

Testaustavat Jyväskylän seudun yrityksissä

Miia Pajunen

Opinnäytetyö
Toukokuu2015

Tietojenkäsittelyn koulutusohjelma
Luonnontieteiden ala





Tekijä Pajunen, Miia	Julkaisun laji Opinnäytetyö	Päivämäärä 11.05.2015
	Sivumäärä 40	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Testaustavat Jyväskylän seudun yrityksissä		
Koulutusohjelma Tietojenkäsittely		
Työn ohjaaja Kiviaho, Niko		
Toimeksiantaja Jyväskylän ammattikorkeakoulu		
Tiivistelmä <p>Työn tarkoituksena oli toteuttaa Jyväskylän ammattikorkeakoululle kirjallinen selvitys Jyväskylän seudun yritysten testauksen tasosta. Materiaalina käytettiin Jyväskylän seudun yritysten haastatteluiden pohjalta saatua tietoa. Työn tavoitteena oli saada tietoa siitä, miten testauskoulutusta tulisi kehittää.</p> <p>Opinnäytetyöhön sisällytettiin teoriaosuus ohjelmistotestauksesta ja testausautomaatioita. Teoriaosuus käsittelee tärkeimpiä osa-alueita ohjelmistotestaukseen liittyen, ja täten auttaa ymmärtämään tuloksia käsittelevän osan opinnäytetyöstä. Opinnäytetyössä käytettiin tutkimusmenetelmänä kvalitatiivista eli laadullista tutkimusta.</p> <p>Työn tuloksena saatiin aikaan raportti, josta nähdään testauksen tila tutkituissa yrityksissä, ja jota voidaan käyttää testauksen koulutuksen parantamiseen.</p>		
Avainsanat (asiasanat) Ohjelmistotestaus		
Muuttiedot		



Author Pajunen, Miia	Type of publication Bachelor's thesis	Date 11.05.2015
		Language of publication: Finnish
	Number of pages 40	Permission for web publication: x
Title of publication Testing styles in companies around Jyväskylä region		
Degree programme Business Information Systems		
Tutor Kiviaho, Niko		
Assigned by JAMK University of Applied Sciences		
Abstract <p>The goal of this assignment was to produce a written report for JAMK University of Applied Sciences about testing styles in companies around Jyväskylä region. Material that was used was collected via interviews from companies. The goal was to receive information how to improve testing education.</p> <p>Theory section about software testing and automated testing was included in the thesis. The theory section covers important parts related to software testing and helps to understand result section of the thesis. Qualitative examination was used as a research method in the thesis.</p> <p>The result of this assignment was a report that shows the shape of testing in the companies that were included in the interviews. The report can also be used to improve testing education.</p>		
Keywords/tags (subjects) Software testing		
Miscellaneous		

Sisältö

1	Johdanto	3
2	Tutkimusasetelma	3
2.1	Opinnäytetyön tavoitteet ja rajaukset.....	3
2.2	Tutkimus-/kehittämismenetelmät.....	3
2.3	Tutkimuskysymykset.....	4
3	Ohjelmistotestaus	4
3.1	Testauksen periaatteet.....	4
3.2	Testauksen V-malli.....	6
3.3	Ketterä testaus.....	8
4	Testausautomaatio	9
5	Tulokset	14
5.1	Taustatiedot ja tuotekehitysmalli.....	14
5.2	Testauksen organisointi.....	20
5.3	Testausprosessi.....	24
5.4	Testauksen suunnittelu.....	27
5.5	Testaustavat.....	29
5.6	Työkalut.....	33
5.7	Virhekäsittely.....	34
5.8	Kehitystarpeet.....	35
6	Pohdinta	37
	Lähteet	40

Kuviot

Kuvio 1. V-malli: vesiputouksen vaiheet ja niihin assosioidut testausvaiheet	6
Kuvio 2. Yritysten henkilömäärät tuotekehityksessä.....	14
Kuvio 3. Tyypillinen projektissa työskentelevä henkilömäärä	15
Kuvio 4. Yhtäaikaisesti meneillään olevien projektien määrä.....	16
Kuvio 5. Yrityksillä käytössä olevat ohjelmistokehityksen menetelmät	16
Kuvio 6. Käytössä olevat roolit ja niiden jakautuminen yrityksiin.....	17
Kuvio 7. Käytössä olevat seremoniat ja niiden jakautuminen yrityksiin.....	18
Kuvio 8. Projektien minimi- ja maksimipituudet yrityksittäin.....	18
Kuvio 9. Projektin uuden version minimi- ja maksimijulkaisuväli yrityksittäin	19
Kuvio 10. Testauksen ulkoistaminen yrityksissä.....	21
Kuvio 11. Testaajien minimi- ja maksimimäärät projektissa yrityksittäin	22
Kuvio 12. Testisuunnitelman tekeminen yrityksissä.....	27
Kuvio 13. Yritysten testaukseen käyttämä aika projektin kokonaiskestosta.....	29
Kuvio 14. Katselmointien järjestäminen yrityksissä	30
Kuvio 15. Testauksen automatisointi yrityksissä	32
Kuvio 16. Yrityksissä käytössä olevat testauksen työkalut	33

Taulukot

Taulukko 1. Testauksen periaatteet.....	5
Taulukko 2. Manuaalinen testaus vastaan automaattinen testaus	10
Taulukko 3. Automatisoitaviksi sopivat testit	12
Taulukko 4. Automatisointiin sopimattomat testaukset	12

1 Johdanto

Tuotteiden testauksen pitäisi olla merkittävä osa tuotteiden kehitystä ennen asiakkaalle toimittamista. Testauksella ehkäistään viallisten tuotteiden päätyminen asiakkaalle asti. Kaikkia virheitä ei pystytä mitenkään löytämään, mutta hyvällä testauksella virhemarginaali saadaan todella pieneksi.

Opinnäytetyön ideana on kartoittaa, millä tasolla Jyväskylän seudun yrityksissä testaus todellisuudessa on. Tällöin on mahdollista kehittää menetelmiä, joilla testausta saataisiin parannettua ja toimeksiantajan, Jyväskylän ammattikorkeakoulun, järjestämää Testaus-opintojaksoa kehitettyä. Opintojakson kehittäminen auttaisi opiskelijoita oppimaan enemmän testaamisesta ja ajankohtaisista testausmenetelmistä.

2 Tutkimusasetelma

2.1 Opinnäytetyön tavoitteet ja rajaukset

Opinnäytetyön toimeksiantajana toimii Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma. Aihe sai alkunsa syksyllä 2012 järjestetyllä Testaus-opintojaksolla, jonka harjoitustöiden tuotoksia käytetään opinnäytetyössä materiaalina. Tarkoituksena on kartoittaa Jyväskylän seudun yritysten tavat toteuttaa testaus, mitkä asiat ovat hyvällä mallilla ja mitkä asiat kaipaavat kehitystä.

Kerätystä materiaalista rakennetaan yksi kokonaisuus, jossa vertaillaan yritysten testautapoja, -menetelmiä jatestauksen tasoa.

2.2 Tutkimus-/kehittämismenetelmät

Tutkimusmenetelmänä on kvalitatiivinen eli laadullinen tutkimus, jossa on kuitenkin määrällistä aineistoa mukana. Tutkimuksessa ei ole kuitenkaan tarkoituksena analysoida aineistoa määrällisen, vaan laadullisen tutkimuksen menetelmillä. Laadullisessa

tutkimuksessa ei käytetä tilastollisia menetelmiä tai muita määrällisiä keinoja vaan sanoja ja lauseita, kun taas määrällinen tutkimus perustuu lukuihin ja yleistyksiin. Laadullisessa tutkimuksessa tutkittavan ja tutkijan välillä on suora kontakti esimerkiksi haastattelun muodossa. Tutkimusmenetelmät vaikuttavat saatuihin tuloksiin ja kvalitatiivisen päättelyn logiikassa edetään yksittäisistä havainnoista tuloksiin. (Kananen 2008, s. 24.)

Haastatteluiden pohjana toimi ennalta valittujen kysymysten joukko, josta valittiin joko kaikki tai osa esitettäväksi haastateltaville yrityksille.

2.3 Tutkimuskysymykset

Tutkimuskysymyksiksi päädyttiin valitsemaan seuraavat:

Mitä on ohjelmistotestaus?

Miten Jyväskylän seudun yrityksissä ohjelmistotestausta suoritetaan?

3 Ohjelmistotestaus

3.1 Testauksen periaatteet

Ohjelmiston testaus on ohjelmiston elinkaarelle välttämätöntä, jos ohjelmistosta halutaan saada toimiva kokonaisuus. Testausmäärän tarve riippuu ohjelmiston koosta ja vaikeusasteesta. Esimerkiksi omaksi tai lähipiirin huvitukseksi tehty pieni peli ei vaadi lähellekään yhtä paljon ja yhtä tarkkaa testausta kuin pankille soveltuva asiakkaiden hallintajärjestelmä. Näillä kahdella on jo ohjelmiston koon puolesta huima ero, mutta myös tietoturvallisuus tulee kysymykseen pankille tarkoitettussa järjestelmässä.

Testauksen suorittaminen mahdollistaa myös ohjelmiston laadun mittauksen eri vaatimusten ja ominaisuuksien suhteen. Ohjelmiston virheettömyys ja löydettyjen vir-

heiden korjaaminen nostavat ohjelmiston laadun luotettavuutta.(Müller, Beer, Klonek & Verma 2010.)

