

Janita Sallanko

**Käyttöliittymän regressiotestauksen automatisointi:
toimintatutkimus**

Tietotekniikan pro gradu -tutkielma

28. marraskuuta 2019

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Janita Sallanko

Yhteystiedot: janita.h.sallanko@student.jyu.fi

Ohjaajat: Antti-Juhani Kaijanaho ja Ville Isomöttönen

Työn nimi: Käyttöliittymän regressiotestauksen automatisointi: toimintatutkimus

Title in English: Automation of regression testing on user interface: action research

Työ: Pro gradu -tutkielma

Opintosuunta: Ohjelmisto- ja tietoliikennetekniikka

Sivumäärä: 57+0

Tiivistelmä: Teollisuudessa regressiotestauksen automatisointi on perusteltua, koska se säästää aikaa ihmisen tekemältä testaukselta ja allokoii testaajien resursseja olennaisempiin testitapauksiin. Testiautomaation avulla voidaan varmistaa, että testaus suoritetaan säännöllisesti ohjelmistolle ja varmistetaan ohjelmiston toiminnallisuuden eheys. Tutkielman tavoitteena on saada aikaan muutos organisaatiossa, jotta testiautomaatio saataisiin käyttöön käyttöliittymän regressiotestauksen helpottamiseksi. Muutos toteutuu toimintatutkimuksen muodossa ja sen vaikutuksia arvioidaan organisaation ilmapiirin ja asenteiden muutoksia tarkastelemalla. Toimintatutkimuksen kautta saadaan lisätietoa testiautomaation käyttöönoton vaikutuksista lyhyellä ajanvälillä ja kartoitetaan siihen liittyviä haasteita.

Avainsanat: Regressiotestaus, automaatio, käyttöliittymä, toimintatutkimus

Abstract: In the industry, automating regression testing makes sense because it saves time from human-made testing and allocates testers' resources to more relevant test cases. Test automation can help to ensure that testing is performed regularly to the software, and to ensure the integrity of the software functionality. The purpose of this thesis is to bring about change in the organization when testing automation will be introduced to facilitate user interface regression testing. The change will take the form of action research and the impact is assessed on the basis of change in organization's atmosphere and attitudes. The action research provides more information on the effects of introducing test automation in a short

space of time and challenges.

Keywords: Regression, testing, automation, user interface, action research

Esipuhe

Tämä tutkielma lähti alun perin käyntiin kiinnostuksesta automaattista käyttöliittymätestausta kohtaan, mutta muuttuikin tutkimukseksi, jolla tuntui olevan merkitystä isommassa mittakaavassa. Olen iloinen, että niin kävi. Pääsin tutkimuksessa yhdistämään teoriaa ja käytäntöä tavalla, jota en osannut odottaa. Haluan kiittää kaikkia tutkimuksessa mukana olleita ihmisiä, joita ilman tämä tutkimus ei olisi toteutunut.

Suuri kiitos ohjaajilleni Antti-Juhani Kaijanaholle ja Ville Isomöttöselle, joiden ansiosta ajatus toimintatutkimuksesta muuttui todeksi. Ohjauksenne ja tukenne auttoivat pääsemään tähän pisteeseen.

Kiitos haastateltaville ja työpajoihin osallistuneille, jotka jaksoitte käyttää aikaanne tutkimukseen ja halusitte olla osana muutosta. Kiinnostuksestanne tutkimusta kohtaan oli suuri apu.

Kiitos myös lähipiirilleni kannustavasta tuesta ja rakentavista palautteista tutkimuksen aikana.

Viimeisenä kiitos sinulle, joka luet tätä. Jos pääsit tälle riville asti, olet lukenut tutkielmani ensimmäisen sivun. Kiitos siitä.

Jyväskylässä 28. marraskuuta 2019

Janita

Kuviot

Kuvio 1. V-malli (Ammann ja Offutt 2016, 23).....	4
Kuvio 2. Tutkimuksen vaiheet kaaviona	12
Kuvio 3. Temaattinen verkko regressiotestauksen merkitykselle	19
Kuvio 4. Temaattinen verkko testauksen rajoitteille	23
Kuvio 5. Temaattinen verkko muutoksen tarpeesta	25
Kuvio 6. Temaattinen verkko ajan roolista muutoksessa	33
Kuvio 7. Temaattinen verkko muutoksesta regressiotestaamisessa	36
Kuvio 8. Temaattinen verkko ajan tulevaisuuden näkemyksistä	38

Sisältö

1	JOHDANTO	1
2	OHJELMISTOTESTAUKSEN TERMISTÖÄ	3
	2.1 Ohjelmistotestaus.....	3
	2.1.1 Ohjelmistotestaus teollisuudessa.....	3
	2.2 Regressiotestaus	5
3	REGRESSIOTESTAUKSEN AUTOMATISOINTI.....	7
	3.1 Regressiotestauksen automatisoinnin haasteet.....	8
4	TUTKIMUKSEN TOTEUTUS	10
	4.1 Metodologia	10
	4.2 Toimintatutkimuksen vaiheet	11
	4.2.1 Alkuhaastattelut.....	13
	4.2.2 Loppuhaastattelut	14
	4.3 Tutkimuksen validiteetti	14
	4.4 Tutkimuksen eettisyys	17
5	ALKUHAASTATTELUN TULOKSET	19
	5.1 Regressiotestauksen merkitys testauksessa	19
	5.1.1 Regressiotestaus käsitteenä.....	20
	5.1.2 Ihmisen tekemä testaus	21
	5.1.3 Testiautomaation rooli testauksessa.....	22
	5.2 Testauksen rajoitteet.....	23
	5.2.1 Organisaation resurssit	24
	5.3 Tarve muutokselle	25
	5.3.1 Tyytyväisyys nykyhetkeen	26
	5.3.2 Odotukset tulevaisuutta kohtaan	26
6	INTERVENTIO	28
	6.1 Intervention tapahtumat	28
	6.1.1 Infrastruktuurin alustus	29
	6.1.2 Työpajat	30
7	LOPPUHAASTATTELUN TULOKSET	33
	7.1 Ajan rooli muutoksessa	33
	7.1.1 Ajan rajallisuus	34
	7.1.2 Testiautomaation vaikutus ajankäyttöön.....	34
	7.1.3 Ajan järjestäminen	35
	7.2 Regressiotestauksen muutokset.....	35
	7.2.1 Käsitys regressiotestauksesta.....	36
	7.2.2 Testiautomaation hyödyt	36
	7.2.3 Testiautomaatioon liittyvät haasteet	37
	7.3 Näkemys tulevaisuudesta	38

7.3.1	Parannus aikaisempaan verrattuna	38
7.3.2	Odotukset tulevaisuutta kohtaan	39
7.3.3	Parannettavaa jatkossa	40
8	POHDINTA	41
9	JOHTOPÄÄTÖKSET	45
	LÄHTEET	46

1 Johdanto

Ohjelmistokehityksessä uuden toiminnon lisääminen ohjelmistoon on aikaa vievää, jos joko kaisen muutoksen jälkeen pitää testata ohjelmisto varmistaakseen, että ohjelmiston toiminnallisuus on edelleen yhtenäinen uusien muutosten kanssa. Vaikka ohjelmiston uudet ominaisuudet testattaisiin huolellisesti, muutokset saattavat rikkoa aikaisempaa toiminnallisuutta. Mitä myöhemässä ohjelmistokehityksen vaiheessa ohjelmistovirhe huomataan, sitä kalliimmaksi myös sen korjaaminen muuttuu (Haikala ja Mikkonen 2011).

Testaus olisi hyvä automatisoida, jotta voidaan varmistua siitä, että testaus suoritetaan tarpeeksi usein (Antawan Holmes ja Marc Kellogg 2006). Säännöllisin väliajoin suoritettavien testien avulla voidaan varmistaa, että ohjelman toiminnallisuus on pysynyt yhtenäisenä. Automatisoidut testaustyökalut ovat hyödyllisiä myös siksi, että ne ovat kykeneviä suorittamaan testejä, raportoimaan lopputuloksia sekä vertailemaan testituloksia aikaisempiin testiajoihin (Yadav ja Kumar 2015).

Perusteellinen automatisoitu regressiotestaus on olennaista myös siksi, että ohjelmistot kehittyvät jatkuvasti. Vaikka ohjelmalla ajaisi yksikkötestit ja kattavuustestit onnistuneesti, ne ei välttämättä todenna riittävästi ohjelmiston todellista toimintakykyä. (Bruns, Kornstadt ja Wichmann 2009).

Kaikkia käyttöliittymän toiminnallisuuksia ei välttämättä pystytä testaamaan esimerkiksi yksikkötesteillä, jos toiminnallisuus on vahvasti sidoksissa käyttöliittymän komponentteihin. Nykypäivänä kehitettävissä ohjelmistoissa käyttöliittymä on keskeinen osa ohjelmistoa, ja voi viedä jopa puolet ohjelmistoon käytettävästä koodista (Memon ja Soffa 2003). Memonin ja Soffan (2003) mukaan käyttöliittymän oikein toimivuus on ratkaisevaa koko ohjelmiston toimivuuden kannalta, minkä takia on aiheellista kiinnittää huomiota käyttöliittymän regressiotestaukseen.

Ihmisen tekemänä käyttöliittymän regressiotestaus voi olla puutteellista ja kuluttaa resursseja. Regressiotestauksessa käydään läpi kaikki uuden ohjelman toiminnallisuudet, mutta manuaalisesti käsin testaaminen on vaivalloista ja aikaa vievää (Bruns, Kornstadt ja Wichmann 2009). Manuaalisessa testauksessa on ongelmana myös se, ettei käyttöliittymän kom-

ponenttien testaajat tai käyttäjät osaa välttämättä valita sopivia testitapauksia alkuperäisestä testisuunnitelmasta testatakseen muuttunutta ohjelmistoa (Singh ym. 2014). Testien automatisointi jättää vähemmän varaa inhimillisille virheille, ja ajan saatossa allokoit testaaajien resursseja muihin ongelmiin. Yksi suurimmista hyödyistä automaattisten testien hyödyntämisessä on se, että automatisoidut käyttöliittymätestit vapauttavat testaajat tylsistä ja toistuvista tehtävistä (Berner, Weber ja Keller 2005).

Teollisuudessa ohjelmistotestauskäytännöt eivät ole kehittyneet niin pitkälle kuin mitä on akateemisessa ympäristössä tutkittu. Vaikka ohjelmistokehityksessä tekniikat ovat kehittyneet ajan myötä, teollisuudessa testauskäytännöt voivat olla vielä alkutekijöissään (Juristo, Moreno ja Strigel 2006). Lisäksi ohjelmistotestauskäytännöt ovat erilaisia eri organisaatioiden välillä (Garousi ja Zhi 2013). Janssonin (2013) mukaan ihmisen käyttäytyminen rakentuu sosiaalisista käytännöistä. Käyttäytymisen muuttaminen tarkoittaa tällöin myös sitä, että asenteita ja arvoja muutetaan samalla (Hodgkinson 1957). Regressiotestauksen automatisointi muuttaa organisaation jo olemassa olevia testauskäytänteitä ja näin vaikuttaa sen sosiaaliseen ilmapiiriin, jonka takia sosiaalisten vaikutusten tutkiminen on perusteltua.

Tässä tutkimuksessa muutetaan suomalaisen ohjelmistoyrityksen testauskäytänteitä toimintatutkimuksen muodossa. Yritykseen, jossa tutkimus tapahtuu viitataan sanalla organisaatio anonymiteetin säilyttämiseksi. Tutkimuksessa selvitetään käytänteiden muuttamisella lyhytaikaiset vaikutukset organisaation ilmapiiriin ja asenteisiin, sekä pyritään intervention kautta helpottamaan regressiotestausta organisaation sisällä. Käytännön hyödyn lisäksi tutkimuksen voidaan nähdä lisäävän oman kontribuutionsa tieteelliseen keskusteluun, sillä tarvitaan lisää raportoitua tietoa regressiotestauksen automatisoinnin soveltamisesta käytäntöön. Tutkimuksessa raportoidaan yhden toimintatutkimussyklin tulokset.

Luvussa 2 käsitellään ohjelmisto- ja regressiotestausta käsitetasolla ja pohditaan ohjelmistotestauksen ilmenemistä teollisuudessa. Luvussa 3 käsitellään regressiotestauksen automatisointia käsitteenä ja siihen liittyviä haasteita. Neljännessä luvussa kuvataan tutkimuksen toteutusta. Luku 5 avaa ennen interventiota toteutetun alkuhaastattelun analyysin tuloksia. Kuudennessa luvussa selitetään intervention toteutus. Intervention jälkeen toteutettiin loppuhaastattelut, joiden tuloksia luku 7 käsittelee. Luvussa 8 pohditaan tulosten merkitystä tutkimuksen kannalta ja luvussa 9 esitetään johtopäätökset.

2 Ohjelmistotestauksen termistöä

Regressiotestauksen automatisoinnin vaikutusten ymmärtäminen vaatii ensin ymmärrystä siitä, mitä regressiotestauksella pyritään testaamaan ja miten se poikkeaa muusta testaamisesta. Tämän ymmärtämiseksi määritellään, mitä ohjelmistotestaus on ja miten ohjelmistotestauskäytännöt näkyvät teollisuudessa.

2.1 Ohjelmistotestaus

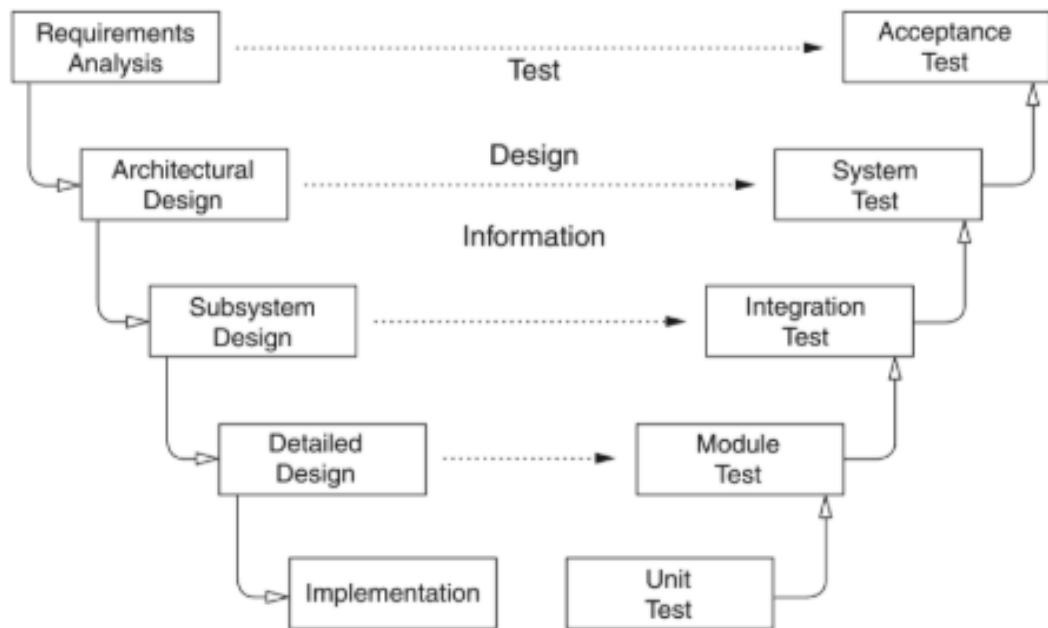
Ohjelmistotestaus on prosessi, jossa ajetaan testit ohjelmistolle tai sovellukselle virheiden löytämiseksi. Ohjelmistovirheiden paikantaminen on olennaista, jotta ohjelmistokehityksen tuloksena olisi virheetön ohjelmisto (Sneha ja Malle 2017). Virheiden löytämisen lisäksi testauksen avulla voidaan varmistaa kehitettävän ohjelmiston laatu. Testauksella voidaan varmistaa, että kehitetty ohjelmisto vastaa vaatimuksia ja toimii odotusten mukaisesti (Singh 2012).

Ohjelmistokehityksen elinkaaren kuvaamista varten on olemassa monia eri malleja kuten vesiputousmalli ja V-malli. Tietojenkäsittelytieteissä V-malli on yksi perinteisimmistä malleista ohjelmistotestauksen elinkaaren kuvaamiseen (Mathur ja Malik 2010). Mallin tarkoituksena on heijastaa kehitys- ja testausaktiiviteettien välistä suhdetta kuvion 1 mukaisesti.

V-malli kehittyi 1980-luvulla ja pohjautuu aikaisemmin ohjelmistokehityksessä käytettyyn vesiputousmalliin. V-mallissa testaus jakautuu neljään eri tasoon: yksikkö-, integraatio-, järjestelmä- ja hyväksymistestaukseen (Mathur ja Malik 2010). Mallissa ohjelmoinnin jälkeen yksikkö-, integraatio- ja järjestelmätestaukset tapahtuvat peräkkäin (Balaji ja Murugaiyan 2012).

