



jamk

Onnistuneen hyväksymistestauksen saavuttaminen monitoimittajaympäristössä

Jenna Eilo

Opinnäytetyö, AMK

Huhtikuu 2024

Tieto- ja viestintätekniikan tutkinto-ohjelma

Eilo, Jenna

Onnistuneen hyväksymistestauksen saavuttaminen monitoimittajaympäristössä

Jyväskylä: Jyväskylän ammattikorkeakoulu. Huhtikuu 2024, 40 sivua.

Tieto- ja viestintätekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: Kyllä

Tiivistelmä

Opinnäytetyön aiheena oli onnistuneen hyväksymistestauksen saavuttaminen monitoimittajaympäristössä. Tutkimuksen päätavoitteena oli löytää kriittisiä kehityskohteita, joiden parantamisen avulla voitaisiin taata parempi hyväksymistestauksen sujuminen monitoimittajaympäristössä. Opinnäytetyössä tutkittiin, mitkä ovat hyväksymistestauksen ja monitoimittajaympäristön haasteet sekä mahdollisuudet, niin yhdessä kuin erikseen.

Opinnäytetyö tehtiin toimeksiantona Tietoevry Finland Oy:lle. Opinnäytetyö toteutettiin kyselytutkimuksen muodossa. Kyselyhaastattelu suoritettiin anonymisti alan asiantuntijoille, joilla on kokemusta monitoimittajaprojekteista ja hyväksymistestauksesta. Opinnäytetyössä hyödynnettiin havainnointia sekä aiheeseen liittyviä artikkeleita ja tutkimuksia.

Opinnäytetyön tuloksena saatiin rajattua osaa-alueet, joita tulee huomioida, jotta saavutettaisiin parempi kyvykkyys hyväksymistestaukseen monitoimittajaprojekteissa. Esimerkkejä näistä osa-alueista olivat prosessin suunnittelu ja hallinta, sekä toimittajien yhteistyö ja kommunikointi. Tutkielman johtopäätöksenä oli, ettei onnistuneeseen hyväksymistestaukseen monitoimittajaympäristössä ole vain yhtä oikeaa ratkaisua. Tutkimuksen avulla syntyi myös käsitys siitä, ettei vastuu hyväksymistestauksen epäonnistumisesta ole vain yhdellä henkilöllä, vaan se on aina laajemman ihmisryhmän käsissä.

Avainsanat (asiasanat)

Hyväksymistestaus, Kyselytutkimus, Monitoimittajaprojekti, Monitoimittajaympäristö, Ohjelmistoprosessi, Ohjelmistotestaus, Prosessi, Testaus

Muut tiedot (salassa pidettävät liitteet)

-

Eilo, Jenna

How to achieve successful user acceptance testing in multivendor environment

Jyväskylä: JAMK University of Applied Sciences, April 2024, 40 pages.

Degree Program in Information and Communication Technology. Bachelor's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

The topic of the thesis was how to achieve successful acceptance testing in a multivendor environment. The main purpose of the study was to identify critical areas that needed to be improved to guarantee better acceptance testing in a multivendor environment. The thesis investigated the challenges and opportunities of acceptance testing and multivendor environments, both collectively and individually.

The thesis was commissioned by Tietoevry Finland Oy, and it was organized in the form of a survey research. The survey interviews were operated anonymously for experts, who have experienced in multivendor projects and acceptance testing. Observation, relevant articles and studies from the topic were also utilized in the research.

The key areas were delineated in order to achieve better capability in acceptance testing in multivendor projects as a result of the thesis. Examples of these areas included process planning and management, as well as vendor collaboration and communication. The conclusion of the thesis was that there is no single correct solution for successful acceptance testing in a multivendor environment. The research also led to the understanding that the responsibility for the failure of acceptance testing lies not only with one individual but within a broader group of people.

Keywords/tags (subjects)

Acceptance testing, User acceptance testing, Survey research, Multivendor project, Multivendor environment, Software process, Software testing, Process, Testing

Miscellaneous (Confidential information)

-

Sisältö

1	Johdanto	3
2	Tutkimusasetelma	4
2.1	Toimeksiantajan esittely	4
2.2	Tutkimuksen sisältö ja menetelmät	4
2.3	Tutkimuskysymykset	5
3	Ohjelmistotestaus	5
3.1	Ohjelmistotestauksen määritelmä	5
3.2	Testauksen peruseriaatteet	6
3.2.1	Testien suunnittelu	6
3.3	Testauksen eri tasot	7
3.3.1	Yksikkötestaus	8
3.3.2	Integraatiotestaus	8
3.3.3	Järjestelmätestaus	9
4	Hyväksymistestaus	9
4.1	Hyväksymistestauksen määritelmä ja ominaisuudet	9
4.2	Hyväksymistestauksen hyödyt	11
4.3	Hyväksymistestauksen haasteet	11
5	Ohjelmistoprosessi	12
5.1	Ohjelmistoprosessin määritelmä	12
5.2	Elinkaarimallit ohjelmistokehityksessä	13
5.2.1	Elinkaarimallit	13
5.2.2	Ketterä kehittäminen	14
5.2.3	Scaled agile framework	14
6	Monitoimittajaympäristö	15
6.1	Monitoimittajaympäristö termin määritelmä	15
6.2	Monitoimittajaympäristön mahdollisuudet	16
6.3	Monitoimittajaympäristön haasteet	17
7	Tutkimusosio	17
7.1	Kyselyhaastattelu	17
7.2	Epäonnistuneen hyväksymistestauksen vaikutukset	18
7.3	Monitoimittajaympäristön tuomat haasteet hyväksymistestaukseen	19
7.4	Ohjelmistoprosessin kehitysehdotukset	20
7.4.1	Hyväksymistestaus osaksi prosessia	21

7.4.2	Hyväksymistestauksen hallinta.....	21
7.4.3	Kouluttaminen ja tuki	22
7.4.4	Aikataulut ja resurssit	23
7.4.5	Palaute ja jatkuva kehitys	24
7.4.6	Menetelmät käytännön tasolle	24
7.5	Toimintatapojen ja käytänteiden kehitysehdotukset.....	25
7.5.1	Testausympäristö ja data.....	25
7.5.2	Yhteisymmärrys testattavasta kohteesta	26
7.5.3	Yhteistyö ja kommunikaatio	26
7.5.4	Dokumentointi ja raportointi.....	28
8	Päätäntä.....	29
8.1	Tutkimuksen eettisyys.....	29
8.2	Tutkimuskysymysten tarkastelu.....	29
8.3	Jatkokehitysehdotukset	30
8.4	Johtopäätökset.....	31
8.5	Yhteenveto	31
	Lähteet	33
	Liitteet	36
	Liite 1. Kyselyhaastattelu	36
	Kuviot	
	Kuvio 1. Testauksen tasot	8
	Kuvio 2. Monitoimittajaympäristön mallinnus	16

1 Johdanto

Yhä useammat ohjelmistot rakentuvat IT-alalla nykypäivänä monitoimittajaympäristöissä. Monitoimittajaympäristöissä ohjelmisto tai palvelu rakentuu eri toimittajien tai palveluntarjoajien toimesta. Ideana on, että organisaatio hyödyntää monitoimittajaympäristöä täyttääkseen tarpeitaan tai toteuttaakseen monimutkaisia projekteja, johon tarvitaan esimerkiksi eri osa-alueen ammattilaisia. Monitoimittajaprojekteissa toimittajia voi olla muutamasta toimittaja organisaatiosta aina useisiin kymmeniin.

Monitoimittajaympäristö tuo paljon mahdollisuuksia, mutta se voi myös tuoda lukuisia eri haasteita projekteille. Haasteita voi muodostua projektin eri osa-alueille, kuten esimerkiksi testaukselle ja eritoten hyväksymistestaukselle. Hyväksymistestaus on jo itsessään haastava testauksen taso, joten ei ihme, että monitoimittajaympäristö voi tehdä siitä usein entistä haastavamman. Jotta monitoimittajaprojektit etenisivät alusta loppuun saakka jouhevasti, on tiedostettava monitoimittajaympäristön tuomat mahdolliset haasteet ja hallittava ongelmakohtat onnistuneen lopputuloksen aikaan saamiseksi.

Tämä opinnäytetyö toteutettiin toimeksiantona Tietoevry Finland Oy:lle. Kuten moni muukin IT – alan yritys, Tietoevry Finland Oy tekee myös päivittäin töitä yhdessä muiden alan yritysten kanssa erilaisissa monitoimittajaprojekteissa. Salassapitovelvollisuuden vuoksi opinnäytetyössä ei käsitelty tarkkoja asiakkuuksiin tai projekteihin liittyviä yksityiskohtia. Sen sijaan tässä tutkimuksessa keskityttiin kuvaamaan asioita yleisluonteisesti ja löytämään ratkaisuja tutkimuskysymyksiin kunnioittaen salassapitovelvollisuutta.

Opinnäytetyön tavoitteena oli saada selville, mitkä ovat ne tekijät, jotka voivat mahdollisesti vaikuttaa negatiivisesti hyväksymistestaukseen monitoimittajaympäristöissä. Tutkimuksen tarkoituksena oli myös selvittää, miten monitoimittajaprojektien prosessia, käytäntöjä ja tapoja tulisi kehittää, jotta saataisiin onnistunut hyväksymistestaus aikaiseksi monitoimittajaympäristössä. Tutkimuksen tavoitteena oli tarjota käytännön ratkaisuja hyväksymistestauksen haasteisiin.

2 Tutkimusasetelma

2.1 Toimeksiantajan esittely

Opinnäytetyön toimeksiantajana toimi Tietoevry Finland Oy. Tietoevry on suomalainen maailmanlaajuinen teknologiayritys, joka tarjoaa laajasti erilaisia digitaalisia palveluita sekä ratkaisuja asiakkaille. Tietoevry toimii laajasti eri yhteiskunnan osa-alueilla, kuten pankkialalla, terveydenhuollossa, teollisuudessa, kasvatuksessa ja opetuksessa. Tietoevry syntyi vuonna 2019, kun suomalainen Tieto ja norjalainen EVERY yhdistyivät. Kahden pohjoismaalaisen firman yhdistyminen vahvisti entistä enemmän yhtiön asemaa Pohjoismaissa. Tietoevryllä työskentelee yli 24 000 alan asiantuntijaa ympäri maailmaa. Tietoevry palvelee tuhansia sekä julkisen että yksityisen sektorin asiakkaita yli 90 maassa. (Tuemme yhteiskunnan kehitystä vastuullisen teknologian avulla 2023.)

2.2 Tutkimuksen sisältö ja menetelmät

Tämän opinnäytetyön tavoitteena oli tunnistaa hyväksymistestauksen mahdollisia ongelmia monitoimittajaympäristössä. Tutkimuksen tarkoituksena oli koota selkeitä ratkaisuja ja parannusehdotuksia, joiden avulla hyväksymistestaus etenisi jouhevasti alusta loppuun saakka. Tavoitteena oli, että tutkimukseen otetaan mukaan eri kokoisissa monitoimittajaprojekteissa työskentelevien asiantuntijoiden näkemyksiä ja kokemuksia aiheeseen.

Opinnäytetyön tarkoituksena oli saada syvälinen ymmärrys aiheesta. Tämän vuoksi tutkimusvaiheessa käytössä oli kvalitatiivinen kyselyhaastattelu. Opinnäytetyön aiheen vuoksi kyselyhaastattelu haluttiin toteuttaa tiiviille asiantuntijajoukolle. Kysely toteutettiin anonyymisti eri tietotekniikan alan asiantuntijoille, jotka parhaillaan työskentelevät tai ovat työskennelleet aiemmin monitoimittajaympäristössä. Opinnäytetyössä käytin myös apuna omia kokemuksia ja havaintoja aiheeseen liittyen. Olen toiminut muutaman vuoden ajan järjestelmätestaajana ja testiautomaatioasiantuntijana monitoimittajaympäristöisessä projektissa. Olen myös ollut tekemisissä hyväksymistestauksen asiantuntijoiden kanssa ja olen päässyt seuraamaan hyvinkin läheltä, miten hyväksymistestaus on edennyt monitoimittajaympäristössä.

2.3 Tutkimuskysymykset

Opinnäytetyön aihe-ehdotus tuli toimeksiantajan toimesta. Hyväksymistestaus monitoimittajaympäristössä on aiheena hyvin laaja kokonaisuus. Laajan aiheen vuoksi tutkimusaihetta rajattiin työn alkuvaiheessa tutkimuskysymysten avulla. Tutkimuksen näkökulma tuotiin esille tutkimuskysymysten kautta. Opinnäytetyössä keskeisimpiä tutkimusongelmasta syntyneitä tutkimuskysymyksiä oli kolme kappaletta:

1. Mitä pitäisi muuttaa prosessissa ja tiedonkulussa, jotta saadaan parempi kyvykkyys hyväksymistestaukseen monitoimittajaympäristössä?
2. Miten varmistetaan, että kaikkien toimittajien komponentit, testaussuunnittelu ja testaus on tietyn tasoista, jotta hyväksymistestaus onnistuu jouhevasti?
3. Kuinka varmistetaan, että testitapaukset heijastavat todellisia käyttötapauksia ja ovat valideja hyväksymistestausvaiheessa, kun kyseessä on monitoimittajaympäristö?

