konsultti ei onnistu ilman asiakasta. Yrityksessä usein ongelmana on resurssipula, joka johtaa siihen, että aika ei riitä konsulteille. Jos konsulttityöstä halutaan maksimaali hyöty, niin yrityksen edustajia tarvitaan viitoittamaan suuntaa ja ohjaamaan tekemistä. Ilman apua ja tukea parhainkaan konsultti ei onnistu, tai tuottaa vain osan tavoitelluista hyödyistä (Lehtikoinen, 2017).

Konsulttipalveluita käyttävät yritykset voivat saada paljon enemmän irti konsulteista, jos uskaltavat käyttää niihin aikaa ja resursseja. Konsultti ei ole taikuri, joka osaa tehdä mitä asiakas haluaa ilman mitään ohjeistusta, vaan asiantuntija, joka toteuttaa pyydetyt asiat ohjeen mukaan.

3.7 Seurantaviikko 7

Maanantai 01.04.2019

Uusi viikko tuo mukanaan lisää perehdytystä uuteen asiakasprojektiin. Uusi projekti alkaa viikon päästä ja sitä varten on hyvä käydä kaikki mahdolliset materiaalit ennakkoon, jotta ensimmäisenä päivänä ei tarvitse kaikkea opetella kerrallaan. Tulevassa projektissa tulee olemaan monta kehittäjää eri maista, joten on tärkeää pitää työssä järjestys. Tähän on paljon erilaisia ohjelmia, yhteen niistä me tutustutaan tänään.

Asiakkaalla on projektissa käytössä tehtävienhallintaohjelma työn helpottamiseksi. Tällaisia ohjelmia on todella paljon, yksi niistä on esimerkiksi Atlassianin tekemä Jira. Projektin onnistumisen pohjana on yksittäisten tehtävien hallinta, joka hoituu juuri tehtävienhallinnan avulla. Tehtävienhallintatyökalun avulla voidaan myös raportoida erilaisia määräyksiä, virheitä ja tukipyyntöjä. Näin iso projekti olisi pulassa ilman kunnollista tehtävienhallintaa. Suurimmaksi osaksi päivä meni ohjelmaan tutustumisessa.

Tiistai 02.04.2019

Viikon toisena päivänä jatketaan tutustumista projektissa käytettäviin työkaluihin. Tänään pitäisi saada toimimaan yrityksellä käytössä oleva organisaatiowikiohjelmisto. Tällainen ohjelmisto on verkkosivusto, jonka sisältöä voivat työntekijät muokata ja täydentää itse omilla selaimillaan. Sinne on hyvä luoda sivustot erilaisista tunnuksista, sivustoista ja muusta hyödyllisestä informaatiosta yritystä ja projektia koskien.

Tässä projektissa on myös käytössä wikiohjelmisto mihin on kerätty tarpeellinen informaatio. Koko päivä meni sivustoa selaillessa, koska eri osa-alueita on todella paljon ja tekstiä on luettavaksi moneksi tunniksi, etenkin jos ei asiasta vielä ole perillä. Kaikki ei tietenkään jää kerrallaan muistiin, mutta onneksi työntekijät projektissa ovat panostaneet sivustoon niin hyvin, että uuden asiantuntijan on helppoa saada tarvittava tieto.

Keskiviikko 03.04.2019

Tänään meidän muiden tunnusten pitäisi olla toimivia ja me päästään tutkimaan testattavaan ohjelmistoon. Jotta ohjelmisto saadaan pyöriväksi, tarvitaan meidän koneille vpn-yhteys, koska siihen ei kehitysvaiheessa pääse muuten ulkopuolelta. Vpn-yhteyden avulla, pääsemme kuitenkin tutkimaan ohjelmaa, jota tulemme testaamaan ja ehkä vähän manuaalisesti kokeilemaan sen toimivuutta. Tämä ja huomina päivä päätimme pitää etäpäivinä, koska tällainen perehdytys ei vaadi toimistolla oloa.

Tämä päivä meni myös kokonaisuudessa asioihin tutustuessa ja tutkiessa. Ohjelman tutkiminen ennakkoon on tärkeää, jotta sitten testausvaiheessa tietää minkälainen ohjelma on kyseessä ja ei tule turhaan tuhlettua aikaa asioiden löytämiseen. Isossa projektissa yksittäinen asiantuntija hoitaa omaa tärkeää mutta pientä kokonaisuutta, joten on hyvä ennakkoon tutustua testattavaan ohjelmistoon ja materiaaleihin.

Torstai 04.04.2019

Toinen etäpäivä menee varsinaisen projektin parissa. Saamme projektikansion tietokoneillemme tutkittavaksi, jotta voimme vähän tarkastella miten testien kirjoittaminen on toteutettu. Näin isoon projektiin mukaan hyppääminen kesken on hankalaa, koska vie oman aikansa ennen kuin alkaa löytämään tarvittavat asiat toisen tekemistä kansioista, joten on hyvä, että varsinainen projektikansio on myös mahdollista käydä läpi ennakkoon.

Projektikansio oli sen verran laaja, että päivä ei siihen tutustumiseen riittänyt. Kun vertaa tätä projektia aikaisempaan, niin tämä on ihan eri tasolla. Ison yrityksen projektit helposti ovat monien kuukausien tai jopa vuosien työt, joten projektikansiot ovat sen mukaiset. Päivässä sai kuitenkin käsityksen sen rakenteesta ja käytettävästä kirjoitustyylistä, niin että maanantaina olisi helpompaa olla asiakkaalla.

Perjantai 05.04.2019

Tänään heti aamusta meidän pitää osallistua toiseen perehdytykseen muiden konsulttien kanssa ja ensimmäiseen daily scrumiin. Daily Scrumissa tiimin jäsenet kertovat mitä tuli tehtyä edellisenä päivänä ja mitä tulee tekemään loppupäivänä. Meidän