2.1.1 Ohjelmistotestaus teollisuudessa

Ohjelmistokehityksen tutkimuksissa on yleensä kaksi tavoitetta: kustannusten vähentäminen ja kehitettävien ohjelmistojen laadun parantaminen (Kasurinen, Taipale ja Smolander 2010). Testaus nähdään teollisuudessa myös laadunvarmistajana, ja monet ohjelmistokehitysprojek-



Kuvio 1. V-malli (Ammann ja Offutt 2016, 23)

tit luottavatkin testaukseen ohjelmiston luotettavuutta mittaavana tekijänä (Felderer ja Ramler 2017).

Onoman et al. (1998) mukaan yrityksissä testausprosessit muistuttavat hyvin pitkälti teollisuudessa käytettävää ohjelmistokehitysmenetelmää. Hän käyttää artikkelissaan vesiputousmallia esimerkkinä, mutta tämä pätee myös uudempiin kehitysmenetelmiin kuten ketterään kehitykseen ja DevOps-menetelmään.

Teollisuudessa testaaminen, etenkin manuaalinen, tuo omat haasteensa käytännön tasolla. Manuaalisessa testauksessa testaajien rooli on hyvin ratkaiseva ohjelmistovirheiden löytämisen kannalta. Heidän puutteellinen koulutus vaikuttaa testaamiseen ja siihen, kuinka hyvin testaajat pystyvät löytämään virheitä tai puutteita ohjelmasta. On mahdollista, että testaajat eivät ole tarpeeksi hyvin varautuneet testamaan monimutkaisia tuotteita tai ole saaneet riittävää koulutusta testaamisesta ennen varsinaista testausta (Whittaker 2000).

2.2 Regressiotestaus

Regressiotestauksella tarkoitetaan testausta, jolla varmistetaan, että testattavan ohjelmiston aikaisempi toiminnallisuus ei ole rikkoutunut uusien ominaisuuksien myötä. Toisin sanoen sillä yritetään paikantaa ohjelmistoregressioita. Regressiotestauksen pyrkimykset ja testaustekniikat ovat hyvin samanlaisia kuin tavallisessa testaamisessa: virheiden paikantaminen ohjelmistossa ja luottamuksen kasvattaminen ohjelmiston toimivuutta kohtaan (Leung ja White 1989). Näiden lisäksi tavallisesta testauksesta poiketen regressiotestauksella halutaan varmistaa, että tuotettavan ohjelmiston laatu säilyy ja sen jatkuva käyttö on turvattu (Leung ja White 1989).

Regressiotestauksen keskiössä on nimenomaan ohjelmiston uudelleentestaaminen, eikä sen suoritus ole testaustasosta riippuvainen (Haikala ja Mikkonen 2011). Tutkijat ovat painottaneet ainoastaan muuttuneiden ja uusien ominaisuuksien testaamista sen sijaan, että käydään regressiotestauksessa kaikki ohjelmiston testitapaukset läpi uudelleen (Leung ja White 1989). Regressiotestaamisessa tyypillisesti ajetaan joko kaikki testit uudelleen samanaikaisesti tai näiden murto-osa (Homès 2012). Yleensä painotus on ensimmäisessä tapauksessa järjestelmä- ja hyväksymistestaustasolla (Homès 2012). Regressiotestaus on olennainen osa ohjelmiston ylläpitovaihetta, jossa järjestelmän toimintalogiikkaa voidaan oikaista, parantaa tai soveltaa uuteen ympäristöön (Leung ja White 1989). Regressiotestaus on ohjelmistotestauksessa yksi käytetyimmistä testaustekniikoista ja sitä käytetään laajasti teollisuudessa (Onoma ym. 1998). Monista muista testaustavoina poiketen regressiotestaus voidaan suorittaa millä tahansa aikaisemmin mainitun V-mallin tasolla.

Regressio voidaan ajatella jakautuvan kolmeen eri tyyppiin: ohjelmistovirheeseen, joka muodostuu muutoksen myötä, virheeseen, joka ilmenee muokatussa ominaisuudessa mutta on ollut jo valmiiksi ohjelmistossa sekä virheeseen, jotka ilmenee muualla ohjelmassa mutta johon tehty muokkaus ei suoranaisesti vaikuta. Näistä viimeisin on vaikein tunnistaa, sillä tällaisen regression ilmeneminen vaatii kokonaisvaltaisemman testaamisen ohjelmistolle, kun taas muut voidaan todentaa testaamalla muuttunutta ominaisuutta. (Homès 2012).

Leung ja Whiten (1989) mukaan regressiotestaus voidaan hahmottaa kahteen eri kategoriaan: tarkentavaan tai progressiiviseen regressiotestaukseen. Tarkentavalla regressiotest-

tauksella tarkoitetaan testaamista, jossa esiintyvät tietyt piirteet: useita testitapauksia voidaan käyttää uudelleen, ohjelman koodiin vaaditaan pieniä muutoksia, testaus suoritetaan epäsäännöllisin väliajoin sekä testauskohteen yksityiskohdat ovat pysyneet muttumattomana. Tarkentavaa regressiotestausta hyödynnetään tyypillisesti ohjelmistokehityksen aikana ja ylläpitovaiheessa. Progressiivinen regressiotestaus sen sijaan hyödyntää vähemmän uudelleenkäytettäviä testitapauksia, vaatii suuria muutoksia koodiin ja olettaa, että testattavan ohjelmiston yksityiskohdat ovat muuttuneet. Progressiivista regressiotestaamista suoritetaan säännöllisin väliajoin ja tyypillisesti adaptiivisen ylläpidon jälkeen. (Leung ja White 1989).

Regressiotestaus käytännössä tapahtuu Onoman et al. (1998) mukaan seuraavissa vaiheissa: ohjelmistovirheen löytyessä korjataan toiminto, muutetaan ohjelmistoa vastaamaan vaatimuksia, valikoidaan testitapaukset, suoritetaan testit, todennetaan virheeseen päätyneet testitapaukset tulosten perusteella sekä tunnistetaan ohjelmistovirheet ja vähennetään niitä.

Testattavan ohjelmiston ominaisuuksien määrän kasvaessa myös testitapausten määrä lisääntyy. On syytä priorisoida testejä, jos aika ei riitä kaikkien läpikäymiseen. Testitapausten valikoimiseen voi olla monta tekijää taustalla, kuten jo olemassa olevat testitapaukset, käytettävissä oleva testidata, riskiarviointi tai organisaation resurssit kuten aika (Homès 2012).

3 Regressiotestauksen automatisointi

Käyttöliittymän regressiotestauksen automatisointia on jo tutkittu jonkun aikaa akateemisella puolella. Etenkin teoriatasolla regressiotestauksesta on tehty aiemmin tutkimusta, mutta tutkimukset ja käytännöt eivät välttämättä kohtaa toisiaan reaali maailmassa (Engström ja Runeson 2010; Onoma ym. 1998). Regressiotestauksen teoreettiset tutkimukset yleensä käsittelevät testitapausten optimointia ja harvemmin syventyvät pohtimaan regressiotestauksen toteutusta käytännössä.

Testiautomaatiolla tarkoitetaan ohjelmistotestausaktiviteettien automatisointia, johon sisältyy ohjelmistokehitys ja testiskriptien ajaminen, testausvaatimusten varmistaminen ja automaattisten työkalujen hyödyntäminen (Karhu ym. 2009). Yksi suurimmista syistä testauksen automatisointiin on se, että manuaalinen testaaminen on hyvin aikaavievää (Collins, Dias-Neto ja Lucena Jr. 2012). Automatisoitu testaus voidaan kohdentaa joko koodiin tai käyttöliittymään. Näistä jälkimmäisessä hyödynnetään viitekehystä generoimaan näppäimistön näppäilyä ja hiirellä painamista mukailemaan käyttäjän toimintoja, jotta voidaan huomata muutokset ja varmistaa, että ohjelmisto toimii oikein (Sneha ja Malle 2017).

Regressiotestaus kuuluisi ajaa automatisoituna, ja Ammannin ja Offuttin (2016, 304) mukaan voidaan jopa väittää, että manuaalisesti suoritettua regressiotestausta ei voida pitää regressiotestauksena. Haasteena manuaalisessa testauksessa on se, että testaaminen kerryttää työmäärää suureksi ja on kallista (Homès 2012). Ihmisen tekemänä testaus on hidasta ja vaikea toistaa uudelleen, mutta sitäkin suurempi huoli on se, että manuaalinen testaus on altis virheellisille tuloksille (Çelik ym. 2017). Monesti ohjelmiston muutosten jälkeen tarvitaan regressiotestausta varmistuakseen siitä, että ohjelman toiminnallisuus ei ole rikkoutunut. Tällaisia tilanteita ovat korjaavat, täydentävät, soveltuvat ja ehkäisevät muutokset (Ammann ja Offutt 2016, 305).

Mitä suurempi testattava ohjelmisto on, sitä enemmän testitapauksia on käytävä testauksessa läpi. Jos jokainen testitapaus lisätään testisuunnitelmaan, se kasvattaa testisuunnitelmaa niin suureksi, ettei testausta ehditä aina käydä läpi, kun uusia muutoksia lisätään testattavaan järjestelmään. Testit häiritsevät ohjelmistokehitystä jos niiden ajamiseen kuluu aikaa. Toisaal-

ta testisuunnitelmien pitää olla myös tarpeeksi kattavat, sillä pienetkin muutokset järjestelmässä aiheuttavat ongelmia ominaisuuksissa, jotka eivät ole välttämättä suoraan kytköksissä muokattavaan ominaisuuteen. (Ammann ja Offutt 2016, 305).

Regressiotestauksen automatisointia puoltavat myös taloudelliset tekijät. Mitä myöhemmässä ohjelmistokehityksen tuotantovaiheessa regressio huomataan, sitä kalliimmaksi tulee virheiden korjaaminen (Haikala ja Mikkonen 2011). Testiautomaation lisäämisellä voidaan tehostaa testaamista, kun automatisoiduilla testeillä allokoidaan testaajien resursseja kriittisten ominaisuuksien testaamiseen ja monimutkaisempiin testitapauksiin (Kasurinen, Taipale ja Smolander 2010).

3.1 Regressiotestauksen automatisoinnin haasteet

Karhun et al. (2009) mukaan testauksen automatisoinnissa ihanteena olisi saada automatisoitua käytäntöjä mahdollisimman pitkälle, mutta tämä harvemmin toteutuu reaali maailmassa. Viimeistään regressiotestauksen automatisointivaiheessa pitää huomioida, mitä kaikkia testitapauksia olisi järkevää automatisoida ja mitä ei. Kaikkien testitapausten automatisointi ei ole todennäköisin ratkaisu testaukseen (Karhu ym. 2009). Liian yksityiskohtaiset testitapaukset vaativat enemmän ylläpitoa, joka taas vaikuttaa resurssien käyttöön. Täysin kattavaa testiautomaatiota ei ole realistista saavuttaa myöskään sen takia, että ihmisen väliintuloa tarvitaan viimeistään testien tulosten tulkitsemiseen tai ylläpitoon (Karhu ym. 2009).

Ohjelmistotestaamisen automatisointi ei ole matalan kynnyksen sijoitus, sillä se on taloudellisesti raskas ja korkean riskin projekti. Jotta automatisoidusta testauksesta olisi hyötyä ohjelmistokehitysprojekteissa, on muutamia seikkoja jotka täytyy huomioida organisaatiotasolla. Ohjelmistokehittäjien ja vanhojen järjestelmien koodit sekä aikaisemmat organisaation käytännöt eivät välttämättä tue automatisoitua testaamista. (Karhu ym. 2009). Jos yrityksessä on totuttu tekemään manuaalista regressiotestausta automatisoidun regressiotestauksen sijaan, siirtyminen testiautomaatioon voi olla hidasta (Collins, Dias-Neto ja Lucena Jr. 2012).

Olemassa olevien rajoitteiden vuoksi testausautomaatio ei tule ratkaisemaan kaikkia testausongelmia. Automatisoidusta testauksesta on hyötyä erityisesti tilanteissa, jossa testitapausta voidaan toistaa useampaan kertaan. Manuaalista ja automatisoitua testausta ei ole

mielekästä erottaa täysin toisistaan, sillä manuaalinen testaus on edelleen kannattavempaa testitapauksissa jotka eivät ole toistettavia. (Kasurinen, Taipale ja Smolander 2010).

Tutkimuksessa regressiotestaus keskittyy käyttöliittymän testaamiseen, jonka tarpeet poikkeavat perinteisestä ohjelmiston regressiotestaamisesta. Tällöin regressiotestauksessa tarvitaan testitapauksia, jotka eivät ole ohjelmaversiosta riippuvaisia (Memon ja Soffa 2003). Käyttöliittymän muutosten jälkeen osa testitapauksista voi muuttua tarpeettomiksi, ja näiden korjaaminen on kallista (Memon ja Soffa 2003).

Regressiotestauksen automatisoinnin hyötyjen toteaminen ei ole suoraviivaista. Muuhun testaamiseen verrattuna regressiotestauksen toimivuutta on vaikeampi arvioida, sillä regressiotestejä ei voida arvioida samoilla mittareilla kuin muuta testaamista. Tarkoituksena on varmistaa, ettei regressioita tai muita sivuvaikutuksia ilmene. (Homès 2012). Jos regressiotestien ajaminen epäonnistuu, se ei välttämättä ole merkki siitä, että testattavassa ohjelmistossa itsessään on vikoja vaan mahdollisesti testeissä. Tällaisten tilanteiden tutkiminen vie aikaa ja resursseja. (Ammann ja Offutt 2016)

Osa testiautomaation haasteista ilmenee vasta käyttöönoton jälkeen. Jos testiautomaatio otetaan käyttöön pienissä osissa, se edesauttaa vähentämään ylläpitokustannuksia pidemmällä tähtäimellä (Collins, Dias-Neto ja Lucena Jr. 2012). Käyttöliittymän testaukseen liittyy paljon käytännön haasteita, ja käytettävän testaustyökalun käyttö voi selittää osan testausajon aikana ilmenevistä ongelmista. Testaustyökalu itsessään ei saa kuitenkaan muuttua selitykseksi jokaiseen ongelmaan (Collins, Dias-Neto ja Lucena Jr. 2012).

4 Tutkimuksen toteutus

Tavoitteena on helpottaa käyttöliittymän regressiotestausta toimintatutkimuksen viitekehysellä ja tutkia organisaation sisällä muuttunutta ilmapiiriä ja asenteita intervention myötä. On mahdollista, että regressiotestaukseen liittyen ilmenee uusia haasteita, joita ei vielä automatisointivaiheessa huomata. Tästä syystä on hyvä selvittää myös samalla, mitä haasteita ilmenee regressiotestauksen automatisointiin liittyen.

Tutkimus on kvalitatiivinen ja tarkoituksena ei ole löytää yleismaailmallista ratkaisua, vaan tarkastella tutkimusongelmaa tietyssä kontekstissa ja ympäristössä. Testiautomaatiota on tutkittu aikaisemminkin, mutta edelleen on tarvetta empiirisille tutkimuksille jossa selvitetään ohjelmistotestaustekniikoita ja -käytäntöjä (Karhu ym. 2009).

Tutkimusongelmaa lähestytään toimintatutkimusmenetelmällä, sillä tutkimuksessa saadaan lisää empiiristä tietoa siihen, mitä käyttöliittymätestauksen automatisoinnista jo tiedetään. Tutkija toimii tutkimuksessa osana organisaatiota, jonka sisällä toteutetaan interventio ja on siten jatkuvassa vuorovaikutuksessa organisaation jäsenten kanssa.

4.1 Metodologia

Toimintatutkimuksessa tutkimus tapahtuu reaali maailman tilanteessa käytännönläheiseen tutkimusongelmaan perustuen (Edwards ja Willis 2014, 8). Aikaisemmin esitelty tutkimusongelma on hyvin käytännönläheinen ja nousee esiin reaali maailman tilanteesta, joten toimintatutkimuksen lähestymistavan hyödyntäminen sopii tutkimuksen luonteeseen. Kirjallisuuden puolella toimintatutkimus yleensä nähdään joko käytännöllisenä, kriittisenä tai teknisenä toimintana (Zuber-Skerritt 2003, 2-3). Käytännöllisellä näkökulmalla viitataan tutkimustapaan, jossa huomioidaan käytännön ongelmat ja pyritään ymmärtämään näitä toiminnan kautta, mikä kuvastaa parhaiten tämä tutkimuksen tavoitteita. Toisaalta tutkimukseen soveltuu joiltain osin myös kriittinen lähestymistapa, sillä testauksen automatisoinnilla halutaan vapauttaa testaa jia turhasta manuaalisesta työstä. Edwardsin ja Williksen (2014, 28) mukaan toimintatutkimuksen onnistumisen kannalta paikallinen konteksti on ratkaiseva, joten tutkijan pitää kehittää ymmärrystä siitä, mitä tutkimusympäristössä tapahtuu. Tämän saavuttamiseksi-

si kerätään tietoa organisaation sisällä, jossa interventio toteutetaan ja luodaan subjektiivisia näkemyksiä organisaatiossa toimivien ihmisten kanssa.