On olennaista nostaa esille, että tutkimus toteutettiin nimenomaan toimittavan organisaation näkökulmasta. On myös keskeistä huomioida, että vaikka tutkimuskysymykset saattavat sisältää negatiivisia painotuksia, tutkimuksen tarkoituksena ei ollut olettaa, että hyväksymistestaus epäonnistuu aina monitoimittajaympäristössä. Tutkimuksessa haluttiin löytää mahdollisiin ja yleisiin ongelmiin ratkaisuja. Tutkimuksen jatkokehitystä ajatellen myös palvelun tilaavan osapuolen ajatukset ja kokemukset hyväksymistestauksesta monitoimittajaympäristössä voisivat tuottaa huomattavasti erilaisia näkemyksiä aiheeseen.

3 Ohjelmistotestaus

3.1 Ohjelmistotestauksen määritelmä

Ohjelmistojen määrä kasvaa jatkuvasti nykypäivänä. Ei mene päivääkään, ettei ihminen tarvitsisi jotain digitaalista palvelua suorittamaan perusarkiaskareita. Mikäli ohjelma ei toimi odotetulla tavalla, se voi olla hyvin haitallista niin kuluttajalle kuin ohjelman tehneelle yritykselle. Mikäli ohjelmisto toimii odottamattomalla tavalla, ihminen voi menettää aikaa, rahaa tai pahimmillaan ihmisen terveys voi olla uhattuna. (ISTQB® Perustason sertifikaattisisältö 2023, 15.)

ISTQB® Perustason sertifikaattisisältö (2018, 12) määrittelee ohjelmistotestauksen olevan keino vähentää tuotannossa tapahtuvien häiriöiden mahdollisuutta. Ohjelmistotestaus on myös keino, jolla arvioidaan ohjelmiston laatua (ISTQB® Perustason sertifikaattisisältö 2018, 12). JMC Nordic

puolestaan määrittelee blogitekstissään Mitä on ohjelmistotestaus ja mihin ohjelmistotestausta tarvitaan (2021) ohjelmistotestauksen olevan menetelmien kokoelma, joiden avulla on tarkoitus varmistaa, että ohjelma toimii odotusten mukaisesti ja virheettömästi. Ohjelmistotestauksen avulla varmistetaan, että uusi koodi on yhteensopiva vanhan koodin kanssa (Mitä on ohjelmistotestaus ja mihin ohjelmistotestausta tarvitaan 2021).

Ohjelmistotestaus on toimenpiteiden joukko, joiden tarkoituksena on löytää vikoja ohjelmiston testattavista kohteista. Mikäli testauksen suunnittelee, hallinnoi ja sen työmäärän arvioi oikein, testaus voi parantaa ohjelmiston laatua huomattavasti. (ISTQB® Perustason sertifikaattisisältö 2023, 15.) Mitä myöhemmin ohjelmistossa piilevät viat löydetään, sitä kalliimmaksi niiden korjaus tulee (Mitä on ohjelmistotestaus ja mihin ohjelmistotestausta tarvitaan 2021). Blogitekstissä Mitä on ohjelmistotestaus ja mihin ohjelmistotestausta tarvitaan (2021) ohjelmistojen vikojen löytämisen tärkeyttä kuvataan seuraavanlaisesti: ”Julkaisuun päässyt vika on myös laajemman yleisön arvosteltavana kuin kehitysvaiheessa kiinni saadut bugit.”

3.2 Testauksen peruseräatteen

3.2.1 Testien suunnittelu

Testien suunnittelussa dokumentoiminen on yksi tärkeimmistä asioista. Testien suunnitteluun kuuluu esimerkiksi testitapausten luominen. Testitapaus, eli englanniksi test case, on karkeasti määriteltynä kirjalliseen muotoon tehty kokoelma ohjeita. Ohjeiden tarkoituksena on testata esimerkiksi tietyn ohjelmiston osan tai kokonaisuuden toiminnallisuutta. Testitapaukset sisältävät askeleita eli steppejä, joita yksi kerrallaan testaaja suorittaa testauksen aikana. Askel pitää sisällään tehtävän toimenpiteen, eli toiminnon mitä testaajan tulee tehdä, syötteen ja odotetun tuloksen. (Leppäniemi 2014, 22.)

Testaaja käy testitapauksen läpi yksi askel kerrallaan. Jos toteutettu tulos poikkeaa testi askeleesta olevasta odotetusta tuloksesta, voidaan havaita, ettei ohjelmisto toiminut odotetulla tavalla. Poikkeavuus kirjataan ylös ja voidaan alkaa selvittämään, onko kyseessä vika eli bugi. Testitapauksen suorittamisen epäonnistuminen ei aina tarkoita, että ohjelmistossa on vikaa. On aina selvitettävä, testaako testitapaus varmasti oikeaa asiaa ja onko testitapauksen toimenpide, syöte ja oletettu tulos paikkansapitävä. Vika eli bugi on ihmisen virheestä johtuva

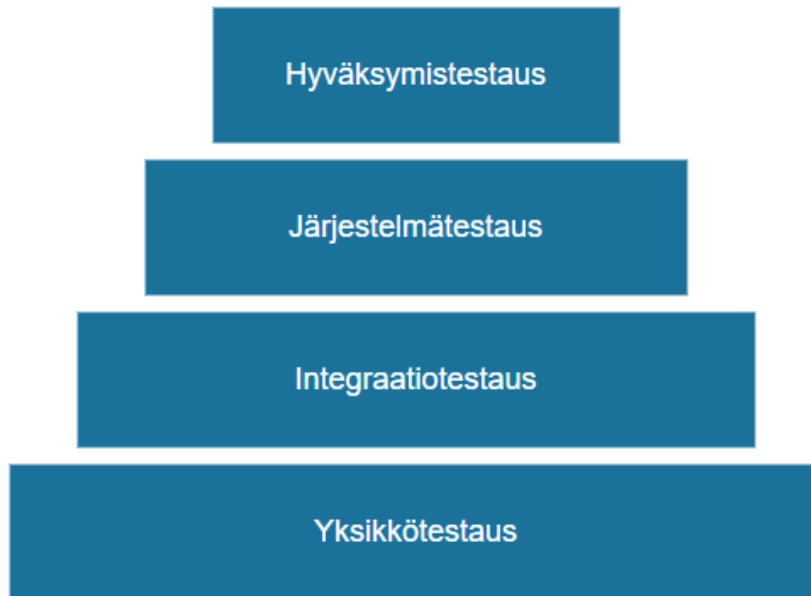
ohjelmistokoodiin syntynyt vika, joka saa aikaan sen, että ohjelmisto ei toimi niin kuin sen pitäisi toimia (ISTQB® Perustason sertifikaattisisältö 2018, 14). Testitapausten suuri etu on siinä, että niiden avulla voidaan heti havainnoida missä kohtaa ohjelmistoa virhe löytyy. Virhe on myös helppo toistaa uudelleen seuraamalla samoja testitapausten askelia. Testitapausten askelten oikeellisuus on testauksen kannalta kriittistä. Jos askeleiden toiminto, syöte tai odotettu lopputulos on kirjattu väärin, raportoidaan mahdollisesti turhia bugeja. Tällaisia testejä kutsutaan vääriksi positiivisiksi testeiksi. Nämä ovat testejä, joista syntyy bugeja, jotka eivät ole oikeasti virheitä. Väärät negatiiviset testit ovat myös ongelmallisia. Ne ovat testejä, joiden olisi pitänyt löytää ohjelmistossa oleva vika, mutta ne eivät löytäneet kyseistä vikaa. Kaikissa tapauksissa viat eivät ole lähtöisin kehittäjien tekemistä ohjelmointivirheistä, vaan virhe voi syntyä jo varhaisessa vaiheessa prosessia. Virhe voi syntyä jo esimerkiksi vaatimusten teko vaiheessa. (ISTQB® Perustason sertifikaattisisältö 2018, 14.)

Hyväksymistestauksessa testien suunnittelussa hyödynnetään käyttötapauksia. Käyttötapaus eli use case kuvaa, kuinka järjestelmän tulisi käyttäytyä tietyissä tilanteissa. Käyttötapauksen tarkoitus on auttaa hahmottamaan selkeästi, mitä tapahtuu, kun käyttäjä käyttää ohjelmaa. Niiden tarkoitus on myös havainnollistaa, miten ohjelma on vuorovaikutuksessa toisiin järjestelmiin. (ISTQB® Perustason sertifikaattisisältö 2018, 53.) Hyväksymistestauksessa käyttötapauksista muodostettujen testitapausten avulla varmistetaan, että järjestelmä täyttää käyttäjien tarpeet ja toimii niin kuin se on suunniteltu.

3.3 Testauksen eri tasot

Testaustasot ovat testaustehtävien joukkoja (ISTQB® Perustason sertifikaattisisältö 2018, 27). Eri testaustasot voivat tapahtua joko perätysten tai ajallisesti myös samaan aikaan omissa ympäristöissä (ISTQB® Perustason sertifikaattisisältö 2023, 28). ISTQB® Perustason sertifikaattisisältö (2023, 28–29) määrittelee, että eri testaustasoja ovat hyväksymistestauksen lisäksi muun muassa komponenttitestaus eli yksikkötestaus, järjestelmätestaus, komponentti-integraatiotestaus ja järjestelmäintegraatiotestaus. Tässä opinnäytetyössä testaustasot on karkeasti jaettu yksikkötestaukseen, integraatiotestaukseen, järjestelmätestaukseen ja hyväksymistestaukseen (ks. kuvio 1). Kuviossa 1 on kuvattu eri testauksen tasot ja havainnoitu,

missä järjestyksessä eri testauksen tasot voivat ilmetä. Hyväksymistestausta tullaan käsittelemään neljännessä kappaleessa tarkemmin.



Kuvio 1. Testauksen tasot (Mitä on ohjelmistotestaus ja mitä hyötyä siitä on 2022, muokattu)

3.3.1 Yksikkötestaus

Yksikkötestauksessa keskitytään testaamaan, että ohjelmiston osa eli yksikkö toimii oikein itsenäisesti (Vornanen 2022, 10). Yksikkötestauksen vastuussa ovat usein ohjelmoijat eli kehittäjät. Yksikkötestaus tapahtuu usein kehitysympäristössä ja se suuntautuu yksittäisiin ohjelman komponentteihin. (ISTQB® Perustason sertifikaattisisältö 2023, 28.) Ideana yksikkötestauksessa on, että se tuo eräänlaista varmuutta siitä, että ohjelmisto pysyy toimintakunnossa, vaikka uusia ominaisuuksia lisättäisiin siihen. Mikäli yksikkötestit menevät edelleen läpi, saa kehittäjä ja projektitiimi varmuutta siitä, ettei ohjelmassa ole mennyt mitään rikki muutoksien aikana. Yksikkötestejä kehitetään jatkuvasti lisää sitä mukaa mitä ohjelman osakin kehittyy eteenpäin ja ne ovat usein automaattisesti ajettavia testejä. (Yksikkötestaus 2023.)

3.3.2 Integraatiotestaus

Vornanen (2022, 10) kertoo, että Choudary (2020) määrittelee, kuinka integraatiotestauksen tarkoituksena on varmistaa, että yksikkötestauksessa testatut osat toimivat oikein, kun ne ovat

liitettyinä toisiinsa. Integraatiotestauksen keskipisteenä on eri järjestelmien ja komponenttien vuorovaikutus eli kuinka data liikkuu lähtöpisteestä päätepisteeseen ohjelmistojen välillä. Integraatiotestauksessa tarkastelun kohteena on rajapintojen laatu. Viat eivät aina piile ohjelmissa, vaan ne voivat myös olla rajapinnoissa. Siksi on elintärkeää testata, että myös rajapinnat toimivat halutulla tavalla. (ISTQB® Perustason sertifikaattisisältö 2018, 28.)

Integraatiotestaus nähdään usein haastavana ja virheen paikantaminen voi olla monimutkaista ja huomattavasti hitaampaa, mitä esimerkiksi yksikkötestauksessa (Seppälä 2020, 6).

3.3.3 Järjestelmätestaus

Järjestelmätestaus eli toiselta nimeltään systeemitestaus tapahtuu testiympäristössä, ja sen suorittavat usein ammattitestaajat. Järjestelmätestausta ohjaavat usein määrittelydokumentit ja järjestelmätestauksessa varmennetaankin, että ohjelma toimii määritysten mukaisesti. Järjestelmätestaus voi olla toiminnallista tai ei – toiminnallista. Toiminnallinen testaus tarkoittaa sitä, että tarkastellaan toimintoja, joita ohjelmiston tulisi suorittaa. Ei – toiminnallisessa testauksessa arvioidaan esimerkiksi ohjelmiston suorituskykyä, käytettävyyttä ja myös tietoturvaa. (ISTQB® Perustason sertifikaattisisältö 2023, 29.) Testattava kohde voi olla jokin suurempi ohjelmiston kokonaisuus tai pieni yksittäinen ohjelmistoon tehty muutos. Testaajilla on käytössään erilaisia menetelmiä ja toimintatapoja, joiden avulla voidaan suunnitella kattava testaus kyseiseen kohteeseen.