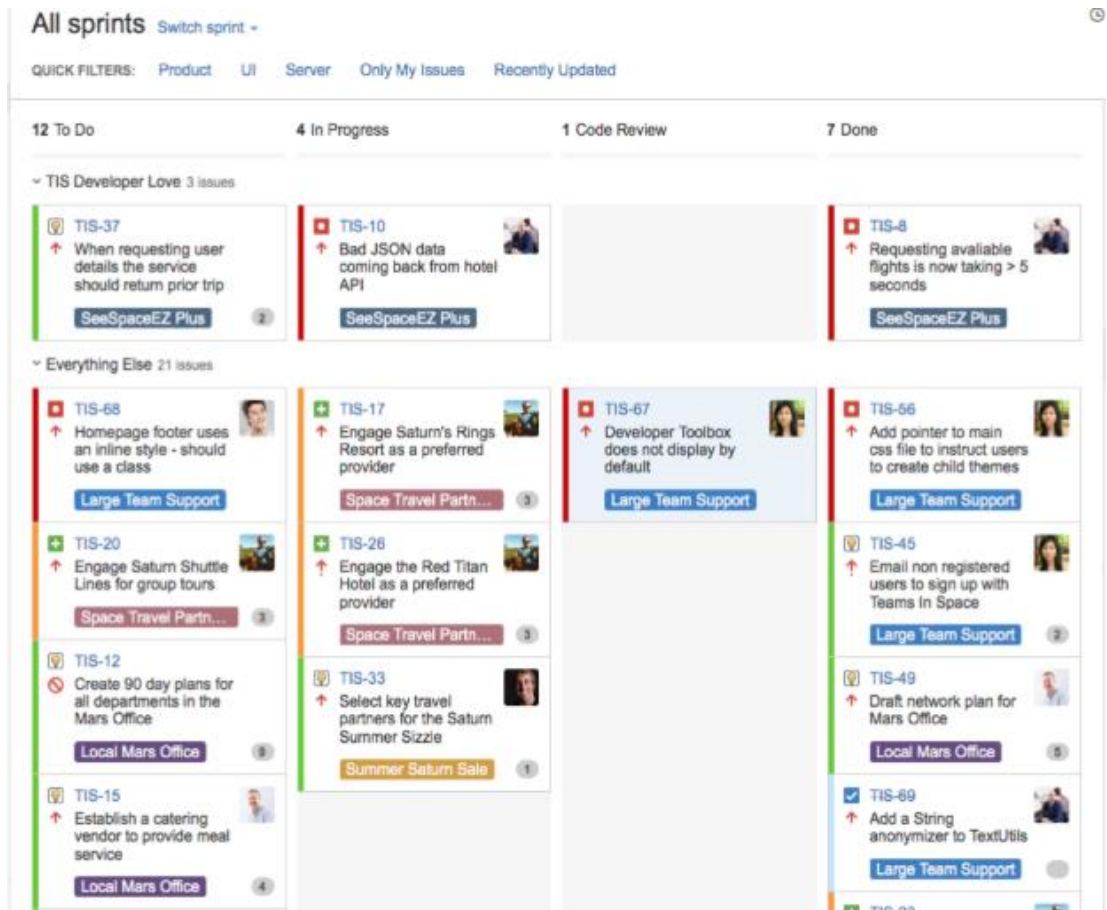
on tarkoitus esittäytyä ja vähän kuunnella tiimin kuulumisia. Loppupäivä sitten menee projektin parissa, kun on viimeinen päivä ennen projektin alkua kyseessä.

Aamupäivästä meillä oli Daily Scrum tulevan tiimin kanssa, jossa pääsimme esittämään. Daily Scrumiin osallistuminen on tärkeää, jotta kaikki tietävät missä mennään ja tämän takia ne pidetään joka ikinen päivä. Ne kestävät yleensä vain noin viisitoista minuuttia, joten siihen ei hirveän paljon aikaa mene. On ihan mielenkiintoista olla yhteyksistä tiimiläisiin, jotka ovat eri puolilla maailmaa ja jutella heidän kanssa.

Viikkoanalyysi

Tämä viikko meni pitkälti perehdytyksen parissa. Viikon aikana pääsin tutustumaan erilaisiin työkaluihin, jotka helpottavat työn seurannan yrityksessä. Esimerkkejä sellaisista työkaluista ovat tehtävienhallintaohjelmisto ja organisaatiowikiohjelmisto. Tehtävienhallintaohjelmiston avulla huolehditaan, että osatehtävät tulevat suoritetuksi ja kokonaisuus valmistuu. Organisaatiowikiohjelmiston tarkoitus on olla oppaana työntekijöille, joka on työntekijöiden voimin kirjoitettu ja sillä auttaa työntekijää työssään.

Yksi esimerkki tehtävienhallintaohjelmistoista on Atlassianin Jira, jonka avulla hallitaan projektin kulkua. Projektien kannalta on tärkeää keskittyä osatehtävien hallintaan. Näin kokonaisuus toimii, tulee valmiiksi, eikä pienempiä, mutta silti tärkeitä tehtäviä jää toteuttamatta (Eficode, 2016). Ohjelma auttaa pitämään projektin kasassa ja kaikki projektissa työskentelevät pysyvät perillä kehityksen edistymisestä. Jiran avulla kehittäjät voivat seurata, järjestellä ja priorisoida bugeja, uusia ominaisuuksia ja parannuksia tiettyihin ohjelman julkaisuihin (Bier, 2013).



Kuva 16. Jira-näkymä (GetApp.com, 2019)

Kuvassa 16 on esitetty näkymä Jirasta, jossa näkyy kaikkien sprinttien tiketit. Siitä voi tarkastella missä vaiheessa mikäkin tehtävä on ja kuka sitä työstää. Näin ainakin scrum master pystyy seuraamaan missä mikäkin tiimi menee, etenkin jos tiimejä on monta erilaista.

Organisaatiowikiohjelmistoja on myös paljon erilaisia, mutta sopiva esimerkki löytyy taas Atlassianilta nimeltä Confluence. Atlassianin mukaan Confluence on sisältöyhteistyöohjelmisto, joka muuttaa nykyaikaisen tiimin työskentelyä (Pendolin, 2018). Confluencea voi verrata Wikipediaan, erona vain se, että Confluence sisältää firman sisäisiä tietoja työntekijöille. Confluence on etenkin uudelle työntekijälle tärkeä työkalu, joka auttaa perehdytyksessä todella paljon, niin kuin kävi itselleni tällä viikolla.

Pages / Demo Project Home

/ Product requirements

User story: Dashboard

Created by Atlassian OnDemand [Administrator] on Dec 19, 2016

Epic link	SOF-62 - Dashboard TO DO
Status	DRAFT
Developer	@Arjan
Designer	@Sami
Product manager	@Des C

Primary user story

As a user, I want to be see a status of everything related to my work in Confluence, so that I can triage my work while on the go.

Requirements

#	Title	User Story	JIRA Issues	Notes	Importance
1	Activity feed	As a user, I want to see the most recent activity related to my work so that I can efficiently	SOF-34 - Activity feed DONE	<ul style="list-style-type: none"> Order pages chronologically with the most recent updates first. 	MUST HAVE

Kuva 17. Confluence-näkymä (Atlassian.com, 2019)

Tavallisin Confluence-näkymä on kuvattu kuvassa 17. Siinä on esimerkiksi kuvattu käyttäjätarina ohjausnäkykymälle, jota voi kehittäjät käyttää mallina työssään. Ohjelmistoon voi tehdä monta erilaista sivua jokaiselle projektin osa-alueelle ja sillä helpottamaan työntekijän arkea.

Työnhallintatyökalut auttavat pitämään projektin tehtävät järjestyksessä ja samalla vähentämään työntekijöiden stressiä. Kun asiat ovat kirjattu selkeään järjestykseen, työntekijät ovat tuottavampia, kun ei mene turhaan aikaa tehtävien selvittämiseen. Tietotyöläisen parempi ajankäyttö tuo enemmän voittoja samaan tapaan kuin tehokkaampi kone tehtaalla (Pulkkanen, 2019).

3.8 Seurantaviikko 8

Maanantai 08.04.2019

Uusi viikko alkaa asiakasprojektissa asiakkaan tiloissa. On todella jännittävää mennä asiakasprojektiin ja tutustua uusiin ihmisiin. Meidän työpäivä alkaa yhdeksältä ja sisältää ainakin pari palaveria, sekä kokouksen liittyen sprintin vaihtumiseen, joka on keskiviikkona. Ensimmäisenä päivänä emme vielä pääse aloittamaan varsinaista projektityöskentelyä, vaan meillä on aikaa perehtyä tiloihin, saada kulkuluvat ja tutustua ihmisiin.

Päivä meni kokonaisuudessa eri kokouksissa ja tapaamisissa. Melkein heti päivän alkuun osallistuimme tiimin daily scrumiin, jonka jälkeen meillä oli tapaaminen projektissa työskentelevien esimiesten kanssa. Meille nämä tapaamiset olivat tärkeitä, koska on parasta tutustua kaikkiin projektissa työskenteleviin asiantuntijoihin heti alussa. Iltapäivästä meillä sitten oli tapaaminen testausasiantuntijoiden kesken, jossa puhuttiin edellisen sprintin asioita ja vähän mitä tuleva sprintti tuo mukanaan. Viimeisinä meillä oli vanhemman testauskonsultin vetämä perehdytystunti, jonka aikana hän avasi projektin sisältöä ja sidosryhmiä.

