

Joni Jauhiainen

LAADUNVARMISTUSINSINÖÖRIN TYÖ SOVELLUSTESTAUKSESSA

LAADUNVARMISTUSINSINÖÖRIN TYÖ SOVELLUSTESTAUKSESSA

Joni Jauhiainen
Opinnäytetyö
Kevät 2024
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, Ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä(t): Joni Jauhiainen

Opinnäytetyön nimi: Laadunvarmistusinsinöörin työ sovellustestauksessa

Työn ohjaaja(t): Jukka Nevalainen, Petri Erälä

Työn valmistuslukukausi ja -vuosi: Kevät 2024

Sivumäärä: 30

Opinnäytetyössä käsitellään sovellustestauksen perusteita sekä seurataan kuuden viikon oppimis-päiväkirjalla sovellustestaajan arkea työviikkojen kuluessa. Työn tavoitteena oli tarjota sovellustestauksesta kiinnostuneille tiivis tietopaketti erilaisista sovellustestauksen menetelmistä sekä tarjota käytännön esimerkkejä sovellusten testaamisesta.

Työn teoriaosuus käsittelee sovellustestauksen peruskäsitteitä, kuten testausstrategioita, testaus-typpejä ja testauksen roolia ohjelmistokehityksessä. Lisäksi tarkastellaan erilaisia testaamisen vaiheita ja prosesseja, kuten testaussuunnittelua, testitapausten luomista ja testausraportointia. Oppimispäiväkirjassa seurattiin kuuden viikon jaksoa, joka koostui käytännön sovellustestauksesta ja sen ympärillä pyörivästä arjesta. Päiväkirjaan kirjattiin ylös havaintoja, oppimiskokemuksia, haasteita ja ratkaisuja, joita koettiin sovellusta testatessa.

Opinnäytetyö tarjoaa kattavan yleiskuvan sovellustestauksen perusteista sekä käytännön esimerkin siitä, miten sovellusten testausta voidaan harjoittaa todellisessa työympäristössä. Työn avulla uudet sovellustestaajat voivat nopeasti perehtyä alaan ja oppia hyödyllisiä käytäntöjä ja menetelmiä sovellusten laadun varmistamiseksi.

Asiasanat: sovellustestaus, päiväkirja, laadunvarmistusinsinööri

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Option of Software Development

Author(s): Joni Jauhiainen
Title of thesis: Quality Assurance Engineer's work in application testing
Supervisor(s): Jukka Nevalainen, Petri Erälä
Term and year when the thesis was submitted: Spring 2024
Number of pages: 30

This thesis offers an overview of software testing fundamentals, followed by a six-week learning diary documenting practical experiences in the field. The objective for this thesis was to give new software testers an easy-to-follow information package with essential insights to testing methodologies and real-world testing.

The thesis serves as a foundational guide to software testing supplemented by practical examples that allow new testers to grasp industry practices and methodologies for ensuring software quality.

Keywords: learning diary, QA-engineer, software testing

SISÄLLYS

1	JOHDANTO	6
2	SOVELLUSTEN TESTAUSMENETELMÄT	7
2.1	Yksikkötestaus	7
2.2	Integraatiotestaus	7
2.3	Systeemitestaus	9
2.4	Hyväksyntätestaus	10
2.5	Muut pienemmät testausmenetelmät	11
3	SOVELLUSTESTAUKSEN TYÖKALUT	13
3.1	Testaustyökalujen tyypit	13
3.2	Yleisimmät sovellustestauksen työkalut	13
3.2.1	Selenium	13
3.2.2	Appium	14
3.2.3	Postman	14
3.2.4	Katalon	15
4	PÄIVÄKIRJA	16
4.1	Viikko 1	16
4.2	Viikko 2	18
4.3	Viikko 3	20
4.4	Viikko 4	22
4.5	Viikko 5	24
4.6	Viikko 6	25
5	POHDINTA	28
	LÄHTEET	29

1 JOHDANTO

Opinnäytetyössä käsitellään sovellustestaajalle liittyvää teoriaa testausmenetelmien ja työkalujen muodossa sekä seurataan kolmen sprintin ja kuuden viikon ajalta sovellustestaajan arkea työn ympärillä. Luvussa kaksi käydään monipuolisesti erilaisia sovellusten testausmenetelmiä ja luvussa kolme käydään sovellustestaukseen liittyviä testaussovelluksia ja niihin liittyviä ohjelmointikieliä. Neljäs luku on sovellustestaajan päiväkirja, jossa seurataan ja kirjataan testaajan työtehtäviä ja niiden onnistumisia sekä mahdollisia puutteita ja ongelmia. Viidennessä luvussa on pohdintaosuus, jossa käydään päiväkirjaosuus läpi ja tarkkaillaan testausmenetelmien kehitystä kuuden viikon ajalta.

Toimeksiantajana toimii Invian Oy, joka kehittää helppokäyttöistä ja monipuolista toiminnanhjousjärjestelmää, joka on suunniteltu hoito- ja sosiaalialan tarpeisiin. Opinnäytetyössä sovellustestaaja käy läpi järjestelmän sovelluksia läpi Internet-, Flutter- sekä Android -ympäristöissä.

Töissä testaaja tarvitsee monipuolisia ongelmanratkaisu- ja ongelmanetsimistaitoja. Näitä taitoja testaajalla ei voi olla ikinä tarpeeksi, ja päiväkirjan avulla näitä on mahdollista parantaa entisestään miettimällä ja keskittymällä kehityskohteisiin. Tavoitteena opinnäytetyössä on viedä testausmenetelmiä ja -tapoja eteenpäin, koska työn sovellustestaaja on ollut alkuperäisesti yrityksen ensimmäinen sovellustestaukseen erikoistunut työntekijä ja yrityksessä ei ole keskitytty sovellustestaukseen täysipäiväisesti.

Päiväkirjaosuudessa seurataan työntekoa kahden viikon sprinteissä, joissa aina kahden viikon jälkeen sovelluksen uudet päivitykset esitellään muille tiimeille. Näissä palavereissa myös käydään tulevien kahden viikon kehitysaiheet ja mahdolliset ongelmat, mihin sovellustestaajat käyttävät resurssejaan.

2 SOVELLUSTEN TESTAUSMENETELMÄT

2.1 Yksikkötestaus

Yksikkötestaus on sovellusten luomisen ja kehittämisen kannalta tärkein osuus, koska sillä varmistetaan koodin laatu ja sen toiminta kooditasolla. Yksikkötestausta tehdään ohjelmistoon pieninä toiminnallisina yksikköinä. Siten koodia kehittäessä samoja testejä voidaan ajaa ja mahdollisten ongelmien löytyessä niillä on helppo paikallistaa ongelman lähde.

Yksikkötestauksen toiminnalliset yksiköt ovat koodilohkoja, jolla eristetään sovelluksen pienempi koodilohko ja tarkastaa, että tämä toimii kehittäjän teoreettisen logiikan mukaisesti. Testi pystyy olemaan vuorovaikutuksessa vain syötteiden ja tosi- tai epätosi -väittämien kanssa. (Amazon Web Services 2024.) Jos yksikkötestauksen koodilohko vaatii ulkopuolisia tietoja esimerkiksi tietokannoista, näiden testaaminen onnistuu käyttämällä datan paikkamerkkejä, kun yksikkötestaus toimii eristyksissä sovelluksen omista tietokannoista. Datan paikkamerkit ovat korvikkeita alkuperäiselle datalle, kun se on siirretty toissijaiseen tallennustilaan, ja tämä mahdollistaa yksikkötestauksen käyttämättä alkuperäistä dataa. (Komprise 2024.)

