

Saara Männikkö

PILVIPOHJAISTEN TUKIASEMIEN SUORITUSKYKYTESTAUS

PILVIPOHJAISTEN TUKIASEMIEN SUORITUSKYKYTESTAUS

Saara Männikkö
Opinnäytetyö
Syksy 2023
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, laite- ja tuotesuunnittelu

Tekijä(t): Saara Männikkö

Opinnäytetyön nimi: Pilvipohjaisten tukiasemien suorituskykytestaus

Työn ohjaaja(t): Teemu Leppänen

Työn valmistuslukukausi ja -vuosi: syksy 2023

Sivumäärä: 25

Tässä päiväkirjaopinnäytetyössä seurattiin viiden viikon ajan työtäni testausinsinöörinä Nokialla pilvipohjaisten tukiasemien suorituskykyä testaavassa PET-tiimissä. Tavoitteena oli antaa lukijalle kattava kuva testausinsinöörin monipuolisista työtehtävistä sekä arvioida käytössä olevia toimintamalleja niiden kehittämiseksi.

Opinnäytetyön alussa esiteltiin työympäristö, urakehitykseni ja tulevaisuuden tavoitteeni sekä sidosryhmät. Lisäksi tarkasteltiin teoreettisesti ohjelmiston suorituskykytestausta ja esiteltiin testaukseen käytettävää testausympäristöä. Päiväkirjaosio keskittyi viiteen viikkoon, joiden aikana käsiteltiin suorituskykytestaukseen kuuluvia tehtäviä. Näitä olivat regressiotestaus, vikojen raportointi ja korjaus, testausympäristön muutostyöt sekä stabiilisuustestaus. Lisäksi jokaisen viikon pohjalta arvioitiin käytössä olevia toimintamalleja sekä mahdollisia vaihtoehtoisia työskentelytapoja.

Tämä päiväkirjaopinnäytetyö osoitti, että testausinsinöörin työtehtävät vaativat muun muassa suunnitelmallisuutta, ammattitaitoa, organisointikykyä sekä yhteistyötä muiden asiantuntijoiden kanssa oman ammatillisen kehityksen tukemiseksi. Systemaattisia työskentely- ja toimintatapoja kehittämällä työtä voidaan tehdä tehokkaasti, luotettavasti sekä turvallisesti ja välttämään aikaa vievältä metatyöltä.

Asiasanat: 5G, päiväkirjaopinnäytetyö, suorituskykytestaus, testausinsinööri, tukiasemat

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information and Communication Technology, Option of Device and Product Design

Author(s): Saara Männikkö

Title of thesis: Performance testing of cloud base station

Supervisor(s): Teemu Leppänen

Term and year when the thesis was submitted: Autumn 2023

Number of pages: 25

This diary thesis followed my work as a test engineer in Nokia in a Performance Entity Testing - team for five weeks. The goal was to provide an overview of test engineer's daily tasks and to assess current procedures for their further development.

The thesis introduced Nokia as a workplace, my career development and future goals, and stakeholders. Performance testing was introduced considering theory, and test environment used in my daily work was showcased. The diary section of the thesis included five weeks, that focused on performance testing tasks. The tasks were regression testing, reporting and fixing issues, test environment changes, and stability testing. Based on every week, current procedures were evaluated, and possible alternative methods were discussed.

This diary thesis showcased that test engineer's work requires careful planning, expertise, organizing skills, and collaboration with other experts to develop and support the skills needed. By developing systematic ways of working, testing can be done efficiently without having to concentrate on unnecessary meta work.

Keywords: 5G, base stations, diary thesis, performance testing, test engineer

SISÄLLYS

1	JOHDANTO	7
2	TESTAUSINSINÖÖRIN TYÖ NOKIALLA	8
2.1	Nokia Oulussa	8
2.2	Testausinsinöörinä PET-timissä	8
2.3	Sidosryhmät	10
3	TYÖN TAVOITTEET	11
3.1	Ohjelmiston suorituskykytestaus	11
3.2	Testaukseen käytettävä testausympäristö.....	12
4	PÄIVÄKIRJA.....	14
4.1	Viikko 1.....	14
4.2	Viikko 2.....	15
4.3	Viikko 3.....	17
4.4	Viikko 4.....	19
4.5	Viikko 5.....	21
5	POHDINTA.....	23
	LÄHTEET.....	25

SANASTO

4G	Neljännän sukupolven mobiiliverkko
5G	Viidennen sukupolven mobiiliverkko
APO	Area Product Owner, vastaa projektin raportoinnista projektin johdolle
F1-Interface	F1-linkki, yhdistää vCU:n ja vDU:n
LPO	Local Product Owner, vastaa PET-tiimin sisäisestä testausaikataulusta
PET	Performance Entity Testing, tukiaseman suorituskykyyn erikoistunut testaus
RF	Radio Frequency, radiotaajuus
STLC	Software Testing Life Cycle, ohjelmistotestauksen elinkaari
Testausympäristö	Testausinsinöörin käyttämä testauslaitteisto
Trafiikkimalli	Erialaista liikennettä sisältävä UE-simulaattorin testitapaus
UE	User Equipment, laite, joka on yhteydessä verkkoon
vCU	Virtualized Centralized Unit, keskitetty yksikkö, joka on osa 5G-tukiasemaa
vDU	Virtualized Distributed Unit, CU:n ohjaama 5G-tukiaseman hajautettu yksikkö

1 JOHDANTO

Tässä päiväkirjaopinnäytetyössä seurataan viiden viikon ajan työtäni Nokian Oulun toimipisteellä testausinsinöörinä Performance Entity Testing (PET) -tiimissä. Testausinsinöörin työtehtäviin kuuluu tukiaseman ohjelmistojen suorituskykytestaus, jonka tavoitteena on varmistaa niiden suorituskyvyn ja asetettujen vaatimusten kohtaaminen. Näin pyritään varmistamaan, että asiakkaille ei pääse ohjelmistoja, joista löytyy suorituskykyä heikentäviä vikoja.

Oman PET-tiimini päävastuualueena on pilvipohjaisten tukiasemien suorituskykytestaus. Tämä pitää sisällään tukiaseman ohjelmistojen uusien ominaisuuksien testaamisen, regressiotestauksen sekä stabiilisuuden testaamisen. Testausinsinöörin työ PET-tiimissämme on hyvin monipuolista ja vaihtelevaa, jonka vuoksi opinnäytetyö kattaa päiväkirjaraportoinnin useasta eri suorituskykytestaukseen liittyvästä tehtävästä. Tehtäviä ovat stabiilisuustestaus, vikojen raportoiminen ja vikojen korjausprosessi, regressiotestaus sekä testausympäristöjen muutostyöt.