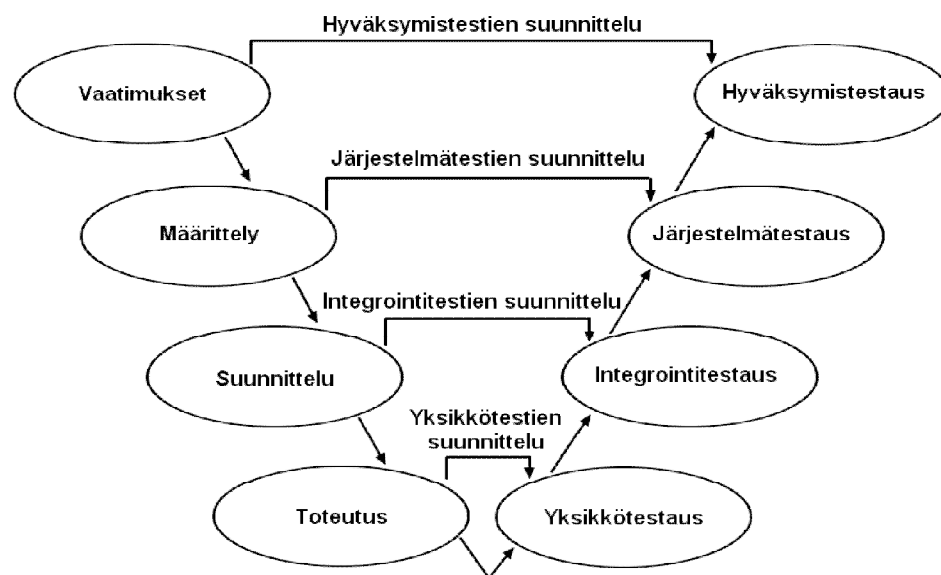
Taulukko 1. Testauksen periaatteet(Graham 2008)

Periaate 1	Testaus näyttää virheiden olemassaolon.	Testaus voi todistaa virheiden olemassaolon, mutta ei sitä, että niitä ei ole. Testaus vähentää löytämättömien virheiden olemassaoloa ohjelmistossa, mutta vaikka virheitä ei löydy, se ei ole todiste virheettömyydestä.
Periaate 2	Tyhjentävä testaus on mahdotonta.	Kaiken testaaminen (kaikki syöteyhdistelmät ja -ehdot) ei ole mahdollista kuin yksinkertaisissa tapauksissa. Tyhjentävän testauksen sijaan käytetään riskejä ja prioriteetteja testauspyrkimysten kohdistamiseen.
Periaate 3	Aikainen testaus	Testaustoimet tulisi aloittaa mahdollisimman aikaisessa vaiheessa ohjelmiston tai järjestelmän kehityksen elämäntaakkaa, ja ne tulisi keskittää määriteltyihin tavoitteisiin.
Periaate4	Virhekerääntymä	Pieni määrä moduuleista sisältää suurimman osan virheistä, jotka löytyvät julkaisua edeltävässä testauksessa ja jotka paljastavat suurimman osan operatiivisista vioista.
Periaate5	Pestisidiparadoksi	Jos samat testit toistetaan useaan kertaan, jossain vaiheessa nämä testit eivät enää löydä virheitä. Tämän pestisidiparadoksin ylitsepääsemiseksi täytyy testitapaukset käydä lävitse ja uusia säännöllisesti sekä uudet, erilaiset testit kirjoitettava ja otettava käyttöön ohjelmiston tai järjestelmän eri osissa mahdollisten lisävirheiden löytymiseksi.
Periaate6	Testaus on kontekstiriippuvainen.	Testaus tehdään eri tavoin eri konteksteissa. Esimerkiksi turvallisuuskriittinen ohjelmisto testataan eri

		tavoin kuin verkkokauppasivusto.
Periaate 7	Harhaluulo virheiden poissaolosta	Virheiden löytäminen ja korjaaminen eivät auta, jos rakennettu järjestelmä on käyttökelvoton eikä täytä käyttäjän tarpeita ja odotuksia.

3.2 Testauksen V-malli

Monet traditionaaliset testausmallit perustuvat V-malliin, joka puolestaan perustuu vahvasti vesiputousmalliin. V-mallin lähestymistapaa organisoituun testaukseen on myös käytetty muissa ohjelmistonkehitysmalleissa tuottaen monia variantteja. Kuvio 1 esittää tyyppillisen tulkinnan V-mallista. (Watkins 2009.)



Kuvio 1. V-malli: vesiputouksen vaiheet ja niihin assosioidut testausvaiheet (Watkins 2009)

V-malli tarjoaa tehokkaat apukeinot testauksen ammattilaisille testauksen siirtämissä ohjelmiston elinkaaren aikaisempiin vaiheisiin ja rohkaisee tiheämpään testaukseen. V-mallin suuri dogmi on, että testauksen suunnittelun pystyy aloittamaan jo vaatimusten kirjoitusvaiheessa. Testauspäällikkö käy lävitse vaatimukset määritelläk-

seen tarvittavat resurssit testaukselle, ja testaussuunnittelija käy vaatimukset lävitse määritelläkseen niiden virheet identifioinnin ja testattavuuden näkökulmasta. Yleisesti ottaen voidaan V-malli-lähtöisessä testauksessa identifioida neljä perustestaus-tasoa tai -vaihetta.(Watkins 2009.)

Yksikkötestaus

Yksikkötestaus tunnetaan myös nimillä kehittäjätestaus ja komponenttitestaus. Kysymyksessä on alimman tason testaus, joka normaalisti identifioidaan ja tyypillisesti suoritetaan kehittäjän toimesta, jotta voidaan varmistua ohjelmakoodin vastaavan vaatimuksia. Historiallisesti yksikkötestaus on manuaalinen prosessi, joka usein sisälsi testivaljaat tai jonkin muun tavan simuloida muita suunniteltuja ohjelmakomponentteja, jotka kommunikoisivat testissä olevan komponentin kanssa mutta joita ei vielä ollut toteutettu. Nykypäivänä yksikkötestaus yleensä suoritetaan käyttäen yhtä kehittäjän testaustyökaluista, joita on tarjolla monia.(Watkins 2009.)

Integrointitestaus

Integrointitestaus tunnetaan myös nimellä moduulitestaus. Sitä käytetään demonstroimaan testattavana olevan ohjelman moduulien rajapintaa ja niiden yhdessä toimivuuden oikeellisuutta. Yleisenä sääntönä integrointitestaus toteutetaan kehittäjien toimesta projektipäällikön valvonnassa. Sääntö on kuitenkin riippuvainen projektin koosta ja tärkeydestä. Formaalimpaa lähestymistapaa käytettäessä, esimerkiksi suuressa tai kaupallisesti tärkeässä projektissa, itsenäinen testaaja tai testaustiimi voisi olla osallisena hoitamassa integrointitestausta.(Watkins 2009.)

Järjestelmätestaus

Järjestelmätestausta käytetään vahvistamaan luottamusta siihen, että valmistunut testattavana oleva ohjelma läpäisee hyväksymistestauksen. Järjestelmätestauksen aikana järjestelmän toiminnallinen ja rakenteellinen stabiilius tutkitaan. Myös testattavan ohjelman ei-toiminnalliset näkökulmat, kuten suorituskyky ja toimintavarmuus, tutkitaan. Riippuen projektin koosta ja tärkeydestä suorittaa järjestelmätestauksen tyypillisesti itsenäinen testaustiimi. Asiakas tai käyttäjän edustaja kutsutaan usein todistamaan järjestelmätestaus.(Watkins 2009.)

Hyväksymistestaus

Hyväksymistestaus tunnetaan myös nimellä UAT (useracceptancetest). Sitä käytetään varmistamaan siihen, että testattava applikaatio vastaa vaatimuksia, ja tarjoamaan luottamusta järjestelmän oikeanlaisesta toimivuudesta sekä varmistamaan, että applikaatio on käyttövalmis, ennen kuin se virallisesti luovutetaan loppukäyttäjille. Tyypillisesti hyväksymistestauksen toteuttavat määrätyt käyttäjän edustajat, joskus itsenäisen testaus edustajan tai tiimin avustuksella.(Watkins 2009.)

Regressiotestaus

Regressiotestauksen rooli on myös huomionarvoinen. Sen tarkoituksena on luoda luottamusta, että testattava applikaatio toimii oikein myös uuden version tai julkaisun myötä. Uusi versio tai julkaisu voi johtua esimerkiksi asiakkaan muunnospyyntöistä, ohjelmiston parannuksista tai ohjelmiston suoritusympäristön muutoksista, kuten perustana olevan käyttöjärjestelmän uusi julkaisu.(Watkins 2009.)

3.3 Ketterä testaus

Ketterä testaus on yhteistyöhön perustuvaa testausta, jossa kaikki ovat osallisina suunnittelussa, toteutuksessa ja testisuunnitelman suorittamisessa. Asiakkaat ovat osallisina hyväksymistestien määrittämisessä rajaamalla käyttötapaukset ja ohjelman ominaisuudet. Kehittäjät tekevät yhteistyötä testaajien kanssa rakennettaessa testi-valjaita, jotka testaavat toimivuutta automaattisesti. Ketterä testaus vaatii kaikkien osallistumista testausprosessiin, mikä vaatii paljon kommunikointia ja yhteistyötä. (Myers, Sandler & Badgett 2012.)

Ketterä testaus edellyttää asiakkaan mahdollisimman aikaista osallistumista kehittämisen elinkaareen. Esimerkiksi kehittäjien saatua stabiilin koodipohjan valmiiksi tulisi asiakkaiden aloittaa hyväksymistestaus ja antaa palautetta kehittäjätiimille. Tämä tarkoittaakin sitä, että testaus ei ketterissä menetelmissä ole vaihe, vaan se on sisällytetty kehitystyöskentelyyn aikaansaaden jatkuvan kehityksen. (Myers, Sandler & Badgett 2012.)

Ketterässä testauksessa turvaudutaan automatisoituun testaukseen. Kehityssyklit ovat lyhyitä, joten aika on arvokasta. Automatisoitu testaus on luotettavampaa kuin manuaalinen, sillä manuaalinen testaus on aikaa syövää ja saattaa itsessään aiheuttaa virheitä. Jotkin ongelmat saattavat kuitenkin vaatia tutkivaa manuaalista testausta, vaikka automatisoitu testaus onkin ensisijainen. (Myers, Sandler & Badgett 2012.)

Scrum

Scrum on projektinhallintamenetelmä ketterään ohjelmistokehitykseen ja testaukseen, joka mahdollistaa itsenäisen työryhmän luomisen rohkaisemalla työryhmän jäsenten yhteneväistä sijaintia, mukaan lukien asiakkaan edustaja, yhdistettynä tehokkaaseen verbaaliin kommunikointiin työryhmän jäsenten ja muiden projektissa osallisena olevien kesken. (Watkins 2009.)

Scrumin tärkein periaate on huomioida se mahdollisuus, että asiakkaat mitä todennäköisimmin muuttavat mielipidettään projektin aikana useaan kertaan siitä, mitä he haluavat ja tarvitsevat sekä että tämän kaltaisia asiakkaiden tarpeita ei pystytä käsittelemään onnistuneesti traditionaalisella ennustavalla tai suunnitellulla tavalla. Tämän vuoksi Scrum omaksuu empiirisen lähestymistavan hyväksyen sen, että ongelmaa ei pystytä täysin ymmärtämään tai määrittelemään, ja keskittyen sen sijaan maksimoimaan työryhmän kykyä toimittaa nopeasti ja vastata esille tuleviin vaatimuksiin. (Watkins 2009.)

Suurin syy Scrum-projektien menestykseen ovat tehokkaat kommunikointitavat, kuten päivittäiset lyhyet palaverit, sekä tarkat roolit työryhmän sisällä. Näillä pystytään hallinnoimaan sitä, ketkä antavat panoksensa palavereissa ja millä tavalla. (Watkins 2009.)