Yksikkötestauksia luodaan logiikkatarkastuksille, rajatarkistuksille, virheiden käsittelyyn ja olio-orientoituihin tarkistuksiin. Logiikkatarkastuksissa testataan suorittaako sovellus oikeat laskelmat ja seuraa oikeaa polkua saadessaan odotetun syötteen. Rajatarkistuksissa katsotaan miten sovellus reagoi annettuihin syötteisiin. Syötteet ovat yleensä sovelluksen haluamissa rajoissa, reunatapauksia tai virheellisiä. Jos sovellus odottaa saavansa syötteen 1–10 väliltä, reunatapaus tähän on 1 tai 10 ja virheellinen syöte on luku arvojen 1–10 ulkopuolelta. Virheiden käsittelyllä tutkitaan miten järjestelmä reagoi, kun syötteessä on virhe ja olio-orientoiduissa tarkistuksissa katsotaan, päivittykö olio oikein, jos koodin suorittaminen muuttaa pysyvän olion tilaa. (Amazon Web Services 2024.)

2.2 Integraatiotestaus

Integroiointitestaus on testausta, jossa sovelluksen kaksi eri komponenttia asteittain integroidaan eli yhdistetään toisiinsa ja tämän jälkeen testataan yhdistettynä ryhmänä (Katalon 2024). Monesti

nämä yksittäiset komponentit toimivat yksinään hyvin, mutta yhdistäessä esiintyy ongelmia, johon integraatiotestaus on välttämätöntä. Ongelmat voivat johtua komponenttien eri versioista, tietomuotojen konflikteista ja erilaisista käsittelylogiikoista. Integraatiotestaus tapahtuu yleensä yksikkötestauksen jälkeen ja tällöin tarkentaa nämä ongelmat komponenttien välillä ennen järjestelmätestausta. (Katalon 2024.)

Ohjelmistot koostuvat monesti useista yksittäisistä ohjelmistokomponenteista tai moduuleista. Moduulit voivat yksikkötestauksessa olla toimivia yksinään, mutta yhdistettäessä ne kaatuvat erisyistä. Eri ohjelmoijien koodatuissa moduuleissa voi olla erilaiset logiikat ja lähestymistavat kehitykseen, joten niiden integroimisessa voi esiintyä toiminnallisia tai käytettävyyso ongelmia. Integraatiotestaus varmistaa näiden komponenttien koodien olevan yhtenäistä, mikä johtaa toimivaan sovellukseen.

Sovelluksen muuttuvat vaatimukset esimerkiksi asiakkaan toimesta ovat yrityksillä iso syy, joka vaatii integraatiotestausta. Toiveet muuttavat yhden moduulin koodin muokkaamista, mikä voi aiheuttaa koodilogiikan muuttamista kokonaan ja tämä näkyy koko sovelluksen toimivuudessa. Nämä muutokset ovat juuri niitä, jotka eivät näy yksikkötestauksissa, mutta integraatiotestauksen alkaessa virheet paljastuvat. Tiedon muuttuminen voi tapahtua moduulien välillä tietoa siirrettäessä, ja jos tieto ei ole oikeassa muodossa siirron aikana, sitä ei voi lukea ja käsitellä. Tämä johtaa bugeihin. Esimerkiksi API-, eli komponenttien ja moduulien välinen raja järjestelemässä, ja kolmannen osapuolen palvelut voivat saada virheellistä syötettä tästä tiedonsiirron ongelmasta ja virheelliset syötteen luovat virheellisiä vastauksia. (Alfame 2018.) Kehittäjät koodia luodessaan ottavat huomioon poikkeuskäsittelyn, mutta joskus tarpeelliset käsittelyt menevät kehittäjiltä ohi ja näitä kaikkia ei voida nähdä ennen kuin moduulit on koottu yhteen.

Viimeisenä yleisenä syynä integraatiotestaukseen on ulkoisten laitteistojen vaatimat rajapinnat, missä ohjelmisto-laitteisto-yhteensopimattomuutta ei havaita kuin vasta asianmukaisella integraatiotestauksella (Katalon 2024).

Integraatiotestauksen parhaimmat lähestymistavat ovat Big Bang -lähestymistapa ja inkrementaalinen lähestymistapa (Valagroup 2023). Big Bang -lähestymistavassa integraatiotestausta tehdään vasta, kun kaikki tai suurin osa komponenteista ja moduuleista on kehitetty. Komponentit yhdistetään ja testaus suoritetaan kokonaisuutena järjestelmälle. Tämä yksinkertaistaa testausta ja on nopeampi aloittaa, jos komponentit ovat valmiita. Testaus sopii pienemmille projekteille,

missä komponentteja on pienempi määräkin. Kyseinen tapa aiheuttaa yksittäisten ongelmien tunnistamisen vaikeutta ja eristämistä ison integraation takia. Testaajille myös voi aiheutua tehottomuutta, kun testaamiseen pääsee vasta, kun kaikki moduulit ovat valmiita.

Inkrementaalinen lähestymistapa on, jossa ohjelmisto on jaettu pienempiin moduuleihin tai komponentteihin, jotka testataan vaiheittain. Tätä yleensä käytetään isompien projektien parissa, missä moduulit ja komponentit valmistuvat niin isolla aikavälillä, ettei kaikkien testaaminen kerralla olisi hyödyllistä ajankäytön ja ongelmien löytämisen kannalta. Yksittäiset ongelmat on helppo eristää ja yksittäiset komponentit voidaan testata ja korjata ennen niiden täyttä integrointia. Inkrementaalinen lähestymistapa vaatii kuitenkin huolellista suunnittelua, että yhdistetyt komponentit ovat aina toisiinsa sopivia testauksen kannalta. Integraatio myös voi viedä enemmän aikaa useampien testausten takia. Valinta testaustapojen välillä riippuu testauksen laajuuden, monimutkaisuuden ja aikataulun sekä tiimin mieltymyksistä. (Katalon 2024.)

2.3 Systemitestaus

Systemitestaus, eli järjestelmätason testaus tai järjestelmäintegraatiotestaus, on prosessi, jossa laadunvarmistustiimi testaa, miten sovelluksen eri komponentit toimivat yhdessä kokonaisessa järjestelmässä tai sovelluksessa. Systemitestauksella varmistetaan, että sovellus suorittaa tehtävät suunnitellusti keskittymällä sovelluksen toiminnallisuuteen eikä järjestelmän sisäisiin toimintoihin. Järjestelmätestauksella voidaan esimerkiksi testata, että jokainen käyttäjän syöttämä arvo tai muu syöte tuottaa oikean tuloksen sovelluksen läpi. Systemitestaus on kolmas testausvaihe ohjelmistonkehitysprosessissa ja suoritetaan yleensä integraatiotestauksen jälkeen. (Techtarget 2024.)

Systemitestaus on välttämätön toimenpide järjestelmille, koska mikä tahansa vika ohjelmistossa voi aiheuttaa äärimmäisiä ongelmia kuluttajille sekä suuria kuluja kehittäjälle. Esimerkkinä vuonna 2014 Nissan joutui kutsumaan takaisin yli miljoona autoa turvatyynyn sensorivian vuoksi. (Techtarget 2024). Systemitestauksen hyötyinä ovat vähentyneet virheet ja täten parantunut tuotteen laatu. Joitain virheitä ei löydetä yksikkö- ja integraatiotestauksen aikana niin systemitestaus löytää vielä virheitä, jotka aiempien mainittujen hyötyjen lisäksi myös säästävät kustannuksia ja parantaa tuotteen turvallisuutta ja/tai luotettavuutta. Virheet voivat johtaa tuotteen myynnin menetyksiin, sovelluksen kehittämiseen käytetyn ajan lisääntymiseen ja täten auttaa projektinvetäjää helpompaan budjetointiin. Systemitestauksella tehdään myös havaintoja mahdollisten koodiongelmien kanssa kehityksen aikana ja tämä estää tulevaisuudessa jo sisään ajetun koodin tarpeellista muokkauksista, joka voisi olla vaikeampaa siinä vaiheessa kuin sovelluksen vasta ollessa kehitteillä.