4 Hyväksymistestaus

4.1 Hyväksymistestauksen määritelmä ja ominaisuudet

Hyväksymistestaus on hyvin tärkeä osa ohjelmistokehitysprosessia. ISTQB® Perustason sertifikaatti (2023, 29) määrittelee hyväksymistestauksen seuraavanlaisesti: “Hyväksymistestaus keskittyy kelpuutukseen ja käyttöönottovalmiuden osoittamiseen, mikä tarkoittaa, että järjestelmä täyttää käyttäjän liiketoimintatarpeet”. UAT testaus, eli hyväksymistestaus: Mitä se tarkoittaa ja miksi se on tärkeää (2023) blogiteksti määrittelee puolestaan, kuinka hyväksymistestaus on ohjelmistoprosessissa testauksen viimeinen vaihe ja hyväksymistestaus tapahtuu ennen kuin tuote siirtyy käyttöönottoon eli julkaistaan asiakkaille käytettäväksi.

Hyväksymistestaus eroaa huomattavasti järjestelmätestauksesta. Hyväksymistestauksessa esimerkiksi tavoitteet ja näkökulmat eroavat paljon järjestelmätestauksesta.

Hyväksymistestauksessa esimerkiksi vikojen löytäminen ei useinkaan ole yksi päätavoitteista. Liiallinen vikojen löytyminen vasta hyväksymistestauksen aikana voi olla suuri riski projektille ja projektin valmistumiselle. Hyväksymistestauksen aikana saadaan hyvin tärkeää tietoa siitä, onko ohjelmisto lainkaan julkaisukunnossa. (ISTQB® Perustason sertifikaattisisältö 2018, 32.) Oleellista on, että on muodostettu juuri oikeanlaiset hyväksymistestauksen aloitus - ja lopetus kriteerit (Rinne 2016, 39). Jo ennen hyväksymistestauksen käynnistymistä, on oltava valmiina hyväksymiskriteerit, jotka ovat muodostettu asiakkaan odotusten pohjalta (Keskitalo 2012, 10).

Hyväksymistestauksen voi suorittaa esimerkiksi loppukäyttäjä, sidosryhmän edustaja, asiakas tai joku muu työhön määrätty henkilö. Ihanteellisessa tapauksessa hyväksymistestauksen suorittaisivat kuitenkin ohjelmiston lopulliset käyttäjät (ISTQB® Perustason sertifikaattisisältö 2023, 29). Jokainen, joka suorittaa hyväksymistestauksen on kuitenkin tiedostettava, mitkä ovat järjestelmän liiketoimintatarpeet ja täyttään järjestelmä sen odotukset ja vaatimukset.

Koska eri testausasoilla käytetään ihanne tapauksessa omia testausympäristöjä, joissa testaus suoritetaan, hyväksymistestaus tapahtuu luonnollisesti hyväksymistestausympäristössä. Jotta hyväksymistestaus olisi mahdollisimman onnistunut, se vaatii aina oman ympäristön testaukselle, ja testausympäristön tulisi olla mahdollisimman samankaltainen tuotannon ympäristön kanssa. (ISTQB® Perustason sertifikaattisisältö 2018, 27.)

Hyväksymistestausta on useaa eri muotoa. On olemassa asiakkaan hyväksymistestaus, käyttöön soveltuvuuden hyväksymistestaus, toimittajan hyväksymistestaus, tuotannollinen hyväksymistestaus ja käyttäjän hyväksymistestaus eli englanniksi user acceptance testing (ISTQB® testaussanasto 2012). Käyttäjän hyväksymistestaus tunnetaan lyhenteellä UAT. Tässä opinnäytetyössä termillä ”hyväksymistestaus” tarkoitetaan nimenomaan käyttäjän hyväksymistestausta eli UAT-testausta.

Käyttäjän hyväksymistestauksen tarkoituksena on varmistaa, soveltuuko ohjelmisto käyttöön juuri ohjelmiston käyttäjien näkökulmasta. Tavoitteena on varmistaa, että ohjelmisto on sellainen mitä on pyydetty vaatimuksissa ja että ohjelmisto vastaa sidosryhmän tarpeita. (ISTQB® Perustason

sertifikaattisisältö 2018, 32.) Hyväksymistestaus tuo aina uutta näkökulmaa testaukseen ja sen aikana voidaan huomata uudenlaisia virheitä tai puutteita, kun testaus tapahtuu loppukäyttäjän näkökulma huomioon ottaen.

4.2 Hyväksymistestauksen hyödyt

Kuten aiemmin on jo mainittu järjestelmätestauksella ja hyväksymistestauksella on eri tavoitteet. Koska hyväksymistestauksen tavoitteet ovat erilaisia, hyväksymistestauksen hyödyt eroavat myös järjestelmätestauksesta. Mahdolliset ohjelmiston virheet voidaan saada vielä kiinni hyväksymistestauksessa. Tämä minimoi kuluja, koska virheet huomataan ja voidaan korjata jo ennen tuotantoon vientiä. (ISTQB® Perustason sertifikaattisisältö 2018, 32.)

Hyväksymistestauksella on näin suuri vaikutus asiakastyytyväisyyteen. Hyväksymistestaus vaikuttaa asiakkaan näkemykseen positiivisesti luottamuksen kohottamisen kautta (ISTQB® Perustason sertifikaattisisältö 2018, 32). Luottamuksen kautta asiakastyytyväisyys ja varmuus ohjelmistoa kohtaan kasvavat huomattavasti, eikä tuotantoon vientiin liity niin paljon epävarmuutta.

Hyväksymistestauksen mentyä läpi, ohjelmiston katsotaan olevan oikeanlainen, laadukas ja täyttää kaikki halutut vaatimukset ja tarpeet (Vuori 2010, 5). Hyväksymistestauksen kautta saadaan myös varmuus, että kaikki tuotteen osat ja toiminnallisuudet ovat haluttujen lakien ja määräysten mukaisia. Näin varmistetaan, ettei yrityksille aiheudu julkaisun jälkeen juridista vastuuta. (What Is User Acceptance Testing? (With Benefits and How-to) 2022.) Hyväksymistestaus vaikuttaa niin asiakkaan kuin toimittavan osapuolen tyytyväisyyteen. Kun toteutus menee läpi hyväksymistestauksesta, myös ohjelmiston toteuttaja saa varmistuksen siitä, että asiakas on tyytyväinen lopputuloksesta ja antanut hyväksynnän tehdylle työlle. (Vuori 2010, 5.) Tämä antaa toimittajalle itsevarmuutta ja luottoa omaan tekemiseen.

4.3 Hyväksymistestauksen haasteet

Huonosti suunniteltuna ja hallittuna hyväksymistestaus voi muodostua ongelmalliseksi kehitysprosessin vaiheeksi. Pahimmassa tilanteessa haasteiden huomioimatta jättäminen voi olla vaaraksi koko prosessin onnistumiselle. Projektien aikataulut tuottavat usein haasteita testauksen eri osa-alueille. Usein projekteissa joudutaan taipumaan aikataulupaineen vuoksi.

Hyväksymistestaus on usein prosessissa se vaihe, jonka tulee joustaa sovittujen aikataulujen vuoksi (Rinne 2016, 54). Valtava kiire ja se, ettei työmäärään osata varautua, ovat suuria kompastuskiviä hyväksymistestauksessa. Myös haastava kokonaisuus ja jatkuvasti muuttuvat vaatimukset voivat tehdä hyväksymistestauksen suunnittelusta hankalaa. Muuttuvien tilanteiden vuoksi voi olla, ettei hyväksymistestauksen aikana tiedetä mikä toiminto pitää vielä paikkaansa ja mikä on muuttunut matkan varrella. Hyväksymistestauksessa kokonaisuus voi olla myös todella hankala sisäistää ja ymmärtää. (Vuori 2010, 15.) Kommunikoinnilla ja tiedon kulkeutumisella on siis suuri vaikutus hyväksymistestauksen onnistumiselle.

Jotta hyväksymistestaus onnistuisi, tulee oleelliseksi muodostaa oikeanlaisia testitapauksia. Hyväksymistestausvaiheessa testitapaukset tulisi kuvastaa reaali maailman käyttötapauksia (Magalhães 2015). On varmistettava myös, että testitapaus on tarpeeksi kattava (UAT testaus, eli hyväksymistestaus: Mitä se tarkoittaa ja miksi se on tärkeää 2023). Virheellinen hyväksymistestauksen testisuunnittelu voi johtaa pahimmissa tapauksissa päällekkäisyyksiin järjestelmätestauksen ja hyväksymistestauksen kesken (Magalhães 2015).

Kivi (2020, 39) kertoo, miten Axelos (2019) kuvaa teoksessaan, kuinka integraatio ja muut erinäiset liittymät eri komponenttien välillä on ymmärrettävä hyväksymistestausvaiheessa. Ohjelmiston valmius hyväksymistestausta varten on myös yksi suuri ongelma. On siis varmistettava, onko ohjelma todella valmis hyväksymistestausta varten. On myös selvitettävä, onko aiemmat testausasot jo toteutettu ja todennettu läpi menneiksi. (UAT testaus, eli hyväksymistestaus: Mitä se tarkoittaa ja miksi se on tärkeää 2023.)

5 Ohjelmistoprosessi

5.1 Ohjelmistoprosessin määritelmä

Paakki (2014, 4) määrittelee termin prosessin siten, että prosessi kuvaa, kuinka jokin asia tulisi toteuttaa. Projekti on eräänlainen prosessin ilmentymä ja projektissa seurataan useampaa tai yhtä prosessia (Paakki 2014, 4). Ohjelmistoprosessi tarkoittaa eri vaiheita, joiden avulla rakennetaan toimiva ohjelmisto vaihe vaiheelta. Ohjelmistoprosessin vaiheita ovat esimerkiksi määrittely ja suunnittelu, ohjelmointi, testaus, käyttö sekä ylläpito. (Laine & Paakki n.d, 1.)

Ohjelmistoprosessi vaikuttaa sekä ohjelmiston laatuun että kustannuksiin. Systemaattisuus eli prosessin selkeys näkyy kustannuksissa ja prosessin lopputuloksen laadussa. Prosessi on oltava sopiva tietyn tuotteiden tekoon, jotta saadaan mahdollisimman hyvä lopputuote aikaiseksi. (Paakki 2014, 4.) Mikäli prosessi ei ole oikeanlainen tai selkeä, on suuri riski, että lopputuotteen laatu kärsii (Paakki 2014, 6). Prosessia tulisi jatkuvasti hioa tarpeen tullen, vaikka kesken projektin, mikäli havaitaan, että kyseinen projekti tarvitsee toisenlaisen prosessin onnistuakseen.

5.2 Elinkaarimallit ohjelmistokehityksessä

5.2.1 Elinkaarimallit

Ohjelmistokehityksen elinkaarimalleja on useita erilaisia. Elinkaarimallilla on suuri vaikutus testaukseen. Ohjelmistokehityksessä elinkaarimalli määrittelee sen, miten eri tehtävät toteutetaan ajallisesti suhteessa toisiinsa (ISTQB® Perustason sertifikaattisisältö 2018, 25). ISTQB® Perustason sertifikaattisisältö (2018, 25) jakaa ohjelmistokehityksen elinkaarimallit peräkkäis-, iteratiivisiin- ja inkrementaalisiin kehitysmalleihin.

Peräkkäismalleista esimerkkejä ovat vesiputousmalli ja V-malli. Vesiputousmallissa eri työtehtävät seuraavat toisiaan peräkkäin ja toinen tehtävä alkaa vasta kun edeltävä tehtävä on päättynyt. Esimerkiksi määrittelyn valmistumisen jälkeen aloitetaan ohjelmointi, ohjelmoinnin jälkeen aloitetaan järjestelmätestaus ja niin edelleen. V-mallissa testaus otetaan mukaan jo prosessin alku pisteestä lähtien. V-mallissa jokaisen toteutusvaiheen vastinparina on eri testaustaso, ja testaustasot voivat toteutua joko perätysten tai osittain myös päällekkäin. (ISTQB® Perustason sertifikaattisisältö 2018, 25.) Mäkinen (2020, 4) kertoo diplomityössään, kuinka vastinpareja voi olla esimerkiksi toteutus ja yksikkötestaus, suunnittelu ja integraatiotestaus, määrittely ja järjestelmätestaus, sekä viimeisenä vaatimukset ja hyväksymistestaus.

Iteratiivisissa kehitysmallissa eri vaiheet toteutuvat tietyn mittaisen vaiheen eli iteraation aikana. Iteraation aikana syntyy toimiva ohjelmisto. Seuraavissa iteraatioissa voidaan lisätä tai muokata ominaisuuksia ohjelmaan. Iteraatiot voivat olla varsin lyhyitä ja muutokset vastaavasti hyvinkin pieniä. Inkrementaalisisissa kehitysmallissa ohjelmisto rakentuu pala kerrallaan, eli eri vaiheet, kuten suunnittelu, testaus ja toteutus tehdään pienissä osissa. (ISTQB® Perustason

sertifikaattisäiltö 2018, 25.) Sekä iteratiivisia ja inkrementaalisia kehitysmalleja voidaan hyödyntää ketterässä kehittämisessä (Lehtonen 2015).