Tiistai 09.04.2019

Tämä päivä on meillä etäpäivä, koska olemme sopineet, että pari päivää viikossa olemme etänä ja loput sitten asiakkaan tiloissa, niin etäpäivinä säästetty työmatka-aika kompensoi pitkää matkaa asiakkaalle. Päivän tarkoituksena on perehtyä viime sprintin aikana tehtyihin ominaisuuksiin, koska huomenna on seuraavan sprintin suunnittelu.

Päivän alkuun oli kuluvan sprintin viimeinen daily scrum, jonka jälkeen oli kokonainen päivä perehtymiseen. Kävin itsenäisesti läpi tehtävienhallintaohjelmistosta eri tiimien tehtävät ja selvittelin missä vaiheessa testaus on. Projekti on todella laaja, joten ominaisuuksia tulee paljon, reilusti enemmän kuin oli testaaajia, minkä takia me tultiin projektiin mukaan.

Keskiviikko 10.04.2019

Tämä päivä menee kokonaan seuraavan sprintin suunnitteluun. Tämä on minun ensimmäinen suunnittelupäivä, joten minkäänlaista kokemusta tai tietoa asiasta ei ole. Suunnittelupäivä järjestetään Helsingissä yhden kehittäjäryityksen tiloissa ja siihen osallistuvat kaikki mahdolliset työntekijät eri puolilta maailmaa, suuri osa kuitenkin Skypen välityksellä. Aikaa tähän on varattu aamuyhdeksästä iltaviiteen, joten mitään muuta tänään me ei tulla tekemään.

Aamulla lounaaseen asti aika meni retrospektiivin parissa, jossa katsottiin mitkä asiat menivät hyvin ja mitkä olisivat voineet mennä paremmin. Retrospektiivit ovat tärkeä osa kehitystä ja tiimin hyvinvointia, koska retroja säännöllisesti pitävät tuotekehitystiimit nauttivat työstään enemmän ja kehittyvät koko ajan toimintatavoiltaan paremmiksi kuin muut tiimit (Kiiskinen, 2017). Lounaan jälkeen olikin sitten aika ryhtyä seuraavan sprintin suunnitteluun. Siinä jokainen tiimi suunnitteli omaa työtään omassa tilassa, joten jokainen sai rauhassa käydä tulevat asiat läpi. Lopuksi vielä jokainen tiimi piti omia demojaan, joihin menikin loppupäivä. Demojen tarkoitus oli näyttää muille tiimeille mitä kukin on saanut aikaisiin, koska kehityksen aikana ei siihen ole aikaa.

Torstai 11.04.2019

Tänään alkaa uusi sprintti, joten päivän aikana on tarkoitus suunnitella mitä me tulemme tekemään sprintin aikana. Eilisen päivän suunnittelusessiot olivat päämääräisesti pyhitetty kehitykselle, joten meidän testaustiimin on pidettävä oma suunnittelu-palaveri. Yhteensä meillä on kolme tiimiä kehityksessä ja meidän on pidettävä palaverit kaikkien kanssa, joten siihen on varattu aikaa tänään ja huomenna.

Päivä meni pitkälti sprintin suunnittelussa testauksen näkökulmasta. Aamusta oli taas daily scrum, jossa kuitenkin oli aika vähän puhuttavaa, koska sprintti vasta alkoi. Lounaan jälkeen iltapäivällä meillä oli sitten ensimmäinen testaussuunnittelu tälle sprintille. Meillä uusilla testaajilla ei vielä mitään sanottavaa siihen ollut, joten me vain kuuntelimme ja yritimme pysyä asioissa perillä. Näiden lisäksi me saimme pari omaa tehtävää aamupäivästä hoidettavaksi ja omat tiketit huomisen alkaen.

Perjantai 12.04.2019

Tänään on viikon toinen etäpäivä ja taas kokouksia täynnä oleva työpäivä. Lounaan jälkeen meillä on sovittu vanhemman asiantuntijan kanssa käytäntöjen läpikäynti, joten tiedämme miten projektissa toimintaan ja miten testejä kirjoitetaan. Jokaisella on oma tyyli kehityksen suhteen ja on parasta, että kaikki asiantuntijat pitävät kiinni samasta tyylistä projektissa, näin kirjoitettu koodin on luettavampaa. Sen jälkeen meillä on kaksi suunnittelupalaveria testeihin liittyen kahden muun tiimin kanssa.

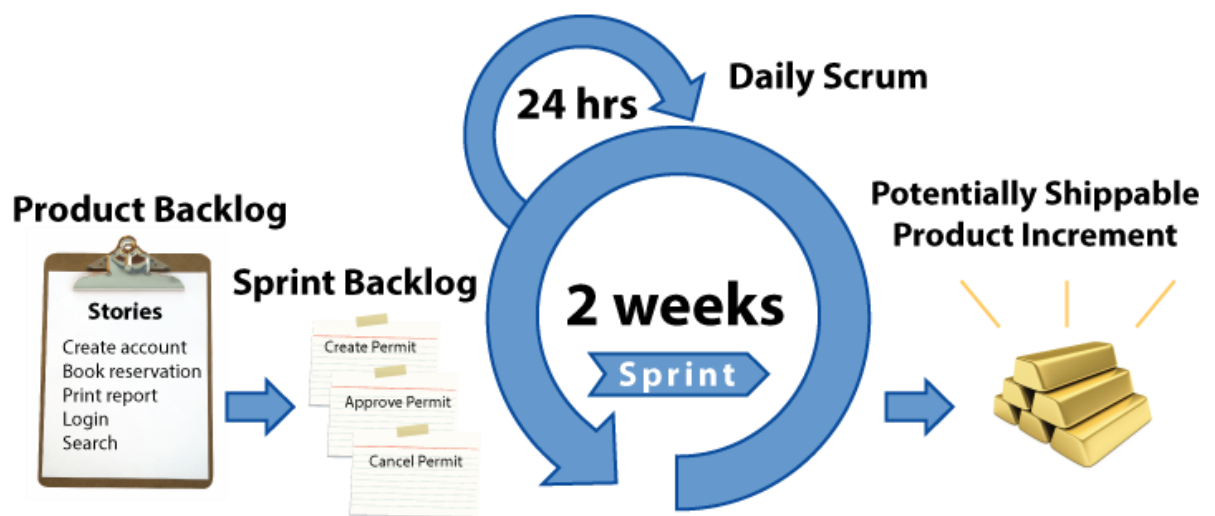
Melkein koko päivä meni kokouksissa, melkein kuusi tuntia Skype oli päällä tauotta. Aamulla dailyn jälkeen hetki aika meni tiketin parissa, lounaan jälkeen sitten alkoi kokousputki. Alussa kävimme läpi projektin käytännöt ja miten asiat ollaan tehty aikaisemmin, jotta voimme pitää järjestyksen meidän kehittämässä asioissa. Tämän jälkeen meillä oli kaksi suunnittelupalaveria liittyen testaukseen tässä sprintissä. Minut ja toisen uuden testaajan jaettiin kahden tiimin kesken, jotta jatkossa jokaista tiimiä kohtaan on ainakin yksi testaaja. Näihin sitten meni loppupäivä ja tiketin teko siirtyy ensi viikolle.

Viikkoanalyysi.