Systeemitestausta tehdään, jotta saadaan selville sovelluksen suorituskykyä, nähdäkseen miten nopeasti sovellus toimii, keskimääräiset latausajat ja sovelluksen vakautta ja vastausaikoja erilaisten rasiusten alla. Sovellusta testatessa myös katsotaan, millaisen kokemuksen käyttäjä saa tästä ja miten helppo sovellusta on käyttää. Tätä testaamista tukevat erilaiset tiedot, kuten käyttäjän virhearvot, sovelluksen tehtävien onnistumisprosentit, käyttäjän käyttämät ajat tehtävien suorittamiseen ja käyttäjätyytyväisyydet. Systeemitestauksessa tehdään myös regressiotestausta, eli testataan, ovatko testauksen aikana tai äskettäin tehdyt koodimuutokset luoneet uusia ongelmia jo valmiisiin komponentteihin. Regressiotestaus on siis helpommin sanottuna jo valmiiksi testattujen asioiden testaamista.

2.4 Hyväksyntätestaus

Hyväksyntätestaus on menetelmä, jossa järjestelmä testataan sen toimivuus yrityksen vaatimuksiin nähden ja arvioidaan, onko se jo valmis toimitukseen. (Geeksforgeeks 2024). Testausta suoritetaan käyttäjien tarpeiden, vaatimusten ja liiketoimintaprosessien mukaisesti, joilla voidaan selvittää täyttääkö sovellus käyttäjän hyväksyntäkriteerit ja mahdollistaa valtuutettujen tahojen päättää, onko järjestelmä hyväksyttävissä.

Hyväksyntätestausta on useita erilaisia, yleisimpänä käyttäjän hyväksyntätestaus, missä selvitetään, toimiiko tuote käyttäjän kannalta oikein. Testaus keskittyy pääasiassa tarkkoihin vaatimuksiin, joita asiakkaat käyttävät. Liiketoiminnan hyväksyntätestaus selvittää, täyttääkö tuote liiketoiminnan tavoitteet ja käyttötarkoitukset. Testaus keskittyy pääasiassa liiketoiminnan voittoihin, mikä voi olla haasteellista markkinan ja uusien teknologioiden muuttuessa, jolloin nykyistä toteutusta joudutaan muokkaamaan, joka johtaa ylimääräisiin kustannuksiin. (Geeksforgeeks 2024.)

Sopimuksen hyväksyntätestaus on menetelmä, jossa luodaan tuotteen käyttöön otossa sopimus, joka määrittää ennalta määrätyn ajan, jonka kuluessa tuote testataan ja tämän on läpäistävä kaikki hyväksyntäkäyttötapaukset. Sopimuksen kanssa kehittäjälle maksetaan sovelluksesta vain, jos tuote on linjassa vaatimusten kanssa. Sopimuksessa pitäisi löytyä testauksen ajanjakso, testausalueet, myöhemmässä vaiheessa ilmenevien ongelmien ehdot ja maksut selvästi määriteltynä molemmille osapuolille. Toiminnallista hyväksyntätestausta käytetään tuotteen toiminnallisen valmiuden määrittämiseen ja se on ei-toiminnallista testausta. Tähän sisältyy pääasiassa järjestelmän palauttamisen, yhteensopivuuden, ylläpidettävyyden ja luotettavuuden testausta. Toiminnallinen

hyväksyntätestaus varmistaa tuotteen vakauden ennen sen julkaisua tuotantoon. (Geeksforgeeks 2024.)

Alfa- ja betatestaus ovat hyväksyntätestausmenetelmiä, joissa alfatestauksessa testaus keskittään kehitysympäristöön ja testauksesta vastaa erikoistunut testaajatiimi. Betatestauksessa tuote altistetaan sen todellisille loppukäyttäjille, palautetta kerätään käyttäjiltä ja näiden avulla korjataan mahdolliset virheet. Betatestaus myös auttaa parantamaan ja korjaamaan käyttäjäkokemusta.

Hyväksyntätestauksissa, kuten kaikissa muissa menetelmissä, on etunsa ja haittansa. Testaus auttaa projektitiimiä saamaan vaatimuksia suoraan käyttäjiltä, koska se osallistaa käyttäjiä testaukseen ja samalla tuo luottamusta ja tyytyväisyyttä asiakkaille, kun he saavat olla mukana testausprosessissa. Hyväksyntätestauksen avulla käyttäjän on myös helpompi kuvata tarpeensa, kun koko sovellus ja sen toiminnallisuus testataan. Testausprosessi myös voi vaikeutua, kun käyttäjät eivät aina halua osallistua testausprosessiin ja mahdollinen palaute osallistuneilta testaajilta voi kestää kauan, kun kaikilla on eriäviä mielipiteitä niin niiden kasaaminen yhteen vie aikaa.

2.5 Muut pienemmät testausmenetelmät

Muita pienempiä, mutta yhtä tärkeitä sovellustestausmenetelmiä, on muun muassa, staattinen ja dynaaminen analyysi, black ja scripted box -menetelmät ja agile. (Stackify 2023). Staattisessa analyysissä ei varsinaisesti suoriteta koodia vaan tutkitaan mahdolliset tapahtumat, jotka saattavat ilmetä suorituksen aikana. Ohjelmistotestaajat suorittavat staattista analyysiä tarkastelemalla asiakirjoja, jotka sisältävät määrittelyvaatimukset, mallit, suunnittelun arkkitehtuurin ja perintökoodit. Staattiseen analyysiin kuuluu koodin tarkastelu ja tarkastus, algoritmin analyysi ja oikeellisuuden todistaminen.

Dynaaminen analyysi sisältää ohjelman suorituksen virheiden ja toiminnallisuuksien epäonnistumisten löytämiseksi ja samalla kattaa ohjelmiston käyttäytymis- ja suoritusominaisuudet. Kun staattinen analyysi on virheiden tunnistamista ohjelmistonkehityksen varhaisessa vaiheessa ennen koodin ajoa, niin dynaaminen analyysi toimii hyvin rinnakkain kehityksen kanssa.

Black box -menetelmä käsittelee sovellusta kuin mustana laatikkona, eli testaajat suorittavat testausta tietämättä ohjelmiston kokonaisrakennetta. Prosessi noudattaa yksinkertaista syötä-tulos -mallia missä testaaja tietää tarvittavat syötteet ja odotetun tulosteen, mutta ei prosessia. Black box

-testaus testaa sovelluksen toiminnallisuutta vaatimusmäärittelyn perusteella ja ei keskity sovelluksen koodiin tai muuhun sisäiseen rakenteeseen mitenkään.

Scripted box -menetelmässä testaaja seuraa suunnitelmaa eli ns. skriptiä. Tämä on olennainen osa testausmenetelmää, koska skripti sisältää testitapaukset ja niihin liittyvät testivaiheet. Nämä suunnitellut vaiheet ovat suunniteltuja täyttämään tietyt vaatimukset tietystä osasta ohjelmaa. Tätä testausta voi käyttää helposti toiminnallisen ja ei-toiminnallisen testauksen tukena. (Stackify 2023.) Agilea voi pitää nykyään tunnetuimpana testausmenetelmänäkin, vaikka sana yleensä liitetään ketterän ohjelmistokehityksen periaatteisiin. Kuin kehityksessä, agile-testaus toimii sprinteissä ja testaussuunnitelmissa tapahtuen ohjelmistokehityksen kanssa samanaikaisesti. Ohjelmistokehityksen muuttuessa enemmän testivetoiseksi, enemmän tiimejä alkaa adoptoimaan agile-menetelmää ja tämä kannustaa testitiimejä testaamaan aikaisemmissa vaiheissa kehitysprosessia.

3 SOVELLUSTESTAUKSEN TYÖKALUT

3.1 Testaustyökalujen tyypit

Sovellustestauksen työkalujen tyyppiä on kahdenlaisia, staattisia ja dynaamisia testityökaluja. Molempia on neljä erilaista ja kaikilla on eri tehtävänsä sovellustestauksen kanssa. (GeeksforGeeks 2024.) Staattisilla testityökaluja käytetään sovellusten staattisiin testausprosesseihin. Nämä eivät testaa sovelluksen todellista suorittamista ja eivät vaadi tiettyjä syötteitä. Staattisia testityökaluja ovat flow analyzers, mitkä tarjoavat joustavuutta datan siirrossa syötteestä lähtöön. Path tests, millä löydetään käyttämätön koodi ja epäjohdonmukaisuudet ohjelmistossa. Kattavuusanalyysillä varmistetaan, että kaikki järkevät polut testataan ohjelmistossa ja rajapinta-analyysillä tarkistetaan muuttujien ja datan siirron seuraukset moduulien välillä. (GeeksforGeeks 2024.)