Tämän päiväkirjaopinnäytetyön tavoitteena on antaa lukijalle kattava kuva testausinsinöörin työstä pilvipohjaisiin tukiasemiin erikoistuneessa PET-tiimissä sekä arvioida käytössä olevia toimintamalleja niiden kehittämiseksi. Opinnäytetyön alussa esitellään testausinsinöörin työympäristöä Nokialla, perehdytään tarkemmin työskentelyyn PET-tiimissäni ja esitellään yrityksen sidosryhmät. Lisäksi tarkastellaan teoreettisesti ohjelmiston suorituskykytestausta ja esitellään työssä käytettävää testausympäristöä. Tämän jälkeen siirrytään viikoittain raportoitavaan päiväkirjaan. Päiväkirjassa esiteltävien tehtävien lisäksi tavoitteena on kunkin viikon pohjalta arvioida käytössä olevia toimintamalleja sekä pohtia, kuinka kyseisiä tilanteita voidaan kehittää. Päiväkirjaa seuraa pohdintakappale, jossa käydään läpi aikaansaannokset, haasteet ja mahdolliset kehitysehdotukset.

2 TESTAUSINSINÖÖRIN TYÖ NOKIALLA

2.1 Nokia Oulussa

Nokia Oyj on suomalainen maailmanlaajuisesti toimiva teknologiayhtiö, joka työllistää noin 87000 henkilöä noin 130 maassa, joista noin 7000 henkilöä työskentelee Suomessa. Suomessa toimipisteet sijaitsevat Espoossa, Tampereella ja Oulussa. Yrityksen päätoimiala on langattomat matkapuhelinverkkopalvelut, kuten niiden laitteistot, ohjelmistot sekä verkkojen suunnittelu ja toteutus. (Nokia Oyj 2023a.)

Nokian Oulun toimipiste sijaitsee tällä hetkellä Ruskossa ja se työllistää noin 2850 henkilöä. Oulun toimipisteessä ovat radio- ja tukiasematuotteiden sekä niiden ohjelmistojen tutkimus- ja tuotekehitysyksiköt. Tuotekehitys työllistää Oulussa noin 1800 henkilöä. Lisäksi Oulussa sijaitsee radio- ja tukiasematuotteiden tuotantoyksikkö. Oulun toimipisteestä käytetäänkin nimitystä "Home of Radio". (Nokia Oyj 2023a.) Oulussa sijaitsee myös testauslaboratorio Otava eli Over the Air Validation Area, jossa 4G- ja 5G-tukiasemille tehdään erilaisia ilmarajapinnan testaustauksia. Nokian Oulun toimipiste tulee muuttamaan vuonna 2025 uudelle rakenteilla olevalle Polaris 5G -älykampukselle Oulun Linnanmaalle, mutta testauslaboratorio Otavan toiminta tulee pysymään Ruskossa.

2.2 Testausinsinöörinä PET-tiimissä

PET-organisaatio kuuluu Mobile Networks -liiketoimintayksikön alle ja koko PET-organisaation toiminta on sijoitettu Nokian Oulun yksikköön. PET-organisaatio koostuu 13 tiimistä, joilla on kaikilla omat vastualueensa, kuten asiakasspesifinen suorituskykytestaus, uusien ominaisuuksien testaus, automaatio sekä pilvipohjaiset tukiasemat.

PET-tiimissä työskentelee testausinsinöörejä, joilla on kaikilla oma vastualueensa testauksessa sekä testausympäristö, jolla testausta suoritetaan. Tiimissä työskentelee myös tiimin sisäisiä teknisiä asiantuntijoita, jotka muun muassa auttavat asiantuntemuksellaan testausinsinöörejä teknisissä ongelmissa. Local Product Owner (LPO) vastaa projektin testausaikataulusta sekä testauksen kulusta. Tiimin kanssa läheisesti työskentelee myös Area Product Owner (APO), joka vastaa muun muassa projektin aikataulusta ja laadusta sekä raportoinnista projektin johdolle. Tiimiin kuuluu myös oma esihenkilö.

Oma polkuni testausinsinööriksi sai alkunsa keväällä 2021 eli olen nyt työskennellyt tässä tehtävässä noin kaksi ja puoli vuotta. Aloitin testauksen oman perehdyttäjäni työparina hänen testausympäristöllään, jolla suoritimme muun muassa tukiaseman maksimaalisen käyttäjämäärän testausta. Pääsin pian tekemään testausta itsenäisesti ja myöhemmin olen rakentanut itse kaksi pilvipohjaista testausympäristöä, joista toisella suoritan nykyisin stabiilisuustestausta ja toinen on kollegani käytössä. Olen myös itse toiminut perehdyttäjän roolissa uusille työntekijöille sekä harjoittelijoille ja pyrin auttamaan heitä aina tarvittaessa.

Tämänhetkinen työnkuvani on tukiaseman ohjelmistojen stabiilisuustestaus. Lyhyesti tukiaseman stabiilisuustestauksella tarkoitetaan tukiaseman pitkäkestoista kuormittamista erilaisilla trafiikkimalleilla. Tavanomainen työpäiväni alkaa aamupalaverilla, jossa käydään läpi tiimimme testausympäristöjen tilanne eli mitä on tehty ja mitä tullaan tekemään. Palaverin jälkeen teen parannuksia UE-simulaattorin (engl. User Equipment) testitapauksiin ja tarkkailen kuinka muutokset vaikuttavat suorituskykyyn. Parannuksia voivat olla esimerkiksi erilaisen liikenteen lisääminen trafiikkimalliin, tai vaikka käyttäjämäärän nostaminen. Päivällä saattaa olla myös muita palavereita ja koulutuksia.

Nokialla työntekijällä on mahdollisuus vaikuttaa omaan urapolkuunsa ja tulevaisuudessa minua kiinnostaisi olla enemmän asiantuntija kuin esimiestehtävissä. Olen työskennellyt monen eri asiantuntijan kanssa vikatutkimuksissa ja testausympäristön konfigurointiongelmiin parissa. Haluaisin tulevaisuudessa itsekkin osata auttaa muita osaamisellani sekä tiedollani.

2.3 Sidosryhmät

Testausinsinöörin työssä tehdään paljon yhteistyötä erilaisten sidosryhmien kanssa. Viankorjausprosessissa testausinsinööri työskentelee tiiviisti korjausvastuussa olevien ohjelmistokehitystiimien kanssa. Kommunikointi tapahtuu Microsoft Teamsin välityksellä ja käytettävä kieli on englanti.

UE-simulaattoreiden tukihenkilöt auttavat testausinsinöörejä UE-simulaattoreihin liittyvissä ongelmissa. Tilanteita, joissa UE-simulaattoreiden tukihenkilöitä tarvitaan, voivat olla esimerkiksi simulaattorin ohjelmiston kaatuminen, uuden simulaattorin konfiguroiminen käyttöön tai uuden ominaisuuden konfiguroinnissa testaustapaukseen tapahtuvat ongelmat. UE-simulaattoreiden tukihenkilöitä työskentelee niin etänä kuin Oulun yksikössä ja heihin otetaan yhteyttä tekemällä tiketti eli työpyyntö.