4 Testausautomaatio

Yksinkertaisesti ilmaistuna automaatiotestaus on manuaalisen testausprosessin automatisointia käyttämällä hyväksi kehitettävää ohjelmistoa. Manuaalinen testaus, eritoten regressiotestaus, voi olla uuvuttava, aikaa vievä ja pitkäväteinen prosessi,

joka on ristiriidassa nykypäivän lyhyempien ohjelmistojen elinkaarien kanssa. Täten ohjelmistoa on vaikea testata läpikotaisin, mikä aiheuttaa kriittisten virheiden löytymättömyyden. Manuaaliset testit ovat myös taipuvaisia ihmisistä johtuviin virheisiin ja epäjohtonmukaisuuksiin, jotka voivat vääristää testituloksia. On kuitenkin muistettava, että automaatiotestaus ei ole täydellinen manuaalisen testauksen korvaaja, vaan tietyt osa-alueet on parempi testata manuaalisesti. Automaatiotestaus siis täydentää manuaalista testausta niiltä osin, kuin se soveltuu ohjelmiston testaukseen. Automaatiotestaus ei välttämättä tee testejä yhtään paremmin kuin ihmistestaaja, mutta oikein käytettynä se nopeuttaa testausprosessia huomattavasti, ja testaaja voi käyttää aikaansa tärkeämpiin ja kriittisempiin osa-alueisiin. (Halili 2008.)

Taulukko 2. Manuaalinen testaus vastaan automaattinen testaus(Halili 2008)

Manuaalinen testaus	Automaattinen testaus
Testien ajaminen (ja uudelleen ajaminen) manuaalisesti voi olla todella aikaa vievää.	Kustannustehokas, jos testejä pitää toistaa useampaan kertaan.
Kaikki vaaditut testit pitää ajaa uudestaan jokaisen version jälkeen. Aikanaan siitä tulee tylsää ja väsyttävää ja se uuvuttaa testaajan.	Sallii automatisoinnin usein kehittyvään koodiin. Sopii parhaiten ketterällä ohjelmistokehitys rungolla kehitetyn koodin testaukseen.
Manuaaliset testit täytyisi ajaa peräkkäin.	Automatisoidut testit voidaan ajaa samanaikaisesti eri koneilla.
Aikaa vievää ja pitkäveteistä, jos kyseessä on suuri testimatriisi, sekä erittäin virheisiin altistuvaa.	Auttaa suuren testimatriisin testauksessa.
Jos testi ajetaan vain kahdesti koodauksen virsintanpylväessä, pitäisi sen todennäköisesti olla manuaalinen, koska siitä on vähemmän kustannuksia kuin automatisoidusta.	Automatisoinnin kehyksen ja testitapausten pystyttäminen ja konfigurointi kustantaa enemmän.
Sopii paremmin, jos testattavana on käyttöliit-	Visuaalista informaatiota ei pystytä automatisoimaan. Sopii paremmin ei-

tymä.	käyttöliittymätesteihin.
Sallii testaajan tekevän enemmän tilapäistä/sattumanvaraista testausta, mikä nostaa mahdollisuuksia löytää todellisia käyttäjävirheitä.	Automatisoinnin testaustyökalut ovat ohjelmistoja itsekin, eikä täydellistä ohjelmistoa ole olemassa. Virheitä voi paljastua myös näistä työkaluista.
Testaaja voi tehdä testausta ilman automatisointia.	Sopii ainoastaan osaan testausprosessista.

Automaatiotestaus on parasta suorittaa silloin, kun samoja testejä toistetaan tai simuloidaan, kuten normaalisti regressio- ja suorituskykytestauksessa tehdään. Kaikki testauksen lähestymistavat eivät kuitenkaan sovi automatisoitaviksi. (Halili 2008.)

Taulukko 3. Automatisoitaviksi sopivat testit(Halili 2008)

Testaustapa	Kuvaus
Hyväksymistestaus	Virallinen testaus, jonka näkökulmana ovat käyttäjän tarpeet, vaatimukset ja yritysprosessi. Otetaan selvää, tyydyttääkö järjestelmä hyväksymisen arvosteluperusteet, sekä annetaan asiakkaalle mahdollisuus päättää, hyväksytäänkö järjestelmä.
Yhteensopivuustestaus	Prosessi, jolla selvitetään ohjelmiston yhteentoimivuus.
Kuormitustestaus	Yksi suorituskykytestauksen tyyppi, jolla arvioidaan komponentin tai järjestelmän käyttäytymistä, kun kuormitus kasvaa, jos on esimerkiksi useampi käyttäjä samaan aikaan. Määritetään, minkälaisen kuormituksen komponentti tai järjestelmä pystyy käsittelemään kerralla.
Kapasiteettitestaus	Testataan, missä järjestelmä altistuu suurelle määrälle tietoa.
Suorituskykytestaus	Testauksen prosessi, jolla määritetään ohjelmiston suorituskyky.
Luotettavuustestaus	Testauksen prosessi, jolla määritetään ohjelmiston luotettavuus.
Rakennetestaus	Komponentin tai järjestelmän sisäisen rakenteen analyysiin pohjautuvaa testausta. Tunnetaan myös nimellä white-box-testaus.
Regressiotestaus	Testaus tehdään jo aikaisemmin testattuun ohjelmistoon, johon on tehty muutoksia. Tällä varmistetaan, että muutosten myötä ei ole ilmestynyt virheitä alueille, joihin ei ole tehty muutoksia.
Poikkeustestaus	Testataan komponentin tai järjestelmän reaktioita virheellisiin syötteisiin ihmiskäyttäjältä, toiselta komponentilta tai järjestelmästä tai johtuen sisäisestä viasta.
Kielteinen testaus	Testit pyrkivät osoittamaan, että komponentti tai järjestelmä ei toimi.

Taulukko 4. Automatisointiin sopimattomat testaukset(Halili 2008)

Testaustapa	Kuvaus
Asennustestaus	Testaus, joka keskittyy siihen, mitä asiakkaiden pitää tehdä asentaakseen uuden ohjelmiston onnistuneesti.
Kokoonpanotestaus	Testataan ohjelmiston asennettavuus ja konfigurointi.
Yhteensopivuustestaus	Testaus, jolla arvioidaan sovelluksen yhteensopivuus tietojenkäsittely-ympäristöön.
Dokumentaatiotestaus	Testataan dokumentaation laatu (käyttöohjeet, asennusopas).
Virhekäsittelytestaus	Testaus, jolla määritellään sovelluksen järjestelmän kyky käsitellä virheelliset transaktiot oikein.
Palautumistestaus	Testataan, kuinka hyvin ohjelmisto palautuu järjestelmän kaatumisista, laitteiston virheistä ja muista vastaavista ongelmista.
Lokalisointitestaus	Testaus, joka keskittyy ohjelmiston kansainvälistymisen ja lokalisoinnin näkökulmiin globalisoidussa sovelluksessa, jota mukautetaan tiettyyn kulttuuriin tai tapahtumapaikkaan.
Käytettävyydestaus	Testaus, jolla määritetään laajuus ohjelmiston ymmärrettävyydestä; helppo oppia, helppo käyttää ja on puoleensavetävä käyttäjille määritellyissä olosuhteissa.

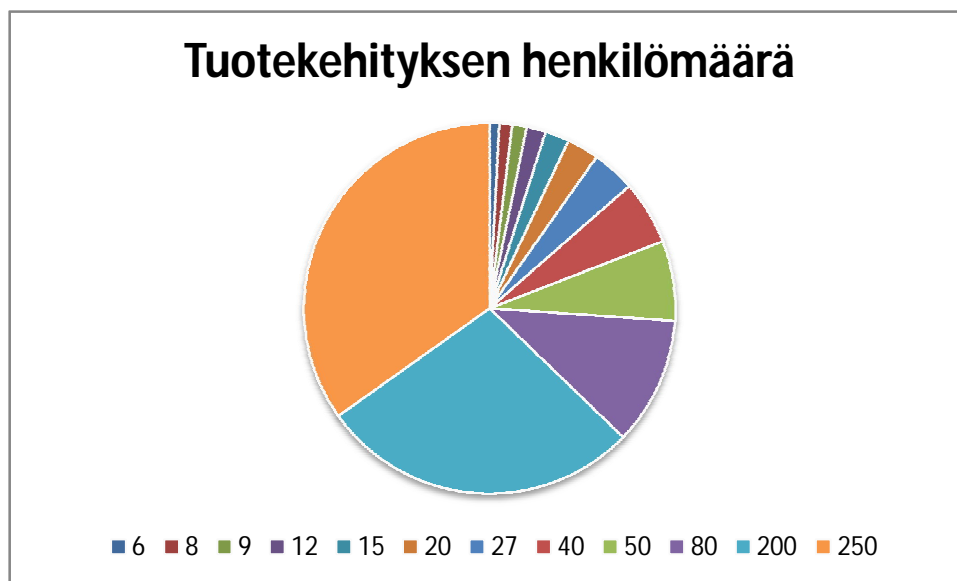
Automaatiotestauksen tehokkuus riippuu vahvasti siitä, onko automatisointi sitä, mitä testaustiimi todella tarvitsee. Jos testaustiimi kokee testien automatisoinnin turvalliseksi, voi se tehdä ihmeitä. Tehokkaasti ja oikeissa paikoissa käytettynä säästää automatisointi aikaa ja rahaa. (Halili 2008.)

5 Tulokset

Luvussa käydään lävitse tutkimuksen tulokset. Tulokset on kerätty Jyväskylän seudun yrityksiltä haastatteluina, ja ne käsitellään nimettöminä. Tutkimukseen osallistui 13 yritystä.

5.1 Taustatiedot ja tuotekehitysmalli

Haastatellut yritykset ovat hyvin erikokoisia, joten heidän tuotekehityksessä olevien henkilöidensä määrä vaihtelee suuresti, kuten kuviosta 2 näkyy. Pienimmillään määrä oli 5–6 henkilöä ja suurimmillaan 250. Yksi yrityksistä kertoi tuotekehityksen henkilömäärän arvioinnin olevan vaikeaa yrityksen suuren koon vuoksi, joten tältä yritykseltä ei ole lukua.