Dynaamiset testausprosessit suoritetaan dynaamisilla testityökaluilla. Nämä testaavat ohjelmistoa olemassa olevilla tai ajantasaisilla tiedoilla. Näihin työkaluihin kuuluvat testiajurit, jotka tarjoavat syöttötietoja testattavalle moduulille. Kokeilualustat, joka näyttää lähdekoodin samanaikaisesti suoritettavan ohjelman kanssa. Emulaattorit, jotka tarjoavat vastausominaisuuksia jäljittelemään järjestelmän osia, joita ei ole vielä kehitetty sovellukseen ja mutaatioanalyysit, joita käytetään testaamaan järjestelmää antamalla tarkoituksella virheitä sovelluksen koodiin. (GeeksforGeeks 2024.)

3.2 Yleisimmät sovellustestauksen työkalut

Luvussa käydään läpi nykypäivän suosituimmat sovellustyökalut QA-timien arjessa.

3.2.1 Selenium

Selenium on ilmainen automatisoidun web sovellusten testaamiseen käytetty avoimen lähdekoodin työkalu. Seleniumia käytetään kirjoittamalla skriptejä, jotka matkivat käyttäjän toimintoja sovelluksessa ja nämä voidaan automatisoida ajettavaksi sovellukseen. (Selenium 2024.) Skriptien ansiosta regressiotestaaminen on hyvin suosittua Seleniumin kanssa. Työkalun avainominaisuudet

ovat sen lisäksi mahdollisuudet testata eri nettiselaimissa helposti samoilla skripteillä. Usean ohjelmointikielen tuki, eli Seleniumin kanssa voidaan työskennellä Javalla, Pythonilla, C#, Rubylla sekä JavaScriptillä, mikä mahdollistaa käyttäjän valitsemaan sopivin ja ammattitaitoisin kieli. Selenium on myös helppo integroida käytettäväksi muiden työkalujen ja frameworkkien kanssa, kuten TestNG, JUnit ja Jenkinsin kanssa, parantaen sen testausmahdollisuuksia sovellusten ympärillä. (Selenium 2024.)

3.2.2 Appium

Appium on ilmainen mobiilisovellusten testaukseen erikoistunut työkalu, mikä mahdollistaa sovelluksen testaamisen useilla alustoilla ja laitteilla. (Appium 2024). Kuten Seleniumissa, Appium on suosittu sen monipuolisuuden alustojen ja laitteiden valintojen määrän välillä. Android, iOS ja Windows sovellukset toimivat samoilla skripteillä. Myös Appium on suosittu sen eri ohjelmointikielten mahdollisuuksien takia. Sovellus tukee Javaa, Pythonia, Rubya, Javascriptiä ja muita useita ohjelmointikieliä, että käyttäjällä on mahdollisuus tässäkin työskennellä oman kielensä kanssa. Appium ei myöskään vaadi muokkaamaan sovellusta millään tavalla tai lisäämään ekstra koodia tai kirjastoja käyttäkkeen automatisoituja skriptejä. (Appium 2024.)

3.2.3 Postman

Postman on suosittu alusta ohjelmointirajapintojen kehitykseen. Se yksinkertaistaa kehitysprosessin, testaamisen ja rajapintojen dokumentoinnin. (Postman 2024.) Rajapinnat mahdollistavat eri sovellusten kommunikoinnin toisensa kanssa ja Postman tarjoaa käyttäjäystävällisen liittymän näiden toimintaan. Testauksen kannalta Postmanilla voi luoda skriptejä JavaScriptiä käyttäen vahvistaakseen käyttäytymisen rajapintojen pyynnöistä automaattisesti. Postman voi luoda näistä automatisoiduista skripteistä dokumentaatioita sinun rajapinnastasi, joka mahdollistaa muiden kehittäjien helpon tutustumisen ja oppimisen rajapinnastasi. (Postman 2024.) Postmanilla voi luoda myös koeservereitä, jotka simuloivat rajapintojen vastauksia ja ei vaadi, että oikean rajapinnan pitää olla koko ajan toiminnassa.

3.2.4 Katalon

Katalon on alustariippumaton testaus automaatioon kehitetty alusta suunniteltu tukemaan netti-, mobiili- ja työpöytäsovelluksien testaamista. Katalon on hyvin käyttäjäystävällinen, mikä tekee siitä helposti käytettävän aloittajista kokeneempiin kehittäjiin. (Katalon 2024.) Sovelluksen monipuolisuus alustojen ja käyttöliittymien kanssa tekee siitä suositun yhtenä työkaluna kaiken testaamiseen. Sovelluksen ominaisuus, mitä muilla kilpailijoilla ei ole, on syöttöjen nauhoitustoiminto, jolla voit nauhoittaa tekemäsi asiat sovelluksessa ja sen jälkeen toistaa ne automatisoituina testeinä. Kokeneimmille käyttäjille on mahdollisuus kirjoittaa omat skriptinsä käyttäen Groovy-ohjelmointikieltä, joka on Javan syntaksiin yhteensopiva kieli. Katalon myös tarjoaa mahdollisuuden kirjoittaa testitapauksia luonnollisella kielellä, mikä on helppoa ymmärtää myös ei-tekniikasta näkökulmasta. Tätä ei muut testaus työkalut tarjoa käyttäjälle, mikä tekee sovelluksesta hyvin erilaisen verraten kilpaileviin sovelluksiin. (Katalon 2024.)

4 PÄIVÄKIRJA

Päiväkirjaosuudessa seuran omia työtehtäviäni ja kirjaan niiden mahdollisia kehityskohteita. Työn seuraamiseen päiväkirja on todella hyvä keino seurata, mitä tehdään työpäivien ja -viikkojen aikana. Mielestäni kuuden viikon päiväkirja antaa enemmän kuin tarpeeksi mahdollisuuksia nähdä mikä menee töissä hyvin ja missä olisi parantamisen varaa. Päiväkirjaosuuden on kirjoitettu blogi-tyylisellä tekstillä toisin kuin teoriaosuus, joka oli akateemista tekstiä.

4.1 Viikko 1

Viikko on sprintin 66 aloitus ja luvassa tulee olemaan lyhyt viikko torstain itsensäkehittämisspäivän ja pitkäperjantain takia, mutta ei anneta tämän häiritä työntekemistä kovin paljoa. Odotettavissa on hiukan kiireisempi ja työntäyteisempi viikko tämän takia, mutta onneksi tekee töitä mistä todella tykkää.

Maanantai 25.3.2024

Päivä alkoi katsomalla sprinttitaululta testausalustassa olevat tiketit ja käymällä läpi mahdollisia tikettejä, jotka ovat minun testialueellani. Viime viikolla aloitin tikettiä, mikä koskee uusia lastensuojelulomakkeita. Kun ei ollut uusia kiireellisiä tikettejä, jatkan näiden lomakkeiden testaamista ja lomakkeiden kehittämis- ja korjaamisalueiden kirjaamista. Puolenpäivän jälkeen sain lomakkeet katsottua kunnolla läpi ja jaettua muistiinpanot kehittäjille tulevaisuutta varten. Lounaan jälkeen aloin katsomaan tikettiä tehtävienhallinta bugista, missä tehtävän data ei päivittynyt oikein tietyssä näkymässä. Tämä bugi oli korjattu ja päivän kuluessa tutkiessa samaa näkymää huomasin uuden ongelman, missä asiakkaan nimi oli kadonnut ja tästä kirjoitin nopean tiketin, joka siirtyikin heti sprintille korjattavaksi. Päivän loppuun vielä oli kaksi tikettiä samasta ongelmasta asiakkaan lomakkeen tulostamisesta back- ja frontendissä. Tutkittuani ja kommentoituani nämä korjatuiksi pystyin lopettamaan päivän.