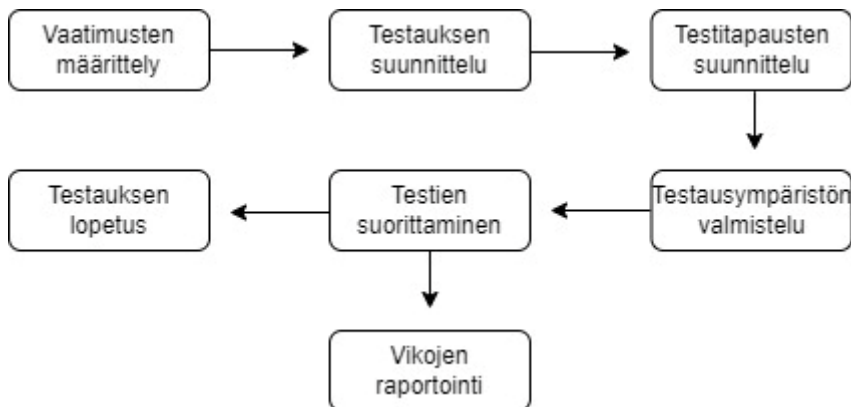
Paikalliset eri laboratoriotiimit vastaavat esimerkiksi testausympäristöjen sähkötoista, varaston ylläpidosta ja sekä testausverkoista. Paikalliseen laboratoriotiimiin ollaan yhteydessä esimerkiksi uusien testausympäristöjen tilan varauksessa, etätietokoneiden asennuksissa tai testausverkkoon liittyvissä ongelmissa. Lisäksi Intiassa sijaitsee laboratorioverkkotiimi, joka vastaa verkkoon tehtävistä huolloista ja päivityksistä.

3 TYÖN TAVOITTEET

Opinnäytetyön tavoitteena on antaa lukijalle kattava kuva testausinsinöörin työtehtävistä sekä arvioida käytössä olevia toimintamalleja niiden kehittämiseksi. Henkilökohtainen tavoitteeni on kehittyä vikojen etsimisessä sekä lokitiedostojen keräämisessä. Opinnäytetyö on toteutettu päiväkirjamuotoisena raporttina, jossa seurataan työtäni viiden viikon ajanjaksolla. Ajanjaksolle ajoittuvia suorituskkytestauksen tehtäviä ovat stabiilisuustestaus, vikojen raportoiminen ja korjausprosessi, regressiotestaus sekä testausympäristöjen muutostyöt. Jokaisen viikon jälkeen on lyhyt reflektointikappale, jossa pohditaan esiintyneitä ongelmia ja mahdollisia vaihtoehtoisia toimintatapoja.

3.1 Ohjelmiston suorituskkytestaus

Ohjelmistotestauksen yleisiä vaiheita voidaan kuvata ohjelmistotestauksen elinkaaren eli Software Testing Life Cyclen (STLC) avulla (kuva 1). STLC sisältää ohjelmistotestauksen vaiheet, joita voidaan soveltaa testattavan tuotteen ja tarpeen mukaan. Vaiheet ovat vaatimusten määrittely, testauksen suunnittelu, testitapausten suunnittelu, testausympäristön valmistelu, testauksen suorittaminen ja vikojen raportointi sekä testauksen lopetus. (Hooda & Chhillar 2015, 10.)



KUVA 1. Ohjelmistotestauksen vaiheet (Hooda & Chhillar 2015, 10).

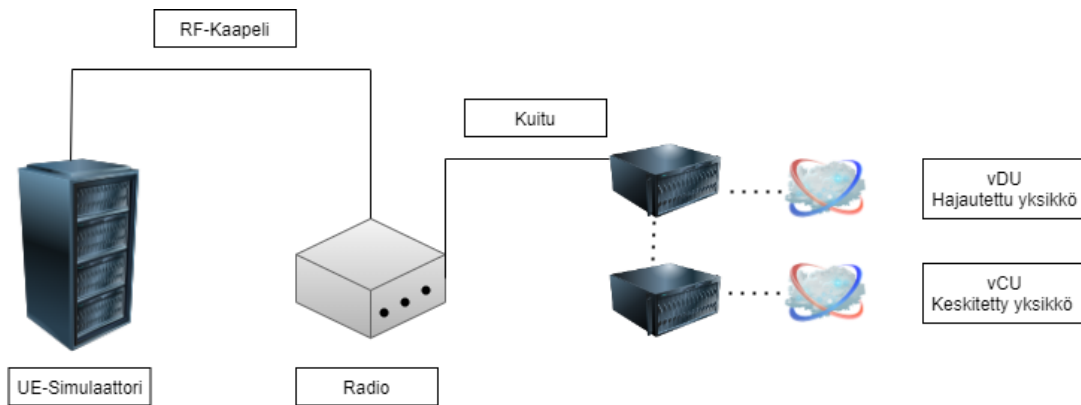
Suorituskykytestaus mukailee samoja vaiheita kuin STLC (Khan 2010, 13). Suorituskykytestauksella pyritään varmistamaan, että ohjelmiston suorituskyky täyttää sille asetetut vaatimukset (Naik & Tripathy 2011, 209). Käytännössä siis tukiaseman ohjelmiston suorituskykytestauksella varmistetaan, että tukiaseman suorituskyky kestää esimerkiksi suurten käyttäjämäärien ja dataliikenteen aiheuttaman kuormituksen. On myös tärkeää huomioida, että yritykset muotoilevat testausprosessin omien tuotteidensa ja tarpeidensa mukaan (Hooda & Chhillar 2015, 10). Omassa PET-tiimissäni suorituskykytestauksessa keskitytään pilvipohjaisten tukiasemien uusien ominaisuuksien testaukseen, regressiotestaukseen, sekä stabiilisuustestaukseen. Regressiotestit ovat ajoittain toistettavia testejä, joilla pyritään varmistamaan, etteivät ohjelmistoon tuodut uudet ominaisuudet heikennä ohjelmiston suorituskykyä. Stabiilisuustestauksen tarkoituksena on varmistaa, että tukiaseman ohjelmisto kestää monipuolisen verkkoliikenteen tuoman pitkäaikaisen rasituksen. (Naik & Tripathy 2011, 17, 213.)

STLC toteutuu esimerkiksi stabiilisuustestauksessa seuraavan kuvauksen mukaisesti. Ensin projektin johto määrittelee ohjelmistoversiot, joilla stabiilisuustestausta halutaan tehdä. Testauspyyntö lähetetään stabiilisuustestausta suorittaville testausinsinööreille, jotka valmistelevat testaukseen määrätyt testausympäristöt eli päivittävät halutut ohjelmistoversiot. Lisäksi kehitetään trafiikkimallitapaukset eli ajettavat testit, jotka sisältävät monipuolista ja tukiasemaa kuormittavaa dataliikennettä, kuten esimerkiksi erilaisia puheluita ja käyttäjien liikkumista solujen välillä. Testin aikana testausinsinööri tarkastaa määrätyin väliajoin suorituskyvyn, muistienkulutuksen sekä resurssit. Jos tarkastettavissa kohteissa havaitaan poikkeamia, selvitetään poikkeaman syy ja siitä kirjataan vikaraportti, joka käynnistää vian korjausprosessin. Vikaraportti menee ohjelmiston osasta vastaavalle tutkimustiimille, joka selvittää yhteistyössä testausinsinöörin kanssa, kuinka vika korjataan. Jos poikkeamia ei havaita, jatketaan testiä kohti seuraavaa tarkistuspistettä. Lopuksi kirjataan testin tulos testausjärjestelmään.