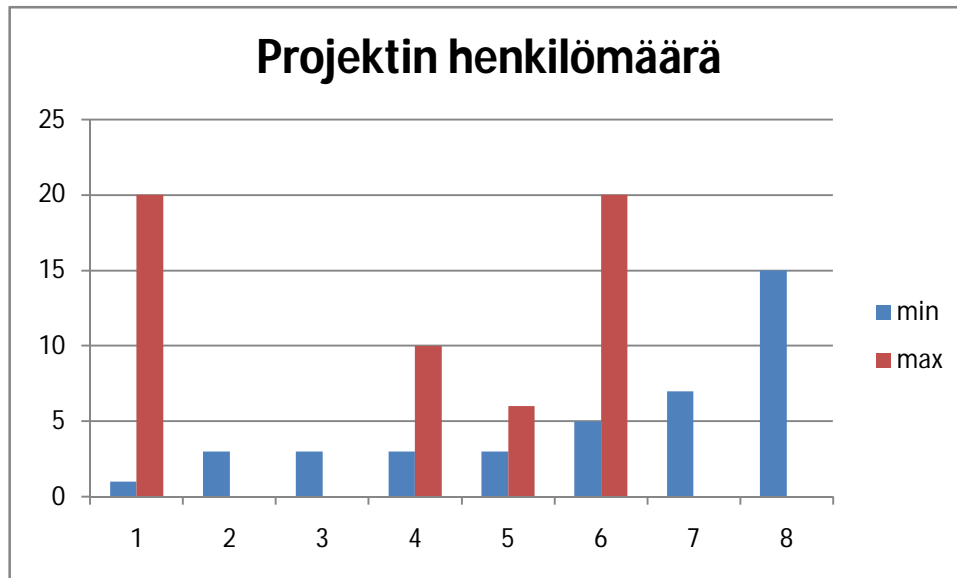


Kuvio 2. Yritysten henkilömäärät tuotekehityksessä

Projektin henkilömäärä

Yhdessä projektissa työskentelevien henkilöiden määrä vaihtelee yhden ja 20:n välillä. Kuviosta 3 näkyvät kyseiset minimi- ja maksimimäärät yrityksittäin. Viideltä yritykseltä ei lukua saatu. Projektissa työskentelevien henkilöiden määrä vaihtelee, koska projektit ovat erikokoisia. Useimmat pienet ja lyhyet projektit eivät vaadi isoa työ-

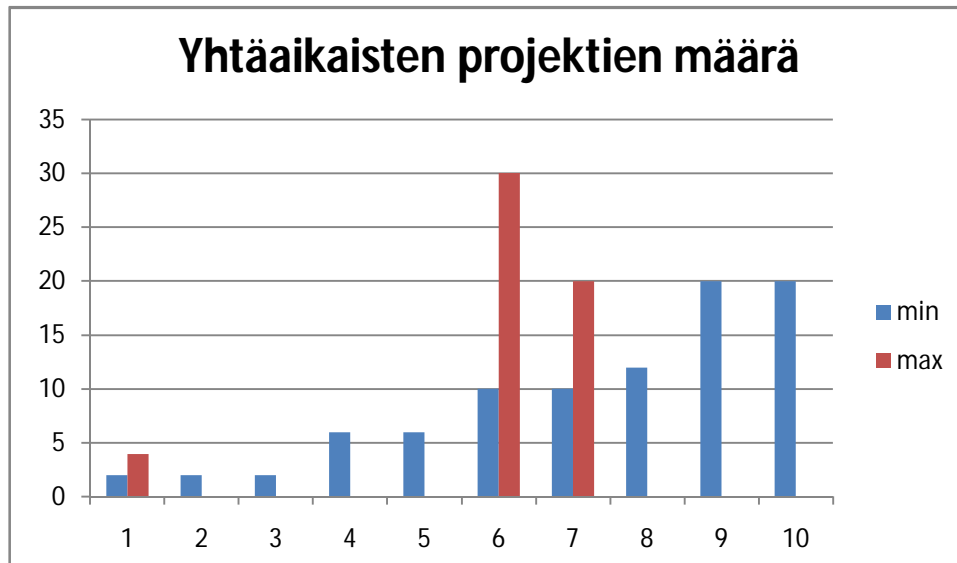
ryhmää toteuttamaan projektia, vaan se on mahdollista toteuttaa yhden henkilön työmäärällä esimerkiksi kuukaudessa.



Kuvio 3. Tyypillinen projektissa työskentelevä henkilömäärä

Yhtäaikaisten projektien määrä

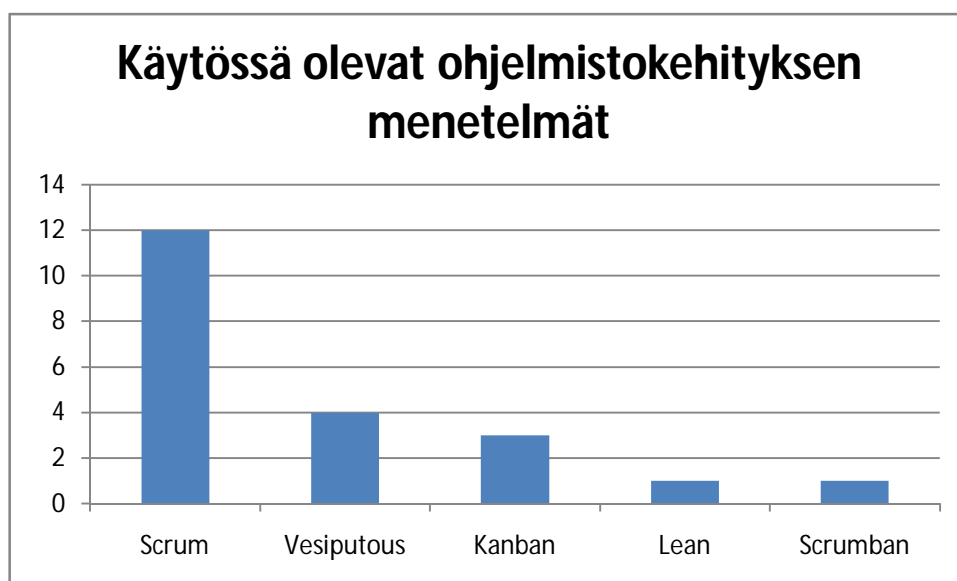
Yrityksien yhtäaikaisten projektien määrä vaihtelee suuresti riippuen yrityksen koosta. Pienimmillä yrityksillä projekteja on kahdesta kuuteen yhtäaikaisesti meneillään, kun taas isoimmilla 10:stä 30:een. Kuviossa 4 näkyvät määrät ja vaihteluvälit yrityksittäin. Useimmat yritykset pystyivät kertomaan yhden luvun, joka pääsääntöisesti pitää paikkansa, mutta oli myös muutamia, jotka joutuivat antamaan vaihteluvälin yhtäaikaisten projektien määrästä. Kaksi yritystä ei kyennyt antamaan minkäänlaista lukua, ja yksi kertoi projekteja olevan useita kymmeniä.



Kuvio 4. Yhtäaikaisesti meneillään olevien projektien määrä

Ohjelmistokehityksen menetelmät

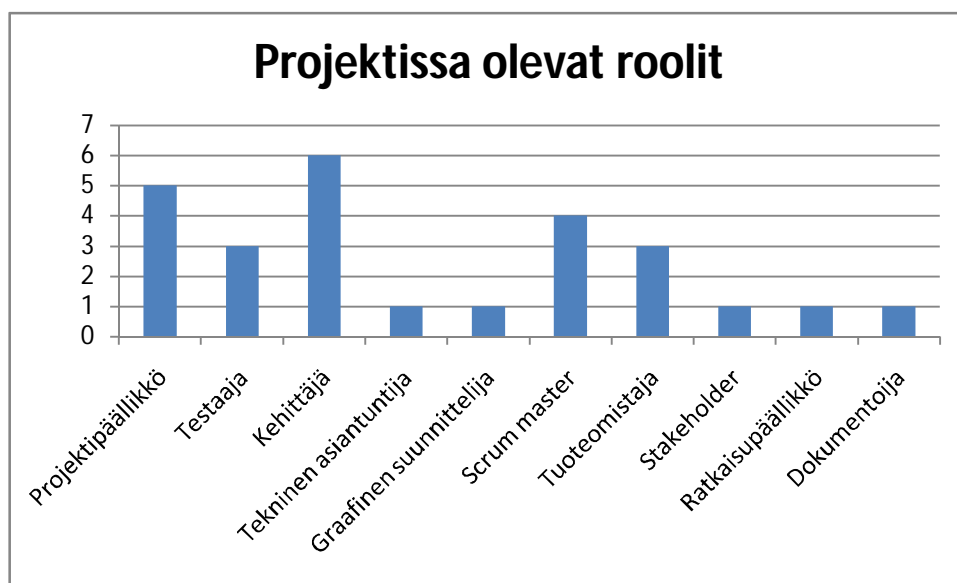
Ohjelmistokehityksen menetelmistä selkeästi käytetyin on Scrum, jota käyttää 13 haastatellusta yrityksestä 12. Yhdellä yrityksellä käytössä oli pelkästään vesiputous. Kuudella yrityksellä oli kuitenkin Scrumin rinnalla muita menetelmiä, joista Kanban ja vesiputous olivat suosituimpia. Yhdellä yrityksellä oli rinnalla Lean ja yhdellä Scrumban, joka on Scrumin ja Kanbanin risteytys. Menetelmien jakautuminen näkyy kuviossa 5.



Kuvio 5. Yrityksillä käytössä olevat ohjelmistokehityksen menetelmät

Projektissa olevat roolit

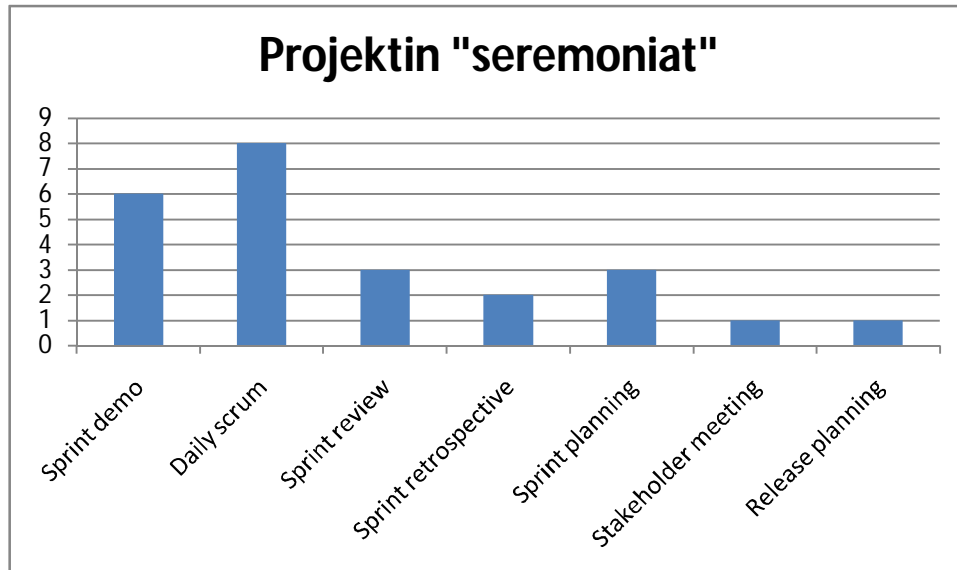
Projekteissa olevia rooleja oli yhteensä 10 erilaista. Puolet koostui rooleista, jotka olivat käytössä ainoastaan yhdessä yrityksessä, ja toinen puoli rooleista, joita oli kolmessa tai useammassa yrityksessä. Erittely rooleista ja siitä, kuinka monessa yrityksessä ne ovat käytössä, näkyy kuviosta 6. Vastaus tähän kysymykseen löytyi vain seitsemältä yritykseltä. Eniten käytössä olevat roolit olivat normaaleja Scrumin rooleja, mutta muutamia poikkeuksiakin löytyi, kuten dokumentoija, tekninen asiantuntija ja ratkaisupäällikkö.