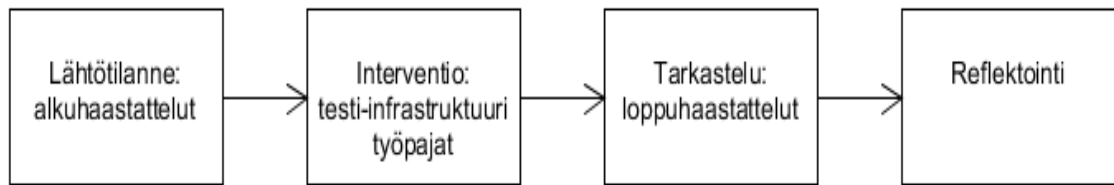
Toimintatutkimuksen määritelmästä ei olla yhtä mieltä, mutta toimintatutkimukselle ominaisia piirteitä on määritelty eri lähteissä (Petersen ym. 2014). Kurt Lewiniä pidetään alkuperäisen toimintatutkimuksen mallin kehittäjänä (Adelman 1993). Hänen kehittämän mallin mukaan tutkimuksen tietojenkeruu koostuu neljästä eri osasta: toiminnan arvioinnista, sen vaikutuksista oppimisesta, seuraavan askeleen suunnittelemisesta ja aikaisempien havaintojen heijastamisesta kokonaiskuvaan (Lewin 1946). Toiminnan arviointi osoittaa, onko muutos vastannut odotuksia ja miten toimintaa voidaan kehittää eteenpäin sen pohjalta (Lewin 1946). Tutkimukselle on asetettu selkeä tavoite, johon yritetään pyrkiä ja tutkimuksen tiedonkeruu tapahtuu toiminnan tulosten perusteella (Lewin 1946). Carrin ja Kemmicksen (1983, 165) mukaan toimintatutkimuksen päämääränä on parantaa nykyistä tilannetta ja saada ihmiset, joita muutos koskettaa osallistumaan mukaan muutokseen.

Willisin ja Edwardsin (2014, 307) mukaan tutkimus jakautuu neljään eri vaiheeseen: tutkimukselle luodaan suunnitelma, kerätään havaintoja ja kokemuksia, työstetään toiminnallista prosessia ja reflektoidaan aikaisempien vaiheiden pohjalta tutkimuskysymykseen. Nämä neljä vaihetta käydään tutkimuksessa useaan kertaan läpi.

Toimintatutkimusta käytettiin alun perin pääsääntöisesti pedagogisilla aloilla, ja sen jälkeen se levisi muille aloille. Informaatioteknologian puolella toimintatutkimus on vielä tällä hetkellä vähemmän käytetty tutkimusmetodi (Petersen ym. 2014). Petersenin et al. (2014) mukaan on tarvetta tutkimuksille, joissa relevanssia tarkastellaan käytännön näkökulmasta, mikä osoittaa miksi on aiheellista hyödyntää toimintatutkimusta tietojenkäsittelytieteissä.

4.2 Toimintatutkimuksen vaiheet

Edelliseen luvun toimintatutkimuksen perinteiseen malliin viitaten tämä tutkimus poikkeaa tavallisesta toimintatutkimuksesta siten, että tutkimuksessa käydään sykli ainoastaan kerran läpi. Ensimmäisen syklin jälkeen tarkastellaan toiminnan seurauksia sen sijasta, että reflektoidaan muutoksia aina uutta sykliä varten. Tutkimuksen vaiheet etenevät kuvion 2 kuvalla tavalla. Tutkimuksessa raportoidaan pro gradu -tutkielman puitteissa ainoastaan ly-



Kuvio 2. Tutkimuksen vaiheet kaaviona

hyen aikavälin vaikutuksia organisaatioon ja sen sisällä tapahtuvaan sosiaaliseen muutokseen. Hodgkinsonin (1957) mukaan käytöksen muutos on vaikeaa ilman, että samalla muutetaan asenteita ja arvoja ja tämä osoittaa sen, minkä takia sosiaalisen muutoksen arvioiminen on tärkeää tutkimuksen tavoitteiden kannalta.

Toimintatutkimuksen tutkijan roolissa toteutan intervention, jolla edesautetaan organisaation tavoitetta hyödyntää testiautomaatiota regressiotestauksessa sekä toimin tutkimuksen aikana osana organisaatiota. Interventiossa hyödynnetään aihepiiristä jo valmiiksi löytyvää kirjallisuutta siltä osin, miten ne tukevat interventiota ja tutkimuksen tavoitteita. Tarkoituksena ei ole antaa kirjallisuuden määrätä tutkimuksen suuntaa, vaan enemmänkin tukea toimintaa (Case ja Light 2011). Toimintatutkimuksessa tutkija toimii joko osana sisäpiiriä jota tutkimus koskettaa tai heidän kanssaan (Herr ja Anderson 2014). Tarkoituksena on, että muutos tapahtuu joko ympäristössä, jossa muutokset tapahtuvat tai tutkimukseen osallistuvissa ihmisissä ja tutkijassa itsessään (Herr ja Anderson 2014).

Toimintatutkimukselle tyypillisesti halutaan selvittää, miten tilanne on muuttunut intervention jälkeen aikaisempaan verrattuna. Tämän takia tutkimuksessa järjestetään sekä alku- ja loppuhaastattelu, joissa haastateltavien kanssa keskustellaan regressiotestauksesta. Haastateltavat koostuvat organisaation jäsenistä, joita toimintatutkimuksen interventio koskettaa. Tällaisia henkilöitä ovat testaajat ja ohjelmistokehittäjät. Kaikki osapuolet ovat tasa-arvoisia tutkimuksen näkökulmasta ja osallistetaan mukaan sen vaiheissa (Zuber-Skerritt 2003).

Tutkimuksen kannalta halutaan kartoittaa, millaisia vaikutuksia testiautomaatiolla on manuaaliseen regressiotestaamiseen. Tämän selvittämiseksi haastatteluissa käytetään haastattelukysymyksiä, jotka perustuvat tutkimuksen teemoihin. Haastattelukysymykset pyörivät kol-

men eri teeman ympärillä: miten haastateltavat käsittävät itse regressiotestauksen määritteen, miten he käyttävät organisaation resursseja testaamiseen sekä miten tyytyväisiä he ovat sen hetkisiin regressiotestauskäytäntöihin organisaatiossa. Toimintatutkimukselle on tyypillistä antaa osallistujille mahdollisuus kehittää omia totuuksiaan siitä, mitä on heidän tilanteessaan tapahtumassa (Edwards ja Willis 2014). Tästä syystä haastatteluissa käytetyt kysymykset ovat tyypiltään avoimia ja antavat osallistujille mahdollisuuden tuoda omia näkemyksiään esille.

Haastattelut analysoidaan temaattisen analyysin avulla (Attride-Stirling 2001). Analysointi on aineistolähtöistä, sillä tutkimuksessa halutaan selvittää, mitä haastateltavat itse ovat mieltä organisaation tilanteesta. Tutkimuksen kannalta on kiinnostavampaa se, mitä haastateltavat sanovat kuin esimerkiksi puheen tapa. Attride-Stirlingin (2001) mukaan temaattiset verkot pyrkivät aineistossa esiintyvien ristiriitaisten määritelmien sovittamisen sijasta ymmärtämään tutkittavaa ongelmaa tai ajatuksen tärkeyttä. Analysoinnissa käytetään temaattisia verkkoja analyysin tukena. Temaattinen verkkojen hyödyntäminen kvalitatiivisen tutkimuksen apuna jakautuu Attride-Stirlingin (2001) mukaan kolmeen vaiheeseen: aineisto hajotetaan osiin, tutkitaan litteroitua tekstiä ja tulkitaan sen sisältöä. Temaattisessa analyysissä aluksi aineisto jaetaan eri koodeihin sen perusteella, mistä aihepiireistä aineistossa puhutaan (Attride-Stirling 2001). Esimerkiksi tässä työssä koodi "aika" sisälsi lauseita kuten "Kyl siihen työaikaan sit kuuluu mutta se, se on tietysti se.". Samalla lauseella voi olla myös useampi koodi. Lopulta koodit rajattiin niihin, jotka koskettivat suoraan tutkimuksen aihepiiriä ja nousivat aihepiirinä toistuvasti esille haastatteluissa. Aineistosta valikoitiin teemat aikaisemmin rajattujen koodien pohjalta ja ne ryhmiteltiin kategorioihin, joita Attride-Stirlingin (2001) mukaan on kolmella eri tasolla: perusteemat, ryhmittelyteemat ja globaalit teemat. Aineistosta tunnistettiin ensin perusteemat, joiden pohjalta hahmotettiin ryhmittelyteemat. Tämän jälkeen ryhmittelyteemoja tarkastelemalla tunnistettiin globaalit teemat.

4.2.1 Alkuhaastattelut

Alkuhaastatteluihin osallistui yhteensä kuusi ihmistä. Haastattelut järjestettiin kahdenkeskinä tapaamisina, jossa keskustellaan avoimesti regressiotestauksesta. Haastateltavat vastasivat haastattelukysymyksiin vapaasti omien kokemusten ja subjektiivisten näkemystensä

pohjalta. Haastateltavien kanssa luotiin keskustelun kautta käsitys siitä, mitä regressiotestauksella tarkoitetaan ja miten organisaatiossa pitäisi toteuttaa regressiotestausta. Keskustelut tallennettiin ääninauhoitteina.

Tutkittavista kerättiin haastattelujen aikana mahdollisimman vähän henkilötietoa ja vältettiin haastattelukysymyksiä, joissa henkilötietoja tulee esille. Haastatteluaineisto anonymisoitiin tutkimuksen päätteeksi eikä haastateltaviin voida tutkimuksen jälkeen liittää uutta tietoa.

Alkuhaastattelussa kerättyjä tietoja hyödynnettiin interventiossa. Ennen interventiota alkuhaastatteluissa kartoitettiin nykytilanne organisaatiossa ennen varsinaisten muutosten tekemistä.

4.2.2 Loppuhaastattelut

Loppuhaastatteluihin osallistui viisi ihmistä, joista neljä oli aikaisemmin osallistunut alkuhaastatteluihin. Loppuhaastattelut järjestettiin kahden keskinäisenä tapaamisina samanlaisissa tiloissa kuin alkuhaastattelujen aikana.

Haastattelukysymykset koostuivat samoista teemoista kuin alkuhaastattelussa. Haastattelujen vastauksia verrattiin keskenään todetakseen, miten interventio on vaikuttanut organisaation sisällä ja pohtia, onko muutos todella tapahtunut.

Loppuhaastattelut järjestettiin noin kuukausi sen jälkeen, kun intervention ensimmäinen työpaja oli pidetty ja sillä oli selvästi vaikutusta loppuhaastatteluissa, sillä osa haastateltavista toi työpajat ja yrityksen muutokset keskustelussa oma-aloitteisesti esille. Alkuhaastatteluihin verrattuna loppuhaastattelussa haastateltavien kanssa keskustelu oli paljon suoraviivaisempaa. Haastattelukysymykset herättivät selvästi vähemmän pohdiskelua kuin alkuhaastatteluissa, mutta toisaalta haastateltavat perustivat vastauksensa viime aikoina kertyneisiin kokemuksiin, mikä oli positiivista tutkimuksen muutoksen arvioinnin kannalta.

4.3 Tutkimuksen validiteetti

Haastateltavilla oli haastattelun aikana mahdollisuus tuoda vapaasti omia mielipiteitään ja näkökulmiaan esille, mutta joihinkin aihepiireihin oli todennäköisesti vaikuttanut se, että

haastateltavia oli ennen haastattelua jo tiedotettu organisaatiossa tapahtuvasta interventtiosta. On hyvin mahdollista, että tällä on ollut oma vaikutuksensa keskustelun kulkuun vaikka alkuhaastattelujen aikana haastattelijat ei maininnut interventtiosta haastateltaville. Tällaisia viitteitä tuli esille muutama otteeseen keskusteluissa:

Haastattelijat: Voitko vielä tarkentaa mitä tarkoitat kun sanoit, että saahaan testausta käyntiin?

Haastateltava 3: No esimerkiksi sitä, että tässä kun joku vaikka saa gradunsa aikaseksi niin sitten mahdollisesti tästä niinku tulee jotakin muutoksia tähän, näihin käytäntöihin tai jotain.

Tutkimuksen aineisto koostuu ihmisten haastatteluista, joten aineistossa liikutaan haastateltavien omassa kokemusmaailmassa ja kuunnellaan heidän näkemyksiään. Aineiston sisältö on siis hyvin subjektiivista, eikä sitä voida mitata kvantitatiivisilla menetelmillä. Kvalen (1994) mukaan aineistot, jotka perustuvat haastatteluihin eivät ole luotettavia, koska haastateltavien vastaukset riippuvat esitetystä kysymyksestä. Toisaalta tästä syystä haastattelussa on tietoisesti annettu haastateltavien kertoa vapaasti heidän ajatuksiaan ja keskustelun suunta on määrätynyt sen pohjalta, mitä haastateltava on tuonut esille tai kokenut tärkeäksi.

Kvantitatiivisten menetelmien käyttö on muutenkin vaikeaa silloin, kun puhutaan regressiotestauksesta ja halutaan arvioida automaattisten testien vaikutusta ohjelmistokehitykseen. Regressiotestauksen testejä ei voida arvioida samoilla mittareilla kuin muussa testaamisessa, sillä tarkoitus on varmistaa ettei uusia regressioita synny ja sivuvaikutukset ovat huomioitu ohjelmistokehityksessä (Homès 2012).

Alkuhaastattelun kysymykset olivat suunniteltu kysyttäväksi testaajilta ja ohjelmistokehittäjiltä eikä muokkauksia haastattelukysymyksiin tehty haastateltavan toimenkuvan perusteella. Jälkikäteen ajateltuna olisin toivonut, että kysymykset olisi paremmin osoitettu erikseen kehittäjille ja testaajille yhteisten kysymysten sijasta, sillä testaajat hämmetyivät alun perin kehittäjiä varten suunnitelluista kysymyksistä ja toisinpäin. Kohderyhmät huomioitiinkin alkuhaastattelua paremmin loppuhaastatteluissa, mutta haastatteluteemat pysyivät samoina.

Loppuhaastattelujen aikana moni haastateltavista antoi kysymyksiin hyvin suoria vastauksia, joka näkyy myös loppuhaastattelujen aineiston analysoinnissa. Loppuhaastatteluissa haasta-

teltavat pääsivät kertomaan intervention kautta kertyneistä kokemuksista ja ajatuksista joiden kertominen ei välttämättä vaadi syvällisempää pohdiskelua. Lopputuloksen arvioinnin kannalta oli hyvä, että haastateltavien vastaukset olivat selkeästi tulkittavissa.

Toimintatutkimuksen käytännönläheisyys herättää Hodgkinsonin (1957) mukaan huolen siitä, voiko toimintatutkimusta pitää tieteellisenä tutkimuksena. Lewinin (1946) päinvastoin mainitsee, että tutkimus, joka ei tuota mitään kirjallisuuden lisäksi ei itsessään riitä. Toimintatutkimukseen liittyen nousee herkästi esille kritiikki tutkimuksen validiteetista sen perinteisessä merkityksessä etenkin, jos tutkimuksessa käytetään narratiivista otetta (Heikkinen, Huttunen ja Syrjälä 2007). Tutkielmassa ei raportoida pelkästään itse tutkijan kokemuksia, vaan myös tutkimukseen osallistuneiden haastateltavien ajatuksia ja mielipiteitä. Narratiivisesti raportoitujen tutkimuksien validiteettia ei voida arvioida kvantitatiivisten tutkimusten perusteiden mukaisesti, joten Heikkinen, Huttunen ja Syrjälä (2007) ehdottavat artikkelissaan toisenlaista tapaa arvioida kvalitatiivisia tutkimuksia, niin kutsutun mielikuvia herättävän periaatteen mukaan. Sen perusteella tutkimuksen narratiivista kerrontaa voidaan arvioida evokatiivisuuden eli sen mukaan, kuinka paljon mielikuvia tai tunteita tutkimuksen teemaa kohtaan herää (Heikkinen, Huttunen ja Syrjälä 2007).