3.2 Testaukseen käytettävä testausympäristö

Mobiiliverkko koostuu tukiasemien muodostamista soluista eli maantieteellisistä alueista, joille tukiasemat syöttävät yhtäjaksoisesti radiosignaalia (Miller 2013, 56). Testausinsinöörin käyttämä testausympäristö simuloi käytännössä pientä osaa mobiiliverkkoa. Testausympäristöön kuuluu UE-simulaattori, joka on kytketty radioihin RF-kaapeleilla eli radiokaapeleilla. Radiokaapelit simuloivat

soluja. Testausinsinööri rakentaa UE-simulaattorin ohjelmistolla testaustapauksen, jolla muodostetaan soluille erilaista dataliikennettä. Radioiden tehtävä on lähettää ja vastaanottaa radiosignaalia sekä digitalisoida ja vahvistaa sitä (Nokia Oyj 2023b). UE-simulaattorin lisäksi radiot yhdistyvät optisilla kuiduilla vDU:n pilviserverille eli hajautettuun yksikköön. Hajautetun yksikön tehtävä on ohjata alemman tason ajoitusherkkiä tehtäviä. Hajautettu yksikkö yhdistyy F1-linkityksellä keskitettyyn yksikköön eli vCU:hun, joka vastaa korkeamman tason viestinnästä sekä on yhteydessä runkoverkkoon. (Toskala & Poikselkä 2020, luku 5.) Pilvipohjaisessa arkkitehtuurissa keskitetty yksikkö ja hajautettu yksikkö ovat toteutettu pilviservereille ohjelmistoratkaisuilla (Nokia Oyj 2023b). Kuvassa 2 on havainnollistettu pilvipohjaisen testausympäristön kokoonpano.



KUVA 2. Pilvipohjainen 5G testausympäristö.

4 PÄIVÄKIRJA

4.1 Viikko 1

Raportoinnin aloitusviikolla suoritin kollegani sijaistusta. Kävimme läpi hänen testausympäristönsä perusasiat eli testausympäristön tämänhetkisen tilanteen, tarvittavat yhteydet, UE-simulaattorin testaustapaukset, jotka hän oli kehittänyt kyseisiä testejä varten sekä vastuullani olevat testit. Aloitin tutustumalla tarkemmin testien vaatimuksiin sekä UE-simulaattorin testitapausten rakenteeseen ja toimintaan. Vastuullani oli suorittaa testejä, joissa käyttäjille oli määritetty tietty prioriteetti. Prioriteetti takasi muun muassa tietyn viiveen ja pakettien menetysprosentin.

Aloitin testauksen solun maksimikäyttäjämäärätestillä, jossa tukiasemaa kuormitetaan solun maksimaalisella käyttäjämäärällä. Testejä tehdessäni havaitsin, ettei kaikilla käyttäjillä ollut liikennettä, joten testin tulos ei vielä täyttänyt sille määritettyjä vaatimuksia. Jos testin tulos ei täytä sille asetettuja vaatimuksia, täytyy tehdä muutoksia joko UE-simulaattorin testitapauksiin tai tukiaseman parametreihin. Jos ongelma ei ratkea muutoksilla, tehdään vikailmoitus. Vastaavat testit olivat vaatineet aikaisemmin muutoksia, joten päätin jättää käyttäjämäärätestin myöhemmäksi ja jatkaa eteenpäin.

Vastuullani oli myös testi, jossa testataan määrätyllä käyttäjämäärällä saavutettavaa maksimaalista tiedonsiirtonopeutta. Kuten käyttäjämäärätesteissä, huomasin että muutamalla käyttäjällä ei ollut liikennettä. Päätin tehdä muutoksia UE-simulaattorin testitapaukseen ja kokeilla ajaa testiä uudelleen. Muutosten jälkeen liikenne alkoi toimia kaikilla käyttäjillä, testin tulos täytti sille asetetut vaatimukset ja sain merkitä testin hyväksytyksi testausjärjestelmäämme.

Ensimmäisen viikon aikana omalla testausympäristölläni aloitettiin konfiguraatiomuutokset, jossa testausympäristöni kokoonpanoa pienennettiin. Muutostöiden valmistuttua kävin laboratoriossa tekemässä vaadittavat muutokset radioiden ja serverin välisiin kuitukytkentöihin ja tarkistamassa,

että niiden välinen yhteys alkoi toimia. Kun yhteys muodostui, aloitin testausympäristön konfiguroinnin takaisin käyttövalmiuteen kollegani avustuksella. Hän oli tehnyt vastaavat muutokset vastikään omaan testausympäristöönsä, joten saimme testausympäristöni pian valmiiksi.

Seuraavana aamuna huomasin, että muutostöiden jälkeen yksi testausympäristöni soluista oli vikatilassa. Aloin tutkimaan, mikä aiheutti solun jäämisen vikatilaan. Päätin priorisoida oman testausympäristöni, koska olin saanut pyynnön ajaa 48 tunnin stabiilisuustesti viikonloppuna. Lokitiedoista ei löytynyt mitään selkeää syytä ongelmalle, joten kokeilin käynnistää vDU:n uudelleen ja ongelma siirtyi toiselle solulle. Otin uudet lokit toiselta komponentilta ja tein hakuja viallisen solun tunnuksella. Löysin ongelmaani viittaavia ilmoituksia, jotka lähetin kollegalleni tarkistettavaksi. Hän löysi kaksi parametriä, jotka minun piti muuttaa. Muutoksen jälkeen kaikki solut nousivat viimein käyttövalmiuteen ja sain käynnistettyä stabiilisuustestin.

Yhteenvetona ensimmäinen viikko kului pitkälti regressiotestauksen parissa, kun oma testausympäristöni oli muutostöiden kohteena. Huomasin esimerkiksi maksimikäyttämäärättestissä, että olisi hyödyllistä, jos jokaisella testausinsinöörillä olisi testauspäiväkirja. Päiväkirjaan kirjattaisi mitä muutoksia ja hienosäätöjä on tarvittu tietyn testin suorittamiseen. Päiväkirjan arvo korostuu etenkin regressiotestauksessa, kun testit täytyy pystyä toistamaan. Ensimmäisen viikon testeissä tarvittiin muutoksia UE-simulaattorin testitapauksiin, jotka tiesi vain sillä hetkellä lomalla oleva testausinsinööri. Oman testausympäristöni soluongelman ratkaisutapa oli mielestäni hyvä. Pidän ongelmanratkaisusta ja se on myös yksi vahvuuksistani testausinsinöörinä.

4.2 Viikko 2

Viikko alkoi viikonloppun yli ajetun stabiilisuustestin tulosten tarkistuksella ja kirjaamisella. Ajoin siis 48 tunnin stabiilisuustestin, jossa tukiasemaa kuormitettiin yhtäjaksoisesti monipuolisella liikenteellä. Koska testillä ei ollut päivystysvelvoitetta viikonloppuna, otin testin päätyttyä ajosta lokit. Jos testiä olisi päivystetty, lokit olisi tarkastettu määritetyin väliajoin. Kävin läpi lokit, muistinkulutuksen, resurssit sekä suorituskykyä mittaavat kuvaajat. Lopuksi kirjasin testin tuloksen järjestelmäämme.