Tämä projekti toteutetaan ketterän kehityksen menetelmällä ja Scrum -mallilla. Scrum on siis tekijöidensä Ken Schwaberin ja Jeff Sutherlandin mukaan prosessiviitekehys, jossa on määritelty periaatteita, arvoja, tapahtumia, rooleja ja välineitä, joilla tehdään ohjelmistokehityksestä ketterää: lopputuotosta kehitetään siis pienissä paloissa siten, että siihen kehitettäviä ominaisuuksia säädetään asiakastarpeen mukaisesti jatkuvasti kehityksen aikana (Oinonen, 2017). Minua ja muita testaajia Scrumista eniten koskevat aktiviteetit eli daily scrum ja sprintti.

Daily scrum eli päiväpalaveri, on päivittäin pidettävät tiimin palaveri, jossa kukin tiimin jäsen kertoo mitä on tehnyt edellisenä päivänä, mitä tulee tekemään seuraavan päivän aikana ja onko mitään, mikä hidastaa kehitystä. Palaverin vetää scrum-mestari ja siihen osallistuvat kaikki jäsenet, vaikka muutkin kiinnostuneet ovat tervetulleita. Palaverin tarkoitus on saada kaikille tieto siitä, missä mennään ja mitä ongelmia on tullut vastaan. Daily scrum kestää yleensä 15 minuuttia, jonka jälkeen on varattu 15 minuuttia lisää ongelmien katselmointiin, jos on tarvetta.

Daily scrum on kuitenkin osa sprinttiä, joka alkaa sen suunnittelulla. Kuten Perttu Tolvanen kirjoittaa artikkelissaan Ketteryys haltuun: Scrum pähkinäkuoressa (2019) Sprintin aikana tiimi toteuttaa sprinttiin kuuluviksi valittuja toiminnallisuuksia. Sprintin aikana vaatimusten muuttaminen on kiellettyä, ja tiimillä on täysi vapaus tehdä tarpeelliseksi katsomiaan toimenpiteitä, jotta sovittu sprintin päämäärä voidaan saavuttaa. Tiimi organisoii itsensä parhaaksi katsomallaan tavalla. Jokaisen sprintin lopuksi tiimi esittelee valmista tuotetta tuotteen omistajalle. Sprintin lopuksi tuotteen siis pitäisi olla periaatteessa käyttöönotettavissa: se on toteutettu, testattu, dokumentoitu, käyttöliittymä on valmis ja niin edelleen. Näin omistaja voi päättää joko seuraavan sprintin tekemisestä tai tuotteen käyttöönottamisesta.



Kuva 18. Kuvaus scrum-menetelmästä (Rasmusson, 2019).

Kuten kuvassa 18 näkyy, ketterässä kehityksessä on tarkoin suunniteltu aikataulu, mikä auttaa pitämään projektin kasassa. Sprintin kesto on tarkoin määritetty, jokainen päivä alkaa daily scrumilla ja kaikki tehtävät ovat merkattu backlogiin. Meidän projekti seuraa pitkälti kuvassa esitettyä menetelmää, ainoana erona se, että meidän sprintti on nykyään kolme viikkoa.

Sprintin päätteeksi on myös todella tärkeätä pitää retrospektiivi. Retrospektiivissä pyritään säännöllisesti toistuvilla lyhyemmillä sessioilla saamaan aikaan tehokas, jatkuvasti oppiva ja kehittyvä tiimi. Asioiden parantaminen on helpompaa, kun ei kerätä pitkää listaa parannuskohteita ja palata asiaan puolen vuoden tai vuoden päästä (Kiiskinen, 2017). Pitämällä hyvä retrospektiivi tiimin motivaatio pysyy parempana, koska tekijät huomaavat, että asioille tapahtuu jotain ja jatkossa asioiden korjaaminen on helpompaa.

3.9 Seurantaviikko 9

Maanantai 15.04.2019

Uusi viikko alkoi daily scrumilla, niin kuin kaikki muutkin päivät. Kumpikin meistä uusista asiantuntijoista sai oman tiimin kehittäjästä, joten jokainen osallistuu oman scrumiin ja se mihin minä osallistun siirtyi heti aamuun työpäivän alkuun. Tämä on aika kätevää, koska siitä alkaa päivä ja sitten pääsee työskentelemään ilman häiriötekijöitä. Päivän agendana on tutkia tiketit testejä varten huomista läpikäyntiä uuden tiimin testaajan kanssa ja aloitan myös uuden testin tekemisen.

Päivä meni melkein kokonaisuudessa testin parissa. Kokenut testaaja saisi tehtyä työt paljon nopeammin, mutta näin juniorina siihen menee enemmän aikaa. Jonkun aikaa jouduin myös tutustumaan kehitystiimin tiketteihin, jotta olen huomiossa tapaamisessa paremmin perillä asioista.

Tiistai 16.04.2019

Tämä päivä sisältää testin tekoa ja suunnittelupalaverin kehitystiimin testaajan kanssa. Itse vielä tulen toimimaan enemmän kuunteluroolissa, koska takana on vasta viikko projektissa ja en ole vielä ehtinyt päästä projektiin kiinni kunnolla. Mukana on varsinaisena testaajana pitempään projektissa ollut kokeneempi kollega, joten ei pitäisi palaverin kanssa olla ongelmaa.

Palaverissa kävimme läpi nykytilanteen projektin suhteen ja mitä testejä pitää tehdä. Mukana meidän lisäksi oli testimanageri, joka johti palaveria ja kirjasi ylös tiketit tulevia testejä varten. Suuri osa ominaisuuksista on vielä kesken, joten testaus tapahtuu heti kun on jotain valmiina. Sillä välin teen edellistä testiä loppuun aina kun on vapaa hetki.

Keskiviikko 17.04.2019

Eilisessä palaverissa jaettiin uusia testaustikettejä ja minä sain vastuun testata kehitystiimin uutta osiota, kun se on valmis. Siihen mennessä hoidan loppuun tiketin jonka aloitin viime viikon perjantaina ja vähän pohjustan uusia tikettejä varten. Testattavaa tulee olemaan paljon, joten kollegani tulee auttamaan minua testien tekemisessä, näin saadaan nopeammalla aikataululla testattua tärkeät uudet ominaisuudet.

Päivä alkoi klassisesti daily scrumilla, joka on etenkin tällä hetkellä tärkeää, koska odotamme että ominaisuus valmistuu ja näin saamme uusimman tiedot tilanteesta. Osio ei ollut vielä valmiina integroitavaksi testausversioon ohjelmistosta, joten päivä jatkui aikaisemman testin parissa. Testi on reilusti haastavampi mitä meillä oli perehdytyksessä ja aikaa se vie paljon enemmän. Vanhemmat kollegat yrittävät parhaansa mukaan auttaa, mutta heillä on todella paljon työtä, joten suurimmaksi osaksi joutuu selvittämään asiat itse, mikä on ongelma ainoastaan ajan takia, projektissa kun on aikataulu julkaisulle. Muuten tällä tavalla oppii asiat todella hyvin ja jatkossa prosessi on reilusti nopeampi.

Torstai 18.04.2019

Tänään on viikon viimeinen päivä ja huomisesti alkaen voi hetken aikaa rauhoittua pääsiäisen merkeissä. Vapaat tulivat hyvään hetkeen, koska näiden kahden viikon aikana projektissa tuli sen verran tavaraa, että aivot tarvitsevat nyt pari vapaata päivää aikaa prosessoida kaiken informaation. Kuitenkin tämä päivä menee vielä täysillä kehityksen parissa, vaikka vietämmekin etäpäivää, jotta kaikilla on mahdollisimman lyhyt kotimatka päivän päätteeksi.