Toimintatutkimuksen pyrkimyksenä on ratkaista lokaaleja ongelmia yleistettävien totuuksien löytämisen sijasta (McKay ja Marshall 2000). Vaikka tutkimuksen tapahtumat ja tulokset ovat sidottuna tutkimusympäristöön ja sen olosuhteisiin, voidaan McKayn ja Marshallin (2000) mukaan parin tekijän perusteella miettiä, miten toimintatutkimuksen tuloksia voidaan hyödyntää esimerkiksi muissa muutosta suunnittelevissa organisaatioissa. Ensimmäinen on se, miten toiminnan seurausten pohdinnassa on käytetty teoriaa tulosten selittämisen ja tulkinnan kannalta. Tällöin muut voivat päättää, voidaanko tutkimuksen tuloksia soveltaa muissa asetelmissa, ihmisissä ja interventioissa (McKay ja Marshall 2000). Interventiossa teoria toimii toteutuksen tukena ja tutkimuksessa pohditaan toiminnan tuloksia olemassa olevan kirjallisuuden avulla. Päätöksen tekemistä edesauttaa myös se, kuinka tarkasti tutkimuksen asetelmaa, prosessia ja tuloksia on kuvattu lukijalle (McKay ja Marshall 2000). Luvussa 4.2 kuvattiinkin tutkimuksen toteutusta ja vaiheita. Näiden lisäksi interventiota, tuloksia ja pohdintoja käydään läpi omissa luvuissaan. Interventiota käsittelevässä luvussa kuvataan myös tutkijan omakohtaisia kokemuksia intervention toteutuksen lisäksi. Toisena tekijänä voidaan

pitää toimintatutkimuksessa käytyjen syklien määrää. Mitä enemmän syklejä käydään toimintatutkimuksessa läpi, sitä suuremmalla todennäköisyydellä voidaan luottaa tutkimuksen löydösten siirrettävyyteen (McKay ja Marshall 2000). Tässä tutkimuksessa kuitenkin raportoidaan työmäärän puitteissa vain yhden toimintatutkimussyklin tuloksia.

4.4 Tutkimuksen eettisyys

Moneen muuhun tutkimusmetodologiaan verrattuna toimintatutkimus on hyvin riippuvainen muista ihmisistä ja yhteisöstä, joiden puissa toimitaan (McNiff ja Whitehead 2001). Saman toteaa myös Somekh (2005), jonka mukaan organisaatiossa ihmisten käyttäytymiseen vaikuttaa heidän persoonallisuuden lisäksi myös ympäristö, jossa he toimivat. Tutkimuksen ihmislähtöisyyden takia on tärkeää pohtia sen eettisiä vaikutuksia organisaatiolle, jossa toimitaan (McNiff ja Whitehead 2001).

Ihmisen tekemän työn automatisointi ei ole yhteiskunnallisesti uusi asia. Nykypäivänä monet järjestelmät ovat automatisoituja ja ihmisten tarve tehdä manuaalista työtä on vähentynyt. Sheridan ja Parasuramanin (2005) mukaan automaatio ei kuitenkaan tarkoita ihmisten korvaamista. Työpaikkojen viemisen sijasta automaation käyttö muuttaa ihmisen roolia, mutta myös tavalla, jota ei ole osattu ennustaa (Parasuraman ja Manzey 2010).

Automaation hyödyntäminen ei tarkoita sitä, että sen käyttö on virheetöntä tai automatisoidut prosessit eivät voisi lakata toimimasta. Parasuraman ja Manzey (2010) tuovat myös esille, että automaation toimivuus määrittää sen, kuinka paljon ihminen pystyy luottamaan automaatioon. Vaikka automatisoidut prosessit toimisivat yleensä ongelmitta, on hyvä huomioida realiteetin asettamat rajat. Automaation käytön puoleellisuus voi olla myös haitallista, sillä se vaikuttaa myös niin yksittäisten ihmisten kuin tiimien päätöksentekoon (Parasuraman ja Manzey 2010).

Toimintatutkimus on käytännön tutkimusta, jossa pyritään vastaamaan reaali maailmassa ilmenevään ongelmaan. Tällä halutaan erityisesti painottaa muutosta, jolla parannetaan olosuhteita (Melrose 2001). Tutkimuksessa käytetään testiautomaatiota keinona lähtötilanteen parantamiseen, sillä se on organisaatiolähtöisesti haluttua muutosta eikä vain tutkijasta johtuvaa. Lähtötilanne olisi eri, jos organisaatio ei olisi osana muutosta vaan enemmänkin sen

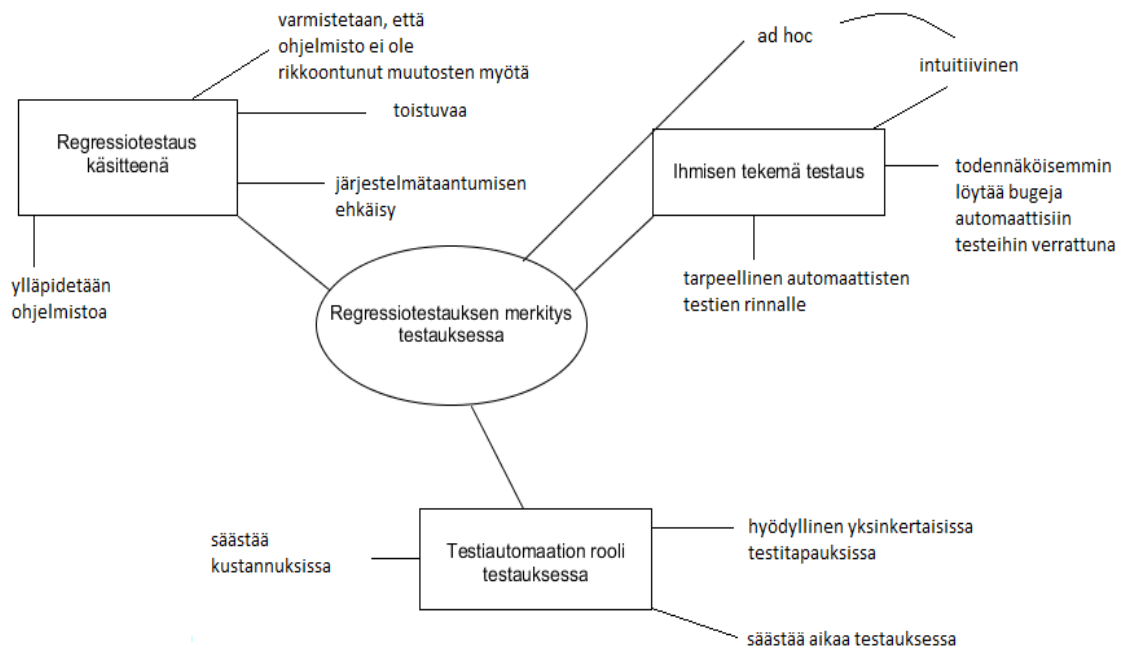
kohteena ilman vaikutusvaltaa, jolloin muutoksen haittavaikutusten pohtiminen olisi perusteltua muutosten kannalta. Tutkimuksessa haastatteluiden perusteella automaatiota ei nähty uhkakuvana tai vastustusta testiautomaatiota kohtaan ei tunnistettu.

5 Alkuhaastattelun tulokset

Alkuhaastattelun aineiston analyysissä tunnistettiin teemat rajatun tekstin pohjalta. Aineistosta löytyi yhteensä kolme globaalia teemaa: regressiotestauksen merkitys testauksessa, testauksen rajoitteet sekä tarve muutokselle. Ensimmäisenä mainitulle teemalle löytyi kolme ryhmittelyteemaa, ja kahdelle jälkimmäiselle globaalille teemalle nousi kaksi ryhmittelyteemaa esiin.

Aluksi käydään läpi regressiotestauksen merkitystä organisaation testauskäytännöissä kuvion 3 mukaisesti. Sen jälkeen käsitellään testauksen rajoitteita, kuten kuvio 4 hahmottaa temaattisena verkkona. Lopuksi avataan muutoksen tarvetta, jota kuvio 5 esittelee.

5.1 Regressiotestauksen merkitys testauksessa



Kuvio 3. Temaattinen verkko regressiotestauksen merkitykselle

Kuvio 3 kuvaa regressiotestauksen merkitystä testauksessa temaattisena verkkona. Seuraa-

vaksi käydään globaalille teemalle löytyneet ryhmittelyteemat läpi, joita on yhteensä kolme.

5.1.1 Regressiotestaus käsitteenä

Haastateltavien mukaan regressiotestauksella tarkoitetaan testausta, jolla pyritään varmistamaan, että testattavan ohjelmiston toiminnallisuus ei ole rikkoontunut muutosten myötä:

Haastateltava 3: [...] Jos nyt puhutaan ohjelmasta, eli jos ohjelmaan tehdään jotain muutoksia, niin sitten regressiotestissä testataan, että ne muutokset ei ole vaikuttanut mihinkään aikasempaan tai siis, niin, aiempiin toiminnallisiin ohjelmassa.

Haastattelussa esiin tulleet näkemykset regressiotestauksen määritelmästä olivat melko yhtenäiset suhteessa tutkimuksen alkuvaiheessa pohdittuihin näkemyksiin siitä, mitä regressiotestaus on. Poikkeuksena oli yksi haastateltavista, joka sanoi assosioivansa regressiotestauksen automaattisena testauksena ja järjestelmätaantumisen välttämisenä:

Haastateltava 6: No mä aattelen regressiotestauksella sitä, että se on testausta jolla pyritään välttämään järjestelmätaantumista. Varmaankin myös automaattista testaamista, ei välttämättä mutta aattelen kuitenkin sen enemmän automaattisena testauksena.

Regressiotestauksesta puhuttaesta tuli myös esille, miten regressiotestauksella varmistetaan testattavan ohjelmiston ylläpidettävyyttä:

Haastateltava 4: Joo, no siis sillä varmistetaan tai ylläpidetään asioita. Se on minusta aika lailla ylläpitojuttu. [...] Ja lähinnä tulee vaan silloin jos kehitetään uutta että tavallaan sillä varmistetaan että asiat ei mene rikki että, että niin.

Regressiotestausta sovelletaankin monesti ohjelmistokehitysvaiheen lisäksi ylläpitovaiheessa, joten haastateltavan näkemys on perusteltua.

Regressiotestauksessa testitapaukset suoritetaan toistuvasti uudelleen läpi, joten testauksen toistettavuus tuli odotetusti esille haastattelujen aikana:

Haastateltava 1: Mm, no sitten niitä toistetaan niinku useamman kerran ja [...] että tämän uudelleen testaan sitä että miten tää nyt menee ja sit kirjaan ittelleni ylös et aa miten tää nyt menikään.

Etenkin testauksen toistettavuus on yksi vahvimista perusteista regressiotestauksen automatisoinnille, sillä testaajien työaika voisi hyödyntää niiden testitapausten testaamiseen, joita ei ole aikaisemmin huomioitu (Berner, Weber ja Keller 2005).

5.1.2 Ihmisen tekemä testaus

Manuaalisesta regressiotestauksessa puhuttaessa tuli monesti esille ihmisen tekemän testauksen intuitiivisuus automaattisiin testeihin verrattuna:

Haastateltava 1: [...] Niin en tiedä onkse sama siinä automatisoinnissa sitten että löytääkse sit niitä mitä manuaalisesti löytyisi.

Ihmisen tekemän testauksen intuitiivisuuden yhteydessä tulikin monesti esille samalla ad hoc tyylinen testaus. Haastateltavien mielestä regressiotestauksen ohella on olennaista tehdä ad hoc testausta, vaikkei se suoraan regressiotestaukseen liittyisi. Ad hoc testauksella viitataan testaamiseen, jossa yritetään paikantaa virheitä testaussuunnitelman ulkopuolelta. Ad hoc testaaminen korostaa testaajan luovuutta keksiä uusia tapoja testata samaa ohjelmistoa (Agruss ja Johnson 2000). Moni haastateltavista sanoikin hyödyntävänsä ad hoc testausta regressiotestauksen lomassa:

Haastateltava 1: Joo. Kuuluu ehtottomasti että poikkeaa välillä sieltä ja et, tota, useemmiten ne löydökset ovatkin sen ulkopuolelta. Elikkä harvoin löytyy siitä et jos seuraa ainoastaan sitä [testaussuunnitelmaa], näin tarkasti. [...] On hyvä poiketa välillä. Ja joka kertahan se on vähän erilaista poikkeamista, emmä joka kerta niiku poikea siihen johonkin tiettyyn mihin mä viime kerralla poikkesin vaan sit se on vähän niinkun semmosta vähän filisjutuakin että et mihin.

Agrussin ja Johnsonin (2000) mukaan ad hoc testaaminen ja regressiotestaaminen tukevat toisiaan, sillä regressiotestaamisessa virheen löytyessä tarvitaan ad hoc metodeita ongelman

analysoimiseen ja virhetilanteen toistamiseen. Toisaalta yksi ad hoc testaamisen haasteista on se, että ohjelmistovirheiden löytäminen vaatii testaajalta ohjelmiston tuntemista.

Haastateltavat näkivät ihmisen tekemän testauksen tarpeellisena, mutta eri tarkoitukseen kuin mihin automaattisia käyttöliittymätestejä tarvitsisi. Haastateltavat näkivät epätodennäköisenä, että ihmisen tekemä testaus muuttuisi automaation myötä tarpeettomaksi:

Haastateltava 6: Mitä enemmän automaattisia testejä niin sitä varmempi voi olla siitä et se järjestelmä toimii ilman manuaalisia testaamista. Kokonaan se ei poista koskaan, aina täytyy testata manuaalisesti.

5.1.3 Testiautomaation rooli testauksessa

Regressiotestauksen yhteydessä testiautomaation esiin tuominen ei ole sinänsä uusi asia. Luvussa 3 oli esitelty lähteitä, jotka pitävät testiautomaatiota itsestäänselvyytenä, kun puhutaan regressiotestauksesta. Regressiotestauksen testitapausten toistettavuus soveltuu luonteeltaan automatisoitavaksi.

Haastateltavien kanssa keskustellessa nousi esiin se, että haastateltavat eivät oleta automaattisten testien kattavan kovin monimutkaisia testitapauksia:

Haastateltava 2: On ja siis mun mielestä se on kun sehän riippuu siitä että mitä se on tavallaan käsketty tekemään. Sen mää tiedostan että ei siinä voi hirveen monimutkaisia tehtäviä ja tavallaan että ei se voi tiedostaa sitä jos siellä on joku semmonen asia väärin mikä on esimerkiksi sillain että tavallaan näet että tuo on selkeesti väärin. Mut tota just tommostet yksinkertaiset asiat niin kyllä mää siis luotan siihen enempi kuin mitä ihmiseen.

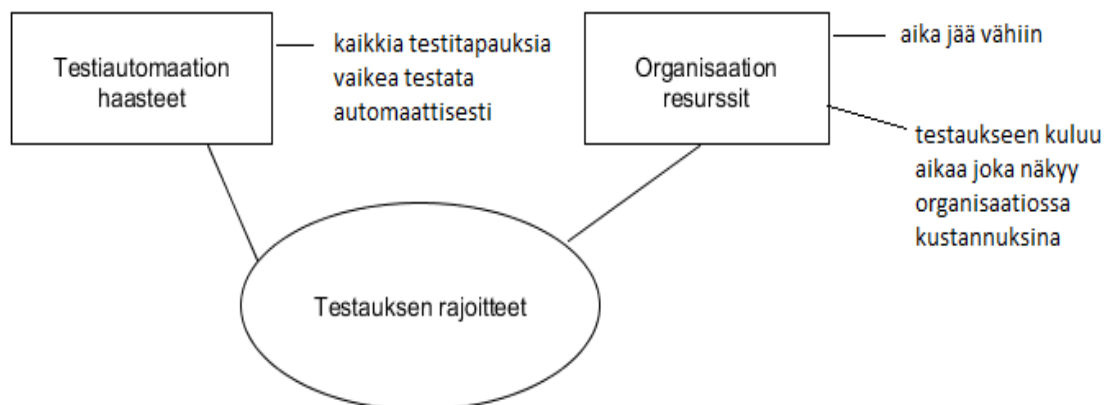
Ajankäyttö nähdään monesti haasteellisena ohjelmistokehityksessä, ja erityisesti regressiotestaukseen käytettävää aikaa ei välttämättä huomioida samalla tavalla muuhun testaamiseen verrattuna (Leung ja White 1989). Haastattelujen aikana ymmärrettävästi ajankäyttö tuli useaan otteeseen esille, ja heräsi keskustelua siitä voisiko testiautomaatio allokoida testaajien aikaa muuhun testaamiseen:

Haastateltava 1: No se tietysti helpottaa, se vie vähemmän aikaa meiltä jotka voisi käyttää sen ajan johonkin muuhun niinkun työtehtävään.