Jatkoin myös tällä viikolla kollegani sijaistamista. UE-simulaattorin hallintatietokone oli jumissa ajetaan automaattisia päivityksiä ja se piti käynnistää uudelleen. Uudelleenkäynnistys aiheutti UE-

simulaattorin ohjelmiston kaatumisen. Kaatumisen seurauksena ohjelmistosta puuttui kaikki testitapaukset sekä yhteydet. Olin törmännyt vastaavanlaiseen tilanteeseen vastikään ja osasin itse palauttaa testitapaukset sekä yhteydet. Jouduin kuitenkin tekemään UE-simulaattorin tukihenkilölle tiketin, koska liikenne ei toiminut testejä käynnistettäessä. Tukihenkilö otti minuun yhteyttä ja auttoi korjaamaan ongelman. Jatkoin testausta viime viikolla kesken jääneillä käyttäjämäärätesteillä. Tein testitapaukseen pieniä muutoksia ja laitoin testin ajoin työpäivän päätteeksi. Tarkistin aamulla testin suorituskykyä mittaavat kuvaajat ja saavutetun käyttäjämäärän. Testin tulos täytti sille asetetut vaatimukset, joten kirjasin sen hyväksytyksi testausjärjestelmäämme. Kollegani palasi myöhemmin tällä viikolla lomalta ja hän jatkoi testausta omalla testausympäristöllään.

Aloitin valmistautumisen pidempään stabiilisuustestiin päivittämällä tukiaseman luontiin käytettävän työkalun, koska uusimmat tukiaseman ohjelmistot vaativat siitä viimeisimmän version. Työkalun päivityksen yhteydessä päivitin myös vCU:n ja vDU:n ohjelmistot uusimpiin julkaistuihin versioihin. Päivitysten jälkeen tein kokeilevaa testausta eri käyttäjämäärillä ja puheluiden pituuksilla ja seurasin, kuinka niiden muuttaminen vaikuttivat suorituskykyyn.

Loppuviikosta minun piti perehtyä uuteen testausympäristöön, koska sijaistan toista kollegaani osan viikosta. Kyseinen testausympäristö oli kokoonpanoltaan hyvin erilainen kuin omani, joten kävimme konfiguraation ja vastuullani olevat testit tarkasti läpi. Lisäksi täytyi tarkistaa yhteyksien toimivuus, jotta voin hallinnoida kyseistä testausympäristöä ja sen osia. Perjantaina tein viimeiset valmistelut pidempää stabiilisuustestiä varten eli päivitin vCU:n ja vDU:n ohjelmistot testipyynnössä ilmoitettuihin versioihin. Lisäksi tein stabiilisuustestin aloituskirjaukset järjestelmään ja laitoin UE-simulaattorille tarpeeksi testitapauksia jonoon, jotta testit pyörisivät ilman pysähtymistä viikonlopun yli.

Yhteenvedona toinen viikko oli stabiilisuustestien kannalta hyvin tyypillinen eli tulosten tarkistusta ja uuteen testiin valmistautumista. Viikolla tapahtui myös tietokoneiden uudelleenkäynnistymisistä johtuvia ongelmia, kuten jumittumisia ja testitapausten katoamisia. Tällaisiin tilanteisiin olisi hyvä varautua varmuuskopioimalla testitapaukset tietyin väliajoin tai vähintään opettelemalla niiden palauttaminen.

4.3 Viikko 3

Kolmas viikko alkoi stabiilisuustestin tulosten tarkistuksella. Tarkistin suorituskyvyn kuvaajat, lokit, muistin kulutuksen sekä resurssit. Tarkistaessani testin suorituskyvyn kuvaajia ja lokeja havaitsin erään komponentin kaatumisen, joka oli aiheuttanut solujen käymisen vikatilassa ja katkoksen liikenteessä. Kun testaaaja havaitsee testissä poikkeavuuden, kuten käyttäjämäärän tippumisen tai jonkin komponentin kaatumisen, tehdään vikaraportti. Tätä varten täytyy kerätä vian tutkimiseen tarvittavat lokitiedostot. Vertailin lokien virheilmoituksia sekä suorituskyvyn kuvaajissa näkyvän notkahduksen aikaleimaa ja löysin oikean virheilmoituksen, josta aloitin raportin teon.

Vian korjausprosessi etenee niin, että ensin kerätään tarvittavat lokit ja täytetään raporttipohja. Raporttipohjassa kuvataan havaittu vika sekä oman testausympäristön perustiedot ja mahdolliset muutokset. Kun raportti on huolellisesti täytetty ja mukaan on liitetty tarvittavat lokit, niin raportti osoitetaan viallisen ohjelmiston osasta vastaavalle tutkimustiimille ja tiimi alkaa tutkia lokeja. Mikäli tutkittavista lokeista ei löydy tarpeeksi tietoa, tekee tutkimustiimi pyynnön lisätietoja varten. Tutkimustiimi voi pyytää testausinsinööriä esimerkiksi uusimaan vian ja keräämään tutkimustiimin tarvitsemat lokit. Kommunikointiin käytetään aina Teams-keskustelua, jossa on mukana vian löytänyt testausinsinööri sekä tutkimustiimin jäseniä. Kun vika on löydetty ja korjattu, testausinsinööri tekee vielä korjauksen vahvistuksen korjatulla ohjelmistoversiolla.

Tällä viikolla minulla oli ohjelmassa myös kollegani sijaistaminen muutamana päivänä hänen ollessa poissa. Olimme osan päivästä Teams-puhelussa hänen kanssaan ja teimme testausta yhdessä, jotta näkisin, kuinka hänen testiympäristönsä toimii. Hänen testausympäristönsä solut eivät siirtyneet käyttövalmiuteen, joten pyysin häntä tekemään parametrimuutoksen. Olin tehnyt vastaavanlaisen muutoksen omaan testausympäristööni aikaisemmin. Muutos toimi myös hänen testausympäristöllään, joten pääsimme aloittamaan hänen konfiguraatiolleen määrätyt testit.

Testaus aloitettiin tiedonsiirtonopeustestillä. Tässä testissä testattiin tukiaseman solun maksimaalista tiedonsiirtonopeutta, toisin kuin ensimmäisen viikon raportissa, jossa testattiin tiedonsiirtonopeutta per käyttäjä. Saimme erinomaisia tuloksia heti ensimmäisillä testeillä ja kirjasimme testin hyväksytyksi. Testausympäristö oli nostettu käyttövalmiuteen nopeasti muutostöiden jälkeen ja seuraavat testit vaativat uuden kaapeloinnin radioille, jotta soluille saadaan enemmän käyttäjiä.