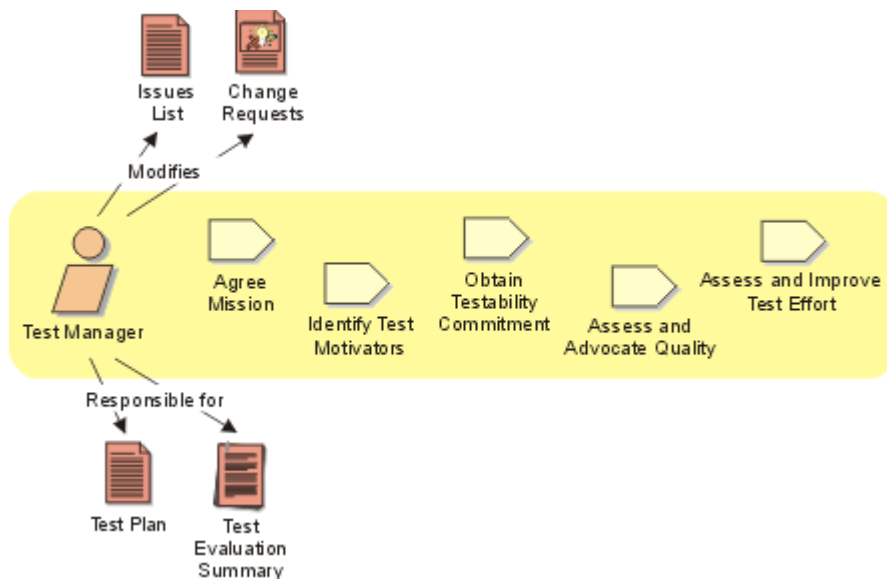
Aamun daily scrumin jälkeen työstin loppupäivän iso testiä, jota olen työstänyt nyt jo monta päivää. Nyt kun me ollaan uusia asiantuntijoita, meille annetaan hyvin aikaa työstää asiat, vaikka autetaankin heti, jos ollaan jumissa meidän tikettien kanssa. Itselläni edistyminen on hidasta, mutta varmaa, vaikka olisinkin toivonut saavani asiat nopeammin kasaan. Tässä testissä tuli paljon minulle uusia temppuja, minkä takia jouduin pitkään tutkimaan ja etsimään ratkaisuja.

Viikkoanalyysi.

Tämä viikko oli pääsiäisen takia lyhyempi työpäivien määrän suhteen, mutta työtä oli ihan riittävästi. Viime viikon sprintin suunnittelu johti siihen, että tällä viikolla suunnitelimme tulevia testejä, kun minut ja toinen uusi konsultti jaettiin eri kehitystiimeihin testaajiksi. Meidän tiimin testaus suunnittelupalaveri oli tiistaina, jossa kävimme nykyistä testitilannetta kehitystiimin tekemässä ominaisuudessa testauspäällikön kanssa. Testauspäällikön osallistuminen on testaajien kannalta todella tärkeää, koska testauspäällikkö suunnittelee testitapaukset ja jakaa testitikit testaajien kesken. Näin testaajat voivat keskittyä pelkästään testien tekemiseen.

Comiqin testauspäälliköt jakautuvat kahteen osaan, testauspäällikkö ja tekninen testauspäällikkö. Molempien avaintehtävät ovat testauksen koordinointi, testausprosessien kehittäminen ja testauspäällikkönä toimiminen. Kuitenkin teknisen päällikön pitää olla kiinnostunut myös testausautomaatiosta, sekä osata testauksen tekniset osa-alueet, kuten testausautomaatio, suorituskykytestaus tai DevOps -puolen toimintamallit (www.comiq.fi, 2019). Testauspäällikkö on tärkeä henkilö projektille, koska hän voi parantaa tuloksia vaatimalla tiettyjä toimintatapoja, seuraamalla aktiivisesti toimintaa ja tarjoamalla valmennusta (Tuovinen, 2019).

Comiqilla on suuri osa testauspäällikköistä, koska idyllisessä tilanteessa, jokainen projekti tarvitsisi sellainen, jotta testaajat voisivat keskittyä testeihin ja projektipäällikkö niiden suunnitteluun, sekä työn kulun seurantaan ja järjestyksen ylläpitoon.



Kuva 19. Testauspäällikön tehtävät (www.sceweb.sce.uhcl.edu, 2019)

Kuten kuvassa 19 näkyy, testauspäällikkö vastaa monesta projektin osasta. Tärkeät tehtävät testauspäälliköllä ovat testien suunnittelu ja tulosten tarkastelu. Hyvällä suunnittelulla hän varmistaa, että testaaajien aika ei mene ylimääräiseen työhön ja testaus voidaan suorittaa mahdollisimman nopeasti. Testitulosten tarkastelu taas auttaa suunnittelemaan tulevat testit paremmin, koska testauspäällikkö näkee miten työt on hoidettu ja mitkä ominaisuudet testattu.

Testauspäällikön työ on myös todella vastuullista, koska on hänen vastuulla varmistaa että testit ovat laajat, tehokkaat, laadukkaat ja että työ etenee aikataulun mukaisesti (www.guru99.com, 2019). Eteen voi tulla myös monia ongelmia, joita testauspäällikkö joutuu hoitamaan. Usein käy niin että testaukseen ei ole varattu tarpeeksi aikaa ja resursseja, projektin aikataulu ja budjetti ovat todella tiukkoja, tiimin jäsenet eivät ole kaikki samassa paikassa tai vaatimukset testeille ovat liian haastavia. On siten testauspäällikön vastuulla hoitaa kyseiset ongelmat ja varmistaa että työn tulos on paras mahdollinen.

Testaajat eivät kyllä saa isompaa kosketusta testauspäällikön työstä, koska nämä kaksi tehtävää pyritään pitää toisistaan erossa. Joissakin tilanteissa testaaja saattaa hoitaa myös testauspäällikön työt, mutta näin käy harvoin. Tämän viikon tapaamisessa näki kuitenkin hyvin testauspäällikön tärkeyden, koska hän johti koko palaveria ja oli parhaiten perillä testien nykytilanteesta ja osasi jakaa työtehtävät hyvin.

3.10 Seurantaviikko 10

Tiistai 23.04.2019

Opinnäytetyön viimeinen seurantaviikko sattui myös olemaan sprintin viimeinen viikko, joten tämän sprintin tehtävät on saatava loppuun. Itselläni on vieläkin sama yksi tiketti kesken, joka on pakko saada tällä viikolla valmiiksi, siihen on mennyt valmiiksi jo liikaa aikaa. Tämä päivä ja huomina ovat etäpäiviä, joten näiden päivien aikana oli tarkoitus saada testikokonaisuus valmiiksi, jotta sitten torstaina paikan päällä voisi testin tarkastuttaa kokeneemmalla kollegalla ennen kuin vie sen yhteiseen kansioon.

Päivä meni kokonaisuudessa iltaan saakka testin parissa. Testit ei periaatteessa ole laaja, mutta siinä on monta vaikeata osaa ja se on aika tärkeä ominaisuus. Työtä hidastaa se, että tämän tyyppisiä ongelmia en ole kohdannut aikaisemmin testien teossa, joten ratkaisuja pitää keksiä työn ohessa. Suurin osa päivästä meni ongelmien ratkaisemiseen, enkä päivän päätteeksi saanut testiä toimimaan, joten päätin jättää testin huomiseksi.