Kuvio 6. Käytössä olevat roolit ja niiden jakautuminen yrityksiin

Projektin seremoniat

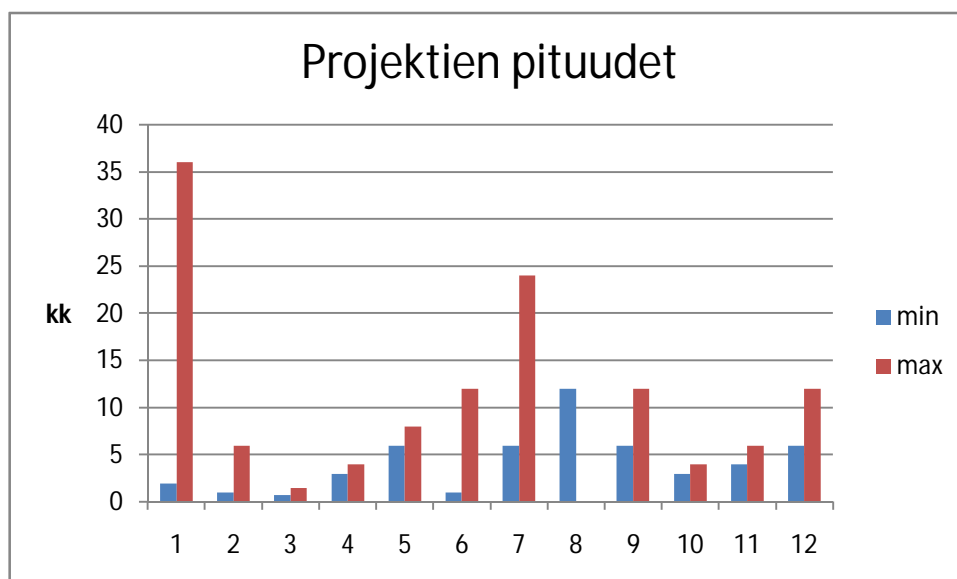
Yrityksissä käytössä olevista seremonioista Daily Scrum ja Sprint demo olivat kaikkein suosituimpia (kuvio 7). Seremonioita oli myös muita Scrumille tyypillisiä, kuten Sprint review, Sprint retrospective ja Sprint planning. Harvinaisempana tuli yhdellä yrityksellä Stakeholdermeeting. Neljän yrityksen mahdolliset seremoniat jäivät mysteeriksi.



Kuvio 7. Käytössä olevat seremoniat ja niiden jakautuminen yrityksiin

Projektien pituudet

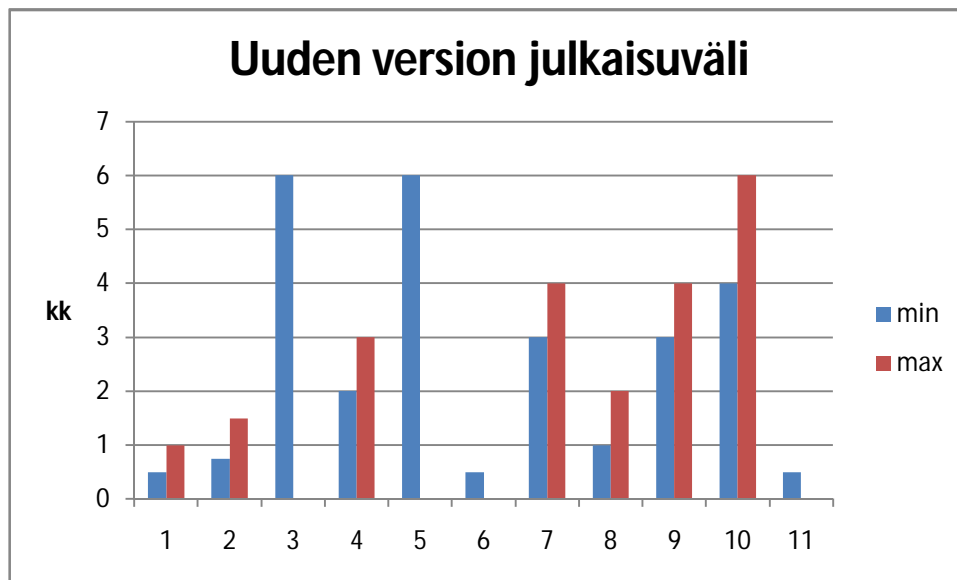
Projektien pituudet ovat täysin projektikohtaisia, ja useimmilla haastatelluista yrityksistä vaihteluväli pituuksissa on huomattava. Suurin ero on kahdesta kuukaudesta kolmeen vuoteen ja pienin kolmesta kuuteen viikkoa (kuvio 8). Kahdella yrityksellä on kaksi eri osastoa, joissa projektien pituuksien vaihteluväli eroaa toisistaan. Projektien pituuksilla ei oikeastaan ole mitään selkeää normia, vaan vastauksista näkee, kuinka paljon ne todellakin heittelevät projektikohtaisesti niin yrityksen sisällä kuin verrattaessa yrityksiä toisiinsa. Vastaukset saatiin kymmeneltä yritykseltä.



Kuvio 8. Projektien minimi- ja maksimipituudet yrityksittäin

Uuden version julkaisuväli

Projektin uuden version julkaisuväli vaihtelee haastateltavissa yrityksissä parista viikosta kuuteen kuukauteen (kuvio 9). Yhdeksästä kysymykseen vastanneesta yrityksestä neljällä väli oli alle kuukauden, loppuilla yli. Kahdessa yrityksistä julkaistaan valmis tuote suoraan, eli projektin aikana ei julkaista mitään. Yksi yritys kertoi välin vaihtelevan projekteittain, ja yhdessä riippui asiakkaasta, julkaistaanko projektin aikana mitään vai julkaistaanko vasta valmis tuote projektin päätteeksi. Yhden yrityksen kahden eri osaston julkaisuvälien ero oli pari kuukautta, ja yhdessä julkaistaan uusi versio joko 2–3 kuukauden välein tai puolivuositain.



Kuvio 9. Projektin uuden version minimi- ja maksimijulkaisuväli yrityksittäin

Tuotteen tai toimialan erityishaasteet

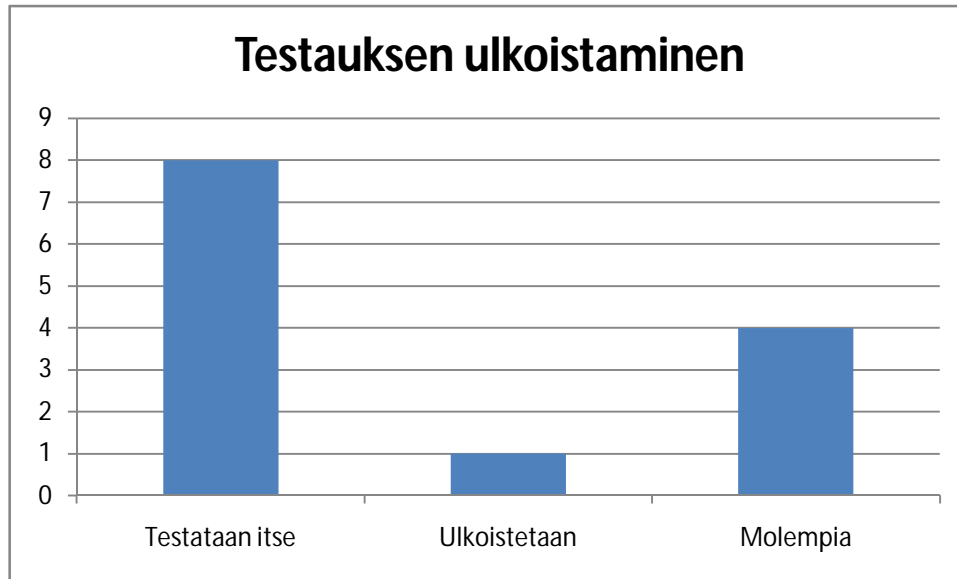
Tuotteen tai toimialan asettamiksi kehityksen ja testauksen erityishaasteiksi koettiin seuraavat asiat:

- kaikkien osapuolien aikatauluttaminen
- kaikkien osapuolien sitouttaminen
- testausta ei tehdä tuotannossa
- tytäryhtiön ongelmista ei kulje tieto
- haastavaa saada asiakas ymmärtämään testauksen tärkeys
- tietoturvan ja asiakkaan tarpeiden täyttäminen

- algoritmit
- testilaitteiston saatavuus kehitettäessä useille alustoille
- ERP-järjestelmien laajuus
- ERP-järjestelmien useat mahdolliset erilaiset käyttötavat
- järjestelmien kriittiset, erityisen tarkkaa testausta vaativat toiminnallisuudet
- testausympäristön luominen, joka vastaa asiakkaan ympäristöä
- asiakkaan toiminnanohjausjärjestelmät
- projektien erilaisuus
- sulautetut järjestelmät
- ilman tietyn alan koulutusta vaikea ymmärtää joitain asiakkaan vaatimuksia
- henkilötietoja käsiteltäessä tietosuojaa
- massojen hallinta

5.2 Testauksen organisointi

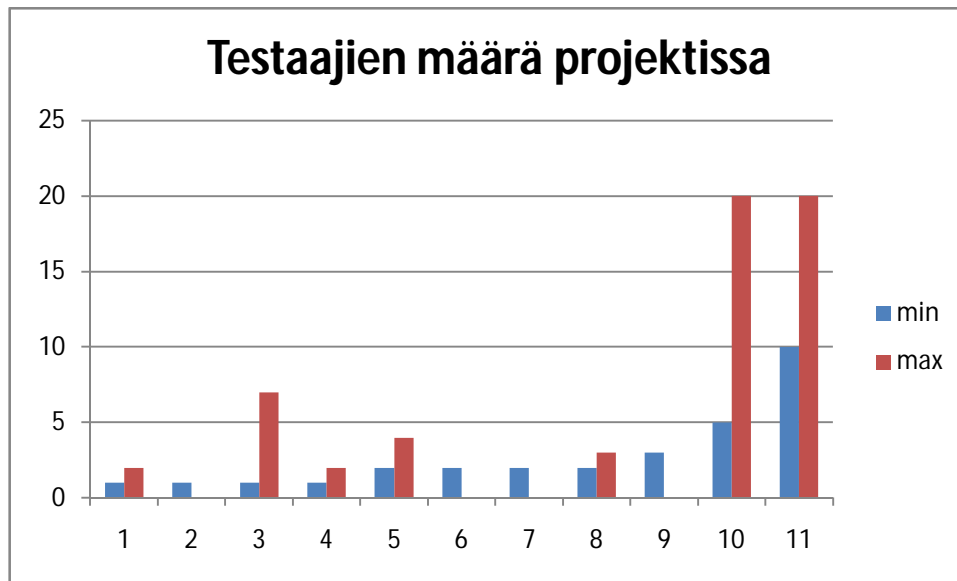
Testauksen ulkoistaminen ei ole yrityksissä kovinkaan suosittua. 13 yrityksestä ainoastaan yksi ulkoistaa testauksensa kokonaan, kahdeksan tekee testauksen kokonaisuudessaan itse, ja neljä tekee pääsääntöisesti itse, mutta ulkoistaa tiettyjä osia alueita, kuten tietoturvatestauksen (kuvio 10). Parilla ulkoistusta käytävällä yrityksellä on haasteita testauksen raportoinnin kanssa. Välttämättä ei saada minkään näköistä raporttia siitä, mitä virheitä on löytynyt ja mitä korjattu. Parilla taas raportointi on hyvinkin kattavaa. Eritoten testauksensa kokonaan ulkoistanut yritys saa säännöllisen raportin sen lisäksi, että heillä on pääsy testaaajien tikettijärjestelmään. Yhdellä yrityksistä testauksen kypsyyttä on vielä lapsen kengissä, mutta sitä pyritään kehittämään koko ajan.