Testausympäristössä oli kolme radiota, jotka olivat hyvin erilaisia kuin oman ympäristöni radiot, koska niissä oli enemmän antenniliitäntöjä. Aloitin kaapelointiin perehtymisen tiedustelemalla kollegoideni testausympäristöjen kaapelointeja ja tutustumalla tämän testausympäristön nykyiseen kaapelointiin sekä logiikkaan. Kävin myös katsomassa paria testausympäristöä vastaavalla kaapeloinnilla. Menin laboratorioon tietokoneeni kanssa ja tein yhden solun käyttäjämäärätestejä muutettuani yhden radion kaapelointia, jotta huomaisin, tuleeko käyttäjämääriin positiivisia vai negatiivisia muutoksia. Muutoksia on hyvä tehdä ensin vain yhdelle radiolle, jotta nähdään vaikutukset nopeasti. Lisäksi näin tiedetään, että mikä oli alkuperäinen tilanne, kun kaikkia radioita ei ole muutettu kerralla.

Seuraavana päivänä työt täytyi aloittaa testauslinjojen tietokoneiden uudelleen käynnistyksellä, koska ne olivat menneet yön aikana jumiin automaattisten päivitysten takia. Sain myös vikaani tutkivalta ryhmältä pyynnön testata uudempaa ohjelmistoversiota, koska he olivat tutkimuksissaan päätyneet siihen, että vastaava vika olisi korjattu edellisenä päivänä. Asensin omaan testausympäristööni uudet ohjelmistot ja kirjasin uuden stabiilisuustestin aloitetuksi. Minun täytyi testata korjaus stabiilisuustestillä, koska edellinen stabiilisuustesti oli epäonnistunut havaitun vian takia. Kun olin saanut stabiilisuustestin käynnistettyä, jatkoin toisen testausympäristön radioiden kaapelointimuutosta, koska edelliset muutokset eivät vielä poikineet tarvittavaa käyttäjämäärää. Olin testannut tässä vaiheessa kollegoideni kaapelointimalleja. Pyysin apua kaapelointiongelmaan eräältä specialistilta, jonka kanssa olin aikaisemminkin tehnyt yhteistyötä. Hän tuli katsomaan radion kaapelointia ja oli sitä mieltä, että sen kuuluisi toimia. Tarkistimme testitapauksen asetukset, joihin tulikin pieniä muutoksia. Muutosten jälkeen käyttäjämäärät ja puheluiden laatu olivat jälleen hieman paremmat mutta käyttäjämäärä oli edelleen vajaa.

Seuraavana päivänä tarkistin stabiilisuustestin tilanteen ja kirjasin sen hetkisen tilan hyväksytyksi ja menin takaisin laboratorioon jatkamaan radioiden kaapelointia. Aloin muuttamaan kahden muun radion kaapelointia vastaavaksi kuin ensimmäisen, koska sillä kaapeloinnilla oli saatu tähän mennessä parhaat tulokset. Radioita kaapeloidessani huomasin yhden radioista olevan kaapeloitu eri tavalla kuin kaksi muuta. Tein pienen muutoksen ja kokeilin, kuinka kaapelointi toimii. Pyysin kollegaani tekemään sen radion solulla käyttäjämäärätestin. Testi osoitti kyseisen radion kaapeloinnin olevan oikea, sillä käyttäjämäärä oli testissä vaadittu ja se oli myös stabiili. Niinpä kaapeloin kaksi muuta radiota samalla tavalla. Testasimme käyttäjämääriä kaikilla kolmella solulla yhtäaikaaisesti. Saimme testin vaatimuksissa määritetyn käyttäjämäärän, joten kirjasimme testin hyväksytyksi.

Otin itselleni viikonlopputöitä, jotka sisälsivät oman testausympäristöni stabiilisuustestin seuraamista vian korjauksen vahvistamista varten. Lisäksi tein regressiotestejä testausympäristöllä, jota olimme kollegani kanssa käyttäneet aikaisemmin kuluneella viikolla. Jatkoisin regressiotestejä eräänlaisella käyttäjämäärätestillä. Tein pieniä muutoksia testitapaukseen, jolla olimme aikaisemmin testanneet solun maksimaalisen käyttäjämäärän. Seurasin käyttäjien resurssimääriä terminaalista, josta määrät näkyivät reaaliaikaisesti. Kun olin tehnyt tarvittavat muutokset testitapaukseen sain resurssimäärät pysymään maksimissa. Jätin testin pyörimään illaksi, jotta siitä piirtyy suorituskykyä mittaavat kuvaajat.

Sunnuntaina tarkistin stabiilisuustestin tuloksen, vahvistin löytämäni vian korjauksen ja ilmoitin tuloksen vikaa tutkineelle tiimille. Lisäksi minun täytyi yrittää testiä, jossa tukiasemaa kuormitetaan mahdollisimman suurella määrällä erilaista liikennettä. Testasin kollegani vanhaa testitapausta mutta sillä ei saavutettu riittävän suurta kuormitusta, joten päätin tehdä oman testitapauksen UEsimulaattorille. Olin jokin aika sitten tutkinut vastaavaa testiä kollegani kanssa ja päätin koittaa samanlaista testitapausta kuin hän oli käyttänyt, koska tiesin että hän oli saanut sillä korkean kuormituksen tukiasemalle. Tein simulaattorille testitapauksen, jolla on aikaisemmin saatu hyvä kuormitus tukiasemalle. Myöhemmin tarkistettuani kuormituksen sain luvan merkitä testin hyväksytyksi.

Yhteenvedona viikko kolme sisälsi työskentelyä kahdella eri testausympäristöllä. Tiimissämme lähes kaikkien testausympäristöjen konfiguraatiot ovat jokseenkin erilaisia, joten työskennellessä eri ympäristöillä täytyy tuntea ympäristön rakenne ja mahdolliset eroavaisuudet omaan ympäristöön nähden. Työskennellessä toisen testausinsinöörin testausympäristöllä täytyy myös huomioida, että muutoksia tehdessä täytyy pitää alkuperäiset tiedostot ja testitapaukset tallessa. Lisäksi muutoksia tulisi tehdä vain kopioihin ja muutokset tulisi kirjata ylös. Lisäksi useammalla ympäristöllä työskentely vaatii organisointikykyä sekä pelisilmää, jos jotain täytyy priorisoida.

4.4 Viikko 4

Tällä viikolla on määrä ajaa uusi 48 tunnin stabiilisuustesti. Testipyynnössä oli ilmoitettu eri ohjelmistojen halutut versiot, jotka täytyy päivittää meidän stabiilisuustestausympäristöihimme. Päivitykseen täytyi tehdä tiketti tukitiimille, joka vastaa servereiden alustojen päivityksestä. Tiketillä tarkoitetaan käytännössä työpyyntöä. Päivityksen vuoksi minun täytyi ajaa oma testausympäristöni alas.