Tiistai 26.3.2024

Eilen työpäivän jälkeen oli tullut kommenttia, että lastensuojelulomakkeiden muistiinpanot pitäisi tiketöidä erikseen niin tästä tuli suoraan päivän ensimmäinen työtehtävä. Tämän kanssa meni hiukan pitempi tovi, että sai muistiinpanot käännettyä englanniksi kunnolla tikettiin. Sitten piti lähteä toimistolle päin kehityskeskusteluun ja testilomakkeiden suunnitteluun QA-tiimin kanssa. Kehityskeskustelu siirtyi myöhemmälle ajankohdalle, mutta näitä sattuu aina sovelluskehitysfirmissä, että muut asiat sovelluksen ympärillä ovat paljon tärkeämpiä. Testilomakkeet ja uusi testiskenaarioitten luonti saatiin hienosti alkuun ja saattaa olla, että loppuviikon työaika käytetään näiden tekemiseen.

Keskiviikko 27.3.2024

Päivä meni kokonaisuudessaan alusta loppuun testiskenaarioitten ja testicasejen suunnittelemisen ja dokumentoinnin kanssa. Pienen tauon puoliltapäivin otin katsoakseen yhden Android-tiketin asiakasraportoinnista. Tiketin ja siihen liittyvän sovelluksen tutkimiseen meni noin tunti ja loppu päiväksi pääsi takaisin testauksen dokumentointiin.

Torstai 28.3.2024

Kuukauden viimeisin torstai on tiimillämme PDD, eli personal development day, ja päivän aikana seurasin Ottomated-nimisen sovelluskehittäjän suoraa lähetystä, jossa hän kehitti sovellusta Rus-tilla tehdäkseen Twitch-chatista "vähemmän ärsyttävän" hänen sanojensa mukaan. Sovellus ei tullut valmiiksi, mutta hänen sanojensa mukaan sovellus sensuroisi ja muokkaisi sanoja ja tekisi sovelluksen ulkonäöstä ihan erilaisen Twitchin tarjoaman chatsovelluksen sijasta. Tein myös tutkimusta NAS-palvelimista, kun kaveri haluaisi hankkia sellaisen, mutta ei ole oikein näppärä tietotekniikan kanssa. Käytin TechTargetin ja -Radarin artikkeleita asiasta lukemiseen ja LinusTech-Tipsin videoita nähdäkseen miten tällaisen voisi rakentaa. Nämä päivät ovat todella mukavia työarjen keskelle, kun saa niin sanotusti luvan kanssa opiskella ja tutustua mahdollisiin asioihin mitkä kiinnostaisivat. Tein myös työtä opinnäytetyön parissa hetken.

Perjantai 29.3.2024

Loma.

Yhteenveto

Lyhyt viikko takana, mutta päiväkirjan ensimmäinen viikko oli vähän kirjoittamisen puolesta hake-
mista, kun ei oikein tiennyt mitä kaikkea laittaa tekstiksi. Viikon kuluessa tämäkin alkoi hahmottu-
maan paremmin ja nyt yhteenvetoa kirjoittaessa huomaa, että tämäkin onnistuu paljon paremmin.
Viikon kohokohtana oli saada testidokumentointi alkuun ja rullaamaan, kun näitä meidän yrityk-
sessämme ei ole vielä ollut ollenkaan. Dokumentoinnin kanssa voi olettaa, että menee todella pit-
käkin aika ennen kuin nämä ovat todella hyödyllisiä testauksen kannalta. Kunhan dokumentit val-
mistuvat, niin voin olla aika varma, että niiden tekeminen on sen arvoista testauksen ja bugien ra-
portoinnin kannalta.

4.2 Viikko 2

Sprintin toinen viikko, joka on taas lyhyempi pääsiäisen takia. Odotettavissa on paljon testidoku-
mentoinnin suunnittelua ja tekemistä ja sprintin päättyessä perjantaina niin voi myös odottaa pal-
jon tikettejä, jotka vaativat testausta varmistaakseen korjaukset.

Maanantai 1.4.2024

Loma.

Tiistai 2.4.2024

Päivä alkoi tutustumalla palkanlaskennan isoon bugitickettiin, missä palkanlaskut sisälsivät liikaa
dataa siihen mitä haussa yritettiin hakea. Tiketissä meni melkein koko päivä ja tämän lisäksi löy-
sin uuden bugin, missä yksi korjaus palkanlaskussa näkyy vain UI:ssa kahdesti, vaikka se ei ole
backendistä haettua tietoa kuin vain kerran, mikä pitääkin. En saanut toistettua tätä bugia helposti
ja lyhyt loppupäivä meni tätä etsiessä ilman tulosta. Suoraan sanottuna ärsyttävää, että löytää
bugin, mutta ei tiedä miten sen toistaa. Onneksi kuitenkin tämä ei ole mitenkään sovellusta rik-
kova bugi, niin jos asiakas tämän löytää ja tietää miten toistaa, siinä ei synny mitään vahinkoa ke-
nellekään.

Keskiviikko 3.4.2024

Keskiviikko laitettiin käyntiin käymällä tiketit testing-listalta läpi ja sieltä bongattiin yksi minkä saa testaukseen. Ennen tämän testausta otettiin kuitenkin palaveri QA-tiimin kanssa tulevista testaus-alueista sovelluksen ympärillä ja puhuttiin näistä aiemmin tehdyistä testicaseista, että mitä kukin tulevaisuudessa alkaa tekemään. Meillä yrityksessä tämä QA-tiimi on vasta ihan alkutekijöissään, niin toiminnan suunnitteleminen ja parantaminen on kyllä todella mielekäästä työtä, kun tämän edistyksen näkee todella nopeasti tämänlaisessa työympäristössä. Hyppäsin tämän päivän ainoan tiketin pariin ennen lounasta ja ennen viime viikolla peruuntunutta kehityskeskustelua. Kehityskeskustelu oli elämäni ensimmäinen, mihin olin osallisena ja tämä oli todella mieltä ja oloa parantava kokemus. Mietti ja keskusteli asioita mihin ei ole koskaan kiinnittänyt huomioita erikseen. Tämän jälkeen oli hyvä hypätä Flutter-sovelluksen pariin isojen työvuorotikettien pariin. Näiden tutkiminen jatkuu huomiseenkin vielä.

Torstai 4.4.2024

Päivä jatkui eilisen Flutterin työvuorotiketin tutkinnasta. Meidän Flutterin kehittäjä oli kommentoinut tikettiin, että toinen kollega oli tutkinut tiketin jo, mutta voisin yrittää löytää mahdollisia ongelmia tästä. Tiketin suurimmat korjaukset eivät näkyneet sovelluksessa aiemmasta tutkinnasta huolimatta ja tästä juteltiin kehittäjän kanssa, jonka jälkeen hän otti löytämäni ongelmat tutkintaan. Flutterin läpikäytyä siirryin lounastauolle, jonka jälkeen vaihdoin Android sovellukseemme ja samanlaisiin työvuorotiketteihin, mitä Flutterissa oli. Androidissa oli kirjautumisongelmia yhdessä kehitystilassa, minkä takia en voinut ihan uusimpia tikettejä tutkia täällä ja tästä laitoin viestiä kehittäjille eteenpäin, mutta RC-julkaisussa (Release Candidate) pystyin katsomaan pari vanhempaa pientä tikettiä läpi. Meillä järjestetään Vismalla etätaukojumppia niin joka torstai pääsee heilumaan vartiksi tämän tahtiin. Tätä odottaa joka viikko mielellään, koska heidän ohjaajansa on todella rauhallisia ja vaikka onkin simppeleitä asioita mitä ohjaavat niin helpompi tehdä mukana, kuin yksin kotona. Vismalla on myös usein kaikenlaisia puheita ja esityksiä ja päivän lopetukseen työnhelle pääsi kuuntelemaan ”Työn merkityksellisyyttä metsästäessä” -Inspirational Talkkia. Päivä päättyi lääkintälistan käyttöoikeus bugin tutkimiseen ja huomisen voi aloittaa puhtaalta pöydältä

Perjantai 5.4.2024

Aurinkoinen perjantai alkoi palaamalla eilisen Flutterin tiketin pariin, kun korjaukset oli saatua ajettua uudestaan sovellukseen. Tutkittuani tiketin uusiksi löysin, että yksi asetukset oli jäänyt väärään oletusarvoon, mutta tästä kollegalle viestiä laitettua meni minuutti ja tiketti toimi täysin oikein ja pystyin sulkemaan tiketin. Hyppäsin tämän jälkeen uuden Flutterin tiketin kimppuun, mikä liittyi melkein samaan asiaan, mutta sisälsi uusia asetuksia, joiden läpikäymisessä meni pitempi hetki. Tiketin jälkeen pääsi suoraan sprinttipalaveriin summatakseen menneen sprintin tiimien kanssa, jossa menikin puoli päivää melkein yhdessä lounaan kanssa. Palavereiden jälkeen suunnittelin tulevaa sprinttiä ja katselin tulevia tikettejä mitä ensi viikolla mahdollisesti valmistuu. Tämän jälkeen pystyinkin hyppäämään viikonlopun viettoon ja lepäämään tulevaa viikkoa varten.