Kuvio 10. Testauksen ulkoistaminen yrityksissä

Testaajien määrä projektissa

Testaajien määrä yhdessä projektissa on yleisimmin 1–3 henkilöä, joskin yhdessä isommista yrityksistä testitiimi koostuu 10–20 henkilöstä (kuvio 11). Jopa testauksen ulkoistavassa yrityksessä yksi henkilö osallistuu testaukseen ja hoitaa kommunikoinnin testausta hoitavan yrityksen kanssa. Vastaus saatiin 11 yritykseltä.



Kuvio 11. Testaajien minimi- ja maksimimäärät projektissa yrityksittäin

Testaajien tausta

Testausta yrityksissä tekevien henkilöiden taustat ovat hyvin kirjavat. Selkeästi eniten oli kuitenkin insinööri- ja IT-alan koulutusta. Yllättäen vain kolmen yrityksen testaajilla on ISTQB-sertifikaatti. Yhden yrityksen testaajista kaikki eivät osaa koodata, eikä se ole mikään selkeä vaatimus työntekijöille. Menetelmien ymmärtäminen on kuitenkin aina eduksi testaajan työssä ja osittain edellytyskin työn onnistuneeseen suorittamiseen. Yhden yrityksen testaajilla taas ei ole minkäänlaista testaajan taustaa, mutta he tietävät koodauksesta vähän.

Testaajien koulutukset, jotka tulivat esille haastatteluissa:

- insinööritutkinto
- tietojärjestelmätieteiden tutkinto

- TestPro-koulutusohjelma (työvoimapolitiittinen koulutus, jossa henkilö on yrityksessä töissä, mutta ei työsuhteessa)
- tietotekniikan insinööri
- tietojenkäsittelyn tradenomi
- mediatekniikan koulutus
- diplomi-insinööri
- tradenomi
- ohjelmistokehitys
- elektroniikka-asentaja
- liikuntatieteiden maisteri
- hyvinvointiteknologian insinööri
- ylempi korkeakoulututkinto.

Testaajien osaamisen kehittäminen

Kaikki kysymykseen vastanneet 11 yritystä tukevat ja rohkaisevat työntekijöiden osallistumista koulutuksiin. Yhdellä yrityksellä tätä varten on varattuna kattava budjetti, ja jokainen työntekijä osallistuu vuosittain johonkin kehittävään koulutukseen, jonka opit hän käy tiivistetysti yrityksen muille työntekijöille läpi. Kaksi yrityksistä kertoo heidän tuotekehityshenkilöstönsä voivan suorittaa esimerkiksi ISTQB-sertifikaatteja, ja sen lisäksi toisesta näistä yrityksistä testaajien osaamista kehitetään sisäisillä ja ulkoisilla koulutuksilla. Seitsemässä yrityksessä kuitenkin pääsääntöinen oppiminen tapahtuu projekteissa työtä tekemällä sekä jakamalla tietoja, taitoja ja ideoita keskenään. Yksi yrityksistä kertoi kustantavansa Advanced Levelin kurssin, jos työntekijältä löytyy kiinnostusta sen suorittamiseen.

Asiakkaan osallistuminen testaukseen

Kysymykseen vastasi yhteensä 10 yritystä. Näistä kymmenestä ainoastaan yhdessä yrityksessä ei asiakas osallistu testaukseen mitenkään ja toisessa asiakkaan kanssa toimitaan tiiviisti yhdessä, mutta asiakas ei virallisesti osallistu testaukseen. Neljässä yrityksessä asiakas osallistuu testaukseen hyväksymistestauksen muodossa. Asiakkaiden joukosta löytyy kuitenkin sellaisia, joilla ei ole resursseja, osaamista tai aikaa edes hyväksymistestaukseen. Yhdessä yrityksessä pyritään mahdollisuuksien mukaan

auttamaan tällaista asiakasta, mutta siihen ei varata erikseen aikaa. Kolmessa yrityksessä asiakas on mukana joko loppukäyttäjän roolissa tai pilottiasiakkaana ja raportoi tuotteesta löytyneet virheet, jotka yritys korjaa. Yhdessä yrityksessä asiakas hoitaa pääosin tuotantotestauksen.

5.3 Testausprosessi

Yrityksiä pyydettiin kuvailemaan testausprosessiaan, ja vastaukset saatiin kaikilta 12 yritykseltä, jotka suorittavat testauksen itse. Tässä on vastaukset listattuna yrityskohteisesti:

1. Testauksessa ei ole vaiheita vaan, se suoritetaan vaatimusten yhteydessä, joissa tehtäviin kuuluu testaaminen. Yksikkötestaus kuuluu kehittäjille, integrointi- ja järjestelmätestaus suoraan käyttäjäkertomukseen.
2. Yrityksen testausprosessi alkaa testaussuunnitelman laatimisesta, joka tehdään erikseen jokaiselle projektille. Suunnitelma ja testitapaukset laitetaan Exceliin, josta ne on helppo kaivaa esiin, kun testaus aloitetaan. Testaus tehdään suunnitteluvaiheessa määriteltyjen testitapausten perusteella. Yksikkötestauksen tekee alihankkija, joka samalla tekee ohjelmistoa. Integroititestaukselta tekee yksi ihminen yrityksessä (black-box), kun tuote on tullut alihankkijalta. Asiakas hoitaa hyväksyntätestauksen. Rajapintoja yritys tutkii määrittelyn mukaan, eli testataan, toimiiko se niin kuin määrittelyvaiheessa on suunniteltu. Suorituskykyä testataan yhtenä tärkeänä testauksena.
3. Yrityksessä on toiminnassa hyväksi todettu menettely testaukseen. Yrityksen käytössä on omien kokemusten kautta koottu ja jatkuvaa ylläpitoa vaativa tarkastuslista. Käytössä oleva tarkastuslista pätee kaikkiin valmisteilla oleviin tuotteisiin ja tarjottaviin palveluihin. Yrityksen toimintaperiaatteisiin kuuluu tuotteistettujen tuotteiden esitarkastaminen sekä valmiin tuotteen tarkastaminen. Tuotteistamisen perusedellytyksiä onkin, ettei ns. runkoa tarvitse testata joka kerta vaan ainoastaan, jos siihen tehdään muutoksia. Valmiin tuotteen testaaminen suoritetaan ennen tuotteen luovuttamista asiakkaalle johdettujen projektien ajallisesta mitasta. Haastateltu henkilö kertoo testaamisen olevan resurssien hukkaamista kesken projektin.

4. Testaus on yksi projektin osa-alueista, ja testausprosessiin kuuluvat yksikkötestaus, integrointitestaus, toiminnallinen testaus, käytettävyydestaus, suorituskykytestaus sekä tietoturvatestaus.
5. Yrityksellä ovat käytössään kaikki perinteiset testausvaiheet: yksikkötestaus, integraatiotestaus, järjestelmätestaus ja hyväksymistestaus. Yksikkö- ja integraatiotestausta suoritetaan tuotteen, ominaisuuden tai toiminnon tuotekehitysvaiheessa. Järjestelmätestaus suoritetaan yrityksessä koko järjestelmäversiolle tuotekehitysprojektin koodausvaiheen päättymisen jälkeen.
6. Yrityksessä ovat käytössä kaikki yleisimmät testaustavat: yksikkötestaus, integrointitestaus, järjestelmätestaus ja hyväksyntätestaus. Käytännössä niiden käytön määrä ja laajuus vaihtelevat projektiokohtaisesti. Joissakin projekteissa hyödynnetään vain joitakin testaustapoja, mutta yleisellä tasolla periaatteena on, että vähintään yksikkötestien on hyvä olla tehtyinä. Joissakin projekteissa ei käytetä muita testausmenetelmiä ollenkaan.
7. Ohjelmoijat yksikkötestaavat, ja testaajat aloittavat testaamisen aina sprintin alusta asti, mutta ennen pilotin alkua testataan vielä kaikki ominaisuudet kerralla. Pilotoinnissa tulee myös asiakkailta virheitä, mutta sen tarkoituksena ei ole antaa asiakkaalle keskeneräistä sovellusta käyttöön. Pilotoinnin jälkeen päivitys ajetaan muillekin asiakkaille. Hyväksymistestausta ei suoriteta, koska yksilöityä toimitettavaa tuotetta ei periaatteessa ole. Käytettävyys on tärkeä, mutta ei ensisijainen. Testaus keskittyy toiminnallisuuksien testaamiseen.
8. Ohjelmoijat testaavat kokonaisuutta eli suorittavat enemmän järjestelmätestausta, ei niinkään yksikkötestausta. Hämmästyttävän vähäinen on asiakkaalle pääsevien virheiden määrä, vaikka testaus on lapsen kengissä.
9. Yksikkötestaus hoidetaan kehittäjätiimeissä. Loput suoritetaan testaustiimissä: integraatiotestaus, järjestelmätestaus ja järjestelmäintegraatiotestaus muihin järjestelmiin.
10. Kehittäjät tekevät ja kehittävät yksikkötestausta. Järjestelmä- ja integrointitestauksen hoitaa testaaja. Puhdasta hyväksymistestausta ei oikeastaan ole.
11. Yrityksen testaukseen kuuluu yksikkötestausta ja järjestelmätestausta. Kehittäjät testaavat omalla tahollaan. Kehittäjä kirjoittaa yksikkötestin ja testaajat testaavat. Tämä tehdään projektin aikana. Projektin lopussa tehdään vielä testaajien toimesta järjestelmätestaus.

12. Yrityksen testaukseen kuuluvat testauksen suunnittelu, yksikkötestit, manuaalitestaus, testien automatisointi ja regressiotestaus.

Testaustavat

Käytössä olevista testaustavoista kysyttäessä listasi 11 yritystä seuraavat

- tutkiva testaus
- skenaariotestaus
- yksikkötestaus
- integrointitestaus
- hyväksyntätestaus
- suorituskkytestaus
- käyttöliittymätestaus
- skaalautuvuustestaus
- käytettävyytestaus
- satunnaiset menetelmät
- inkrementaalinen testaus
- toiminnallinen testaus
- järjestelmätestaus.

Listatuista ehdoton suosikki yritysten keskuudessa on tutkiva testaus, mutta myös hyväksyntätestaus, suorituskkytestaus, käyttöliittymätestaus, käytettävyytestaus ja järjestelmätestaus tulivat esille useamman yrityksen kohdalla. Yhdessä yrityksessä on ajan saatossa huomattu, että tutkiva testaus on ollut erittäin tehokasta ja joskus jopa parempaa kuin tarkkaan suunniteltu testaus.