Tiistaina oli kaksi palaveria, joissa katsottiin stabiilisuustestausympäristöjen testitapausten kehitysvaatimuksia ja testiautomaation tämänhetkistä tilannetta. Palavereista saa hyviä ideoita, mitä omalle testausympäristölle pitäisi kehittää, mutta aika toteutukselle on hyvin rajallista. Kollegani päivitti testausympäristöni testiautomaation ollessani kesälomalla. Minun pitäisi päivittää testiautomaatio, kun sille löytyy sopiva aika. Suunnitelmissani on pitää palaveri kollegani kanssa, jotta opin käyttämään testiautomaatiota tehokkaasti ja voin alkaa ajamaan sillä erilaisia testejä työpäivän päätteeksi. Suunnittelin ajoittavan palaverin seuraavalle viikolle, kun stabiilisuustestit on ajettu.

Keskiviikkona sain luvan nostaa testausympäristöni takaisin käyttöön serverin alustan päivityksen jälkeen. Päivityksen jälkeen ei ilmennyt ongelmia. Päivitin vCU:n ja vDU:n ohjelmistot testipyyntöä vastaaviksi ja seuraavaksi minun piti päivittää vDU:n ja radioiden välinen rajapinta. Tarkistin oikean version ja tein testiautomaatiotyökälulla ohjelmistopakettin, jonka siirsin käyttäjälleni. Päivityskomento ei jostain syystä toiminut. Sain komennon lopulta toimimaan, kun löysin vanhan Teams-keskustelun, jossa oli keskustelua aiheesta. Kun kaikki ohjelmistot oli päivitetty ja testausympäristöni tukiasema oli käyttövalmiudessa, tein tarvittavat kirjaukset testausraportointityökaluun ja käynnistin stabiilisuustestini.

Torstaina tarkistin stabiilisuustestini tilan eli suorituskyvyn kuvaajat, muistin, resurssit sekä lokit. Kun kaikki näyttivät olevan kunnossa, kirjasin testin sen hetkisen tilan järjestelmään. Kun stabiilisuustesti oli käynnissä, minulla oli aikaa auttaa kollegaani hänen testausympäristönsä kanssa. Testausympäristössä oli käyttäjämäärään liittyviä ongelmia ja yritin parhaani mukaan etsiä mahdollisia poikkeavuuksia asetuksista sekä UE-simulaattorilta. Tarkistin välillä, että stabiilisuustestini liikenne pysyi tasaisena. Yhtäkkiä liikenne oli nollassa ja eräällä komponentilla näkyi uudelleen käynnistymishälytyksiä. Kuten aina ongelmien ilmaantuessa, nytkin ensimmäinen tehtävä oli ottaa lokit heti, kun se olisi mahdollista. Paras tapa löytää lokeista oikeat vikailmoitukset, on tarkistaa tukiaseman hälytyshistoriasta aikaleimat ja katsoa lokeista, mitä juuri ennen sitä on tapahtunut. Tutkimukseen käytetty lokipaketti sisältää monia eri lokeja tukiaseman osista. Kollegani opetti minulle tutkimusten ohessa, mistä näkee tarkimmat tiedot uudelleenkäynnistymis tapauksissa. Löysinkin kyseisestä lokista osan, joka oli käynnistynyt uudelleen ensimmäisenä.

Yhteenvetona neljännellä viikolla työskenneltiin pitkälti stabiilisuustestiin valmistautumisen ja stabiilisuustestauksen parissa. Stabiilisuustestit vaativat usein monen eri alustan päivityksiä. Opin

tällä viikolla tekemään itse serverin alustan päivitystiketin sekä nostamaan oman testausympäristöni uudelleenasetuksen jälkeen käyttövalmiuteen. Työssämme oppii paljon muilta testausinsinööreiltä, joten on tärkeää kysyä neuvoja, mikäli jotain ei itse osaa ratkaista.

4.5 Viikko 5

Olin perjantain poissa sairastuttuani, joten kollegani tutki stabiilisuustestissä tapahtuneen uudelleenkäynnistymisen syyn ja teki siitä vikaraportin, josta aloitettiin tutkimus. Stabiilisuustesti oli myös päätetty käynnistää perjantaina uudestaan ja ajaa testiä viikonlopun yli. Tarkistin testin suorituskykyä kuvaavat kuvaajat, muistin kulutuksen sekä resurssit ja tein niistä kirjaukset testausraportointityökaluun. Vikaraportin Teams-keskustelussa tutkiva tiimi etsi komponentilta uudelleenkäynnistymisestä kirjautuvia lokeja, mutta lokit olivat ehtineet nollaantua ajan myötä.

Sain vikaa tutkivalta tiimiltä pyynnön yrittää uusia vian ja kerätä siitä lisää lokeja, joten loin tukiasemani kokonaan uusiksi ja käynnistin testin UE-simulaattorilla. Loin tukiasemani kokonaan uusiksi normaalin uudelleenkäynnistymisen sijasta, koska halusin olla varma, ettei uudelleenkäynnistys vaikuta vian uusiutumiseen. Vikaa uusittaessa täytyy osata ottaa huomioon toiminnot, jotka voivat vaikuttaa vian uusiutumiseen. Erilaiset uudelleenkäynnistykset ovat yksi niistä toiminnoista.

Annoin testin pyöriä päivän ja yön yli. Vika ei uusiutunut, joten ilmoitin asiasta vikaa tutkivalle tiimille. Löysin kuitenkin aivan uuden vian, jonka juurisyytä täytyi alkaa tutkimaan. Kokeneempi kollegani auttoi minua vian tutkimisessa, koska hänellä oli kokemusta vastaavasta viasta ja sen tutkimuksista. Otimme Teams-puhelun, jossa hän neuvoi minulle, kuinka eri komponenteilta kerätään lokeja. Tutkimme, löytyykö lokeista poikkeavuuksia vian ilmaantumishetkellä ja löysimme ainakin osasta keräämistämme lokeista vikaan viittaavia ilmoituksia, joista otin myös kuvankaappaukset. Kun olimme varmoja vian juurisyytä, keräsin kaikki keräämämme lokit kansioon ja tein sähköpostitse alustavan tutkimuspyynnön eräälle tutkimustiimille.

Sain seuraavana aamuna sähköpostia tutkimuspyyntöäni koskien. Toimitin tiimille aikaisemmin keräämäni lokitiedostot. Päätin varmistaa, uusiutuvatko löytämäni viat käynnistämällä stabiilisuustestin viikonlopuksi uusilla ohjelmistoversioilla. Lupasin raportoida tutkimustiimille, uusiutuiko vika viikonlopun stabiilisuustestin aikana.

Yhteenvedona raportoinnin viimeisellä viikolla työskenneltiin stabiilisuustestaukseen liittyvien tehtävien parissa. Tällä viikolla tutkittiin myös kahta eri vikaa, joihin liittyen tehtiin yhteistyötä vikaa korjaavien tiimien kanssa. Tiimeille kerättiin tarvittavia lokitiedostoja ja etsittiin niistä vikoihin viittaavia ilmoituksia. Kollegani auttoi minua lokien keräämisen kanssa. Nyt osaan kerätä vastaaviin vikoihin tarvittavat lokitiedostot ja hakea niistä oikeita vikaan viittaavia ilmoituksia.