5.2.2 Ketterä kehittäminen

Vesiputousmallin vastakohtana on ketterä kehitys. Toiselta nimeltä ketterä kehitys on ketterä viitekehys eli englanniksi agile development. Ketterän kehityksen ominaispiirteitä on, että tätä viitekehystä käyttävät menetelmät ovat sekä inkrementaalisia että iteratiivisia (Lehtonen 2015). Ketterässä kehityksessä kehitys tapahtuu sprinteissä eli sovitun mittaisessa aika jaksossa, jonka aikana ennalta määrätyt tehtävät toteutetaan (Ekholm & Nylén 2021).

Muita ketterän kehityksen tyypillisiä piirteitä ovat esimerkiksi tiivis yhteistyö asiakkaan kanssa, kommunikoinnin tärkeyden korostaminen, niin tiimissä kuin sidosryhmien kanssa. Ketterässä kehityksessä ei pidättäydytä alkuperäisissä suunnitelmissa, vaan mikäli jokin asia ei toimi, suunnitelmia kehitetään parempaan suuntaan lennosta. (Lehtonen 2015.) Ketterässä kehityksessä on tyypillistä, että koodi on jatkuvan muutoksen alla. Tämä aiheuttaa sen, että regressiotestaus on ratkaisevan tärkeää ketterässä viitekehityksessä. (ISTQB® Perustason sertifikaattisäiltö 2018, 36.)

Regressiotestaus on muutokseen pohjautuvaa testausta. Mikäli koodin tiettyä osaa muokataan, se voi aiheuttaa tahattoman muutoksen ohjelmiston toiminnallisuuteen toisaalla. Usein taantuma eli regressio tapahtuu ajan kuluessa huomaamattomasti. Testiautomaatiota käytetään paljon regression havaitsemiseksi, sillä testiautomaatiossa kone ajaa testitapauksia ja näin ihminen ei joudu jatkuvasti toistamaan samoja testejä manuaalisesti. (ISTQB® Perustason sertifikaattisäiltö 2018, 36.) Ketterän viitekehityksen toimintaperiaatteita voidaan hyödyntää suurissakin monitoimittajaprojekteissa.

5.2.3 Scaled agile framework

Scaled agile framework eli lyhennettynä SFAFe tarkoittaa ketterän kehityksen menetelmiä skaalattuna isoille projekteille ja se on usein käytössä sovelluskehityksessä (SFAFe tulee ja pelastaa 2018). CGI Inc luettelee blogitekstissä SFAFe tulee ja pelastaa (2018), että SFAFe:n ydinarvoja ovat muun muassa: "sisäänrakennettu laatu, tulosten aikaansaaminen, täysi läpinäkyvyys ja tekemisen linjaaminen liiketoiminnan tavoitteisiin". Scaled agile framework on skaalautuva ja muunneltavissa

organisaation toimialasta, koosta tai monimutkaisuudesta riippumatta. SAFe pitää sisällään periaatteita ja käytäntöjä ketteristä menetelmistä, DevOps - sekä Lean toimintamalleista. Menetelmien avulla vastataan suurten ohjelmistojen ja järjestelmien toimittamisen ja koordinoimisen haasteisiin. (What Is SAFe® n.d.) Peltolan (2020, 8) mukaan Itewiki (2020) määrittelee, että Lean tarkoittaa ajattelutapaa ja johtamiskäytäntöjä, jonka keskiössä on asiakaskeskeisyys, nopeus, laatu ja olennaiseen keskittyminen. DevOps tarkoittaa toiminta- ja ajattelutapaa, jossa korostetaan ihmisten yhteistyötä, automaatiota ja integraatiota (Peltola 2020, 8). SAFe menetelmää voidaan hyödyntää monitoimittajaprojekteissa.

Scaled agile frameworkin hyötyjä ovat muun muassa se, että SAFe:n ansioista tuotteet valmistuvat ajallaan, työntekijät ovat tyytyväisempiä ja sitoutuneempia, tiimien ja asiakkaan välinen kommunikointi on tehokkaampaa, tuottavuus on suurempaa ja tuotteen laatu ja lopputulos on parempi. Nokia Software on yksi monista esimerkki firmoista, jotka ovat ottaneet SAFe menetelmän käyttöön. (Benefits of SAFe® n.d.)

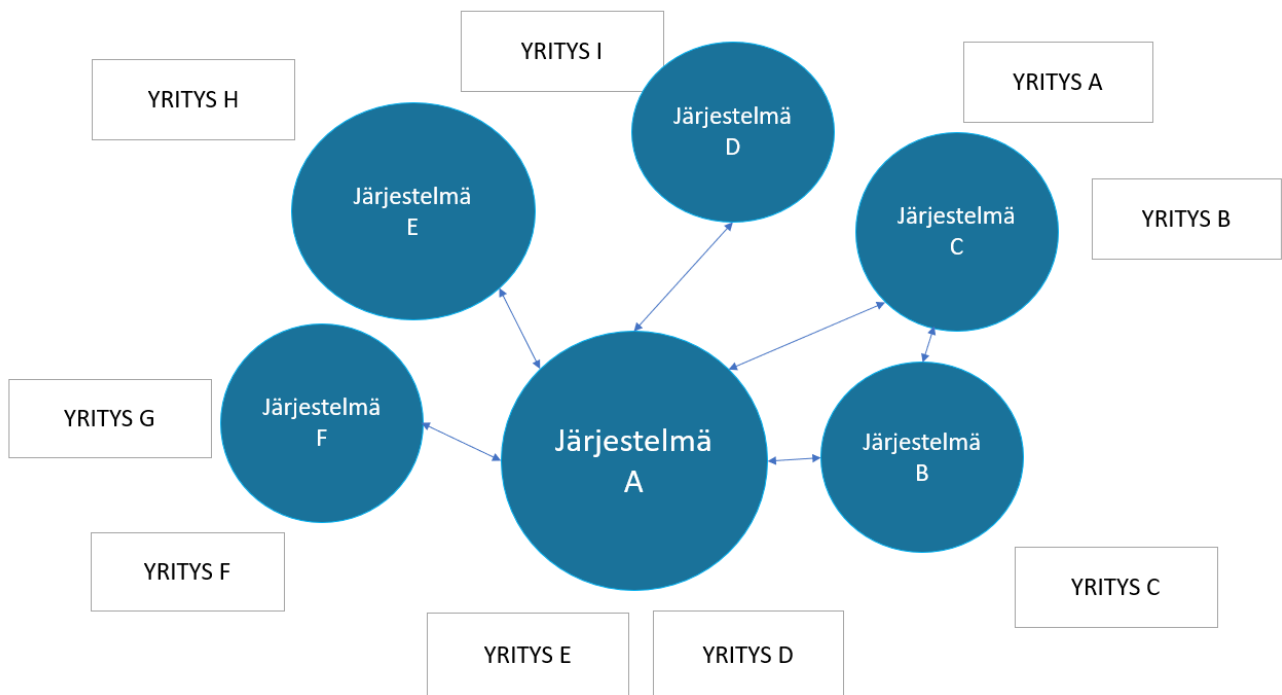
6 Monitoimittajaympäristö

6.1 Monitoimittajaympäristö termin määritelmä

Ohjelmistot syntyvät yhä useammin monitoimittajaympäristöissä. Monitoimittajaympäristö käsitteellä tarkoitetaan nimensä mukaisesti sitä, että yritykset tilaavat ohjelmiston monen eri palveluntarjoajan kautta, minkä myötä ohjelmistot rakentuvat monitoimittajaympäristöissä (Heikkilä 2014, 6). Heikkilä (2014, 14) määrittelee, että monitoimittajaympäristön ominaisuus on muun muassa se, että asiakas voi tilata kokonaisen palvelun tai vain palvelun osia eri toimittajilta. Heikkilän (2014, 14) mukaan Bapna, Barua, Mani ja Mehra (2010) määrittelee monitoimittajaympäristön puolestaan siten, että monitoimittajaympäristöiden ominaisuus on se, että niissä toimittajat ovat riippuvaisia toisistaan ja toistensa rakentamista kokonaisuuksista.

Monitoimittajaympäristössä eri yritykset ovat vastuussa yksin omasta järjestelmästä tai useampi yritys voi yhteistyössä muodostaa järjestelmän yhdessä (ks. kuvio 2). Kuviossa 2 on havainnollistettu miltä voisi näyttää monitoimittajaympäristökokonaisuus suuremman mittakaavan projektissa. Kuviossa voi havainnoida, kuinka eri toimittajien järjestelmät linkittyvät toisiinsa muodostaen järjestelmäkokonaisuuden. Koska monitoimittajaympäristössä eri

järjestelmät ovat kytköksissä toisiinsa, jokainen muutos tulee miettiä niin, että muutoksen jälkeenkin järjestelmäkokonaisuus toimii edelleen suunnitellusti.



Kuvio 2. Monitoimittajaympäristön mallinnus (Grönroos n.d, muokattu)

6.2 Monitoimittajaympäristön mahdollisuudet

Monitoimittajaympäristö tuottaa paljon merkittäviä etuja asiakkaalle. Kivi (2014, 30) kuvaa, kuinka asiakas voi tilata tarvitsemansa ohjelmiston osan tai ohjelmiston toimittajalta, joka on kyseisen osa-alueen asiantuntija. Myös Tsarykau (2021) korostaa puolestaan kirjoituksessaan, kuinka eri toimittajan käytöllä mahdollistetaan se, että lopputuloksen eri palaset ovat tuotettu parhaalla mahdollisella osaamisella. Monitoimittajaympäristön etuna on myös se, että asiakas ei ole riippuvainen vain yhdestä palveluntarjoajasta (Tsarykau 2021).

Koska projektin parissa työskentelee useampi yritys ja tiimi, lopputuote on myös valmis aikaisemmassa vaiheessa, sillä useat eri firmat tekevät toteutuksia samaan aikaan. Mikäli käytössä olisi vain yksi tai kaksi firmaa, aikaa projektin valmistumiselle menisi huomattavasti kauemmin vähäisempien resurssien takia.

6.3 Monitoimittajaympäristön haasteet

Vaikka monitoimittajaympäristössä on huomattavia positiivisia vaikutuksia palvelun tilaajalle, on sen onnistumisessa paljon haasteita. Monitoimittajaympäristössä järjestelmä kokonaisuus voi muodostua eri palveluntarjoajista, sopimuksista ja teknologioista (Kivi 2020, 26). Eri teknologioista huolimatta eri järjestelmien ja järjestelmän osien tulisivat olla yhteensopivia ja toimia saumattomasti yhdessä. Arkkitehtuurin hallinnalla on suuri rooli monitoimittajaympäristössä. Monitoimittajaympäristössä kokonaisuuden hallinnan tulee olla kuvattu tarkasti ja selkeästi. (Kivi 2020, 26.)

Tsarykau (2021) kuvaa, kuinka monitoimittajaympäristö tuo mukanaan myös tietoturvariskit. Monitoimittajaympäristössä työskennellään useiden eri yritysten kanssa, minkä vuoksi myös mahdollisten tietoturvaongelmien ja tärkeiden tietovuotojen riskien uhka voi kasvaa. Erilaisin sopimuksin ja varatoimenpiteiden avulla tietoturvariskit kuitenkin voidaan minimoida. (Tsarykau 2021.)

Kiven (2020, 31) mukaan Axelos (2019) kuvaa teoksessaan, kuinka monitoimittajaympäristöissä on todella olennaista rajata ja määritellä jokaisen eri toimittajan rooli projektissa. Eli täytyy määritellä tarkasti mitä jokainen tekee ja mistä he ovat vastuussa. On ymmärrettävää, että jokaisen toimittajan tulee tietää mitä itse pitää tehdä ja miten oma tuote vaikuttaa muiden toimittajien tekemiin kokonaisuuksiin. Kiven (2020, 31) mukaan Axelos (2019) kertoo, kuinka eri palveluntoimittajat ja heidän halunsa ja valmiudet yhteistyöhön on aina otettava huomioon ennen kuin aletaan toimia monitoimittajaympäristössä.

7 Tutkimusosio

7.1 Kyselyhaastattelu

Tutkimuksessa suoritettiin kyselyhaastattelu osana opinnäytetyön tutkimusta (ks. liite 1). Liitteessä 1 on näkyvissä kyselyhaastattelun kysymykset. Kyselyhaastattelu toteutettiin Google Forms –ohjelmiston avulla. Kyselyhaastattelu suoritettiin helmikuussa 2024 ja kysely lähetettiin neljälle Tietoevryn asiantuntijalle, jotka ovat työskennelleet monitoimittajaprojekteissa ja hyväksymistestauksen parissa. Kyselyyn vastanneet henkilöt ovat toimineet erilaisissa

testausrooleissa vuosien varrella. Esimerkkeinä näistä rooleista mainittiin testauspäällikkö, hyväksymistestauksen koordinaattori ja toimittajan testausvastaava. Kyselyyn vastanneet ilmoittivat osallistuneensa projekteihin tai hankkeisiin muutamasta vuodesta jopa kymmeneen vuoteen. Puolet kyselyyn vastanneista kertoivat, että he ovat olleet monitoimittajaprojekteissa, joissa toimittajia on ollut yhteensä 3–4 kappaletta ja puolet, joissa toimittajia on ollut 5–7 kappaletta.