Keskiviikko 24.04.2019

Toinen etäpäivä alkoi taas daily scrumilla, jota eilen ei pidetty pääsiäisloman takia. Sen jälkeen on tarkoitus palata eillisen ongelman pariin ja saada asia etenemään. Täällä on myös tehtävien statuksen tarkastelu tiimin kanssa, jotta ongelmat saadaan ratkottua yhdessä, eikä kukaan jää jumiin omien ongelmien kanssa.

Daily scrumissa vähän käytiin kuulumisia, mutta loma vaikutti siihen, että kenelläkään ei hirveästi kerrottavaa ollut. Sen jälkeen hyppäsin ratkomaan oman testin ongelmaa ja sainkin korjattua sitä aamupäivän aikana. Tämä taas todistaa sen, että on pakko pitää taukoa, jos ei ratkaisu meinaa löytyä. Tässä tilanteessa annoin itselleni yön aikaa ja sen jälkeen huomasin heti virheet ja sain ongelmat ratkottua. Iltapäivän statustarkastelussa pystyinkin sanomaan, että testi on korjauksia vailla valmis ja sain siihen pari vinkkiä millä sain testit toimimaan hyvin.

Torstai 25.04.2019

Tämän päivän tarkoitus on saada testi hiottua julkaisukelpoiseksi ja edetä tästä muihin testeihin. Tämä päivä menee taas toimistolla, jotten kollegoilta voi helposti kysyä neuvoa ja pyytää käymään testit läpi mahdollisten korjauksien takia. Sen jälkeen kuin tämä testi on luovutettu, on tarkoitus käydä läpi seuraava testikokonaisuus ja millä aikataululla ne on saatava valmiiksi.

Aamupäivän aikana sain testin viimeistelyä ja iltapäivästä se oli julkaisukelpoinen. Toinen uudempi konsultti oli myös saanut tehtyä omansa samaan aikaan ja teemme seuraavan kokonaisuuden puoliksi, koska siinä on todella monta vaikeata testiä. On mukavaa työskennellä yhdessä, koska siinä tilanteessa voi kysyä mielipidettä toiselta, eikä se häiritse hänen työtä, hän kun tekee samaa kokonaisuutta.

Perjantai 26.04.2019

Tänään on viikon ja seurannan viimeinen päivä. Tänään on tarkoitus aloittaa uusien testien suunnittelu ja ehkä jopa itse testaus iltapäivästä. Perjantai on yleensä meillä etäpäivä ja niin se on myös tällä kertaa. Päivä menee varmaan kokonaisuudessa uusien testien suunnitteluun ja manuaaliseen testaukseen, näin saa paremman kuvan testattavasta ominaisuudesta automaatiotestejä varten.

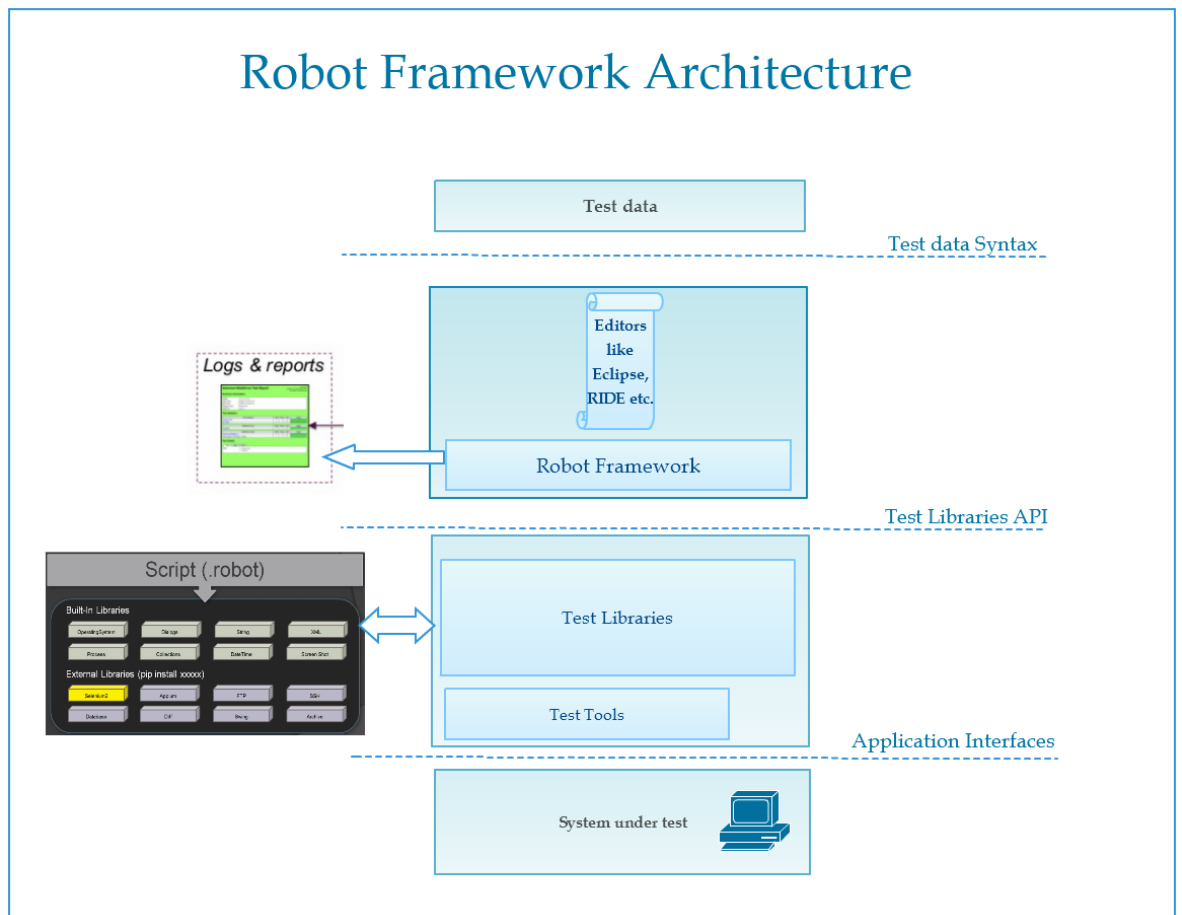
Päivä meni niin kuin olin aikaisemmin suunnittellut. Aamupäivästä suunnittelin testejä ja kävin sivustoa manuaalisesti läpi. Ominaisuus on vielä vähän keskellä ja integraatiovaiheessa, joten testaus tullaan tekemään tämä mielessä. Kun testataan kehitysvaiheessa oleva ominaisuus, se tuottaa tiettyjä ongelmia, koska ominaisuus voi muuttua tai kadota ohjelmasta kokonaan. Tämä taas johtaa siihen, että testit eivät toimi jokaisella ajokerralla.

Viikkoanalyysi.

Tämä viikko meni kokonaisuudessa testien parissa ja on viimeisellä viikolla hyvä palata kaikkien testien ytimeen eli Robot Framework -kieleen ja miksi se on niin suosittu, sekä käytetty. Itse on tähän mennessä käyttänyt kaikissa projekteissa kyseistä kieltä ja se onkin meidän työntekijöiden lisäksi muidenkin suomalaisten testaajien suosikki. Tässä opinnäytetyössä Robot Framework on jo avattu teknisestä näkökulmasta, mutta nyt on hyvä hetki palata siihen, etenkin parin viikon asiakasprojektin jälkeen.