5 POHDINTA

Opinnäytetyössä seurattiin viiden viikon ajan työtäni testausinsinöörinä Nokiolla PET-organisaatiossa pilvipohjaisten tukiasemien suorituskykytestauksen parissa. Valitsin raportointitavaksi päiväkirjaraportoinnin ja viikkokohtaisen lähestymistavan. Kukin viikko keskittyy eri teemoihin, jotta lukija saa mahdollisimman kattavan kuvan testausinsinöörin työstä. Lisäksi kunkin viikon pohjalta arvioitiin käytössä olevia toimintamalleja sekä esitettiin mahdollisia kehitysehdotuksia tehokkaampiin työskentelytapoihin.

Raportoinnin ajanjaksolla suoritin regressiotestausta, toisten testausinsinöörien sijaistusta, stabiilisuustestausta, viankorjausprosesseja, testausympäristön kaapelointia sekä testitapausten muokkauksia. Sain ajettua useamman testin hyväksytysti. Mainittava saavutus on myös toisen testausympäristön onnistunut radioiden kaapelointimuutos. Tein raportoinnin aikana myös useamman stabiilisuustestin, joista sain tehdä vikaraportteja ja osallistua vian korjausprosesseihin.

Ensimmäisellä viikolla tehty regressiotestaus osoitti, että testausinsinöörien olisi hyvä pitää testauspäiväkirjaa, koska regressiotestit täytyy pystyä toistamaan tietyin väliajoin. Tällaisella suunnitelmallisuudella testit voidaan toistaa myöhemmin tehokkaasti ja välttää turhalta työltä esimerkiksi poissaolojen aikana. Toisella viikolla saatiin klassinen esimerkki varmuuskopioiden tärkeydestä, kun tietokoneiden päivitykset kaatoivat UE-simulaattorin testausohjelmiston. Ongelmatilanteita ennakoimalla välttää ei-toivotuilta tilanteilta, joissa voidaan menettää tärkeää tietoa. Kolmannella viikolla työskenneltiin kahdella testausympäristöllä, joista toisella tehtiin stabiilisuustestausta ja toisella regressiotestausta. Työskenneltäessä toisen insinöörin testausympäristöllä olisi tärkeää pitää kirjaa tehdyistä muutoksista sekä tehdä muutoksia vain tiedostojen kopioihin. Kahdella testausympäristöllä työskentely vaatii hyvää organisointikykyä sekä keskittymistä kulloisellakin testausympäristöllä vaadittaviin tehtäviin. Neljänneltä ja viidenneltä viikolta keskeisimpänä asiana nousi esiin yhteistyön merkitys oman ammatillisen kehityksen tukena. Apua pyytämällä ongelmatilanteista päästään nopeammin eteenpäin kuin yksin asiaa pohtimalla. Kunkin viikon pohjalta voidaan todeta, että olisi hyvä tarkastella ja kehittää systemaattisia toimintatapoja työn tehokkuuden lisäämiseksi.

Henkilökohtaisella tasolla asettamani tavoitteet olivat kehittyä eri lokien keräyksessä ja vian etsinnässä. Mielestäni nämä tavoitteet täyttyivät hyvin. Opinnäytetyöprosessin aikana oli useita tehtäviä, joissa minun täytyi kerätä paljon erilaisia lokitiedostoja. Opin kollegoilta uusia tapoja juuri lokien keräykseen ja vikojen etsintään liittyen. Kuten olen raportoinnin aikana maininnut, työsamme oppii paljon toisilta testausinsinööreiltä ja apua kannattaa aina pyytää.

Seuraavaksi minulla on tavoitteena päivittää testausympäristöni testiautomaatio, joka on jäänyt kaikkien muiden tehtävien takia hieman pienelle huomiolle. Lisäksi minun täytyy perehtyä tarkemmin automaation toimintaan, jotta saan siitä kaiken hyödyn irti. Minun olisi myös tarkoitus kehittää tämänhetkisen stabiilisuustestin liikennettä kuormittavammaksi ja monipuolisemmaksi. Voisin esimerkiksi tehdä useamman testitapauksen, joissa kaikissa olisi hieman erilainen liikenne ja kuormitus.

Opinnäytetyössä oli haasteellista salassapitovelvollisuuden tuoma rajoite siihen, kuinka yksityiskohtaisesti asioista pystyi kirjoittamaan, koska opinnäytetyö on julkinen. Tämän vuoksi asiat täytyi ilmaista ammatillisten termien lisäksi myös yleistajuisella tasolla. Haasteita tuotti myös löytää oikeat suomenkieliset vastineet kuvaamaan tietoliikennetekniikan englanninkielisiä termejä ja lyhenteitä. Haasteista huolimatta koen, että opinnäytetyöprosessi vei ammatillista kehittymistäni erittäin paljon eteenpäin. Prosessin aikana minun täytyi useaan otteeseen pysähtyä miettimään, miten jäsentää ja selittää monimutkaisia työtehtäviä selkeällä ja johdonmukaisella tavalla. Tämä vaati oman työn reflektointia sekä oman asiantuntijuuden kehittämistä esimerkiksi perehtymällä alan tekniseen tietoon.

LÄHTEET

Hooda, Itti & Chhillar, Rajender Singh 2015. Software test process, testing types and techniques. International Journal of Computer Applications, 111 (13), 10–14. Hakupäivä: 10.11.2023. <http://dx.doi.org/10.5120/19597-1433>

Khan, Mohd Ehmer 2010. Different forms of software testing techniques for finding errors. International Journal of Computer Science Issues (IJCSI), 7 (3), 11–16. Hakupäivä: 10.11.2023. https://www.researchgate.net/profile/A-Zaidan/publication/46093547_Towards_Corrosion_Detection_System/links/549239a60cf2484a3f3e0b22/Towards-Corrosion-Detection-System.pdf#page=19

Miller, Michael 2013. Wireless Networking Absolute Beginner's Guide. Que Publishing. Hakupäivä: 25.10.2023. O'Reilly Online Learning: Academic/Public Library Edition. Vaatii käyttöoikeuden. <https://learning.oreilly.com/library/view/wireless-networking-absolute/9780133381313/>

Naik, Kshirasagar & Tripathy, Priyadarshi 2011. Software testing and quality assurance: theory and practice. John Wiley & Sons, Inc., Hoboken, New Jersey. Hakupäivä 10.11.2023. <http://ndl.ether-net.edu.et/bitstream/123456789/89778/1/staqtpsn.pdf>

Nokia Oyj 2023 a. Finland. Hakupäivä: 20.09.2023. <https://www.nokia.com/about-us/careers/our-locations/finland/>.

Nokia Oyj 2023 b. Open RAN explained. Hakupäivä: 30.10.2023. <https://www.nokia.com/about-us/newsroom/articles/open-ran-explained/>.

Toskala, Antti & Poikselkä, Miikka 2020. "5G Architecture." Teoksessa 5G technology: 3GPP new radio (toim, Harri Holma, Antti Toskala & Takehiro Nakamura). John Wiley & Sons. Hakupäivä: 25.10.2023. O'Reilly Online Learning: Academic/Public Library Edition. Vaatii käyttöoikeuden. <https://learning.oreilly.com/library/view/5g-technology/9781119236313/>