Kyselyyn vastattiin täysin anonyymisti. Anonyymisyys varmistettiin niin, että kyselyyn vastanneiden ei tarvinnut antaa mitään henkilökohtaisia tietoja vastaus vaiheessa. Vastaajia kehoitettiin myös antamaan vastaukset niin, etteivät ne sisällä yksityiskohtia esimerkiksi asiakkaasta, projektista tai hankkeesta. Vastaajia pyydettiin antamaan vastauksia ei pelkästään yhdestä tietyistä projektista, vaan heitä kehoitettiin kuvailemaan yleisesti omia kokemuksiaan aiheeseen liittyen. Vastaukset esitetään opinnäytetyössä yleisellä tasolla, eikä kyselyyn vastanneisiin henkilöihin tulla viittaamaan edes heidän työnimikkeillään.

Seuraava osio pitää sisällään kyselyn vastaajien ajatuksia, huomioita ja parannusehdotuksia aiheeseen liittyen. Kyselyn tulosten avulla on pyritty syventymään aiheeseen ja selvittämään asiantuntijoiden kokemusten kautta, miten hyväksymistestauksesta saataisiin mahdollisimman onnistunut kokonaisuus monitoimittajaympäristössä. On huomioitava, että kyselyn vastausten rinnalla tuon esiin myös oman kokemukseni aiheesta.

7.2 Epäonnistuneen hyväksymistestauksen vaikutukset

Jotta saavutettaisiin onnistunut hyväksymistestaus, on käytävä läpi mitä vaikutuksia epäonnistuneella hyväksymistestauksella on ja minkä vuoksi onnistunut hyväksymistestaus on tärkeää saavuttaa. Kyselyyn vastanneet listasivat, että epäonnistuneen hyväksymistestauksen vaikutuksia ovat muun muassa tuotteen laadun heikkous, bugien pääseminen tuotantoon ja se, kuinka tuote tai palvelu ei vastaa liiketoiminnan tarpeita.

Kyselyn tuloksissa ilmeni, että virheiden pääseminen tuotantovaiheeseen aiheuttaa merkittävää kuormitusta monitoimittajaprojektin eri toimittajille. Bugien korjaukset, korjausten uudelleentestaus ja hyväksymistestauksen viivästyminen voivat aiheuttaa suuriakin lisäkustannuksia. Monitoimittajaympäristössä kaikki komponentit ja järjestelmät pitää toimia

yhdessä mutkattomasti. Jos hyväksymistestaus on jäänyt puutteelliseksi, voi olla, että järjestelmät tai komponentit ovat yhteensopimattomia keskenään ja tämä vaikuttaa koko järjestelmän toimintaan. Monitoimittajaympäristössä virheiden juurisyiden löytäminen on myös aikaa vievää. Useat eri toimittajat voivat joutua käyttämään aikaa virheiden löytämiseen, mikä kuluttaa vielä enemmän resursseja ja luo työkuormaa eri toimittajille. Virheiden selvittäminen vie huomiota pois muista tärkeistä tehtävistä.

Kyselyn tuloksissa tuli myös esille, että hyväksymistestauksen epäonnistuminen voi aiheuttaa myös tunneperäisiä haittavaikutuksia. Niitä on muun muassa epäonnistumisen tunne toimittajalla, sillä toimittajan voidaan nähdä epäonnistuneen tehtävässään. Lisäksi haittavaikutuksia voi olla esimerkiksi yleinen motivaation laskeminen palvelua ja tuotetta kohtaan. Asiakas voi kokea myös pettymystä ja turhautumista niin toimittajaa kuin palvelua kohtaan. Jos vika on ajautunut tuotantoon saakka, voi palvelun laatu kärsiä pahasti. Monitoimittajaympäristöissä vika voi haitata koko palvelun toimintaa, jos vika piilee esimerkiksi integraatiossa. Pahimmassa tapauksessa epäonnistunut hyväksymistestaus voi aiheuttaa maineen menetyksen ja luottamuksen puutteen toimittavaa organisaatiota kohtaan.

Kyselyn vastauksissa tuli selkeästi esille se, kuinka hyväksymistestauksen epäonnistuminen ei ole koskaan yksittäisten toimijoiden syytä. Toisin sanoen hyväksymistestauksen epäonnistuminen ei ole yksittäisten työntekijöiden, kuten esimerkiksi testaajien vastuulla.

7.3 Monitoimittajaympäristön tuomat haasteet hyväksymistestaukseen

Monitoimittajaympäristö tuo mukanaan lisää haasteita niin hyväksymistestaukseen kuin muihinkin ohjelmistoprosessin vaiheisiin. Kyselyn vastauksissa nostettiin esille, kuinka testauksen eteneminen ja valmistuminen on huomattavasti hitaampaa monitoimittajaympäristössä. Testattavien kohteiden määrä voi olla huomattavasti suurempi, mitä yhden toimittajan projekteissa, mikä vuorostaan tekee testauksesta usein isotöistä. Lisäksi hyväksymistestaukselle on varattava paljon aikaa. Tämän lisäksi testien suunnittelu voi olla hidasta monitoimittajaympäristössä, sillä kokonaisuudet voivat olla hyvinkin laajoja ja käyttötapaukset sekä määrittelyt voivat valmistua hitaasti.

Myös käyttöoikeuksien saaminen laajalle joukolle voi olla hankalaa, mikä hidastaa esimerkiksi sopivan testausympäristön rakentamista. Hybridiympäristön luominen, jossa jokaisen eri toimittajan palvelut ovat käytettävissä, voi olla haastavaa rakentaa ja ylläpitää. Eri toimittajat voivat toimittaa omia muutoksia eri aikoihin, minkä vuoksi useamman testausympäristön käyttö on välttämätöntä. Mikäli hyväksymistestauksen ympäristöjä on useampi, niiden ylläpitäminen ajan tasalla voi myös olla haastavaa. Usean ympäristön käyttö voi johtaa siihen haasteeseen, että turhista virheistä raportoidaan, mikäli virheet johtuvat ympäristöjen eroavaisuuksista.

Kyselyn vastauksissa kommunikointi nostettiin usein yhdeksi onnistumisen avaintekijöistä. Hyväksymistestausvaiheessa tulisi hyödyntää eri toimittajien osaamista ja tarpeen tullen myös osallistuttaa eri toimittajia hyväksymistestauksen suunnitteluun. Toisaalta liiallinen kommunikointi voi tuoda myös omat haasteensa. Hyväksymistestauksen tulee olla erilainen järjestelmätestauksen kanssa. Hyväksymistestausvaiheessa ei saa, eikä voida hyödyntää suoraan järjestelmätestauksen testitapauksia. Näin ei voi toimia edes aikataulupaineen vuoksi. Hyväksymistestauksen tulisi hyödyntää olemassa olevia käyttötapauksia testien suunnittelussa ja tuoda oma näkökulmansa testaukseen.

Kyselyn vastauksissa nostettiin esille, että monen eri toimittajan mukana olemisen ei pitäisi olla huomattavasti haasteellisempaa kuin vain yhden toimittajan. Vika piilee projektin hallinnassa. Mikäli projekti ja prosessi on hyvin suunniteltu ja hallittu, monitoimittajaympäristö ei tuo hyväksymistestaukseen sen suurempia haasteita. On siis kiinnitettävä huomiota mitä muutoksia prosessi ja toimintatavat vaativat, jotta monitoimittajaympäristössä hyväksymistestaus sujuisi yhtä jouhevasti kuin yhden toimittajan projekteissa.

7.4 Ohjelmistoprosessin kehitysehdotukset

Prosessin eli kehityskulun lopputuloksena on asiakkaan kanssa yhdessä määritelty kokonaisuus. Monitoimittajaprojekteissa ohjelmistoprosessi tulisi aina suunnitella ja tehdä monitoimittajaympäristöä huomioonottavaksi. Monitoimittajaympäristön tuomat mahdolliset haasteet ja erityispiirteet tulisi aina ottaa huomioon ja kehittää prosessia ympäristöön sopivaksi.

7.4.1 Hyväksymistestaus osaksi prosessia

Kyselyhaastattelun vastauksissa nostettiin useasti esille se, kuinka hyväksymistestaus tulisi tuoda osaksi prosessia. Ideaali tilanteessa hyväksymistestauksen suunnittelu ja testitapausten teko aloitettaisiin jo projektin alkuvaiheessa. Näin hyväksymistestauksesta ei muodostuisi vain yksi iso rupeama projektin päätteeksi. Hyväksymistestaaajat tai hyväksymistestauksen koordinoijat olisi hyvä ottaa mukaan mahdollisimman aikaisessa vaiheessa projektia. Näin hyväksymistestausvaiheessa työskenteleville muodostuisi mahdollisimman selkeä kuva testattavasta kohteesta. Hyväksymistestaus saisi laajemman käsityksen eri toimittajien tuottamista osista ja kuinka osat tulisi toimia keskenään. Hyväksymistestausvaiheesta vapautettaisiin näin aikaa ja hyväksymistestauksen aikana voitaisiin keskittyä itse testaamiseen. Mikäli hyväksymistestaus olisi mukana jo aikaisessa vaiheessa projektia, hyväksymistestauksen edustajilta voisi tulla arvokkaita mielipiteitä eri asioista. Näin voitaisiin jo projektin alkuvaiheessa tehdä korjausliikkeitä, mikäli huomataan ettei joku asia tule toimimaan käytännössä.

7.4.2 Hyväksymistestauksen hallinta

Kyselyssä ilmeni, miten hyväksymistestauksen hallinnan katsottiin olevan yksi oleellisimpia asioita monitoimittajaympäristöissä. Kyselyn vastauksissa tuotiin esiin ehdotus siitä, kuinka monitoimittajaympäristöissä voitaisiin ottaa mukaan koordinoijia, joilla olisi ymmärrys eri toimittajien toteutuksista. Koordinoijat olisivat täten useiden eri toimittajien tuottamien palveluiden ja suurempien kokonaisuuksien asiantuntijoita. Laajan tietämyksen avulla koordinoijat voisivat avustaa muodostamaan oikeanlaisia testitapauksia hyväksymistestausta varten. Näin varmistettaisiin, että testitapauksilla testataan juuri oikeaa asiaa hyväksymistestauksen näkökulmasta. Koordinoijilla olisi tärkeää olla myös hyvät kommunikointitaidot. He voisivat olla tiedon välittäjinä eri toimittajaorganisaatioiden välillä.

Kyselyn vastauksessa täsmennettiin myös hyväksymistestauksen etenemisen seurannan tärkeyttä ja sen lopputulosten raportointia. Jotta hyväksymistestausten löydöksiin tartuttaisiin mahdollisimman nopeasti, on hyväksymistestauksen seuranta ja raportointi oltava kunnossa. Hyväksymistestauksen aikana tehdyt virrehavainnot ovat välttämätöntä saada hyvin nopeasti selvitykseen ja mahdollisesti korjattavaksi oikealle toimittajalle. Sidosryhmiä on myös hyvä pitää ajan tasalla kaikesta, mitä hyväksymistestauksessa tapahtuu. Kyselyn vastauksessa kuvattiin sitä,

miten hyväksymistestausta ajatellen on aina oltava varasuunnitelmia odottamattomien tilanteiden varalle. Näin aikataulussa pysyminen on todennäköisempää, kun on osattu etukäteen varautua muuttuvia tilanteita varten.

Yhtenä ideana kysely vastauksissa oli, että hyväksymistestaukselle olisi myös hyvä ennakkoon määritellyt testaussyklit. Hyväksymistestaukselle kerrotaisiin hyvissä ajoin mikä ja millainen kokonaisuus on tulossa seuraavaksi testattavaksi. Hyväksymistestaukselle ilmoitettaisiin myös tarkasti millä aikataululla kokonaisuus tulisi olla testattuna. Näin hyväksymistestaukselle jäisi tarpeeksi aikaa suunnitella testausta ja saada laaja ymmärrys eri toimittajien toimittamasta kokonaisuudesta jo ennen varsinaisen testauksen alkamista.

Kyselyn vastauksissa nostettiin esille, että yksi hyväksymistestauksen haasteista monitoimittajaympäristössä on se, ettei ole yhteistä näkemystä siitä mitä halutaan hyväksymistestaukselta. On siis tärkeää, että jokaisella on yhtenäinen näkemys siitä, mikä on hyväksymistestauksen rooli ja kuinka hyväksymistestaus tullaan toteuttamaan. On mietittävä etukäteen kuka hoitaa hyväksymistestauksen, eli onko hyväksymistestauksen tekijä loppukäyttäjä, asiakkaan asiantuntija vai ammatti testaaaja.

7.4.3 Kouluttaminen ja tuki

Jotta hyväksymistestauksesta saataisiin mahdollisimman onnistunut, hyväksymistestauksesta vastaavan osapuolen on tiedettävä miten ja mitä he testaavat. Kyselyn vastanneet kuvailivat, kuinka hyväksymistestausta varten pitää tarpeen tullen varata aikaa testaajien kouluttamiseen. Koska hyväksymistestauksen voi tehdä niin loppukäyttäjä, asiakkaan asiantuntija tai ammattitestaaja, kouluttaminen muodostuu entistä tärkeämmäksi. Kuka tahansa hyväksymistestauksen toteuttaa, on tiedettävä testattavasta järjestelmästä ja testattavasta kohteesta. Kouluttaminen on keskeistä, sillä usein monitoimittajaympäristön järjestelmät voivat olla monimutkaisia ja ne voivat sisältää paljon haasteellisia toiminnallisuuksia.