Työajan säästäminen käytännössä tarkoittaisi sitä, että organisaation resurssit vapautuisivat kehittävämpään toimintaan. Ajan lisäksi kustannukset olivat toinen aihepiiri, joka nousi keskustelussa esille, kun puhutaan resurssien säästämisestä:

Haastateltava 6: [...] [Automaattisten testien] ylläpitäminen minun käsityksen mukaan ei ole niin kallista jolloin se on paljon tuottavampaa tehdä niitä automaattisesti ja siksi niitä varmasti myös tehdään automaattisesti.

5.2 Testauksen rajoitteet



Kuvio 4. Temaattinen verkko testauksen rajoitteille

Kuvio 4 esittää testauksen rajoitteita temaattisen verkon muodossa. Testauksen rajoitteille löytyi kaksi ryhmittelyteemaa, joita seuraavat alaluvut käsittelevät.

5.2.1 Organisaation resurssit

Haastatteluissa yhtenä suurimmista haasteista testauksessa oli selvästi aika ja erityisesti sen rajallisuus, kuten aikaisemmin kävi ilmi. Kaikilla haastateltavilla nousi aika esille regressio-testauksen suurimpana haasteena. Ajankäytön tuomat haasteet myös herättivät huolta testauksesta, kuten eräs haastateltava kuvailee asian:

Haastateltava 2: No joo, tai siis monesti oon sillä tavalla että varsinkin kun se aika on rajallinen niin on just se että onkohan mää, ehinköhän mää nyt riittävän tarkalla tasolla testata jonkun asian ja että sitten se on se yleisempi huoli ehkä siinä.

Ohjelmistokehityksessä on tyypillistä, että testaukseen varattu aika on rajallinen. Etenkin teollisuudessa on normaalia, että testaukseen käytettävä aika sovitaan jo ennen kehitettävän tuotteen toteutusta (Leung ja White 1989). Leungin ja Whiten (1989) mukaan regressio-testaukseen käytettävää aikaa ei välttämättä huomioida suunnitteluvaiheessa muuhun testaamiseen verrattuna, jonka takia testaaja joutuu käytännössä kiirehtimään testausta.

Kustannusten rajoitteet näkyvät epäsuorasti ajankäytössä. Erään haastateltavan mukaan tämä voi näkyä esimerkiksi testattavien testitapausten valitsemisessa:

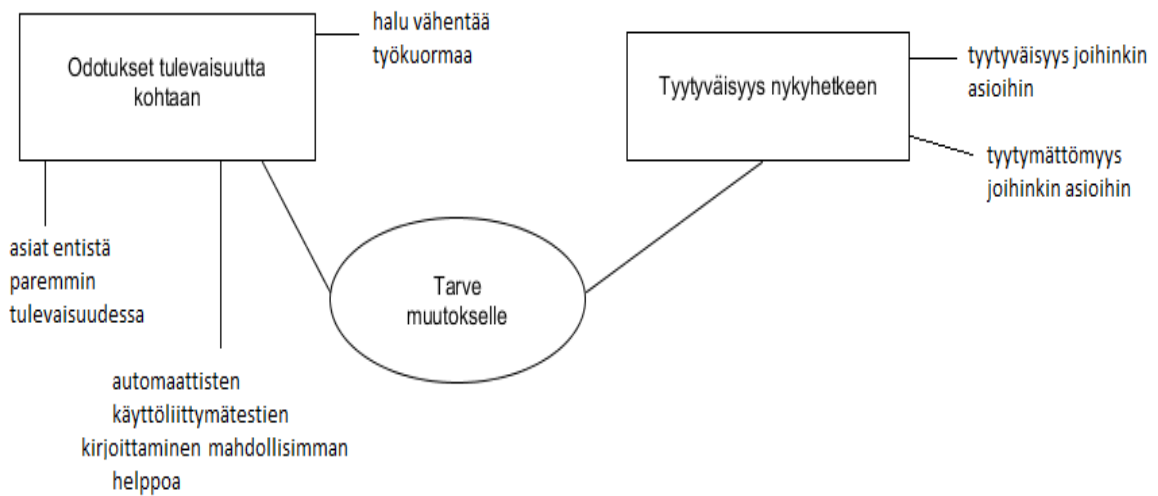
Haastateltava 6: Ja sitten jos se testitapaus ei ole hyödyllinen niin sit siinä tulee taas että me käytetään paljon rahaa ja saahan suhteellisen vähän hyötyä, eli kaikki testitapaukset ei ole minusta järkeviä, ei ole järkevää tehdä jotain mikä sitoo meidän resursseja hirveesti. Siinä tapauksessa tai sillä aatellen kaikki testit ei oo hyviä. Että sit siinä just se että sanoisin että ne testit jotka on mahdollisimman tuottavia mutta edelleen, aika vaikea se on silleen sanoa suoraan mitkä on tuottavat testit tai mitä kannattaa, se on aina, aina tulee jonkun ihmisen jonkunlaisen kokemuspohjan tai muuhun perustuen.

Kuten aikaisemmissa luvuissa oli käsitelty, myös testiautomaatio itsessään tuo omia haasteita mukanaan. Automaattisten käyttöliittymätestien kirjoittamisessa täytyy huomioida testitapauksista ne, joita on mielekästä automatisoida ja mitä ei:

Haastateltava 6: Voi olla että kaikki testit menee läpi mutta ohjelma ei käynnisty ollenkaan tai tai siinä testataan asioita jotka tai ei testata jotain sellaista ihmisen kannalta olennaista asiaa jonka ihminen näkee ensisilmäyksellä. Kaikkia ei voi automatisoida koska se on äärettömän kallista mutta paljon voidaan automatisoida.

Haastateltavan toteamus regressiotestauksen kalliudesta on perusteltua, jos kaikki testitapaukset pitää käydä läpi. Saman toteavat myös Yoo ja Harmanin (2012) tutkimuksessaan. Jos testauksessa pitää käydä ohjelmisto kattavasti läpi, testitapausten määrä kasvaa sitä mukaan, kun testattava ohjelmisto laajenee. Jos resursseja tällaiseen on rajoitetusti, pitää pohtia miten regressiotestaukseen käytettyä vaivaa saa vähennettyä (Yoo ja Harman 2012).

5.3 Tarve muutokselle



Kuvio 5. Temaattinen verkko muutoksen tarpeesta

Kuvio 5 kuvaa muutoksen tarvetta temaattisena verkkona. Muutoksen tarpeen ryhmittelyteemoja käydään läpi seuraavissa alaluvuissa, joita on yhteensä kaksi.

5.3.1 Tyytyväisyys nykyhetkeen

Kaksi kolmasosaa haastateltavista kertoivat olevansa tyytyväisiä organisaation sen hetkisiin testauskäytäntöihin, kun taas loput ilmaisivat olevansa tyytymättömiä. Käsite tyytyväisyydestä on hyvin subjektiivinen, ja haastateltavien näkemys siitä, mihin he ovat tyytyväisiä tai tyytymättömiä voivat paljon vaihdella heidän oman kokemuksen tai näkökulman perusteella. Haastateltavat kokivat, että organisaatiossa oli käytäntöjä joihin he olivat tyytyväisiä, mutta saattoi olla joitain jotka kaipaisivat vielä parantamista. Ajan rajallisuus oli yksi asia, joihin haastateltavat eivät pääosin olleet tyytyväisiä:

Haastateltava 2: No siis, mä oon oikeestaan tyytyväinen varmaankin kaikkeen muuhun paitsi just siihen että kun siihen [testaamiseen] ei ole aikaa [...].

Toisaalta tyytyväisyyttä herätti myös se, että haastateltavat olivat tietoisia organisaatiossa tapahtuvista muutoksista ja parannuksista:

Haastateltava 4: Olen tyytyväinen siihen että asiat ovat muuttumassa ja muuttuneet ja se on mikä vähän niinku pelastusta tähän tunteeseen että tuntuu että asiat muuttuvat ja on toivoa että asiat paranee [...].

Yhtä haastateltavaa lukuunottamatta kaikki haastateltavat toivoivat tästä huolimatta lisää käytäntöjä, jotka tukisivat tai helpottaisivat regressiotestaamista.

5.3.2 Odotukset tulevaisuutta kohtaan

Haastateltavista suurin osa mainitsi oma-aloitteisesti testiautomaation haastattelun aikana. Regressiotestauksen automatisoinnin avulla työkuorman vähentäminen nousi esille haastattelussa, kun oli keskustelua testiautomaation hyödyistä:

Haastateltava 2: [...] mä en nyt oleta tietenkään että se voisi kaiken mahdollisen testata mutta kaikkea mitä se voi testata niin otan innolla vastaan. Koska sit se on meiltä muilta pois.

Haastateltavien odotukset myös tulevaisuutta kohtaan nousivat haastatteluissa esille, kun haastateltavat mainitsivat intervention vaikutuksista organisaation sen hetkisiin käytäntöihin.

Tällaiset puheet jäivät kuitenkin hyvin yleiselle tasolle:

Haastateltava 4: [...] On ajatus ja tunne että asiat paranee ja muuttuu ja on ajatukset että millä tavalla ja mihin suuntaan kannattaa ja vois mennä.

Alkuhaastattelujen jälkeinen interventio koskettaa hyvin suoranaisesti ohjelmistokehittäjiä, sillä intervention jälkeen heillä on tarvittavat työkalut automaattisten testien kirjoittamiseen. Testien kirjoittamisen helppous oli toivottu piirre:

Haastattelija: Millaisia toiveita?

Haastateltava 6: Ainakin että käyttöliittymätestien tekeminen ois sen jälkeen suhteellisen helppoo, ois jollain tavalla hyvin ymmärrettäviä eli testitapauksista ei tulis hirveän monimutkaisia koska se vie aina aikaa ja niiden ylläpito vaikeutuu sitä kautta siis, helppouden myötä myös ylläpidettävämpiä. Myös silleen että se ei vaadi järjettömiä muutoksia ohjelmaan mutta ehkä kuitenkin niin että ne myös varautuu siihen että tulevaisuudessa voi tulla jotain muutoksia [...].

Käyttöliittymätestien kirjoittamisen helppous ei nouse tyypillisesti esille, kun mietitään testiautomaatiota. Se on kuitenkin tärkeää huomioida interventiossa, jos halutaan varmistaa, että testejä tulee kirjoitettua. Chenin et al. (2012) mukaan testien epäselvät yksityiskohdat tai testien kirjoittajan huono tuntemus testattavasta ohjelmistosta vaikuttavat siihen, kuinka luotettavia käyttöliittymätestejä kirjoitetaan.

6 Interventio

Alkuhaastattelut vahvistivat teoriaa siltä osin, että regressiotestauksessa testiautomaation hyödyntäminen on perusteltua ja tukee ihmisen tekemää työtä säästämällä työaika. Alkuhaastattelujen lisäksi myös olemassa oleva tutkimus tukee näkemystä, jossa testiautomaation hyödyntäminen regressiotestauksessa on aiheellista. Automaattisissa käyttöliittymätesteissä nousee monesti esille kriittisiä ohjelmistovirheitä, joita ei voisi muuten todeta yksikkötestien tai manuaalisen testaamisen kautta (Klammer ja Ramler 2017).

Organisaation käytäntöjen parantamiseksi luotiin interventio, jossa regressiotestausta automatisoitiin testiautomaation avulla. Regressiotestauksen automatisoinnilla haluttiin helpottaa organisaation jäsenten työskentelyä ja parantaa sisäisiä käytäntöjä. Tarkoituksena oli muuttaa organisaation käytäntöjä siten, että regressiotestaus tulisi suoritettua testattavalle ohjelmistolle kehittäjien toimesta automatisoitujen testien avulla ennen, kun ohjelmisto päätyy testaajille testattavaksi. Collins et al. (2012) mukaan testaaminen ei ole pelkästään testaajien vastuulla, joten ohjelmistokehittäjien tekemä testaus tukee tätä. Automaattisten regressiotestien ajaminen pitäisi ehkäistä toiminnallisuuden rikkoontumista ja mahdollistaa testaajien ajan investoimisen monimutkaisemmille testitapauksille.

Tutkimuksen interventiossa esitettiin henkilöstölle organisaation sisäisesti uusia käytänteitä ja testaustyökaluja, otettiin ne käyttöön ja työpajan kautta opeteltiin hyödyntämään automaattisia käyttöliittymätestejä. Käytännön tasolla tämä ilmeni testi-infrastruktuurin rakentamisena, jonka jälkeen organisaation jäseniä koulutettiin käyttämään työkaluja, jotka erikoistuvat käyttöliittymän regressiotestien luomiseen.

6.1 Intervention tapahtumat

Intervention toteutus jakaantui kahteen osaan: ensimmäisessä osassa pohjustin testi-infrastruktuuria automaattisille käyttöliittymätesteille, joka sisälsi valmiit testit yleisimmille käyttötapauksille. Toisessa osassa pidin kehittäjille koulutuksen siitä, miten automaattisia käyttöliittymätestejä kirjoitetaan selainpohjaiselle käyttöliittymälle. Testi-infrastruktuuri ja automaattiset käyttöliittymätestit luodaan ohjelmistolle, joka on organisaation kehittämä tuote.

6.1.1 Infrastruktuurin alustus

Käyttöliittymätestien kirjoittamisessa hyödynsin vapaan lähdekoodin kirjastoa, joka erikoistuu hyväksyntätestien kirjoittamiseen. Kirjasto hyödyntää Selenium -nimistä työkalua käyttöliittymätesteissä. Automatisoitujen käyttöliittymätestien kirjoittamiseen soveltuvan kirjaston valitsemisessa oli muutamia tekijöitä taustalla. Halusin varmistaa, että ohjelmistokehittäjille testien kirjoittaminen olisi mahdollisimman helppoa, jotta kehittäjille kynnys kirjoittaa itse automaattisia testejä olisi alhaisempi. Avoimen lähdekoodin kirjaston käyttäminen oli myös ilmaista ja säästi hieman resursseja organisaatiolta. Kirjaston pohjalta kehitin organisaatiolle testaustyökalua, joka sisältää tuen kaikista yleisimmille testitapauksille testattavassa ohjelmistossa. Tuki lisättiin sen takia, että kehittäjien ei tarvitsisi automaattisten testien kirjoittamisessa luoda kaikkea alusta asti, vaan he pystyvät keskittymään itse testin kirjoittamiseen.

Automaattisten testien testitapausten valitseminen ei ollut suoraviivaista. Voi olla vaikeaa tunnistaa, mitkä testitapaukset ovat uudelleen testattavia (Yoo ja Harman 2012). Sama pätee myös vanhentuneisiin testitapauksiin. Käyttöliittymätestauksen automatisoinnissa pitää miettiä tarkkaan, mitä testitapauksia on kannattavaa testata, jotta testit olisivat mahdollisimman ylläpidettäviä. Testattavan ohjelmiston koodi ja rajapinnat ovat jatkuvassa muutoksessa, joka vaikuttaa vahvasti käyttöliittymän testaamiseen (Collins, Dias-Neto ja Lucena Jr. 2012). Valitsin testattavan ohjelmiston toiminnallisuuden kannalta tyypillisimmät testitapaukset, joita ei olisi mielekästä testata käsin ja ovat helposti automatisoitavissa.

Ei ole itsestäänselvää, miten luodaan luotettavia testiskriptejä ja varmistetaan, että ne ovat virheettömiä (Chen ym. 2012). Käyttöliittymää testatessa monella tekijällä on vaikutusta siihen, miksi testaus päättyy virheeseen vaikka ohjelmisto itsessään toimisi. Testiajon kaatuminen voi johtua muun muassa testissä käytettävästä työkalusta. Testeissä käytetyn työkalun tuomia ohjelmistovirheitä on vaikea välttää, ja testivirheitä analysoidessa pitää huomioida onko testitapauksessa vaiheita, joita testityökalulla on haastavaa suorittaa. Chenin et al. (2012) mukaan suurin osa virheeseen päätyvistä testeistä yleensä johtuu virheellisistä tekstiskripteistä, joten testien laatuun on syytä panostaa. Tämän vuoksi pyrin tekemään mahdollisimman laadukkaita testejä, jotta testaustulokset olisivat luotettavempia sen sijaan, että virheellisiin testitulosten tarkasteluun ja arviointiin kuluisi aikaa.