Sovelluskehityksessä on ollut pitkään haaste, joka on tuttu etenkin ketterässä kehityksessä. Kuinka varmistaa, että uusi ominaisuus ei hajota aikaisempaa kokonaisuutta? Ketterässä kehityksessä ohjelmistoa pitää testata jatkuvasti, ja se taas onnistuu käytännössä vain automatisoimalla testausta (Kotilainen, 2018).

Kuten Kotilainen mainitsee artikkelissaan Diplomityöntekijä keksi ohjelmiston, josta tuli suomalaisten suosikki (2018), yksi parhaista testauksen työkaluista Robot Framework on kehitetty Suomessa. Puolen vuoden työkokemuksen perusteella, Kotilaisen näkökulma on helppo hyväksyä, nimittäin Robot Framework on käyttäjän kannalta luotu mahdollisimman mukavaksi käyttää ja sen opettelu on suhteellisen helppoa, mutta taas ominaisuuksia on monia. Robot Frameworkin on kehittänyt suomalainen Pekka Klärck ja se sai alkunsa vuonna 2004. Ensimmäiseksi Klärck yritti jatkaa hanketta Nokialle ja sen avulla automatisoinnissa onnistuttiin paljon helpommin, niinpä sen käyttö levisi Nokian sisällä ohjelmistotestaukseen. Tämän jälkeen vuonna 2008 siitä julkaistiin avoimen lähdekoodin versio ja sitten vuonna 2015 kehitys siirtyi Robot Framework Foundation -säätiölle. Ohjelmistosta on tullut Suomessa de facto – standardi automaattisessa ohjelmistotestauksessa (Kotilainen, 2018).



Kuva 20. Robot Framework arkkitehtuuri (www.devonblog.com, 2019)

Kuvassa 20 on esitetty Robot Framework arkkitehtuuri. Robot Framework on alun perin kirjoitettu Pythonille, minkä takia sitä voi käyttää kaikilla käyttöjärjestelmillä. Kieli käyttää pohjana kirjastoja, joita voi helposti laajentaa omilla kirjastoilla. Kun testi ajetaan, Robot Framework käsittelee testi datan ja sitten aja testit käyttäen avainsanoja joka valmiista tai omista kirjastoista.

Robot Frameworkin suosion syiksi Pekka Klärck nostaa kolme asiaa. Ensimmäiseksi ohjelman toteutustapa tekee siitä erittäin joustavan ja sillä voi testata, vaikka mobiilisovellusta tai rest-rajapintoja. Toinen etu on testien tekemisen helppous ja esimerkiksi testikomentojen luku tai tekeminen ei vaadi ohjelmointitaita. Kolmas etu on raportointi. Testeistä saadaan aina kattavat raportit ja niiden perusteella on helpompi korjata virheet.

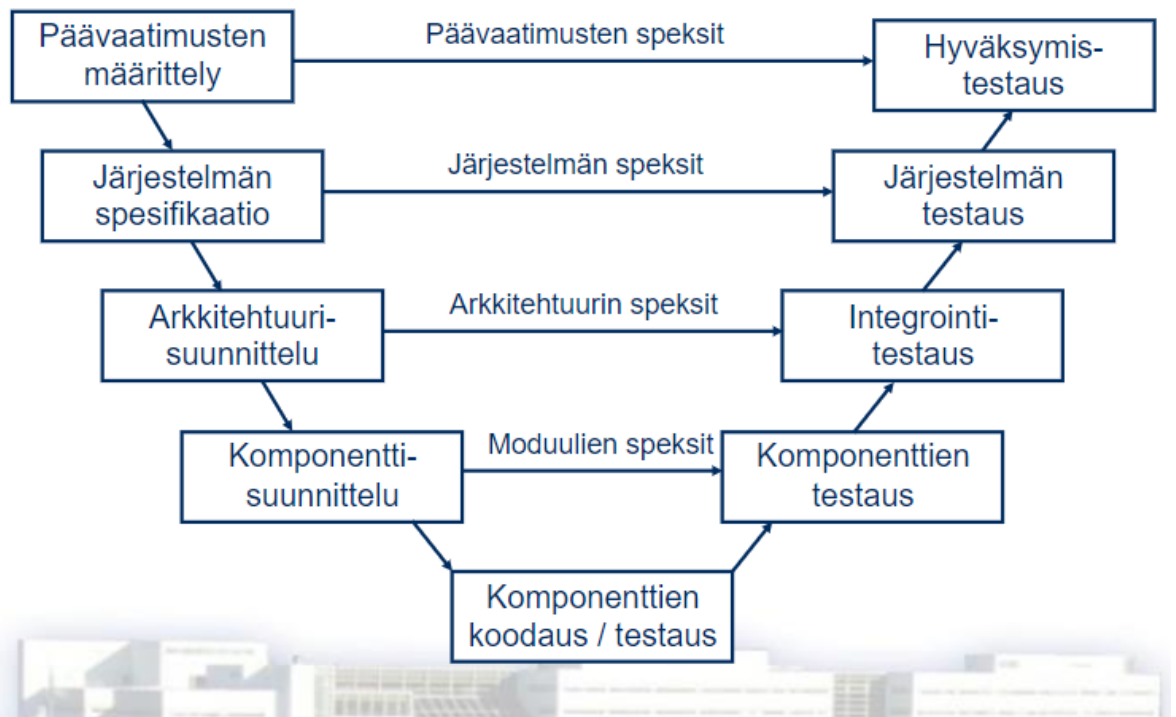
4 Pohdinta ja päätelmät

Kymmenen viikon seurannan jälkeen voin sanoa varmasti, että omat taidot ovat kehittyneet todella paljon. Opinnäytetyön alussa olin suhteellisen tuore asiantuntija ja moni asia vielä ihmetytti ja tuotti ongelmia. Vaikka näiden kymmenen viikon jälkeen joudun silti usein hakemaan apua, pärjään paljon paremmin yksin. Näin ollen tärkein muutos joka on tapahtunut viimeisten kymmenen viikon aikana on se, että uskon enemmän itseeni ja olen itsevarmempi. Tässä työssä iso ongelma usein on epävarmuus omista taidoista ja oman osaamisen vähättely. Uskon että tässä asiassa olen kehittynyt paljon, vaikka vielä on paljon tehtävää asian eteen. Tämän lisäksi uskalla puhua muille kanssakehittäjälle ilman pelkoa, että sanon jotain väärin, mikä vaivasi työn alkukuukausina. Olen ymmärtänyt ei keskustelu auttaa tiimin välistä yhteistyötä ja jos jokin asia vaivaa, siitä pitää kysyä, vaikka se onkin omasta mielestä väärä tai turha kysymys.

Tänä aikana olen päässyt kokeilemaan työskentelyä isossa oikeassa projektissa ja milaista on olla osa kehitystiimiä. Olemme aikaisemminkin pitäneen Daily Scrumeja ja Sprinttejä, mutta ne olivat yleensä vain koulutusmielessä. Tässä projektissa taas Daily Scrumeilla oli oikeasti tärkeä tarkoitus ja siihen osallistui kehittäjiä ympäri maailmaa. Ensimmäiset kerrat olivat todella pelottavaa puhua omista tekemisistä kaikille, kun koko ajan tuntui, että teen todella vähän. Parin viikon jälkeen kuitenkin tajusin, että jokainen tuo oman panoksen projektiin ja välillä tulee ongelmia eteen, eikä hidasta tempoa tarvitse hävetä, kunhan työt on tehty hyvin. Näin sain itse irti enemmän Daily Scrumeista, kun pystyin pysymään paremmin perillä muiden tekemisistä. Tällainen Sprinttityöskentely ei ollut minulle täysin uusia asia, mutta tällä tavalla hyödynnettynä avautui uudella tavalla.