Kyselyn vastauksissa kuvattiin, kuinka tuki on myös ensiarvoisen tärkeää hyväksymistestauksen aikana. Monitoimittajaympäristöt koostuvat useista eri tiimeistä ja täten myös useista eri alan asiantuntijoista. Jokaista toimittajaa tulisi hyödyntää ja hyväksymistestauksen tulisi pyytää tukea hyväksymistestausta varten. Tukea voi olla esimerkiksi tekninen tuki testauksen suunnittelun tai

testauksen aikana. Tuki voi olla myös esimerkiksi tukea testausympäristön ongelmatilanteiden kanssa. Tuen ja koulutuksen avulla hyväksymistestaaajat eivät koe työtänsä kuormittavaksi ja saavat varmuuden tekemästään työstä.

7.4.4 Aikataulutus ja resurssit

Kyselyssä vastanneet olivat yhtä mieltä siitä, kuinka aikataulutus ja resurssit ovat yksi avaintekijä onnistuneelle hyväksymistestaukselle monitoimittajaympäristössä. Koska mukana on useampia toimittajia, aikataulut voivat muuttua hyvinkin nopeasti ja odottamatta. Hyväksymistestauksen suunnittelussa on varauduttava aikataulumuutoksiin ja mukauduttava niihin.

Hyväksymistestaukselle, testauksen koordinoinnille ja jokaiselle projektin prosessin vaiheelle on myös jätettävä hyvin aikaa ja resursseja. Liian kiireinen ja epärealistinen aikataulu voi vaikuttaa palvelun laatuun ja projektin onnistumiseen. Jos määrittely esimerkiksi viivästyy, eikä tuotantoon vientiä voida siirtää syystä tai toisesta eteenpäin, hyväksymistestaus ja testauksen muut vaiheet tulevat kärsimään. Jos testaukselle ei jää tarpeeksi aikaa mahdolliset virheet voivat jäädä huomaamatta ja ohjelmiston virheet voivat edetä tuotantoon asti.

Kyselyn vastauksissa kuvattiin, kuinka aikataulujen luominen on tärkeä osa projektia jokaiselle toimittajalle. Aikataulujen avulla varmistetaan, että jokainen toimittaja ja tiimi edistyy töissään tehokkaasti. Monitoimittajaympäristössä jokaisen toimittajan on sitouduttava sovittuihin aikatauluihin, jotta jatkuvia aikataulumuutoksia ei tulisi. Eri toimittajien palveluiden valmistuksen viivästyminen voi olla kohtalokasta hyväksymistestaukselle. Mikäli yksi toimittaja ei pysy aikataulussa, voi olla, ettei järjestelmää voida kokonaisvaltaisesti testata. Kun aikataulutus on suunniteltu hyvin ja tarkasti, on eri toimittajien helppo kommunikoida toistensa kanssa projektin etenemisestä ja mahdollisista kipukohdista.

Kyselyyn vastanneet painottivat, että hyväksymistestauksessa on tärkeä tehdä priorisointia aikataulutuksen ohessa. Priorisoinnilla varmistetaan, että oleellimmat testauskohteet tulevat testattua ensimmäisten joukossa ja näin saadaan varmistettua, että kriittisimmät toiminnallisuudet toimivat ennen tuotantoon vientiä. Kun aikataulu on selkeä, mutta ei liian tiukka, saadaan paras mahdollinen lopputulos aikaiseksi. Oikealla aikataulun ja resurssin suunnittelulla varmistetaan, että hyväksymistestaus onnistuu ja suoriutuu kiitettävästi niin, ettei kukaan toimija uuvu matkan varrella. Eri toimittajien ja asiantuntijoiden motivaatiota saadaan

myös pidettyä yllä, kun aikataulutusta ei ole tehty mahdottoman tiukaksi jo prosessin suunnitteluvaiheessa.

Asiantuntijat täsmensivät kyselyn vastauksissa, miten olennaista on monitoimittajaympäristöissä viedä tuotantoon kokonaisuuksia mahdollisimman sopivissa ja hallittavissa osakokonaisuuksissa. Näin hyväksymistestaukselle ei tule yhdellä kertaa valtavaa testauskuormaa, vaan testattava kohde on hallittavissa mittasuhteissa ja näin aikataulussa pysyminen ja projektin onnistuminen on realistisempaa. Kun aikataulutusta ja resursseja suunnitellaan, tulee ottaa aina huomioon, onko kyseessä täysin uusi palvelu, jatkokehitys olemassa olevaan järjestelmään vai ylläpidetäänkö vanhaa järjestelmää. Prosessissa ja esimerkiksi testauksen suunnittelussa nämä kaikki tekijät on otettava huomioon.

7.4.5 Palaute ja jatkuva kehitys

Kyselyssä vastauksissa ilmeni, kuinka tärkeänä asiantuntijat pitivät palautteen keräämisen tärkeyttä. Jopa vuosien kestävässä monitoimittajaprojekteissa, on oleellista aika-ajoin kerätä palautetta eri toimittajilta. On ratkaisevan tärkeää keskittyä palautteeseen ja ottaa siitä opiksi. Kesken projektin voidaan tarpeen tullen kehittää prosessia ja toimintatapoja lennosta, jos huomataan, että jokin asia ei toimi asiantuntijoiden mielestä. Kyselyssä kerrottiin, kuinka prosessin tulisi olla jatkuvasti kehittyvää. Palautteen kautta mahdolliset prosessin ongelmat saadaan korjattua tehokkaasti. Palautetta on tärkeä pyytää eri toimittajilta, jotta saadaan laaja käsitys ja jokaisen toimittajan mielipide kuulluksi.

7.4.6 Menetelmät käytännön tasolle

Yksi haaste monitoimittajaympäristöissä on se, että menetelmät ja ideologiat, mitä projektissa tai hankkeessa sovitaan käytettävän, pitäisi tuoda myös näkyväksi ruohonjuuritasolla. Ei riitä, että projektissa tai hankkeessa sovitaan, että käytetään ketteriä menetelmiä, mutta ketterät menetelmät eivät näy mitenkään projektissa suunnitteluvaihetta pidemmälle. Menetelmästä pitäisi siis ottaa kaikki hyödyt irti ja saada hyödyt myös näkymään jokaisessa prosessin vaiheessa. Monitoimittajaympäristössä jokaisen toimittajan tulisi myös lupautua noudattamaan kyseistä menetelmää.

7.5 Toimintatapojen ja käytänteiden kehitysehdotukset

Hyväksymistestaus voi pahimmillaan muodostua pullonkaulaksi eli kuormittaa ja hidastaa prosessia, jos ei ymmärretä mitä ollaan testaamassa. Tietyillä toimintatapojen muutoksilla voidaan tehostaa hyväksymistestausta monitoimittajaympäristöissä ja saada siitä toimiva kokonaisuus.

7.5.1 Testausympäristö ja data

Jotta hyväksymistestauksessa löydettäisiin virheet ennen tuotteen menoa tuotantoon, on kriittisen tärkeää, että hyväksymistestausympäristö on samankaltainen tuotannon kanssa. Kyselyssä asiantuntijat kuvasivat, kuinka tuotannon kaltainen testausympäristö mahdollistaa sen, että esimerkiksi ei-toiminnallisten vaatimusten tila tulee ilmi jo hyväksymistestauksen aikana. Tämän kaltaisina esimerkkeinä mainittiin esimerkiksi vasteajat. Järjestelmätestauksessa käytetty ympäristö voi olla hyvinkin erilainen tuotannon ympäristön kanssa, minkä vuoksi juuri hyväksymistestauksen ympäristö tulisi olla tuotannon kaltainen.

Kyselyyn vastanneet korostivat, että on myös olennaista, että jokaisella toimittajalla olisi käytössä ympäristö tai pääsy ympäristöön, jossa nähdään eri toimittajien tuotokset yhdessä. Eli käytössä olisi niin sanottu hybridiympäristö. Ympäristön tulisi olla sellainen, että siellä on liitettyä kaikkien eri toimittajien komponentit, palvelut sekä järjestelmät. Myös integraation tulisi olla olemassa eri komponenttien ja järjestelmien välillä. Ympäristön tulisi olla käytössä aina kaikille eri toimittajille ja mahdollisista katkoksista tulisi aina viestiä koko monitoimittajaprojektille. Ideaali tapauksessa hybridiympäristöön olisi pääsy jo esimerkiksi järjestelmätestausvaiheessa. Näin jo järjestelmätestauksen aikana päästäisiin näkemään, kuinka eri toimittajien komponentit ja palvelut toimivat keskenään. Tällä tavoin vähennettäisiin hyväksymistestauksen työkuormaa ja mahdolliset virheet havaittaisiin jo järjestelmätestauksen vaiheessa. Esimerkiksi mahdolliset yhteensopivuus ongelmat ja integraatiovirheet palveluiden välillä saataisiin jo oikaistua ennen, kun testauksessa edetään hyväksymistestaukseen saakka.

Sekä hyväksymistestaukselle että järjestelmätestaukselle tulisi olla käytössä myös sopivaa testidataa testausta varten. Kyselyn vastauksissa pidettiin tärkeänä, että monitoimittajaprojekteissa kootaan erilaisia testausaineistoja eri toimittajilta. Näin voidaan varmistaa, että testidata on saatavilla, monipuolista ja realistista. Testauksen aikana voidaan

onnistuneesti kokeilla erilaisia skenaarioita testiympäristössä. Tällä tavoin voidaan todentaa, että ohjelmisto toimii eri tilanteissa odotetulla tavalla.

7.5.2 Yhteisymmärrys testattavasta kohteesta

Kyselyssä vastauksissa tuli esille, kuinka hyväksymistestauksessa on olennaista dokumentoida testattavan kohteen sisältö, jotta jokainen osapuoli ymmärtää mitä ollaan testaamassa. On tunnistettava kohde mitä testataan ja on myös tiedostettava mitä ei testata. Olisi hyödyllistä, mikäli eri testausasiantuntijat olisivat katselmoimassa käyttötapauksien tekoa, jotta hyväksymistestauksen testitapausten teko sujuisi tehokkaasti. On tärkeää, että testitapaukset ovat sisällöltään muodostettu oikein. Jotta hyväksymistestaus sujuisi jouhevasti, tulisi testitapausten olla valideja eli vastata todellisia käyttötapauksia mahdollisimman tarkasti. Mikäli jo esimerkiksi testitapausten stepit on toteutettu väärin, pahimmassa tapauksessa oleellisia virheitä jää löytämättä. Vastaavasti voidaan myös raportoida turhista virheistä, jotka hidastavat hyväksymistestauksen etenemistä ja kuormittavat eri toimittajia.

On välttämätöntä myös rajata, mitkä ovat testauksen tehtävät hyväksymistestausvaiheessa. Ottaako testaus kantaa esimerkiksi verkkosivun terminologiaan tai visuaalisiin ratkaisuihin, vai keskitytäänkö esimerkiksi ainoastaan toiminnallisiin. Jokaiseen pieneen yksityiskohtaan ei voida, eikä pidä ottaa kantaa hyväksymistestauksen aikana.

Väärin muodostetut testitapaukset ja turhien virheiden raportointi kuormittaa toimittavia yrityksiä, sillä kaikki raportoidut, virheet on tutkittava. Kyselyn vastauksissa käy ilmi, että hyväksymistestaaajien kouluttaminen ja tukeminen hyväksymistestauksen aikana on yksi ratkaisu tällaisiin haasteisiin.

7.5.3 Yhteistyö ja kommunikaatio

Ihmisten välisen kommunikaation ja yhteistyön tärkeys nousee esille entistä enemmän monitoimittajaympäristöissä. Koska monitoimittajaympäristöiset projektit koostuvat lukuisista eri asiantuntija tiimeistä, olisi hyvä hyödyntää eri osa-alueiden asiantuntijoita hyväksymistestausvaiheessa. Kyselyn vastauksessa tuli ilmi ehdotus, että jokainen toimittaja lukisi kaikki testitapaukset läpi, vaikka osa-alue ei koskisi heitä. Näin voitaisiin jo hyvissä ajoin saada

kiinni virheellisiä testitapauksia. Testitapausten lukeminen voisi myös auttaa hahmottamaan kokonaisuutta paremmin ja mahdollisesti auttaa selkeyttämään tavoitteita entistä enemmän. Kyselyssä nostettiin myös esille se, miten ratkaisevan tärkeää on, että monitoimittajaympäristöissä jokaisen toimittajan olisi hyvä panostaa selkeään ja ymmärrettävään dokumentaatioon ja tiedon välittämiseen avoimesti.

Kyselyssä vastanneet kuvasivat, kuinka yhteistyö ja kommunikoinnin toimivuuden kautta mahdollistetaan esimerkiksi priorisoinnin sujuvuus ja löydösten käsittelyt. Hyvällä kommunikaatiolla voidaan välttää tilanne, jossa useat eri toimittajat selvittävät päivien ajan samoja löydöksiä. Hyvällä yhteistyöllä ja kommunikaatiolla hyväksymistestauksen löydökset ohjataan tehokkaasti vastuussa olevalle toimijalle mahdollisimman nopeasti, niin ettei useat eri toimijat joudu kauan käsittelemään samaa tapausta. Erilaiset ongelmat voivat myös ratketa yhteistyöllä hyvinkin tehokkaasti.