Käyttöliittymätestien ajaminen ei ole kovin käytännöllistä ajaa ohjelmistokoodin käännosten ohella (A. Holmes ja M. Kellogg 2006). Testitapausten määrän kasvaessa testien käsin ajaminen ei ole kannattavaa myöskään sen takia, että siihen kuluu aikaa jos ajettavia testitapauksia on paljon (McMahon 2009). Holmesin ja Kelloggin (2006) tutkimuksessa käyttöliittymätestien ajaminen yöllisenä ajona nähtiin hyvänä valintana, joten ajastin valmiiksi kirjoittamani testit ajamaan joka yö organisaation palvelimilla.

6.1.2 Työpajat

Sen jälkeen, kun olin kirjoittanut automaattisia testejä käyttöliittymälle, pidin työpajat ohjelmistokehittäjille. Jokaiselle työpajalle oli varattu kaksi tuntia aikaa. Työpajassa käsiteltiin tyypillisiä testitapauksia ja kuinka testejä kirjoitetaan aikaisemmin mainitun kirjaston avulla. Käyttöliittymätestien kirjoittaminen on aikaa vievää ja jossain määrin jopa turhaa, jos käyttöliittymä muuttuu ajoittain (Berner, Weber ja Keller 2005). Tästä syystä ohjeistin ohjelmistokehittäjiä keskittymään olennaisiin testitapauksiin ja pohtimaan, millaisien testitapausten testaaminen on järkevää automaattisten käyttöliittymätestien kautta.

Ensimmäinen työpaja kului pääosin teorian läpikäymiseen. Teoriaosuudessa taustoitettiin edellisessä luvussa mainitun vapaan lähdekoodin kirjaston käyttämisen periaatteita ja miten näitä sovelletaan testien kirjoittamisessa. Edellisessä luvussa mainittua työkalua esiteltiin työpajassa ja osallistujille havainnollistettiin, miten sitä voidaan käyttää testien kirjoittamiseen. Selitin työpajaan osallistuneille myös samalla, miten testejä kannattaa kirjoittaa ja annoin käytännön vinkkejä käyttöliittymätestaamiseen. Työpaja sai ohjelmistokehittäjiltä positiivisen vastaanoton. Sen aikana heräsi paljon kysymyksiä, joista suurin osa liittyi joko testien kirjoittamisen käytännön tason ongelmiin tai organisaation käytäntöihin.

Osallistujien toiveesta työpajan jälkeen pidettiin toinen työpaja, jossa jatkettiin aikaisemmasta työpajasta ja harjoiteltiin testien kirjoittamista. Aikaisemmasta työpajasta poiketen työpajassa siirryttiin suoraan käytäntöön ja testattavasta ohjelmasta valikoitiin yksinkertainen testitapaus, jolle osallistujat harjoittelivat testien kirjoittamista. Työpajassa osallistujille annettu tehtävä haluttiin pitää mahdollisimman yksinkertaisena, jotta testit ehdittäisiin kirjoittaa työpajan aikana ja osallistujille jäisi selkeä kokonaiskuva siitä, mitä kaikkea testien

kirjoittamiseen sisältyy. Työpajaan osallistuneet kehittäjät joutuivat harjoittelemaan testien kirjoittamista alusta alkaen edeltävässä työpajassa läpikäydyn teorian pohjalta. Työpajaan oli selvästi varattu liian vähän aikaa, sillä kaikki osallistujat eivät ehtineet kirjoittaa testejä loppuun työpajan aikana. Regressiotestien kirjoittaminen käytännön tasolla vaikutti kuitenkin hahmottavan osallistujille paremmin siltä osin, miten testejä olisi tarkoitus kirjoittaa ja miten se vaikuttaa ohjelmistokehitykseen. Toisen työpajan aikana keskustelu kääntyi osallistujien kesken keskusteluun ohjelmistokehityskäytänteistä, kuten miten ohjelmistokehityksessä voidaan ottaa automaattiset käyttöliittymätestit huomioon.

Kaikki osallistujat eivät päässeet osallistumaan toiseen työpajaan, joten puuttuville osallistujille pidettiin yhteisesti erillinen työpaja, jotta kaikilla kehittäjillä olisi samanlaiset lähtökohdat ja taidot testien kirjoittamiseen. Työpajassa käytiin osaksi samaa teoriaa läpi kuin ensimmäisessä työpajassa. Aikaisemmin kehitettyä työkalua myös esiteltiin osallistujille ja selitettiin, miten testejä olisi kannattavaa kirjoittaa kuten ensimmäisessä työpajassa. Vinkkejä antaessani yritin ohjeistaa, miten joitain käyttöliittymätestauksen haasteita voidaan välttää tai kiertää käytännön tasolla. Ajan säästämiseksi kehitin käytännön osuutta varten harjoitustehtävän ennen työpajaa. Harjoitustehtävä sisälsi testit, jotka olivat pohjustettu valmiiksi samaa testitapausta varten kuin toisessa työpajassa. Työpajan käytännön osuudessa osallistujat täydensivät testit itsenäisesti ja ajoivat ne varmistaakseen, että ne menivät läpi. Edeltävään työpajaan verrattuna osallistujat saivat kirjoitettua testit nopeammin loppuun, ja jokainen osallistuja ehti kokeilla ainakin kerran toimivan testin ajamista työpajan aikana. Kyseisen työpajan aikana saamani palaute käyttöliittymätestien kirjoittamisesta oli pääosin positiivista, mutta entuudestaan vieraan kirjaston käyttämisen opetteleminen vei ymmärrettävästi hetken aikaa. Työpajassa heräsi keskustelua käyttöliittymätestien kirjoittamisen käytännön haasteista.

Omasta mielestäni oli yllättävää huomata, kuinka positiivisesti esittelemäni asiat otettiin vastaan ja selvästi herättivät kiinnostusta. Aluksi olin varautunut pitämään vain yhden työpajan, joten ensimmäisen työpajan jälkeiset tapaamiset tuli järjestettyä vähemmän suunnitellusti. Erityisesti yksi positiiviseksi koetuista asioista oli se, että osallistujien mielestä työpajoissa esitelty tapa kirjoittaa automaattisia regressiotestejä vaikutti helposti käyttöönotettavalta. Tämä oli tutkimuksen kannalta hyvin mielenkiintoinen palaute, sillä interventiossa pyrkimyksenä oli tehdä testien kirjoittamisesta mahdollisimman helppoa ohjelmistokehittäjille.

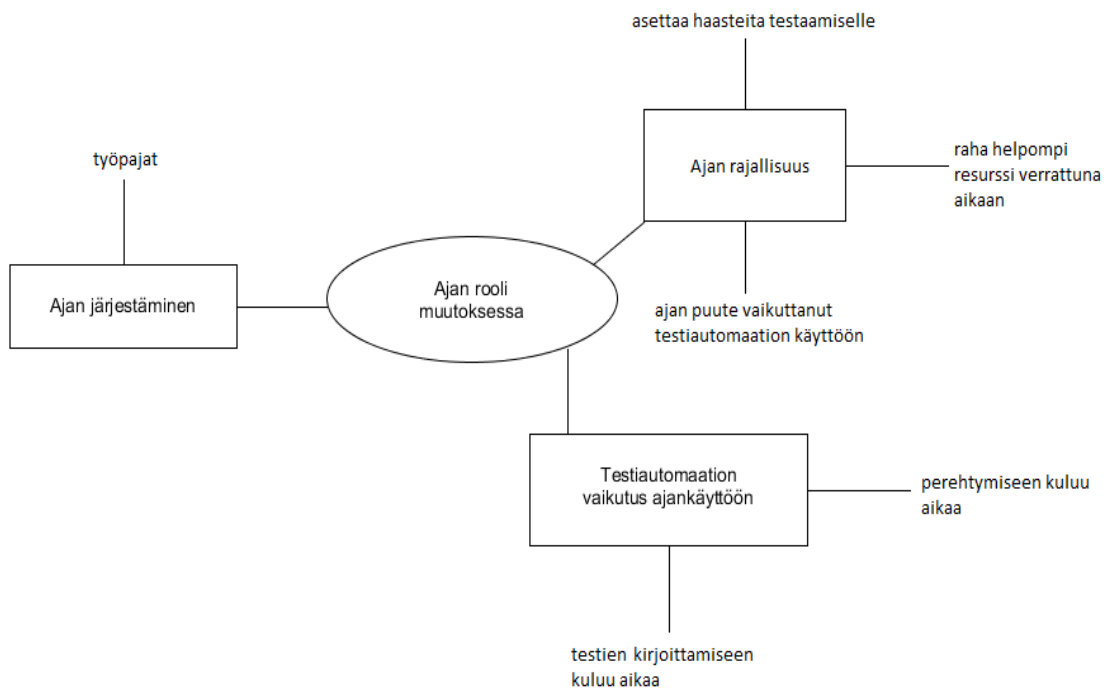
Työpaja herätti osallistujissa selvästi paljon ajatuksia ja keskustelua laajemmassa mittakaavassa kuin osasin alun perin odottaa. Positiivista oli myös se, että kaikilla kehittäjillä kertyi hieman kokemusta testien kirjoittamisesta. Harjoitustehtävän hyödyntäminen kolmannessa työpajassa selvästi auttoi osallistujia kirjoittamaan testit nopeammin loppuun toiseen työpajaan verrattuna. Tämän takia toivon, että olisin kehittänyt harjoitustehtävän jo aikaisemmassa vaiheessa, jotta kaikki kehittäjät olisivat ehtineet kokeilla testien laatimista onnistuneesti vähintään kerran työpajojen aikana. Toisaalta työpajojen järjestäminen useammassa osassa mahdollisti sen, että harjoitustehtävän hyötyjä pystyttiin havaitsemaan. Kolmen työpajan kokemusten pohjalta tiedetään, miten työpajoja kannattaa jatkossa järjestää.

7 Loppuhaastattelun tulokset

Loppuhaastattelussa kartoitettiin organisaation tilanne intervention jälkeen, joka auttaa hahmottamaan miten ilmapiiri ja asenteet muuttuivat alkuhaastatteluihin verrattuna. Aivan kuten alkuhaastatteluissa, loppuhaastattelujen aineistolle tehtiin temaattinen analyysi. Aineistosta oli globaaleja teemoja löytynyt yhteensä kolme: ajan rooli muutoksessa, regressiotestauksen muutokset ja näkemys tulevaisuudesta. Jokaiselle globaalille temalle löytyi kolme ryhmitelyteemaa.

Ensimmäisenä käsitellään ajan roolia muutoksessa kuvion 6 mukaisesti. Tämän jälkeen käydään läpi muutoksia regressiotestauksessa, kuten kuvio 7 esittelee. Lopuksi tarkastellaan haastateltavien näkemyksiä tulevaisuudesta, jota kuvio 8 hahmottaa.

7.1 Ajan rooli muutoksessa



Kuvio 6. Temaattinen verkko ajan roolista muutoksessa

Ajankäyttö ja sen haasteet tulivat poikkeuksetta esille jokaisessa haastattelussa. Haastattelut olivat yksimielisiä siitä, että aikaa on liian vähän käytettävissä regressiotestaamiseen. Kuvio 6 kuvaa ajan roolia muutoksessa temaattisena verkkona.

7.1.1 Ajan rajallisuus

Haastateltavien kanssa haasteista keskustellessa aika ja erityisesti sen rajallisuus tuli monesti esille:

Haastateltava 1: No haaste tuntuu olevan se aika, ajankäyttö. [...] No muuten joo, mutta siis se aika on vaan niinku rajallista että se siihen ei pysty käyttämään niin paljon aikaa ja mitä pitäis pystyä käyttämään [...].

Leungin ja Whiten (1989) tutkimus tukee tätä väitettä. Teollisuudessa monesti tehdään regressiotestausta rajallisen ajankäytön vuoksi silloin, kun se on välttämätöntä eikä aikaa jää uudelleen testaamiseen (Leung ja White 1989). Erään haastateltavan mukaan aikaa on jopa haastavampi järjestää kuin rahaa:

Haastateltava 4: Että usein siihen tarvitaan aikaa ja rahaa [...]. Tuntuu että se ehkä raha on se helpompi, aika on sitten vähemmän helpompi.

Ajan puutteen vuoksi kehittäjillä työpajojen ulkopuolella testiautomaatioon tutustuminen ja testien kirjoittaminen on jäänyt vähemmälle:

Haastattelija: Tuota tuota, onko ollut mitään muita seikkoja kuin se aika minkä takia et ole päässyt vielä kirjoittamaan?

Haastateltava 3: Ei, kyllä sitä olisi ihan hauska testata mutta ei vaan jotenkin... on vaan niin kiire kaikkien muitten kanssa. Niin se taitaa vähän aina olla, pitäis vaan jotenkin ottaa joku aika jostain.

7.1.2 Testiautomaation vaikutus ajankäyttöön

Regressiotestausta varten kehittäjille oli työpajoissa esitelty organisaatiolle kehitettyjä testaustyökaluja ja testiautomaatiota. Ymmärrettävästi uusien työkalujen käytön opetteluun on

kulunut aikaa:

Haastattelija: No, mitkä filikset sulla nousee esille tästä frameworkista tai mitkä oli filikset itse testien kirjoittamisesta?

Haastateltava 7: No se siis vaikutti tosi hyvältä. Mutta vaatis vähän enemmän aikaa perehtymiseen.

Testaustyökalujen opettelu on käytännössä tarkoittanut myös sitä, että testien kirjoittaminen on aikaa vievää:

Haastateltava 3: Testien kirjoittamisessa sinänsä ei ole mitään ongelmaa, se tuntuu vaan aikaa syövältä ongelmalta.

7.1.3 Ajan järjestäminen

Teollisuudessa aika tulee olemaan jatkossakin yksi suurimmista ongelmista, mikä tarkoittaa sitä, että aikaa pitää järjestää yhteisesti silloin, kun organisaation jäsenillä ei ole aikaa opetella testiautomaatiota itsenäisesti. Kehittäjät kokivat tutkimuksessa pidetyt työpajat hyvänä keinona järjestää aikaa testaustyökalujen käytön opetteluun:

Haastattelija: Aivan. Niin tota ei ole tullut itsenäisesti kokeiltua [testien kirjoittamista] kauheesti?

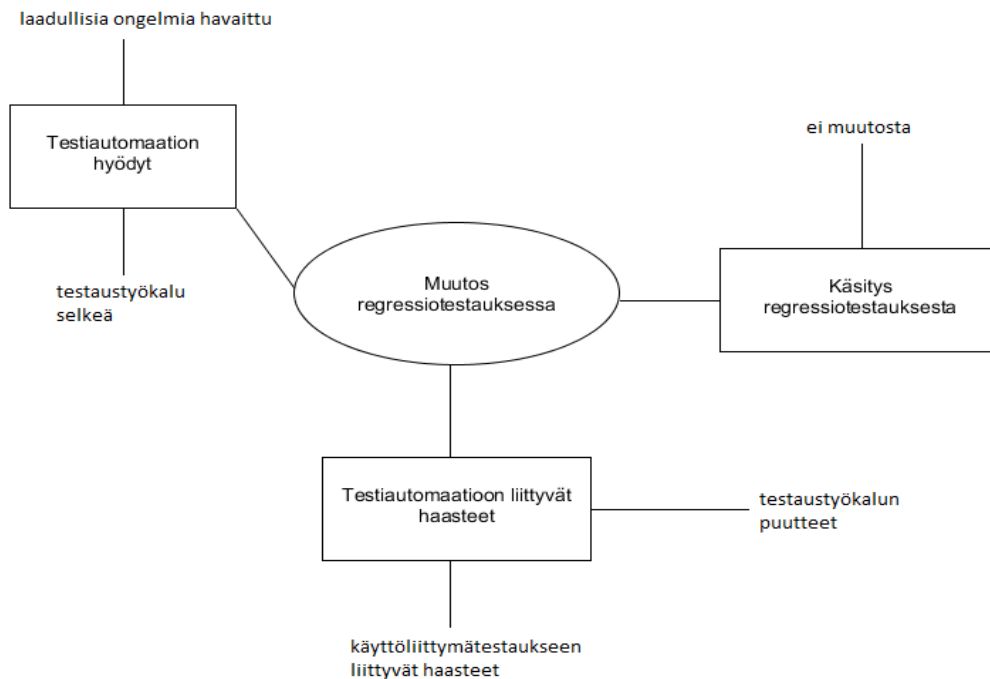
Haastateltava 3: Ei. Ei ole ollut aikaa.

Haastattelija: Aivan.

Haastateltava 3: Siksi se just oiskin ollut hyvä tommonen workshop, siinä olis sitten ollut varattuna aika.

7.2 Regressiotestauksen muutokset

Intervention tuomat muutokset kuten testi-infrastruktuuri ja automaattiset käyttöliittymäesitteit vaikuttivat lähinnä kehittäjien työhön. Erityisesti intervention myötä kehittäjille esitelty testi-infrastruktuuri ja testaustyökalu herättivät paljon keskustelua kehittäjillä. Kuvio 7 esittää regressiotestauksen muutoksia temaattisena verkkona.