Yhteenveto

Viikko oli hyvin puhelinsovellus painotteinen, mutta tämä alkaa olemaan jo erikoistumispuoltani yrityksessä. Olen ainoa testauttiimissämme, joka on tätä paljon käyttänyt niin ihan hyvä, että voin testata Flutterin ja Androidin. Tietokoneen selainversio jää kollegoille testaukseen, niin mikään ei jää testaamatta. Hieno nähdä, kun työskentelee, että kuinka helposti ja nopeasti kollegoihin saa yhteyttä kaikessa tarvittavassa. Flutterin kehittäjä oli viikon aikana muutamaankin otteeseen kyselyjen alla pienistäkin asioista niin aina sai vastauksen, kun sitä tarvitsi. Mainitsin tuosta kehityskeskustelustakin, että miltä se tuntui, niin nyt mietin vain miksi näitä keskusteluja ei ole aiemmin pidetty ollenkaan. Mutta onneksi näitä nyt alkaa olemaan, niin tämä on todella piristävä lisäys työympäristöömme.

4.3 Viikko 3

Kohta ollaankin päiväkirjan puolella välissä ja tälläkin viikolla odotettavissa on paljon suunnittelu- ja dokumentoinnin tekemistä. QA-tiimin muovautuessa omiin rooleihin ja ominaisuuksiin, niin tämän puolesta oma roolikin selkeytyy ja työt helpottuvat, kun saa omat osat mihin keskittyä kunnolla.

Maanantai 8.4.2024

Aamu alkoi testaustikettien läpikäymisellä, mihin ei uutta ollut ilmestynyt viime viikon lopetuksen jälkeen ja aloin kirjoittamaan Flutterin testidokumentointia. Hetken kirjoiteltua tulikin lähtö toimistolla käymään QA-tiimin palaverissa, jossa keskusteltiin tiimin roolituksesta ja työtehtävistä dokumentoinnin suhteen. Kaikille jaettiin omat vastuualueet alkuunsa ja tulevaisuudessa palaverissa määritellään uusia alueita, kun nämä saadaan tehtyä. Loppupäivä menikin Flutterin tehtäväsvun ja tehtävien lisätietojen testidokumentoinnissa. Flutterin dokumentoinnin kanssa voikin mennä melkein koko viikko.

Tiistai 9.4.2024

Päivä menikin alusta loppuun Flutterin testidokumentoinnin kanssa. Odotettavissa huomiseksi on myös samaa, kun testicasejen suunnittelu ja tekeminen on isompi projekti mitä alkuunsa ajatteli.

Keskiviikko 10.4.2024

Päivä alkoi testidokumentoinnin parissa taas, jonka kimpussa menikin heti muutama tuntia. Hetken tauon otin tämän jälkeen katsoakseen yhden testi tilassa olevan tiketin kalenteriin ja kalenterimerkinnän viimeistelyyn liittyen. Tästä löytyi yksi isompi ongelma, mitä en kerennyt dokumentoida tiketiksi ennen iltapäivän palaveria. Palaverissa keskusteltiin Flutterin UI:ta koskevista ongelmista ja tyylivioista. Itse olin vain enemmän kuuntelevassa roolissa tässä palaverissa, että voin tietää mahdollisista tulevista tyyli muutoksista ja myös mahdollisesti miettiä miten kyseiset ongelmat voisi ratkaista paremmin. Palaverin jälkeen kuvasin testivideon löytämästäni ongelmasta ja kirjoitin bugiticketin, jonka laitoin eteenpäin kehittäjälle ennen päivän lopetusta.

Torstai 11.4.2024

Päätin tuossa heti herättyäni, että tänään on välipäivä testidokumentoinnista, kun eilen jo alkoi vähän tuntumaan niiden olevan jo väkinäistä. Olen huomannut ainakin omassa tekemisessä sitä, että jos hetken pitää taukoa isommasta projektista niin seuraavana päivänä tekeminen on paljon tehokkaampaa ja saa tehtyä aiemman päivän kiinni uudella mielellä. Uusia tikettejä ei ollut tullut

testitilaan, mutta kollegat ja asiakkaat ovat puhuneet, että sovelluksen varausosio on ollut ongelmallinen, niin käytin tämän päivän auttaakseen kollegaa testauksessa. Tänään oli myös torstai eli jumppapäivä iltapäivästä, niin sai tehokkaan tauon venyttelyihin ja kehon avaamiseen.

Perjantai 12.4.2024

Perjantai työpäivän osalta koostui ensimmäisen Flutter testiajon pyöryttämisestä ja tutustumalla, miten käytännössä ongelmat merkataan tulevia sprinttejä miettien, että mitkä ongelmista ovat sprinteille nostettavia. Tämän jälkeen sprinttitaululle oli noussut kaksi uutta tikettiä testaustilaan. Yksi pieni Flutterin raporttien suodattamiseen ja toinen isompi palkanlaskun yhteenvedosta. Nämä olivat simppeleitä, eikä noussut mitään erikoista sanottavaa niin pystyi näiden jälkeen jatkamaan suoraan testicasejen kirjoitteluun taas. Siitä oli hyvä myös aloittaa viikonlopun vietto.

Yhteenveto

Puolet päiväkirjasta mennyt kyllä ihan hetkessä läpi. Päiväkirjan alussa meidän QA-tiimimme ja kaikki uudet prosessit olivat tosi alussa, mutta näiden menneiden viikkojenkin aikana on tullut aivan hirveästi edistystä ja tuntuu, että tämä testaaminen ja sen ympärillä pyörivät prosessit paranevat päivä päivältä. Itse melkein kaksi vuotta yrityksessä työskentelevänä vasta nyt sain ympärilleni oman testaukseen erikoistuneen tiimin. Ennen tätä tein yksin näitä hommia ja kaikki eteni vähän vapaalla kädellä, mutta nyt uuden tiimin vetäjän johdolla tämä alkaa muistuttamaan jo ammattimaista tiimityöskentelyä. Testidokumentointi alkaa näyttämään jo sellaiselta minkä kanssa voi tulevaisuudessa tehdä töitä.

4.4 Viikko 4

Tällä viikolla ei ole oikein mitään suunniteltua toimintaa, niin jokainen päivä mennään yksi kerrallaan eteenpäin. Tiistain tiimipalaverissa varmaan selviää, onko asioita mihin keskittyä, vai pitääkö itse kehittää omat tehtävät

Maanantai 15.4.2024

Henkilökohtaisten menojen takia viikko alkoi lyhyemmällä työpäivällä, mutta silti sai puolikkaan päivän tehtyä. Viime yönä ennen nukkumaanmenoa mietin, että miten Flutter-sovellus käyttäytyisi, jos

käyttäjät tietyissä kohdissa käyttöä menettäisi internet-yhteyden niin käytin tähän pitemmän hetken testatakseni ja kysellessäni Flutterin kehittäjältä, että onko näihin jotain protokollia kehiteltynä. Tämä olikin hyvä muistutus kehittäjän mukaan, että tästä pitää keskustella lisää tulevaisuudessa. Se on iso ongelma sovelluksen arkkitehdin puolesta ja korjaaminen ei ole helppo tehtävä. Flutterista irrottua olikin meidän yrityksemme All hands -palaveri, joka on kerran kuukaudessa ja tämän palaverin agendana on käydä yrityksen yleisiä asioita läpi ja kerrotaan aina päivitykset yrityksen myynneistä, hyvinvoinnista, tiimien lisäyksistä tai mahdollisista muutoksista ja muista arkeen liittyvistä asioista.