Kyselyn vastauksissa nostettiin esille, kuinka hyvällä kommunikoinnilla voidaan myös selvittää mahdolliset konfliktit nopeasti. Kommunikoinnin tulee olla myös selkeää, erityisesti silloin, kun viestintä liittyy esimerkiksi muuttuneisiin tilanteisiin. Kyselyn vastauksessa painotettiin, kuinka kaikille sidosryhmille viestintä on oltava selkeää, ja sisältää kattavasti tietoa vaikutuksista, syistä ja tavoitteista. Muuttuvilla tilanteilla voi olla suuriakin vaikutuksia esimerkiksi juuri hyväksymistestaukseen. On siis oleellista, että sidosryhmät ja eri toimittajat pysyvät ajan tasalla tilanteissa jatkuvasti.

Monitoimittajaympäristössä hyväksymistestaus voidaan aloittaa vasta kun jokaisen eri toimittajan tuotteet ovat valmiita. Mikäli kyseessä on laajempi kokonaisuus, joka on riippuvainen eri toimittajien komponenteista tai palvelusta, voi hyväksymistestaus joutua odottamaan testauksen aloittamista pitkään. Kyselyssä kehitysehdotuksena oli, että kommunikoitaisiin paremmin siitä, onko eri toimittajien tuotteet valmiudessa testausta varten. Mikäli kommunikointi ja yhteistyö ei toimi, niin tieto ei kulje hyväksymistestaukselle keskeneräisestä toteutuksesta. Näin ollen turhia virhe löydöksiä muodostuu paljon, mikäli hyväksymistestausta toteutetaan ympäristössä, jossa ei ole vielä kaikkien toimittajien tuoreimmat tuotokset.

Tiedon ja osaamisen jakaminen on hyvin ratkaisevassa asemassa. Voi olla, että monitoimittajaympäristössä on hyvin erilaisten osa-alueiden yrityksiä mukana. Tämän vuoksi olisi tärkeää hyödyntää jokaisen toimittajan osaamista ja tuoda ne julki. On hyvä myös muodostaa ilmapiiri, missä uskalletaan keskustella ja kysyä asioista rohkeasti ja jokainen toimittaja auttaa toisiaan ongelmatilanteissa. Myös mahdolliset työkalut ja apukeinot on hyvä kertoa eteenpäin. Kyselyn vastauksissa kuvattiin, kuinka etukäteen on hyvä sopia kommunikointi kanavat, missä minkäkin tyyllisiä aiheita on käsiteltävä. Myös säännöllisten kokousten avulla voidaan pysyä ajan tasalla asioista.

7.5.4 Dokumentointi ja raportointi

Monitoimittajaympäristössä on kriittisen tärkeää, että määriykset, käyttötapaukset ja muut dokumentaatiot olisivat kaikkien toimittajien saatavilla. Myös testaajilla ja testauksen koordinaattoreilla tulisi olla pääsy dokumentteihin, jotta hyväksymistestauksen testitapauksista saataisiin mahdollisimman onnistuneita. Ilman avointa dokumentaatioiden jakelua paljon oleellista tietoa jäisi piiloon ja pahimmassa tapauksessa testaus ja hyväksymistestaus voisi jäädä hyvin suppeaksi. Kyselyssä vastattiin, kuinka dokumenttien sisältö tulisi olla laadukasta jokaisen toimittajan puolesta.

Kyselyvastauksissa tuli esiin, kuinka eri toimittajien olisi hyvä sopia yhteisistä dokumentointitavoista. Jokaisella yrityksellä ja tiimillä voi olla omia tapoja raportoida asioista. Olisi kuitenkin oleellista, että esimerkiksi ohjelmistoista löydetyt havainnot raportoidaan sovitulla tavalla ja projektissa osallinen noudattaa samoja raportointitapoja. Mikäli testauksen tuloksista raportoidaan epäselvästi, tuloksista ei ole mitään hyötyä. Kyselyyn vastanneiden mielestä dokumentaatioon ja sen laatuun on siis hyvin tärkeää panostaa.

Kaikki monitoimittajaympäristön toimittajien olisi hyvä kommunikoida ja tehdä yhteistyötä avoimesti. Monitoimittajaympäristössä olisi hyvä valita yhteinen raportoinnin työkalu, jota kaikki käyttävät ja jonka avulla voidaan tavoittaa eri toimittajat. Alustana voi olla esimerkiksi Microsoft Teams, Jira, Azure Devops tai mikä tahansa ohjelmisto. Ohjelmiston avulla voidaan jakaa löydöksiä ja keskustella avoimesti kaikkien toimittajien kesken. Mikäli jokainen toimittaja työskentelee omissa oloissaan, eikä viestit ja tieto välity kaikille toimittajille, voi paljon oleellista tietoa jäädä jakamatta.

Kyselyn tuloksista nousi esille erityisesti yhtenäisen raportoinnin tärkeys. Jotta raportoinnin työkaluista olisi hyötyä, tulee kaikkien toimittajien noudattaa yhteistä linjaa. Koska monitoimittajaympäristössä kaikki toimittajat ovat jollain asteikolla riippuvaisia toisistaan, vaatii yhteistyö kaikkien panostuksen. Raportointi mahdollistaa myös sen, että jokainen toimittaja näkee toteutuksen ja testauksen etenemisen ja tulokset helposti ja nopeasti. Löydösten selkeä ja yhtenäinen raportointi on myös erittäin olennaista.

8 Päätäntä

8.1 Tutkimuksen eettisyys

Tutkimuksen tulisi aina noudattaa avoimuutta ja läpinäkyvyyttä, erityisesti eettisyyden kohdalla. On siis korostettava, että tässä tutkimuksessa hyödynsin omia kokemuksia aiheeseen liittyen. Nämä kokemukset näkyvät vahvasti läpi tutkielman, erityisesti opinnäytetyön tutkimusosiossa. Vaikka tutkimuksen aiheeseen oli paljon omakohtaisia näkemyksiä, pyrin silti käsittelemään aihetta objektiivisesti läpi opinnäytetyön. Lisäksi ammatillisten kokemusten tai näkemysten ei ole annettu vaikuttaa missään vaiheessa tutkielmaa, vaan aihetta on käsitelty täysin puolueettomasti. Omien mielipiteiden ei annettu vaikuttaa tutkimuksen sisältöön. Sen sijaan kyselyn tulokset on esitetty sellaisina kuin ne ovat ilmenneet kyselytutkimuksen tuloksissa.

Koska opinnäytetyö toteutettiin vain pienelle asiantuntijajoukolle, on huomioitava, ettei tuloksia voida suoraan yleistää jokaiseen monitoimittajaprojektiin. On kuitenkin tärkeää muistaa, että jokainen monitoimittajaprojekti on ainutlaatuinen. Vaikka kysely olisi toteutettu suuremmalle massalle, kokemukset voivat vaihdella merkittävästi erilaisten projektien välillä.

8.2 Tutkimuskysymysten tarkastelu

On hyvä tarkastella aiemmin esitettyjä tutkimuskysymyksiä tulosten pohjalta. Opinnäytetyössä kaikkiin alkuperäisiin tutkimuskysymyksiin löydettiin vastaus tutkimuksen ja tutkimustulosten läpikäynnin myötä. Tutkimuskysymykset osoittautuivat erittäin keskeisiksi teemoiksi. Kyselyhaastattelujen tuloksissa tutkimuskysymysten teemat nousivat esille useaan otteeseen. Opinnäytetyön tutkimuskysymyksiä oli siis kolme kappaletta:

1. Mitä pitäisi muuttaa prosessissa ja tiedonkulussa, jotta saadaan parempi kyvykkyyks hyväksymistestaukseen monitoimittajaympäristössä?
2. Miten varmistetaan, että kaikkien toimittajien komponentit, testaussuunnittelu ja testaus on tietyn tasoista, jotta hyväksymistestaus onnistuu jouhevasti?
3. Kuinka varmistetaan, että testitapaukset heijastavat todellisia käyttötapauksia ja ovat valideja hyväksymistestausvaiheessa, kun kyseessä on monitoimittajaympäristö?

Ensimmäinen tutkimuskysymys osoittautui tutkimuksen aikana jopa työn tärkeimmäksi kysymykseksi. Kommunikointi, yhteistyö ja prosessin hallinta saivat suurta painoarvoa tutkimuksen tuloksissa. Tutkimuksessa ilmeni, että useita eri haasteita voidaan välttää panostamalla näihin kolmeen kokonaisuuteen. Toiseen tutkimuskysymykseen kyselyhaastatteluissa ratkaisuna löytyi kommunikointi, prosessin hallinta ja yhteisten pelisääntöjen muodostaminen toimittajien kesken. Kolmanteen tutkimuskysymykseen ratkaisuna saatiin testaajien koulutus, tiedonjako, osallistuttaminen prosessiin ja testitapausten katselmointi. Tutkimuksen myötä tuli siis selville, että jokaiseen tutkimuskysymykseen ei ollut vain yhtä oikeaa vastausta.

8.3 Jatkokehitysehdotukset

Jatkokehitystä varten tutkimuksessa voitaisiin pureutua vielä tarkemmin käytännön tasolle. Käytännön tasolla voisi tarkastella jokaista tutkimuskysymystä. Olisi hyödyllistä selvittää, kuinka käytännössä prosessi tulisi suunnitella ja kuinka varmistettaisiin esimerkiksi eri toimittajien testauksen tasokkuus. Jatkokehitystä varten olisi tutkimukseen hyvä saada mukaan esimerkiksi projekti ja tarkastella projektia ja sen prosessia tarkemmin. Näin tutkimuksessa nähtäisiin selkeästi prosessissa ilmenevät haasteet konkreettisesti ja saataisiin ongelmakohtiin käytännön ratkaisuja.

Jatkokehitystä varten tutkimukseen voisi ottaa mukaan useita eri toimittajaorganisaatioita, asiakkaita ja hyväksymistestaajia. Asiakkaiden ja hyväksymistestaajien näkemykset aiheeseen voisivat tuoda uutta näkökulmaa ja hyödyllisiä ideoita aiheeseen. Tutkimuksen kyselyhaastattelua voitaisiin myös jakaa huomattavasti laajemmalle asiantuntijajoukolle. Tällä tavoin havaittaisiin toistuvatko tietyt samat haasteet jokaisessa monitoimittajaprojektissa ja tutkimuksen tulokset olisi mahdollista yleistää. Näin voitaisiin myös selvittää, toistuvatko tässä tutkimuksessakin esiin nousseet teemat laajemman mittakaavan kyselytutkimuksissa.

8.4 Johtopäätökset

Tutkimuksessa on tullut selväksi, että onnistuneen hyväksymistestauksen saavuttaminen monitoimittajaympäristössä vaatii monen eri palasen onnistumisen. Ohjelmistoprosessin suunnittelu ja hallinta ovat kriittisen tärkeitä, sillä ne muodostavat perustan onnistuneelle hyväksymistestaukselle monitoimittajaympäristössä. Prosessin avulla varmistetaan, että projekti etenee tehokkaasti ja aikataulussa jokaisessa toimittavassa organisaatiossa. Prosessi määrittelee, kuinka tuloksista raportoidaan eri toimittajien kesken, miten dokumentoidaan ja miten resurssit jaetaan. Prosessin kautta voidaan tunnistaa ja hallita mahdollisia riskejä ja haasteita, jotka voivat tulla esiin monimutkaisissa monitoimittajaympäristöissä. Hyvin hallittu ja suunniteltu prosessi tuo asiantuntijoille rauhan tehdä omaa työtään, ilman ylitsepääsemättömiä aikataulupaineita ja lopputuloksesta tulee näin ollen myös parempi. On tärkeää jalkauttaa prosessi monitoimittajaprojektin kaikille toimittajille ja sitä tulee myös noudattaa, jotta voidaan maksimoida prosessin tarjoamat hyödyt.

Onnistunut hyväksymistestaus monitoimittajaympäristössä edellyttää jatkuvaa yhteistyötä, etenemisen seuranta ja muuntautumiskykyä uusien odottamattomien tilanteiden tullessa vastaan. On hyvä myös tiedostaa toimintatavat ja käytännöt, jotka eivät toimi ja kehittää niitä jatkoa ajatellen. Jatkuva muutos ja halu kehittyä paremmaksi ovat tärkeässä asemassa onnistumisen kannalta. Monitoimittajaympäristössä on lisäksi myös välttämätöntä huomioida kaikkien toimittajien tarpeet. On myös realistisesti arvioitava kokonaisuuden laajuus. On olennaista saada tarkka käsitys siitä, kuinka iso muutos on kyseessä ja miten monimutkainen ympäristökokonaisuus on kyseessä. Yhteisen tavoitteen ymmärtäminen ja avoin kommunikaatio kaikkien osapuolten välillä ovat avainasemassa hyväksymistestauksen onnistumisessa monitoimittajaympäristöissä. Menestyksekkään lopputuloksen saavuttaminen vaatii jokaisen projektissa mukana olevan organisaation sitoutumisen yhteisiin pelisääntöihin.