Kuvio 7. Temaattinen verkko muutoksesta regressiotestaamisessa

7.2.1 Käsitys regressiotestauksesta

Kaikilla haastateltavilla käsitys regressiotestauksesta on pysynyt muuttumattomana koko tutkimuksen ajan molemmissa kohderyhmissä. Haastateltavien vastaukset regressiotestauksen käsitteen ymmärtämisestä olivat hyvin samankaltaisia:

Haastattelija: [...] miten sä käsität regressiotestauksen?

Haastateltava 5: No varmaan samalla tavalla kun ennenkin.

7.2.2 Testiautomaation hyödyt

Vaikka loppuhaastattelujen aikana oli kulunut suhteellisen vähän aikaa interventiosta, joi-takin hyötyjä pystyttiin havaitsemaan. Automaattista regressiotestaamista varten kehittyjen työkalujen ja työpajojen myötä oli huomattu testattavassa ohjelmistossa laadullisia ongel-mia:

Haastateltava 7: Noo, hyviä muutoksia. Oikeaan suuntaan. Mut mä en oo sinänsä vielä et onko mä vielä tästä automaattisesta regressiotestauksesta niin ollaanko me vielä hyödytty. Tai havaittu silleen mitään. [...] Se nyt tietysti käytettävyysongelmia ollaan silleen havaittu.

Positiivista on myös ollut se, että kehittäjät ovat kokeneet testauksessa käytettävän työkalun mielekkääksi, joka edesauttaa sitä, että he pääsevät kirjoittamaan testejä itse:

Haastateltava 4: Regressiotestauksesta, mm, no, sanoisin [...] että se työkalu mistä sää näytit meille se näyttää hyvältä, toimivalta, selkeältä. Siis sanoisin että mikä on tullut uutena se, että tuo kyseinen työkalu mielestäni näyttää todella hyvältä.

7.2.3 Testiautomaatioon liittyvät haasteet

Tutkimuksessa muutokset eivät olleet täysin ongelmattomia. Yksi perusteltu huoli automatisessa regressiotestaamisessa on se, että käyttöliittymä muuttuu ajan myötä, mikä vaikuttaa myös testaamiseen:

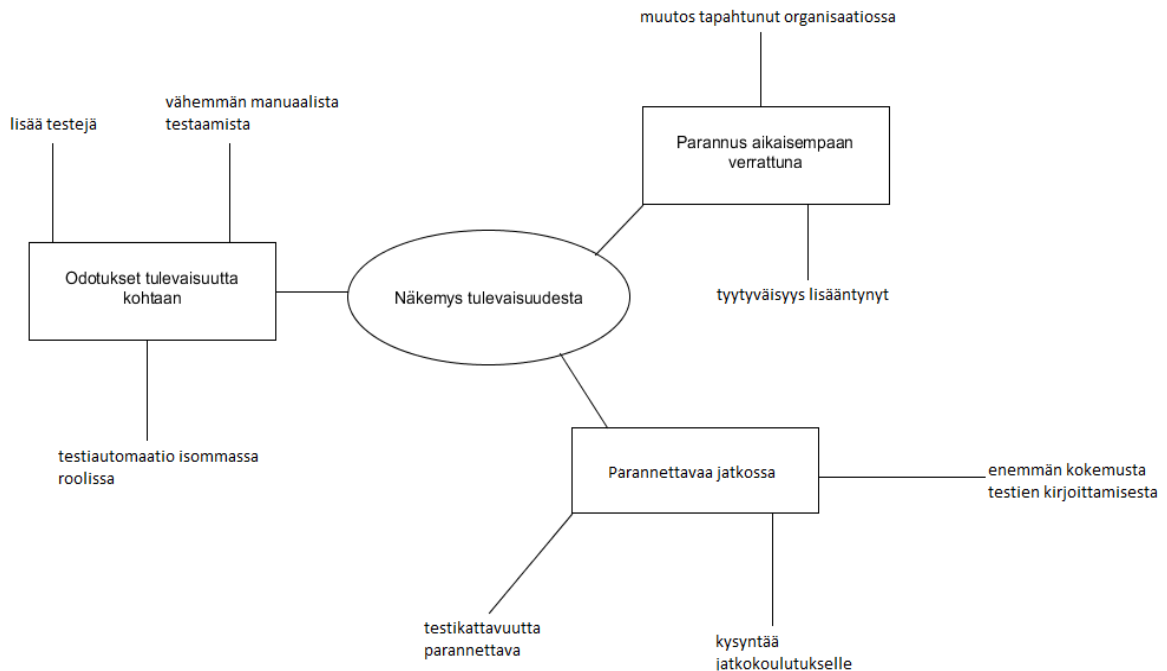
Haastateltava 3: No ne perus semmoset mitkä aina liittyy tohon UI-testaukseen, et sitten kun UI muuttuu ja [käyttöliittymäkomponenttikirjaston] päivitys tehään että kuinka hyvin ne sitten saa sen ne testit ajettua vielä siinä uudessa mutta ei muuta tule kyllä tässä mieleen.

Käyttöliittymän muutosten lisäksi myös testauksessa käytettävä kehys (engl. framework) toi omia haasteita testaamiseen:

Haastateltava 7: No siinä [kehyksessä] on hyvä pohja mistä sitten laajentaa. Tietysti kyllä siellä on jotain puutteita sitten, mutta...

Interventiossa pyrittiin vähentämään testauskehuksesta johtuvia ongelmia mahdollisimman paljon, mutta käyttöliittymätestauksessa kaikkia testausmateriaaleihin (engl. testware) liittyviä virheitä on hankala saada kiinni.

7.3 Näkemys tulevaisuudesta



Kuvio 8. Temaattinen verkko ajan tulevaisuuden näkemyksistä

Pääsääntöisesti haastateltavilla oli positiivinen näkemys tulevaisuudesta ja siitä, että testiautomaation rooli regressiotestauksessa tulee kehittymään jatkossa. Kuvio 8 kuvaa tulevaisuuden näkemystä temaattisena verkkona.

7.3.1 Parannus aikaisempaan verrattuna

Muutoksen tarve on lähtenyt itse organisaatiosta, ja tutkimuksessa intervention kautta on onnistuttu osittain vastaamaan siihen tarpeeseen. Kehittäjien näkökulmasta on ollut ratkaisevaa se, että muutosta on tapahtunut testiautomaation käyttöönoton suhteen:

Haastateltava 3: Öö, no olihan se aikakin että saahaan jotakin tommosta ihan siihen vaiheeseen että sitä pystyy jotenkin käyttämään. Sitä on niin monet vuodet vaan puhuttu että pitäis tehdä sitä ja pitäis testata tämä [...]. Nyt sen sitten on joku tehnyt.

Kun haastateltavilta kyseltiin heidän tyytyväisyyttä organisaation regressiotestauskäytäntöihin, moni heistä kertoi näkevänsä vielä tarvetta parannukselle. Toisaalta tyytyväisyyttä herätti myös se, että parannuksia on tapahtunut organisaation käytänteissä. Haastateltava toi esille sen näkökulman, että käytänteet ovat vielä alkuvaiheessa intervention jälkeen:

Haastateltava 3: Jaa, no onhan aina varmasti jotain parannettavaa mutta tuota, on tää nyt kuitenkin sillä lailla parempi että [...] niitä oikeasti kirjoitettaisiin niin onhan se hyvä alku.

7.3.2 Odotukset tulevaisuutta kohtaan

Testiautomaation käyttöönotto herätti sekä kehittäjillä että testaaajilla toiveita ja odotuksia tulevaisuutta kohtaan. Etenkin testaaajat toivoivat, että tulevaisuudessa testiautomaation myötä tulisi manuaalinen regressiotestaus helpottumaan heidän työssään:

Haastateltava 1: [...] automatisointi niin kuulostaa ihan älyttömän hyvältä ja tommoset ton tyyppiset asiat pystyis sitten testaamaan silleen automaattisesti jos tommoset regressiotestausasiat mitkä on hyvin tommosta kaavamaisia.

Kehittäjät toivoivat käyttävänsä tulevaisuudessa enemmän aikaa automaattisten käyttöliittymätestien kirjoittamiseen. Haastateltavat ilmaisivat olevansa motivoituneita hyödyntämään testiautomaatiota, kun heidän kanssaan keskusteltiin siitä, kuinka paljon he haluaisivat investoida aikaa automaattisten käyttöliittymätestien kirjoittamiseen:

Haastateltava 4: Ai siis miten paljon aikaa?

Haastattelija: Niin.

Haastateltava 4: Mä sanoisin että aika paljon. Tai siis haluaisin.

Keskusteluiden myötä kävi myös ilmi, että osa haastateltavista uskoo testiautomaation roolin muuttuvan jatkossa:

Haastattelija: Uskotko että automaattiset testit tulevat olemaan paljon merkittävämmässä roolissa jatkossa?

Haastateltava 7: Joo. Kyllä ne tulee.

7.3.3 Parannettavaa jatkossa

Muutoksen myötä tuli keskusteluissa esille myös asioita, joita tulisi kehittää jatkossa. Yksi näistä on se, että uusien asioiden opettelu vie oman aikansa, mikä hidastaa työtä alussa. Testien kirjoittaminen vaatii harjoittelua, kuten eräs haastateltavista toi keskustelussa esille:

Haastateltava 3: No siis sillä lailla että nyt kun sen tää testien kirjoittaminen ei ole ollut silleen kauhean pinnalla oleva asia tässä yrityksessä aikaisemmin, tai ainakaan semmonen mitä olis sitten varsinaisesti tehty, ehkä puhuttu. Sillä lailla ei ole semmosta kovin kattavaa kokemusta asiasta niin sitten tässä kun alkaa työkalut helpottumaan ja ohjelman rakenne paranee ja pystyy sitten tekemään niitä testejä niin sitten oppis tekemään noita testejä niin sit siinä oppis tästä ite testauksesta mitä kannattaa tehdä ja että sitten tulis sillä lailla parempaa sitä testikoodia.

Toisena tuli esille se, että kehittäjillä oli työpajojen ulkopuolella hyvin rajatusti aikaa tutustua testien kirjoittamiseen ja olisi mahdollisesti tarvetta vielä uudelle työpajalle:

Haastateltava 7: No, lähinnä se että pääsis nyt niitä ite tekemään. Jaa toivottavasti siihen nyt sitten saadaan varattua aikaa. Ja ehkä joku vielä joku workshoppi vois olla.

Haastattelija: Millaisia asioita toivoisit että käydään workshopissa läpi?

Haastateltava 7: No ehkä silleen että kaikki pääsis vauhtiin suunnittelemaan.

Kolmantena kehittämisen kohteena nähtiin se, että interventiossa alustetut testit kaipaavat täydennystä, jotta ohjelmistoa voidaan testata kattavammin:

Haastateltava 7: [...] Musta se on varsin silleen niinkun miten sen nyt sanois, rakenteellisella tasolla hyvä. Että tietysti kattavuutta voidaan lisätä.

8 Pohdinta

Tutkimuksessa raportoidaan organisaatiossa aloitetun muutoksen lyhytaikaisia vaikutuksia tiedostaen, että muutokset vievät yleensä aikaa ja näkyvät konkreettisesti vasta pidemmällä aikavälillä. Saman ovat todenneet Collins et al. (2012), joiden mukaan testiautomaation hyötyjä on haastavaa kartoittaa vielä alkuvaiheessa, kun testiautomaatio on otettu käyttöön. Toimintatutkimuksessa on tärkeämpää saada muutosta ja parannuksia reaali maailmassa aikaan kuin pyrkiä kehittämään yksittäistä teoriaa, jonka paikkaansapitävyydestä voidaan väitellä akateemisella puolella (Melrose 2001). Toimintatutkimuksen näkökulmasta tavoite toteutuikin, mutta on tärkeää pohtia, ovatko aikaan saadut muutokset vienyt kehitystä haluttuun suuntaan.

Tutkimuksessa onnistuttiin osittain vastaamaan organisaation tarpeisiin, mikä herätti selvästi tyytyväisyyttä etenkin kehittäjien näkökulmasta. Koko tutkimuksen tarkoituksena oli pyrkiä toteuttamaan muutos, joka organisaation jäsenten haastatteluiden perusteella oli aiheellinen. Siinä mielessä tutkimuksen kannalta on ollut positiivista se, että ilmapiirin muutosta on ollut havaittavissa. Kehittäjät myös kokivat testien kirjoittamiseen tarkoitettujen työkalujen selkeäksi eikä testien kirjoittamisen periaatteet tuoneet kehittäjille vaikeuksia, mikä itsessään on edesauttanut perehdyttämistä. Testaustyökalun avulla pystyttiin paikantamaan laadullisia ongelmia testattavassa ohjelmistossa, mikä parantaa testattavan ohjelmiston laatua ja ehkäisee ohjelmistovirheitä syntymästä.

Muutosta organisaation ilmapiirissä oli havaittavissa pääsääntöisesti kehittäjillä, sillä muutokset kohdistuivat suoraan heidän työhön. Testaajilla varsinaista muutosta ei vielä tutkimuksen raportoinnissa ajassa ehditty havaitsemaan, mutta sitäkin suuremmalla syyllä korostuivat odotukset tulevaisuutta kohtaan. Testaajien kannalta on tärkeää myös se, miten muutokset tulevat heijastumaan heidän työhön. Sen lisäksi, että toimintatutkimuksen prosessissa ratkaistaan ongelmia voidaan tutkimuksen tietoa arvioida myös sen perusteella, kuinka tutkimukseen osallistuvien ihmisten kontrolli omaan tilanteeseen on kasvanut (Melrose 2001).

Uuden teknologian käyttöönotto voi organisaatiossa aiheuttaa vastustusta organisaat-

tion jäsenissä. Tällöin voidaan huomata, että muutoksessa on toimittu väärin ja tilanne pitäisi oikaista sen sijaan, että vastustus nähdään ylitsepäästäväenä ongelmana (Markus 1983). Tutkimuksen aikana ei tunnistettu organisaatiossa resistanssia interventiota ja sen tuomia muutoksia kohtaan. Toisaalta tutkimuksessa ilmeni piirteitä, jotka Markuksen (1983) teorian mukaan ovat suositeltavia käytäntöjä vastustuksen vähentämiseksi käyttöönotossa: haastateltavat olivat osallisena ja pystyivät vaikuttamaan muutokseen, sekä kohderyhmä, jota muutos koski, opetteli käyttämään uutta teknologiaa. Edellä mainittujen tekijöiden lisäksi kaikista tärkein huomio on se, että tutkimuksessa tehty muutos oli organisaatiolähtöistä ja siihen osallistuneet organisaation jäsenet olivat motivoituneita muutosta kohtaan. Toimintatutkimuksen kautta muutoksen tekeminen on ainakin tämän tutkimuksen valossa todettu onnistuneeksi tavaksi saada muutosta aikaan organisaation sisäisesti. Tästä havainnosta on varmasti myös hyötyä muille organisaatioille, jotka suunnittelevat muutosta regressiotestauskäytännöissä.

Tutkimuksessa joitakin testiautomaation käyttöönoton haasteita kävi ilmi loppuhaastatteluiden myötä. Haastateltavilla oli ajan puutteen vuoksi hyvin rajatusti tilaisuuksia perehtyä automaattisten testien kirjoittamiseen intervention aikana pidettyjen työpajojen ulkopuolella. Testiautomaatioon liittyvät haasteet kehittäjillä painottuvat pääsääntöisesti siihen, ettei testien kirjoittamisesta ollut tarpeeksi kokemusta. Tästä syystä nähtiin tarpeellisena järjestää lisäkoulutusta, jotta aikaa tulisi järjestettyä muilta töiltä. Testien kirjoittamiseen kuluu oletettavasti kehittäjillä vähemmän aikaa tulevaisuudessa, kun on karttunut enemmän kokemusta automaattisten käyttöliittymätestien kirjoittamisesta. Toinen huomio oli myös se, että intervention aikana luotuja automaattisia käyttöliittymätestejä ei ollut tarpeeksi kattamaan koko ohjelmiston toiminnallisuutta, jonka takia testiautomaatiolla ei vielä testata yhtä kattavasti kuin olisi tulevaisuudessa tarkoitus. Tutkimuksen interventio toi omat muutoksensa organisaation sisäisiin käytäntöihin, jotka ovat vielä muotoutumassa kuten eräs haastateltavistakin huomautti loppuhaastattelussa:

Haastateltava 4: [...] niin se on aika lailla alkuvaiheessa vielä ja just nää [testi]ympäristöön liittyvät jutut että niin, mä sanoisin että nää käytännöt ovat muotoutumassa vielä aika merkittävästi [...].