Opinnäytetyön päätteeksi koen, että päiväkirjamuotoisesta opinnäytetyöstä oli paljon hyötyä työssä. Pitämällä päiväkirjaa palasin uudestaan tekemiini asioihin ja jouduin tutkimaan erilaisia alueita tarkemmin käyttäen ammattikirjallisuutta, josta sitten opin uusia asioita. Päiväkirjan avulla usein palasin asiaan josta en olisi muuten muistanut hakea tietoa tai ongelmaan, jonka muuten olisin unohtanut ratkaista.

Testauksen V-malli



Kuva 21. Testauksen V-malli (Turunen, 2015)

Tärkeintä mitä haluaisin tällä opinnäytetyöllä tuoda esille, on testauksen tärkeys ja kompleksisuus. Taakse jäivät ajat, kun testaus sisälsi vain manuaalista nappien painamista ja se helposti laiminlyötiin, kun ei koettu sen olevan erityisen hyödyllistä. Tämän opinnäytetyön tarkoitus on näyttää, että testaus on iso ja tärkeä kehityksen osa, jolla saa myös säästettyä aikaa ja resursseja. Tätä asiaa voi tutkia vielä laajemmin, koska automaatiotestaus on vasta alkuvaiheessa ja sen hyödyt eivät ole kaikille selvillä. Tutkimalla esimerkiksi automaatiotestauksen avulla säästettyjä resursseja voidaan saada selville tarkka etu joka automaatiotestauksesta tulee.

Kaikin puolin olen tyytyväinen valintaani ja siihen mitä näiden kymmenen viikon päätteeksi sain aikaiseksi. Olen tämän opinnäytetyön avulla päässyt tarkastelemaan omaa kehitystä, sekä huomioimaan omat heikkoudet, jotka voin nyt jatkossa korjailla. Työn analysointi on antanut mahdollisuuden paremmin tarkastella omaa jatkamista työssä ja itselleni parhaiten sopivia menetelmiä. Uskon että tällä opinnäytetyöllä saavutin tavoitteeni eli tuoda testauksen tärkeys lukijoiden tietoon.

Lähteet

James Turnbull, 2014 (Päivitetty 2018) The Docker Book: Containerization is the new virtualization

Teemu Malinen, 2014 Magento verkkokauppa – Mitä se maksaa?,
<https://www.sofokus.com/fi/blogi/magento-verkkokauppa-mita-se-maksaa/>, luettu 18.02.2019

Chromatic, 2003 Extreme Programming Pocket Guide. O'Reilly Media.

Don Wells, 1999, 2000, 2001, 2004 ja 2009 (Viimeisin päivitys 2013) Extreme Programming: A Gentle Introduction, <http://www.extremeprogramming.org/index.html>, luettu 20.02.2019

Agile Alliance, 2019
[https://www.agilealliance.org/glossary/xp/#q=~\(infinite~false~filters~\(postType~\(~'post~'a_a_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'a_a_video\)~tags~\(~'xp\)\)~searchTerm~'~sort~false~sortDirection~'asc~page~1\)](https://www.agilealliance.org/glossary/xp/#q=~(infinite~false~filters~(postType~(~'post~'a_a_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'a_a_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1)) luettu 23.02.2019

Dzmitry Ihnatseyu, 18.07.2018 Robot Framework: The Ultimate Guide to Running Your Tests. <https://www.blazemeter.com/blog/robot-framework-the-ultimate-guide-to-running-your-tests>, luettu 02.03.2019

Ilkka Toivanen, 06.02.2017 DevOps jatkuvan kehittämisen tukena.
<https://www.octo3.fi/devops-jatkuvan-kehittamisen-tukena/>, luettu 07.03.2019

Software Testing Fundamentals, <http://softwaretestingfundamentals.com/integration-testing/>, luettu 16.03.2019

Guru99, 2019 <https://www.guru99.com/integration-testing.html#2>, luettu 16.03.2019

Atlassian Bitbucket, 2019 Git Hooks. <https://www.atlassian.com/git/tutorials/git-hooks>, luettu 18.03.2019

Justin Ellingwood, 12.08.2014 How To Use Git Hooks To Automate Development and Deployment Tasks, <https://www.digitalocean.com/community/tutorials/how-to-use-git-hooks-to-automate-development-and-deployment-tasks>, luettu 19.03.2019

Tuija Ekholm, 19.01.2017 Millainen on hyvä konsultti? <https://www.invenco.fi/millainen-on-hyva-konsultti/>, luettu 30.03.2019

Ville Tolvanen, 16.10.2013 Konsultti, <https://www.villetolvanen.com/fi/2013/10/16/konsultti/>, luettu 30.03.2019

Riitta Lehikoinen, 28.09.2017 Neljä syytä miksi yritys on tyytymätön konsultointiin <https://www.winway.fi/blogi/2017/09/28/21015>, luettu 01.04.2019

Aleksi Pulkkanen, 2019 3 tuottavan työn avainkysymystä, joihin vain harva osaa vastata. <https://www.agendum.com/post/3-tuottavan-tyon-avainkysymysta-joihin-vain-harva-osaa-vastata>, luettu 06.04.2019.

Harri Pendolin, 20.03.2018 Confluence – ohjelma vai alusta? <https://contribute.fi/2018/03/20/confluence-ohjelma-vai-alusta-2/>, luettu 06.04.2019

Eficode, 04.10.2016 Rikkinäiset workflowt: syyt ja seuraukset <https://www.eficode.com/blogi/blogi/rikkinaiset-workflowt-syyt-seuraukset>, luettu 07.04.2019

Nils Bier, 2019 Six Reasons Why I Use Jira In My Role as a Technical Writer <https://www.k15t.com/blog/2014/04/six-reasons-why-i-use-jira-in-my-role-as-a-technical-writer>, luettu 07.04.2019

Arto Kiiskinen, 03.10.2017 Retrospektiivi on elintärkeä tuotekehitystiimeille – muille riittääkin viinanhöyryinen kehityspäivä kerran vuodessa? <https://contribute.fi/2017/10/03/tuotekehitystiimeilta-vaaditaan-retrospektiiveja/>, luettu 10.04.2019

Annika Oinonen, 17.08.2017 Ketterän projektin projektisuunnitelma – Osa 1: Kaikki haluavat olla ketteriä, sillä kukapa haluaisi olla jäykkä? https://www.projekti-instituutti.fi/blogi/ketteran_projektin_projektisuunnitelma_-_osa_1_kaikki_haluavat_olla_ketteria_silla_kukapa_haluaisi_olla_jaykka.2762.blog, luettu 13.04.2019

Perttu Tolvanen, 2019 Ketteryys haltuun: Scrum pähkinänkuoressa
<https://www.meteoriitti.com/2013/06/06/ketteryys-haltuun-scrum-pahkinankuoressa/>, luettu
14.04.2019

Rekrytointi, Comiq Oy, www.comiq.fi, luettu 05.05.2019

Antti-Pekka Tuovinen, 18.04.2013 Testauksen hallinta ja johtaminen, luettu 07.05.2019

Samuli Kotilainen 16.05.2018 Diplomityöntekijä keksi ohjelmiston, josta tuli suomalaisten suosikki. <https://www.tivi.fi/uutiset/diplomityontekija-keksi-ohjelmiston-josta-tuli-suomalaisten-suosikki/ccb0d491-794f-3cd5-b17e-53ff1c932c64>, luettu 12.05.2019.

Robot Framework kotisivu, 2019 www.robotframework.org