8.5 Yhteenveto

Tässä opinnäytetyössä tarkoituksena oli selvittää, kuinka monitoimittajaympäristössä saavutetaan onnistunut hyväksymistestaus. Tutkimuksen vastauksissa tuli esille, että monitoimittajaympäristö ei väistämättä heti tarkoita epäonnistunutta hyväksymistestausta. Vastaukset antoivat myös hyvän käsityksen siitä, että monitoimittajaympäristö tuo väistämättäkin omanlaisia haasteita prosessin

eri vaiheille. Haasteiden välttämisen kuvattiin olevan kuitenkin aina mahdollista, kunhan vain huomioidaan monitoimittajaympäristön ja hyväksymistestauksen ominaispiirteet ja ongelmakohdat.

On tärkeää huomata, että tässä tutkimuksessa esille tuodut asiantuntijahavainnot osoittavat, ettei ole olemassa yhtä ainoaa ratkaisua onnistuneelle hyväksymistestaukselle monitoimittajaympäristössä. Hyväksymistestauksen epäonnistuminen tai onnistuminen ei ole myöskään yhden ihmisen harteilla. Yhteenvetona voidaan kuitenkin sanoa, että prosessin suunnittelulla ja sen hallinnalla sekä toimittajien yhteistyöllä painotettiin olevan suurin merkitys onnistuneen hyväksymistestauksen saavuttamiseksi.

Lähteet

Benefits of SFAFe®. N.d. Artikkelin SFAFe Inc:n www- sivulla. Viitattu 12.01.2024.

<https://scaledagile.com/what-is-safe/scaled-agile-benefits/>

Ekholm, J & Nylén, R. 2021. Ketterän kehittämisen sanakirja – agilea a:sta ö:hön. Blogiteksti Futurice Oy:n www-sivulla. Julkaistu 08.12.2021. Viitattu 20.12.2023.

<https://futurice.com/blog/ketteran-kehittamisen-sanakirja-agilea-a-sta-o-hon>

Grönroos, M. N.d. Tiedon jakaminen ja monitoimittajaympäristö. Artikkelin yrityksen www- sivulla.

Viitattu 21.03.2024. <https://www.vihrearuusu.fi/Jakaminen.php>

Heikkilä, R. 2014. Häiriönhallinnan haasteet ja niiden ratkaisut monitoimittajaympäristössä.

Kandidaatintutkielma. Jyväskylän yliopisto, Tietojärjestelmätieteiden tutkinto-ohjelma. Viitattu

03.12.2023. <https://jyx.jyu.fi/bitstream/handle/123456789/43226/1/Riina%20Heikkil%C3%A4.pdf>

ISTQB® testaussanasto. 2012. International Software Testing Qualifications Board:n

testaussanastostandardin suomenkielinen julkaisu. Viitattu 02.12.2023. [https://tivia-](https://tivia-jasenyhdistykset.fi/fistb-testi/wp-content/uploads/sites/30/2020/12/istqb_sanasto_2015-04-30-2.3-ENG-FI-Koekelpoinen-versio_0.pdf)

[jasenyhdistykset.fi/fistb-testi/wp-content/uploads/sites/30/2020/12/istqb_sanasto_2015-04-30-2.3-ENG-FI-Koekelpoinen-versio_0.pdf](https://tivia-jasenyhdistykset.fi/fistb-testi/wp-content/uploads/sites/30/2020/12/istqb_sanasto_2015-04-30-2.3-ENG-FI-Koekelpoinen-versio_0.pdf)

ISTQB® Perustason sertifiikaattisältö. 2023. International Software Testing Qualifications Board:n julkaisema sertifiikaatin suomenkielinen julkaisu. Julkaistu 1.9.2023. Viitattu 02.12.2023.

<https://fistb.fi/wp-content/uploads/sites/30/2023/09/Kaannos-20230901.pdf>

ISTQB® Perustason sertifiikaattisältö. 2018. International Software Testing Qualifications Board:n julkaisema sertifiikaatin suomenkielinen julkaisu. Julkaistu 10.10.2018. Viitattu 02.12.2023.

<https://tivia-jasenyhdistykset.fi/fistb-testi/wp-content/uploads/sites/30/2020/12/CTFL-2018-Sertifikaattisalto-20181010-1-Valmis.pdf>

Keskitalo, L. 2012. Hyväksymistestaus ohjelmistokehityksessä – Prosessi, tehtävät ja roolit.

Opinnäytetyö, AMK. Haaga-Helian ammattikorkeakoulu, Tietojenkäsittelyn koulutusohjelma.

Viitattu 05.01.2024.

https://www.theseus.fi/bitstream/handle/10024/48825/Keskitalo_Lasse.pdf;jsessionid=21AC9E38E5173213488AA5160DD0EF1B?sequence=1

Kivi, R. 2020. IT-palvelunhallinnan prosessien ja käytänteiden kehittäminen

monitoimittajaympäristössä. Opinnäytetyö, AMK. Haaga-Helian ammattikorkeakoulu,

Tietojärjestelmäosaamisen koulutusohjelma. Viitattu 03.12.2023.

https://www.theseus.fi/bitstream/handle/10024/336805/Kivi_Riku.pdf?sequence=2&isAllowed=y

Laine, H & Paakki, J. N.d. Ohjelmistoprosessi. Helsingin yliopiston luentokalvot. Viitattu

02.12.2023. <https://www.cs.helsinki.fi/u/paakki/ohjuk03-luento2.pdf>

Lehtonen, T. 2015. Sulautettujen järjestelmien ketterä käsikirja. Teoksen verkkojulkaisu. Päivitetty

22.01.2015. Viitattu 20.12.2023. https://tt.utu.fi/embedded_kasikirja/1/1/index.html

Leppäniemi, J. 2014. Testaamisen kehittäminen yrityksen ohjelmistokehityksessä. Opinnäytetyö, AMK. Satakunnan ammattikorkeakoulu, Tietojenkäsittelyn koulutusohjelma. Viitattu 10.12.2023. https://www.theseus.fi/bitstream/handle/10024/73071/Leppaniemi_Jani.pdf?sequence=1&isAllowed=y

Magalhães, I.L. 2015. Challenges of User Acceptance Testing. Julkaistu 18.02.2015. Viitattu 24.01.2024. <https://www.linkedin.com/pulse/challenges-user-acceptance-testing-ivan-luizio-magalh%C3%A3es/>

Mitä on ohjelmistotestaus ja mihin ohjelmistotestausta tarvitaan. 2021. Blogiteksti JMC Nordic Oy:n www- sivulla. Julkaistu 23.11.2021. Viitattu 21.12.2023. <https://www.jmcnordic.fi/blog/mita-on-sovellustestaus-ja-mihin-sovellustestausta-tarvitaan.html>

Mitä on ohjelmistotestaus ja mitä hyötyä siitä on. 2022. Blogiteksti VALA Group Oy:n www- sivulla. Julkaistu 10.11.2022. Viitattu 21.03.2024. <https://www.valagroup.com/fi/blogi/mita-on-ohjelmistotestaus-ja-mita-hyotya-siita-on/>

Mäkinen, S. 2020. Ohjelmistoprosessin laadunvarmistusjärjestelmän suunnittelu. Diplomityö. Tampereen yliopisto, Tietotekniikan DI-tutkinto-ohjelma. Viitattu 12.12.2023 <https://trepo.tuni.fi/bitstream/handle/10024/121106/M%C3%A4kinenSami.pdf?sequence>

Paakki, J. 2014. Ohjelmistoprosessit ja ohjelmistojen laatu. Helsingin yliopiston Tietojenkäsittelytieteen laitoksen verkkoluentoaikalvot. Viitattu 03.12.2023. <https://www.cs.helsinki.fi/u/paakki/Laatu-14-Luentokalvot-2.pdf>

Peltola, R. 2020. SAFen hyödyt ja haasteet järjestelmänkehitystyössä. Opinnäytetyö, AMK. Oulun ammattikorkeakoulu, Tietojenkäsittelyn tutkinto-ohjelma. Viitattu 12.01.2024. <https://www.theseus.fi/bitstream/handle/10024/346145/Peltola%20Riitta.pdf?sequence=2&isAllowed=y>

Rinne, T. 2016. Testauskäytäntöjen kehittäminen Laurean tietojärjestelmäprojektissa. Opinnäytetyö, Ylempi AMK. Tampereen ammattikorkeakoulu, Teknologiaosaamisen johtamisen koulutusohjelma. Viitattu 03.12.2023. https://www.theseus.fi/bitstream/handle/10024/110575/Rinne_Tuuli.pdf?sequence=1

SAFe tulee ja pelastaa. 2018. Blogiteksti CGI Inc:n www- sivulla. Julkaistu 03.05.2018. Viitattu 03.12.2023. <https://www.cgi.com/fi/fi/blogi/safe-tulee-ja-pelastaa>

Seppälä, J. 2020. Integraatiotestaus .NET-palvelinohjelmistolle. Opinnäytetyö, AMK. Jyväskylän ammattikorkeakoulu, Tieto- ja viestintäteknikka tutkinto-ohjelma. Viitattu 05.01.2024. <https://www.theseus.fi/bitstream/handle/10024/354151/Oppari.pdf?sequence=2&isAllowed=y>

Tsarykau, P. 2021. Pros & Cons of Multi-Vendor Relationship. Julkaistu 15.06.2021. Viitattu 24.01.2024. <https://www.linkedin.com/pulse/pros-cons-multi-vendor-relationship-pavel-tsarykau/>

Tuemme yhteiskunnan kehitystä vastuullisen teknologian avulla. 2023. Artikkelitietoevry Oy:n www- sivulla. <https://www.tietoevry.com/fi/meista/tietoevry-lyhyesti/>

UAT testaus, eli hyväksymistestaus: Mitä se tarkoittaa ja miksi se on tärkeää. 2023. Blogiteksti VALA Group Oy:n www- sivulla. Julkaistu 24.10.2023. Viitattu 02.12.2023.
<https://www.valagroup.com/fi/blogi/hyvaksymistestaus/>

Vornanen, M. 2022. ASP.NET-sovelluskehityksen integraatiotestaus. Opinnäytetyö, AMK. Jyväskylän ammattikorkeakoulu, Tieto- ja viestintätekniikan tutkinto-ohjelma. Viitattu 05.01.2024.
https://www.theseus.fi/bitstream/handle/10024/783802/Opinnaytetyo_Vornanen_Mikko.pdf?sequence=2&isAllowed=y

Vuori, M. 2010. Menestyksekkäs hyväksymistestaus. Diasarja kirjoittajan www- sivulla. Julkaistu 21.11.2010. Viitattu 06.03.2024.
https://www.mattivuori.net/julkaisuluettelo/liitteet/menestyksekas_hyvaksymistestaus.pdf

What Is SAFe®. N.d. Artikkelin SAFe Inc:n www- sivulla. Viitattu 12.01.2024.
<https://scaledagile.com/what-is-safe/>

What Is User Acceptance Testing? (With Benefits and How-to). 2022. Artikkelin Indeed Inc:n www- sivulla. Päivitetty 01.10.2022. Viitattu 06.03.2024. <https://ca.indeed.com/career-advice/career-development/user-acceptance-testing#:~:text=Benefits%20of%20UAT&text=Reduces%20cost%3A%20It%27s%20much%20cheaper,such%20products%20from%20the%20market>

Yksikkötestaus. 2023. Ohjelmointi 3: Rajapinnat ja tekniikat - kurssin verkkomateriaali. Viitattu 05.01.2024. https://plus.tuni.fi/graderS/static/compcs140-f2023/modules/09/yksikko_fi.html

Liitteet

Liite 1. Kyselyhaastattelu

1. Mikä on roolisi projektissa? *

Oma vastauksesi

2. Kuinka monta eri toimittajaa projektissa/hankkeessa on?

- 2
- 3-4
- 5-7
- 8 tai useampi

3. Ajallisesti kuinka pitkä projekti/hanke on kyseessä?

Oma vastauksesi

4. Hyväksymistestauksen haasteita on monenlaisia kun on kyseessä monitoimittajaympäristö.

Haasteita on mm:

- Kommunikointi ja yhteistyö
- Testitapausten paikkansapitävyys ja ohjelmiston valmius hyväksymistestaukseen
- Aikataulupaineet ja eri toimittajien aikataulussa pysyminen

Mitä muita vastaavanlaisia haasteita mielestäsi hyväksymistestauksella on monitoimittajaympäristössä?

Oma vastauksesi

5. Mitä lisähaasteita monitoimittajaympäristö mielestäsi tuo hyväksymistestaukseen? *

Oma vastauksesi

6. Millaisia vaikutuksia epäonnistuneella hyväksymistestauksella on mielestäsi? *

Oma vastauksesi

7. Kuinka prosessia tulisi mielestäsi kehittää, jotta hyväksymistestaus olisi onnistunut monitoimittajaympäristössä? *

Oma vastauksesi

8. Mitä toimintatapoja tulisi kehittää tai millaisia toimintatapoja tulisi lisätä, jotta hyväksymistestaus sujuisi mahdollisimman jouhevasti?

Oma vastauksesi