Loppuhaastattelujen tulosten kartoittamisessa esiteltiin kolme temaattista verkkoa, jotka muodostuivat aineiston pohjalta. Näistä kaksi, jotka käsittelevät ajankäyttöä ja regressiotes-

tauksen muutosta, vaikuttavat hyvin vahvasti toisiinsa. Ajankäytön tuomat jännitteet näkyivät siinä, kuinka regressiotestauksen muutokset ilmenivät tutkimuksessa ja epäsuorasti vaikuttivat myös tulevaisuuden näkymiin. Ajan puute selvästi hidasti testiautomaation käyttöönottoa, ja mahdollisesti testiautomaation soveltaminen käytäntöön olisi edistynyt nopeammin ilman tätä rajoitetta. Tällaisessa tilanteessa käyttöönoton vaikutukset mahdollisesti olisivat voineet näkyä jo enemmän loppuhaastattelujen aikana, mutta tätä ei voida tietää varmuudella. Teollisuudessa ajan rajallisuus ei ole uusi ilmiö, mutta jos ajan järjestämiseksi ei tehdä riittäviä muutoksia, testiautomaatiosta saatavat hyödyt jää vähäiseksi tulevaisuudessa. Mahdollisena vaarana tällaisessa tilanteessa on se, että alkuinnostuksen jälkeen automatisoidun testauksen käyttö hiipuu ajan vähyydestä johtuen eikä testauskäytännöissä tai ohjelmiston laadussa havaita parannusta. Tällöin organisaatio jatkaa resurssien tuhlaamista manuaaliseen regressiotestaamiseen automaattisten käyttöliittymätestien kirjoittamiseen säästyvän työajan kustannuksella.

Pysyvän muutoksen tapahtumiseksi tarvitaan organisaatiolta asenteiden ja arvojen pysyvää muutosta. Jopa onnistuneenkin työpajan pitäminen itsessään harvoin riittää pitkäaikaisten muutosten aikaansaamiseksi (Lewin 1946). Yksittäisen ihmisen kyky vaikuttaa asioihin on hyvin rajallinen, sillä hänen toiminta on myös ympäristöstä ja olosuhteista riippuvaista. Käytöksen muuttaminen vaatii sen, että tehdään jotain myös rajoitteille, jotka estävät toimintaa tapahtumasta (Carr ja Kemmis 1983, 183). Tutkielmassa käsiteltiin ainoastaan testaajien ja kehittäjien näkemyksiä, mutta jatkon kannalta olisi hyödyllistä miettiä, miten johtoryhmä voisi vaikuttaa ajankäytön ongelmaan. Työn suunnittelussa olisi tärkeää huomioida automaattiseen regressiotestaukseen kuluva aika, jotta testiautomaatin hyödyt voidaan maksimoida. Automaattisten käyttöliittymätestien kirjoittaminen pitäisi kehittäjien työssä muuttua osaksi vakiintuneita ohjelmistokehityskäytänteitä muun testaamisen kuten yksikkötestauksen tavoin.

Intervention tekemät muutokset ovat tuoneet hyvän lähtökohdan organisaatiolle testiautomaation käyttämiseen regressiotestauksessa. Kaikkia testausmateriaaleihin (engl. testware) liittyviä ongelmia ei pystytty ehkäisemään, mutta isommat haasteet onnistuttiin välttämään. Organisaatiossa käytänteiden muuttaminen harvoin on ongelmaton alkuvaiheessa (Swanson ja Holton 2005). Testiautomaation käyttöönoton haasteet ja hyödyt ovat myös vielä sel-

vitettävä pidemällä aikavälillä, sillä tämän tutkimuksen syklin aikana ei näitä päästy kartoittamaan. Haastateltavilla oli optimistiset näkemykset ja toiveet tulevaisuudesta, joten olisi mielenkiintoista tutkimuksen kannalta kartoittaa muutoksen pitkäaikaiset vaikutukset organisaatiossa. Manuaalinen testaaminen on regressiotestauksessa jatkossakin tarpeellista, mutta ihmisen tekemällä testauksella voidaan kiinnittää enemmän huomiota testitapauksiin, jotka olisivat turhan monimutkaisia tai vaikeita toistaa automaattisilla käyttöliittymätesteillä.

9 Johtopäätökset

Käyttöliittymätestauksen automatisointi auttaa varmistamaan kehitettävän ohjelmiston laatua sekä edesauttaa ylläpitoa varmistamalla, että käyttöliittymän toiminnallisuus ei ole rikkoutunut muutosten myötä. Tutkimuksessa selvitettiin regressiotestaamisen automatisointia organisaation käytäntöjen näkökulmasta sekä tutkittiin sen sosiaalisia vaikutuksia. Tutkimuksen keskiössä on muutosten lyhytaikaiset vaikutukset. Tässä tutkimuksessa osallisena olleiden mukaan tutkimusorganisaatiossa toiveena on saada aikaan pitkäaikaisia muutoksia sisäisissä käytänteissä ja helpottaa manuaalista regressiotestausta testaamalla yksinkertaiset testitapaukset automaattisilla käyttöliittymätesteillä. Tässä tutkimuksessa esille tulleita hyötyjä voidaan käyttää pohjana automatisoidun regressiotestauksen käyttöönottoon liittyvässä keskustelussa niin tutkimusorganisaatiossa kuin myös sen ulkopuolellakin. Kartoittamalla lisää regressiotestauksen käyttöönottoon liittyviä hyötyjä voidaan pohtia, millä muutoksilla muutkin organisaatiot pystyvät helpottamaan testiautomaation hyödyntämistä regressiotestauksessa.

Tutkimuksessa testiautomaation käyttöönoton kannalta oli positiivista se, että kehittäjät kokivat intervention aikana kehitetyn testaustyökalun selkeäksi ja työkalun avulla on havaittu laadullisia ongelmia, joten testattavan ohjelmiston laatua voidaan parantaa lyhyelläkin aikavälillä käyttöönotosta. Haastateltavilla oli positiiviset odotukset tulevaisuutta kohtaan ja testiautomaation oletetaan jatkossa kasvavan isompaan rooliin. Vastaavasti ajankäyttö koettiin haasteelliseksi sekä käyttöliittymätestaukseen liittyviltä haasteilta ei kokonaan vältytty. Ajan puutteeseen voisi organisaation johtoryhmä vaikuttaa allokoimalla enemmän työaika regressiotestaamiseen. Tutkimuksessa raportoidulla aikavälillä ei vielä huomattu eroa testajien työssä, sillä käytänteiden muutokset olivat vasta alkaneet.

Jatkotutkimuksen kannalta olisi mielenkiintoista selvittää, mitä vaikutuksia regressiotestauksen automatisoinnilla on pidemmällä aikavälillä sekä miten automaattinen testaaminen heijastuu manuaaliseen testaamiseen silloin, kun uudet testauskäytännöt ovat ehtineet vakiintua organisaatiossa. Toimintatutkimuksen seuraavaa sykliä ajatellen voidaan suunnitella, miten järjestämällä tarpeeksi aikaa edesautetaan testiautomaation käyttöönottoa. Tutkimuksen tulosten pohjalta saadaan hyvä lähtökohta jatkotoimintaa varten.

Lähteet

- Adelman, Clem. 1993. "Kurt Lewin and the origins of action research". *Educational action research* 1 (1): 7–24.
- Agruss, Chris, ja Bob Johnson. 2000. "Ad Hoc Software Testing: A perspective on exploration and improvisation". Teoksessa *Florida Institute of Technology*, 68–69.
- Ammann, Paul, ja Jeff Offutt. 2016. *Introduction to software testing*. Cambridge: Cambridge University Press.
- Attride-Stirling, Jennifer. 2001. "Thematic networks: an analytic tool for qualitative research". *Qualitative research* 1 (3): 385–405.
- Balaji, S, ja M Sundararajan Murugaiyan. 2012. "Waterfall vs. V-Model vs. Agile: A comparative study on SDLC". *International Journal of Information Technology and Business Management* 2 (1): 26–30.
- Berner, Stefan, Roland Weber ja Rudolf K. Keller. 2005. "Observations and Lessons Learned from Automated Testing". Teoksessa *Proceedings of the 27th International Conference on Software Engineering*, 571–579. ICSE '05. St. Louis, MO, USA: ACM. doi:10.1145/1062455.1062556.
- Bruns, A., A. Kornstadt ja D. Wichmann. 2009. "Web Application Tests with Selenium". *IEEE Software* 26 (5): 88–91. doi:10.1109/MS.2009.144.
- Carr, Wilfred, ja Stephen Kemmis. 1983. *Becoming critical: Knowing through action research*. Victoria: Deakin University.
- Case, Jennifer M., ja Gregory Light. 2011. "Emerging Research Methodologies in Engineering Education Research". *Journal of Engineering Education* 100 (1): 186–210. doi:10.1002/j.2168-9830.2011.tb00008.x.
- Çelik, E., S. Eren, E. Çini ja Ö. Keleş. 2017. "Software test automation and a sample practice for an enterprise business software". Teoksessa *2017 International Conference on Computer Science and Engineering (UBMK)*, 141–144. doi:10.1109/UBMK.2017.8093583.

- Chen, J., M. Lin, K. Yu ja B. Shao. 2012. “When a GUI Regression Test Failed, What Should be Blamed?” Teoksessa *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, 467–470. doi:10.1109/ICST.2012.127.
- Collins, E., A. Dias-Neto ja V. F. d. Lucena Jr. 2012. “Strategies for Agile Software Testing Automation: An Industrial Experience”. Teoksessa *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, 440–445. doi:10.1109/COMPSACW.2012.84.
- Edwards, Claudia, ja Jerry W. Willis. 2014. *Action Research : Models, Methods, and Examples*. Applied Research in Education and the Social Sciences. Charlotte, NC: Information Age Publishing.
- Engström, Emelie, ja Per Runeson. 2010. “A Qualitative Survey of Regression Testing Practices”. Teoksessa *Product-Focused Software Process Improvement*, 3–16. Berlin, Heidelberg: Springer Berlin Heidelberg.
- Felderer, Michael, ja Rudolf Ramler. 2017. “Special issue on collaboration in software testing between industry and academia”. *Software Quality Journal* 25 (4): 1087–1089.
- Garousi, Vahid, ja Junji Zhi. 2013. “A survey of software testing practices in Canada”. *Journal of Systems and Software* 86 (5): 1354–1376. doi:https://doi.org/10.1016/j.jss.2012.12.051.
- Haikala, Iikka, ja Tommi Mikkonen. 2011. *Ohjelmistotuotannon käytännöt*. Helsinki: Talentum Media Oy.
- Heikkinen, Hannu L. T., Rauno Huttunen ja Leena Syrjälä. 2007. “Action research as narrative: five principles for validation”. *Educational Action Research* 15 (1): 5–19. doi:10.1080/09650790601150709.
- Herr, Kathryn, ja Gary L. Anderson. 2014. *The action research dissertation: A guide for students and faculty*. Thousand Oaks, CA: Sage publications.
- Hodgkinson, Harold L. 1957. “Action Research—A Critique”. *The Journal of Educational Sociology* 31 (4): 137–153.

- Holmes, A., ja M. Kellogg. 2006. "Automating functional tests using Selenium". Teoksessa *AGILE 2006 (AGILE'06)*, 6–275. doi:10.1109/AGILE.2006.19.
- Holmes, Antawan, ja Marc Kellogg. 2006. "Automating functional tests using selenium". Teoksessa *AGILE 2006 (AGILE'06)*, 6–pp. IEEE.
- Homès, B. 2012. *Fundamentals of software testing*. London: Hoboken.
- Jansson, Noora. 2013. "Organizational change as practice: A critical analysis". *Journal of Organizational Change Management* 26 (6): 1003–1019.
- Juristo, N., A. M. Moreno ja W. Strigel. 2006. "Guest Editors' Introduction: Software Testing Practices in Industry". *IEEE Software* 23 (4): 19–21. doi:10.1109/MS.2006.104.
- Karhu, K., T. Repo, O. Taipale ja K. Smolander. 2009. "Empirical Observations on Software Testing Automation". Teoksessa *2009 International Conference on Software Testing Verification and Validation*, 201–209. doi:10.1109/ICST.2009.16.
- Kasurinen, Jussi, Ossi Taipale ja Kari Smolander. 2010. "Software Test Automation in Practice: Empirical Observations". *Advances in Software Engineering* 2010.
- Klammer, C., ja R. Ramler. 2017. "A Journey from Manual Testing to Automated Test Generation in an Industry Project". Teoksessa *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 591–592. doi:10.1109/QRS-C.2017.108.
- Kvale, Steinar. 1994. "Ten standard Objections to Qualitative Research Interviews". *Journal of Phenomenological Psychology* 25 (2): 147–173. doi:10.1163/156916294x00016.
- Leung, H. K. N., ja L. White. 1989. "Insights into regression testing (software testing)". Teoksessa *Proceedings. Conference on Software Maintenance - 1989*, 60–69. doi:10.1109/ICSM.1989.65194.
- Lewin, Kurt. 1946. "Action Research and Minority Problems". *Journal of Social Issues* 2 (4): 34–46. doi:10.1111/j.1540-4560.1946.tb02295.x.
- Markus, M. Lynne. 1983. "Power, Politics, and MIS Implementation". *Commun. ACM* (New York, NY, USA) 26 (6): 430–444. doi:10.1145/358141.358148.

- Mathur, Sonali, ja Shaily Malik. 2010. "Advancements in the V-Model". *International Journal of Computer Applications* 1 (12): 29–34.
- McKay, Judy, ja Peter Marshall. 2000. "Quality and rigour of action research in information systems". *ECIS 2000 Proceedings*: 38.
- McMahon, C. 2009. "History of a Large Test Automation Project Using Selenium". *Teoksessa 2009 Agile Conference*, 363–368. doi:10.1109/AGILE.2009.9.
- McNiff, Jean, ja Jack Whitehead. 2001. *Action Research in Organisations*. London: Routledge.
- Melrose, Mary J. 2001. "Maximizing the Rigor of Action Research: Why Would You Want To? How Could You?" *Field Methods* 13 (2): 160–180. doi:10.1177/1525822X0101300203.
- Memon, Atif M., ja Mary Lou Soffa. 2003. "Regression Testing of GUIs". *SIGSOFT Softw. Eng. Notes* (New York, NY, USA) 28 (5): 118–127. doi:10.1145/949952.940088.
- Onoma, Akira K., Wei-Tek Tsai, Mustafa H. Poonawala ja Hiroshi Suganuma. 1998. "Regression Testing in an Industrial Environment". *Communications of the ACM* 41:81–86. doi:10.1145/274946.274960.
- Parasuraman, Raja, ja Dietrich H. Manzey. 2010. "Complacency and Bias in Human Use of Automation: An Attentional Integration". *Human Factors* 52 (3): 381–410. doi:10.1177/0018720810376055.
- Petersen, Kai, Cigdem Gencel, Negin Asghari, Dejan Baca ja Stefanie Betz. 2014. "Action Research As a Model for Industry-academia Collaboration in the Software Engineering Context". *Teoksessa Proceedings of the 2014 International Workshop on Long-term Industrial Collaboration on Software Engineering*, 55–62. WISE '14. Vasteras, Sweden: ACM. doi:10.1145/2647648.2647656.
- Sheridan, Thomas B., ja Raja Parasuraman. 2005. "Human-Automation Interaction". *Reviews of Human Factors and Ergonomics* 1 (1): 89–129. doi:10.1518/155723405783703082.

- Singh, Ashima, ym. 2014. "Efficient regression test selection and recommendation approach for component based software". Teoksessa *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1547–1553. IEEE.
- Singh, Yogesh. 2012. *Software testing*. Cambridge: Cambridge University Press Cambridge.
- Sneha, K., ja G. M. Malle. 2017. "Research on software testing techniques and software automation testing tools". Teoksessa *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 77–81. doi:10 . 1109 / ICECDS . 2017 . 8389562.
- Somekh, Bridget. 2005. *Action research: A methodology for change and development*. Berkshire: McGraw-Hill Education.
- Swanson, Richard A, ja Elwood F Holton. 2005. *Research in organizations: Foundations and methods in inquiry*. San Fransisco, CA: Berrett-Koehler Publishers.
- Whittaker, J. A. 2000. "What is software testing? And why is it so hard?" *IEEE Software* 17 (1): 70–79. doi:10 . 1109 / 52 . 819971.
- Yadav, Preeti, ja Ajay Kumar. 2015. "An Automation Testing using Selenium Tool". *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, ISSN: 2278–6856.
- Yoo, Shin, ja Mark Harman. 2012. "Regression testing minimization, selection and prioritization: a survey". *Software Testing, Verification and Reliability* 22 (2): 67–120.
- Zuber-Skerritt, Ortrun. 2003. *New directions in action research*. London: Routledge.