Tiistai 16.4.2024

Tänään tehdään hiukan rennommin töitä, kun terveystila tuntuu huonommalta ja keskitytään vain Flutterin hyvin black box-tyyliseen testaukseen. Käytiin QA-tiimin kanssa palaverissa katsomassa tulevan viikon tekemiset ja jakamassa omat ajatukset taas menneeltä viikolta.

Keskiviikko 17.4.2024

Kipeänä.

Torstai 18.4.2024

Kipeänä.

Perjantai 19.4.2024

Menipäs se viikko pilalle kipeyden takia. Meillä integroitiin meidän yrityksemme Slack-kanavat Visman Slackkiin niin käytin aamusta pienen hetken tutustuakseni kaikkiin uusiin kanaviin ja liityin kaikkiin kiinnostaviin, esimerkiksi peliaiheiselle kanavalle. Tein tehokkaan aamun parin uuden pienen tiketin kanssa, mitkä olivat tulleet testaustilaan. Kollega oli sairastelun aikana kommentoinut ja merkinnyt minua tarkastamaan muutaman vanhan tiketin backlogilta, että onko nämä vielä ajankohtaisia ja virheellisiä. Suurin osa oli tullut korjatuksi syystä tai toisesta, mutta backlogin siivoaminen ja tarkastaminen aika ajoon on tärkeää, että sovelluksen ongelmat pysyvät ajan tasalla. Tänään oli myös taas sprinttipalaverit, missä käytiin mennyt sprintti läpi ja esiteltiin uudet kehitykset. Pala-

verien jälkeen oli pari isompaa korjausta tehty kalenteriin, joten katsoin nämä läpi ja testasin perustoimivuuksia kalenterista loppupäivän, ettei mitään mennyt rikki uusien päivitysten myötä. Näihin touhuihin olikin hyvä lopettaa päivä ja aloittaa taas uusi viikonloppu.

Yhteenveto

Viikko oli hyvin vajaan kaiken kanssa, mutta sen ajan mitä töihin sai käytettyä, oli hyvin tehokasta. Flutterin testaaminen alkaa olemaan jo todella loppusuoralla, mutta testidokumentointi vielä pitää viimeistellä kevään aikana.

4.5 Viikko 5

Maanantai 22.4.2024

Maanantai alkoi käymällä tiketit läpi ja oli kiva nähdä ensimmäistä kertaa hetkeen, että ei ole testauksessa yhtään tikettiä. Nämä hetket ovat aika harvinaisia, mutta nyt sain koko päivän vapaaksi kirjoittaakseen Flutterin asiakassivuille testidokumentointia ja samalla testattua tämän ominaisuuden läpi.

Tiistai 23.4.2024

Aamusta oli QA-tiimin palaveri, jossa käytiin viikon statukset läpi ja juteltiin, että pitää keskittyä lääkelistan testicasejen dokumentointiin ja testaamiseen niin sain nämä päivän aikana hyvin aloitettua ja edistyttyä. Kollega myös pyysi apua backlogin tarkastamisen ja puhdistamisen kanssa, kun siinä on kovasti hommaa. Sitä ehtii katsomaan lääkelistan casejen dokumentoinnin jälkeen. Lääkelistan dokumentoinnista jatkuu huomenna ja näillä näkymin kerkeää iltapäivään mennessä saamaan valmiiksi tämän.

Keskiviikko 24.4.2024

Päivä jatkui juuri siitä mihin eilen jäi. Kovasti pääsi vääntämään testidokumenttien kanssa ja piti soittaa kollegan kanssa yhdestä ominaisuudesta lääkelistassa, kun en itse ollut ikinä kuullutkaan kyseisestä asiasta. Testidokumentit sain hyvälle mallille, että pääsee perjantaina käymään nämä

läpi ja mahdolliset ongelmat ja viat kirjaamaan ylös. Loppupäivä meni tuossa backlogin tarkastamisessa, että pääsi auttamaan työkaveria kuorman kanssa.

Torstai 25.4.2024

Kuukauden viimeinen torstai eli kuukausittainen PDD päivä oli tänään. Käytin aikani opinnäytetyön edistämiseen ja etenkin keskityin kirjoittamisvirheiden korjaamiseen.

Perjantai 26.4.2024

Päivä alkoi kokoamalla aiemmin kirjoitetut lääkelistan testidokumentit testruniksi ja ajamalla nämä läpi. Ongelmat kirjasin ylös ja laitoin eteenpäin, että PO (Product Owner) näkee runin ajetuksi ja voi nostaa täältä tarvittavat korjaukset tuleville sprinteille. Tämän jälkeen minun nimiini oli laitettu Androidin käyttäjän aktivointiticketti ja pääsin tutustumaan tarkemmin, kuinka sovelluksen aktivointi ja deaktivointi toimii. Tästä tuli kirjoitettua pitempi tekstikappale kehittäjälle mielipiteitäni ja huomioita ja yhtäkkiä huomasinkin, että viikko alkaa olemaan taas ohi.

Yhteenveto

Viikko oli loppujen lopuksi aika yksitoikkoinen, kun suurin osa ajasta meni lääkelistan ympärillä joko dokumentoidessa tai testatessa. En myös alku viikon jälkeen kertaakaan eksynyt testaustilan tiketteihin, niin ensi viikolla varmaan sieltäkin taas tehtävää löytyy. Tämänkaltaiset viikot ovat hyvää vaihtelua arkeen, että saa keskittyä yhden asian ympärille kunnolla. Ensi viikolla alkaakin päiväkirjan viimeinen viikko ja valmistuminen lähenee päivä päivältä, hyvin jännittävää!

4.6 Viikko 6

Maanantai 29.4.2024

Viimeinen viikko päiväkirjaa käytiin täydellä työvoimalla. Käytiin Androidin ja Flutterin kehittäjän kanssa keskustelua tuosta viime viikon käyttäjän aktivointiticketistä ja siitä, miten tätä voisi parantaa. Kehittelimme sitten päivän mittaan hiljalleen sovelluksen toiminnallisuudesta selvemman asiakkaan näkökulmasta. Keskustelujen ohessa Androidiin oli uusittu ja päivitetty GPS:n ja karttaominaisuuden toiminta niin päivä meni kuin siivillä, kun sai testata tämän toiminnallisuuden kokonaan.

Eli koko päivä menikin melkein puhelimen äärellä ja testiasiakkaiden osoitteita vaihtaessa, että jokainen luotu reitti olisi uusi, eikä aiemmin välimuistiin tallentunut reitti.

Tiistai 30.4.2024

Päivä alkoi kunnolla tiiminvetäjän soitolla, kun hän jakoi minulle tietoa tuollaisesta ominaisuusinfo Excel-tilustusta. Taulukkoon piti täyttää tietoja omista osaamisalueista ja taidoista liittyen jokaiseen sovelluksen ominaisuuteen. Hän myös antoi tehtäväksi katsoa ja täytellä mitkä kaikki ominaisuudet on tehty ja saatavilla Androidin ja Flutterin sovelluksiin. Näiden kanssa meni oma tovin, jonka jälkeen sainkin kuulla, että lomapäivät pitäisi merkitä lomakalenteriin niin sitten sai ne miettiä nopeasti läpi. Hyppäsin tehtävästä toiseen ja lomakalenterin selvittämisen jälkeen olikin aika katsoa taas palkanlaskentaan liittyviä tikettejä ennen tiimipalaveria. Nämä tiketit jäivät kesken ja pääsen niistä jatkamaan torstaina vapun jälkeen.