Missä vaiheissa testataan

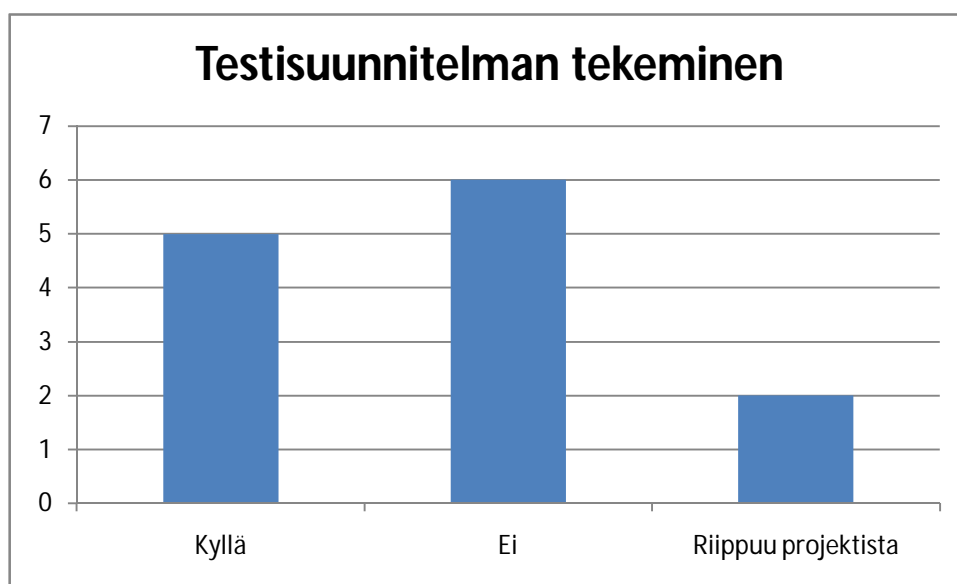
Vastaus löytyi 12 yritykseltä. Näistä yhdeksässä testausta suoritetaan ohjelmistokehityksen kaikissa vaiheissa, joskin yhdessä näistä yrityksistä toteutetaan testaus pienissä maksimissaan kahden viikon mittaisissa projekteissa vasta tuotteen valmistumisen jälkeen. Järjestelmätestit painottuvat yleisesti projektin loppupäähän, ja ne tehdäänkin usein vasta koodausvaiheen päätyttyä.

Yhdessä yrityksistä valmista runkoa testataan ainoastaan, kun siihen tehdään muutoksia, ja valmis tuote testataan ennen luovutusta asiakkaalle. Tämä johtuu projektien lyhyestä kestästä. Yrityksessä ollaan sitä mieltä, että testaus kesken projektin on resurssien hukkaan heittämistä, kun kyseessä ovat lyhyet projektit.

Toisaalla testaus suoritetaan aina sprinttien päätteeksi ja julkaisun yhteydessä tai kun saadaan jokin ominaisuus valmiiksi ja tekniset tiedot ulos.

5.4 Testauksen suunnittelu

Testisuunnitelman tekemisestä kysyttäessä saatiin vastaus kaikilta 13 yritykseltä. Testisuunnitelman tekeminen jakaantui tasaisesti yrityksiin, joissa se tehdään, ja yrityksiin, joissa sitä ei tehdä (kuvio 12). Joukosta löytyi kuitenkin kaksi yritystä, joista toisessa testisuunnitelman kirjoittaminen riippuu projektin koosta ja se tehdään ainoastaan isoissa projekteissa, ja toisessa työryhmän omasta päätöksestä, millaista dokumentaatiota he tuottavat. Testisuunnittelu aloitetaan projektin vaatimusmäärittelyn yhteydessä. Tämä sisältää testisuunnitelman kirjoittamisen niissä yrityksissä, joissa se on käytössä. Yrityksissä, joissa testisuunnitelmaa ei tehdä, ovat käytössä vakiintuneet käytännöt, yleinen testaussuunnitelma, tarkastuslista tai tikettijärjestelmä testauksen suorittamisen apuna.



Kuvio 12. Testisuunnitelman tekeminen yrityksissä

Testitapaukset

Kahdeksassa yrityksessä 10 vastanneesta testitapaukset kirjoitetaan ennen testauksen aloittamista. Suurimmalla osalla testitapaukset kirjoitetaan jo vaatimusmäärittelyn yhteydessä, mutta yhdellä yrityksellä näin tapahtuu vain isojen kokonaisuuksien kohdalla. Yhdessä yrityksessä testitapaukset suoritetaan aikalailla rinnakkain testauksen kanssa, koska sovelluksen tekniset tiedot ja toteutus valmistuvat yleensä samaan aikaan, jolloin testitapauksia ei edes voisi suunnitella paljoo etukäteen. Toisessa yrityksessä testitapauksia ei tehdä ollenkaan.

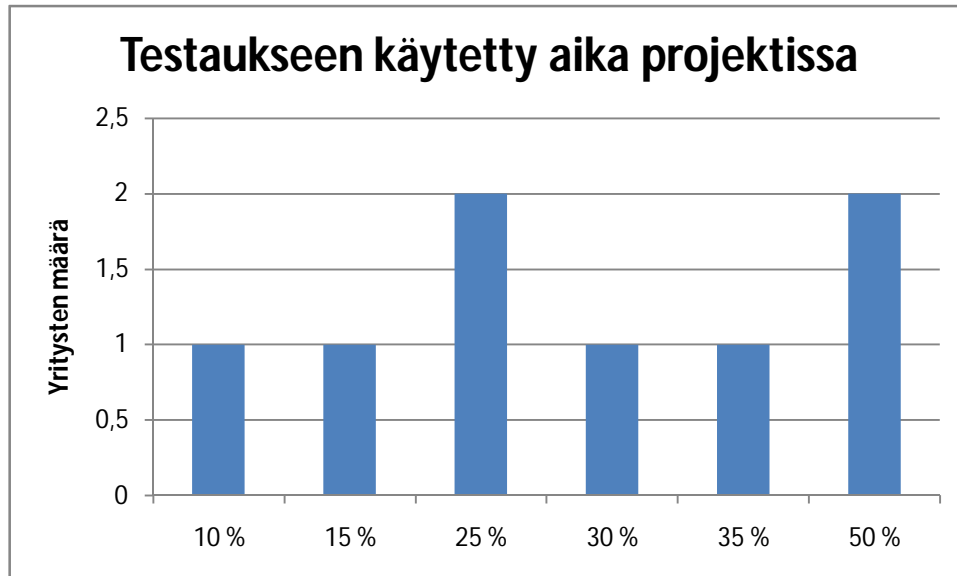
Testauksen vaihtelu

Neljässä yhdeksästä kysymykseen vastauksen antaneesta yrityksestä eivät testauskäytännöt juurikaan muutu. Itse testaus on toki projektikohtaista ja vaihtelee, mutta yrityksissä pyritään pitämään tietyt käytännöt aina samanlaisina. Viisi muuta yritystä kertoi testauksen vaihtelevan enemmän tai vähemmän eri projektien välillä. Yhdellä yrityksellä skenaariotestien automatisointi on kuitenkin antanut mahdollisuuden tehdä samoja testejä uudelleen tulevissa projekteissa. Toisella yrityksellä taas testauskäytännöt vaihtelevat kansainvälisesti eri projektien välillä, mutta saman tuotteen sisällä testauksen pääperiaatteet säilyvät lähes samanlaisina projektista toiseen.

Yhdessä yrityksessä testauksen kerrottiin vaihtelevan todella paljon eri projektien ja jopa osastojen välillä. Toisilla osastoilla testausta tehdään paremmin ja toisilla huonommin. Kyseisessä yrityksessä yritetään kuitenkin kehittää projekteille testauspohjaa, jota voidaan käyttää jatkossa.

Testaukseen käytetty aika

Kokonaisuudessaan vastauksen kysymykseen testaukseen käytetystä ajasta antoi yhdeksän yritystä, mutta kuvion 13 laatimiseen pystyttiin käyttämään näistä vain kahdeksaa, sillä yhdessä yrityksessä testaukseen käytettävä aika vaihtelee todella paljon projektien välillä. Yrityksen pitää miettiä asiakkaan liiketoimintamallia ja logiikkaa, koska vaikka toteutus olisi helppo kehityksen näkökulmasta, ei se sitä välttämättä ole sitä testauksen kannalta. Kaikille yrityksille testaus tuntui olevan tärkeä osa projektia, ja suurimmassa osassa siihen käytetään myös hyvin aikaa.



Kuvio 13. Yritysten testaukseen käyttämä aika projektin kokonaiskestosta

5.5 Testaustavat

Yrityksistä 10 antoi vastauksen, kun kysyttiin, keskittyykö testaus johonkin tiettyyn laatu näkökulmaan. Ainoastaan yksi yritys kertoi yhden laatu näkökulman, johon keskitytään, ja se oli luotettavuus. Kaikilla muilla yrityksillä oli useamman laatu näkökulman yhdistelmiä tai keskityttiin kokonaisuuteen ja sen laatuun. Laatu näkökulmia, joihin yritykset kiinnittävät huomiota, ovat

- luotettavuus
- siirrettävyys
- käytettävyys
- suorituskyky
- ylläpidettävyys
- skaalautuvuus
- virheettömyys.

Listatuista eniten esille nousivat luotettavuus, suorituskyky ja käytettävyys. Projektien erilaisuuden vuoksi käytetään kuitenkin vaatimuksia hyödyksi päätettäessä, mitä testataan eniten. Laatu näkökulmat ovat yrityksille tärkeä testauksen kriteeri.

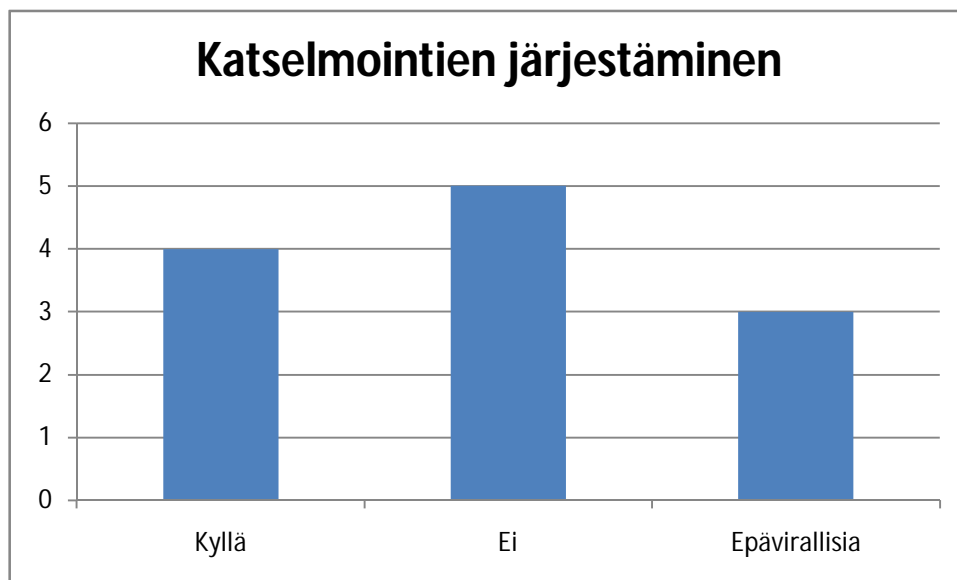
Katselmointien järjestäminen

Katselmointien pitämisestä kysyttäessä vastaukset jakaantuivat melko tasaisesti kolmeen kategoriaan: pidetään, ei pidetä, pidetään epävirallisia (kuvio 14). Vastaus saatiin 12 yritykseltä.