Keskiviikko 1.5.2024

Loma

Torstai 2.5.2024

Päivä alkoi tekemällä backlogin siivousta loppuun. Kovin montaa tikettiä ei ollut mitkä vaativat varmistusta niiden validiteetista, mutta ne olivat hiukan pitempiä ja joutui kehittäjiä kanssa keskustella parista asiasta. Nämä veivät oman aikansa ja niiden jälkeen siirryin tiistaina kesken jääneisiin palkanlaskennan tiketteihin. Ne olivat ihan hyvällä mallilla jo ja hetken työskentelyn jälkeen molemmat saatiin merkittyä tehdyiksi. Minulle merkittiin kalenteritiketti loppuviikoksi, joka on aika iso kokonaisuus. Huomina tulee siis koostumaan tästä tiketistä sprinttipalaverien ympärillä ennen kuin saan päiväkirjaosuuteni loppuun kokonaan.

Perjantai 3.5.2024

Aamu alkoi eilisen kalenteriticketin kanssa. Kokonaisuus oli hyvin monimutkaisesti kasattu niin oman aikani käytettyä asian kanssa otin yhteyttä testauskollegaani, jonka kanssa käytiin tiketti läpi. Heti parityöskentelynä saimme asian nopeasti vauhtiin ja löydettyä kasan ongelmia, mitkä kehittäjille kirjoitettiin ylös. Tämän jälkeen olikin nopeampi lounas ja pääsi istumaan sprinttipalaverin ääreen

näkemään ja kuulemaan viime sprintin kehitetyt ja korjatut asiat. Kuuli myös uuden kollegan esittelyn, mikä on aina yhtä mukava asia, että saa uusia työkavereita! Palaverin jälkeen hyppäsin pidemmän tauon jälkeen kirjoittamaan Flutterin testidokumentaatiota. Sen parissa menikin loppupäivä ja pystyi aloittamaan ensimmäisen viikonlopun päiväkirjaosuuden jälkeen.

Yhteenveto

Viimeinen viikko ja sen alku oli aika erilainen muihin viikkoihin nähden siinä mielessä, että sain alkuvuikosta itse miettiä ja kehitellä Flutterin kehittäjän kanssa tuon aktivointi osuuden kokonaan uusiksi. Tämä käyttökokemuksen kehittäminen olisi tulevaisuudessa hyvin mielenkiintoista mihin voisi erikoistua enemmänkin. Alkuvuikkoa lukuun ottamatta olikin hyvin tavallinen viikko ilman mitään erikoisuuksia. Tämä oli myös viimeinen viikko päiväkirjaa ja alkaa tuntumaan jo oikealta, että voi valmistua koulusta. Hiukan epärealistinen tunne, mutta eiköhän sekin realiteetiksi muutu tässä tulevien viikkojen aikana.

5 POHDINTA

Kuuden viikon ajanjakso meni ohi hyvin nopeasti, kun seurasi päivittäin omaa tekemistä tekstin muodossa. Isoimpana projektina päiväkirjaan otimme sovellustestaustiimimme kanssa aloittaa ja viedä pitkälle testidokumentointia eri mahdollisten sovellusten ja niiden ominaisuuksien ympärillä. Itse mobiilisovellusten eli Flutter- ja Android-sovellusten asiantuntijana näiden testidokumenttien kirjoittaminen jäi omalle vastuulleni hyvin isosti. Testidokumentointia ei ole yrityksen olemassaolon aikana ollut ollenkaan, kun ei ole testaukseen erikoistuvia työntekijöitäkään ollut ennen yritykseen sisääntuloani. Nyt kuuden viikon aikana saimme niitä kehitettyä todella pitkälle, ja vaikka työ ei vielä ole ohi niiden kanssa, niin meillä on nyt todella hyvä pohja mistä jatkaa niiden kehittämistä päivä päivältä.

Tiimityöskentely nousi viikkojen aikana myös isoon rooliin, mihin en ollut tottunut, kun olen aina ollut se yksinäinen testaja ja apua on saanut kehittäjäkollegoilta vain silloin, kun heillä on liennyt aikaa muun työn sivusta. Nyt ympärillä on kaksi muuta yhtä sosiaalista ja energistä työntekijää samojen ongelmien kimpussa, joten aina, jos apua tarvitsee, sitä myös saa. Tätä kaikki työntekijät jokaisessa yrityksessä tarvitsevat ympärilleen. Opin myös paremmin ja pienemmällä kynnyksellä laittamaan viestiä muille kollegoille mahdollisesta avusta, jos en ymmärtänyt sovelluksesta jotain. Esimerkiksi meidän Flutterin kehittäjä sai paljon kyselyitä ja oli hieno huomata, että sieltä sai aina tarpeellisen tiedon, jos testaustiimistä muut eivät osannetkaan auttaa.

Yrityksessä alkoi ensimmäistä kertaa myös kehityskeskustelut, mikä jäi näiltä menneiltä viikoilta todella vahvasti mieleen. Esihenkilömme kävi hyvin tarkasti läpi oloilamme yrityksessä ja sen ulkopuolella, kyseli mielteitä menneisiin ja tuleviin asioihin ja muutenkin oli hyvin kannustava kaiken suhteen. Mainitsin päiväkirjassakin, että keskustelu oli elämäni ensimmäinen missään työympäristössä ja mielestäni näitä pitäisi järjestää kaikissa työpaikoissa työntekijän hyvinvoinnin ja mielipiteitten kuuntelun takia.

Sain myös tehdä päiväkirjan loppupuolella pienen määrä Androidin käyttökokemusta parantavaa työskentelyä yhdessä kehittäjän kanssa ja tästä huomasin, että siihen voisi tulevaisuudessa jopa hiukan enemmänkin erikoistua. Kaiken kaikkiaan päiväkirjan ansiosta opin arvostamaan hyvää työyhteisöä ja kollegoita. Heidän ansiostaan työnteko on helpompaa ja sitä myös oikeasti odottaa, kun tekee jotain, josta tykkää samanmielisten ihmisten kanssa.

LÄHTEET

Alfame 2018. API:t käytännössä - selkokielen katsaus. Hakupäivä 22.4.2024. <https://www.alfame.com/ajankohtaista/apit-kaytannossa-selkokielen-katsaus>

Amazon Web Services 2024. What is Unit Testing? Hakupäivä 15.2.2024. <https://aws.amazon.com/what-is/unit-testing/>

Appium 2024. Appium. Hakupäivä 21.5.2024. <https://appium.io/docs/en/latest/>

Geeksforgeeks 2024. Acceptance Testing – Software Testing. Hakupäivä 27.2.2024. <https://www.geeksforgeeks.org/acceptance-testing-software-testing/>

Geeksforgeeks 2024. Software Testing tools. Hakupäivä 21.5.2024. <https://www.geeksforgeeks.org/software-testing-tools/>

Katalon 2024. Katalon docs. Hakupäivä 21.5.2024. <https://docs.katalon.com/>

Katalon 2024. What is Integration Testing? Definition, How-to, Examples. Hakupäivä 22.2.2024. <https://katalon.com/resources-center/blog/integration-testing>

Komprise 2024. Stubs. Hakupäivä 15.2.2024. https://www.komprise.com/glossary_terms/stubs/

Postman 2024. What is Postman? Hakupäivä 21.5.2024. <https://www.postman.com/product/what-is-postman/>

Selenium 2024. The Selenium Browser Automation Project. Hakupäivä 21.5.2024. <https://www.selenium.dev/documentation/>

Stackify 2023. Best Software Testing Methods. Hakupäivä 5.3.2024. <https://stackify.com/best-software-testing-methods/>

VALA Group 2023. Integraatiotestaus: Mitä se tarkoittaa ja mitä hyötyä siitä on? Hakupäivä 22.2.2024. <https://www.valagroup.com/fi/blogi/integraatiotestaus/>

Yazar, K. 2024. What is system testing? TechTarget. Hakupäivä 23.2.2024. <https://www.tech-target.com/searchsoftwarequality/definition/system-testing>