Yritykset, joissa katselmointeja ei järjestetä, kokevat sen johtuvan ajan puutteesta tai siitä, että niitä ei koeta tarpeellisiksi ja työntekijät kokevat ne hankaliksi. Yksi yritys kuitenkin totesi, että katselmoinnit olisivat varmasti hyödyllisiä.

Yritykset, joissa katselmointeja järjestetään, kokevat saavansa niistä suuren hyödyn. Katselmointeja järjestetään jokaisen sprintin jälkeen, ja niissä käydään läpi sprintin aikana toteutetut ominaisuudet.

Epävirallisia katselmointeja käyttävissä yrityksissä pidetään katselmoinnit esimerkiksi taukojen yhteydessä, jolloin ilmapiiri on rento. Pakollisia katselmointeja on yritetty, mutta ne on koettu puuduttaviksi. Näille epävirallisille katselmoinneille ei ole mitään virallisia prosesseja, vaan työntekijät pitävät niitä tarpeen mukaan. Katselmointien on huomattu olevan uusille työntekijöille oppimisprosessi.



Kuvio 14. Katselmointien järjestäminen yrityksissä

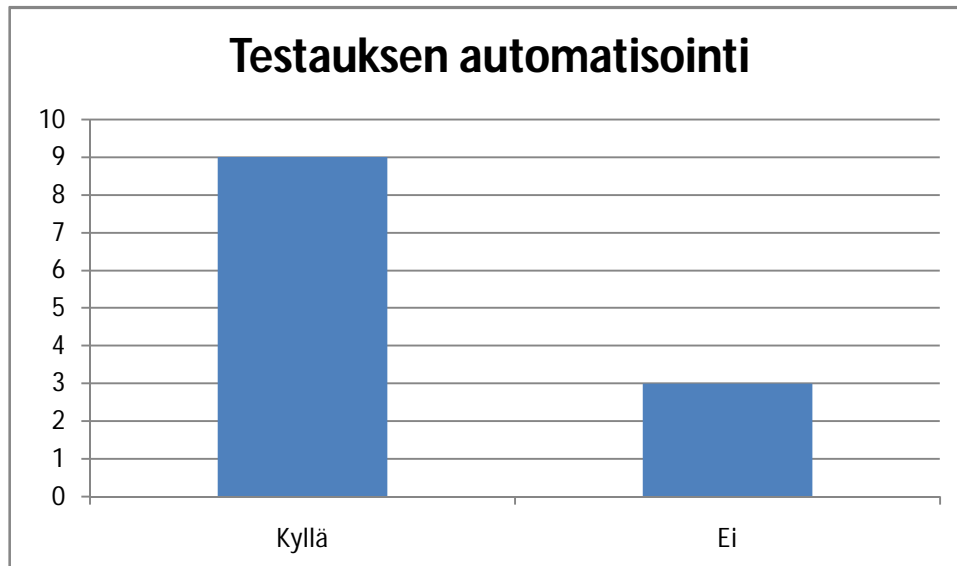
Testauksen automatisointi

12:sta kysymykseen vastanneesta yrityksestä ainoastaan kolme ei automatisoi testausta ollenkaan. Yhdellä syynä tähän on se, että projektit vaihtelevat suuresti, joten sille ei nähdä tarvetta. Toisella testausten automatisointi ei sovellu projektien lyhyen keston vuoksi, koska automatisoinnin valmistelu veisi enemmän aikaa kuin manuaalinen testaus.

Testauksen automatisointia tekevät yritykset pyrkivät automatisoimaan aina, kun se on mahdollista ja järkevää. Yhdellä yrityksellä tavoitteena on automatisoida 80 % testeistä. Automatisoituja testejä yrityksillä ovat:

- koodin tai merkkauškielen syöttäminen validaattoriin
- yksikkötestit
- integraatiotestit
- regressiotestit
- perustoiminnallisuudet.

Automatisointia käytetään testauksen tehostamiseen. Yritykset tuntuvat olevan myös hyvin perillä siitä, milloin automatisointia ei kannata tehdä. Monet mainitsivatkin, että lyhyissä projekteissa automatisointia ei käytetä, koska se on liian työlästä saatuun etuun nähden. Tämä pätee myös joihinkin testauksen osa-alueisiin. Haasteeksi yksi yritys koki sen, että automatisoinnissa käytettävä ohjelmisto aiheuttaa liian suurta ylläpitotyötä. Lähes kaikki yrityksistä pyrkivät lisäämään ja parantamaan automatisointia.



Kuvio 15. Testauksen automatisointi yrityksissä

Ylläpitotestaus

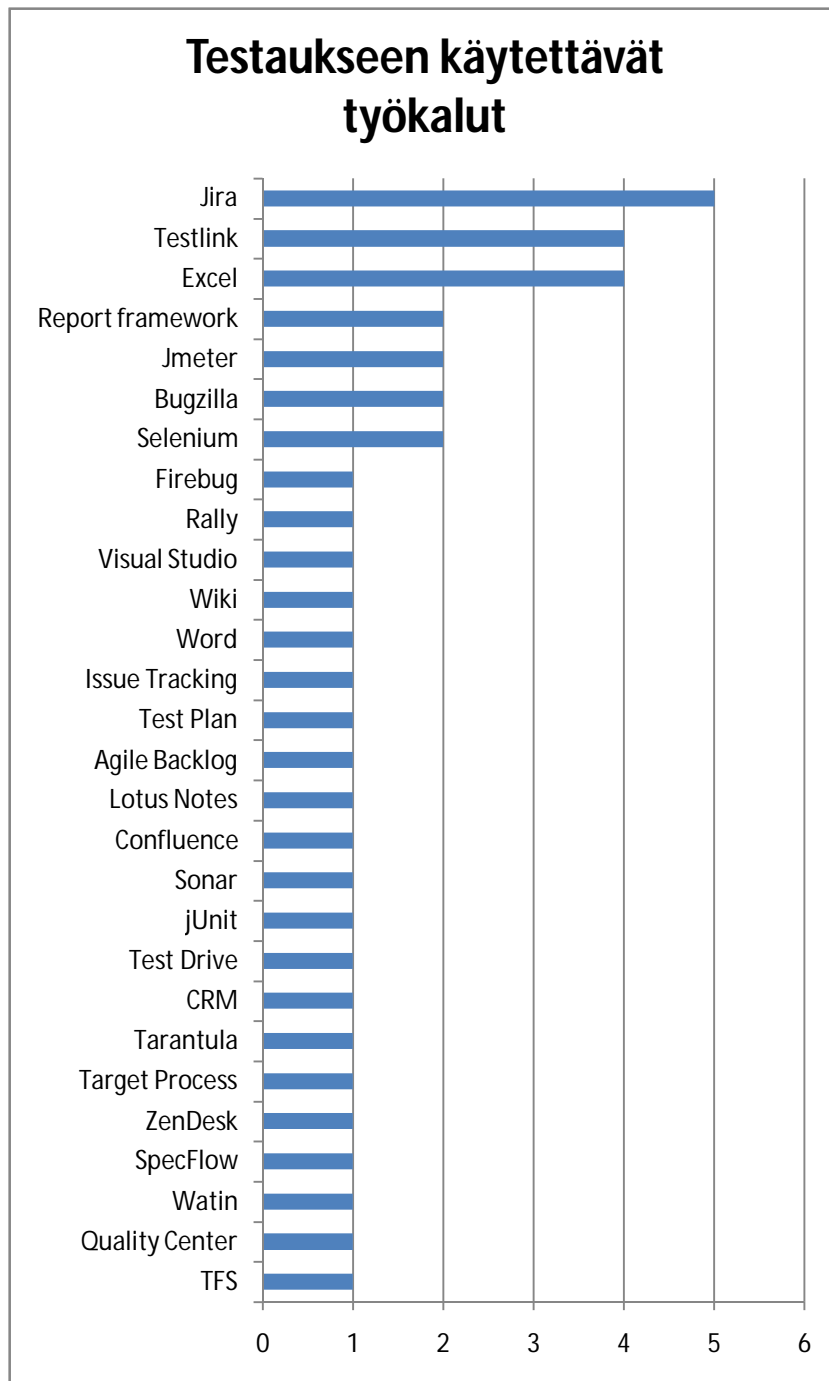
Ylläpitotestauksesta kysyttäessä vastaus saatiin kuudelta yritykseltä. Yhdellä näistä yrityksistä on ylläpito-organisaatio erikseen, joka hoitaa sen osa-alueen. Kuitenkin joskus ison projektin toteuttaminen ja ylläpito käyvät käsi kädessä, ja näissä tapauksissa ylläpitotestausta saattaa olla jonkin verran. Toisella yrityksellä taas erillistä ylläpitotestausta ei ole, mutta heillä on vanhoihin projekteihin meneillään jatkuva kehitystyö, johon kuuluu myös testaus. Neljällä muulla yrityksellä ylläpitotestausta tehdään ainoastaan, kun siihen ilmenee tarvetta.

Testauksen lopettaminen

Testauksen lopettamisen kriteereistä kertoi 12 yritystä. Yhdeksässä näistä 12:sta testauksen katsotaan olevan ohi, kun sprintin tarinat, testitapaukset, vaatimukset tai muut vastaavat käytössä olevat määrittymiset on testattu ja hyväksytty toimiviksi. Yhdessä yrityksessä testaus päättyy yleensä ajan loppumiseen, ja he kertoivat näin tapahtuvan käytännössä aina, jolloin kaikkia haluttuja asioita ei ehditä testaamaan. Toisessa yrityksessä ohjelmisto vain todetaan demopalaverissa toimivaksi, sillä katselmointi antaa kuitenkin hyväksymiselle tietyn pohjan. Yrityksellä, jolla testaus on kokonaisuudessaan ulkoistettu, ei ole määritetty varsinaisia lopetuskriteerejä. Testaukseen käytettävä aika on kuitenkin rajallinen, ja testauksen hoitava yritys ilmoittaa lisätestauksen tarpeesta.

5.6 Työkalut

Testauksen apuna käytettävien työkalujen valikoima oli haastatelluissa yrityksissä kirjava. Yrityksissä käytössä olevia työkaluja kertyi yhteensä 28, joista selkeästi suosituimpia olivat Jira, Testlink ja Excel (kuvio16). Yrityksistä 12 kertoi käytössä olevat työkalut.



Kuvio 16. Yrityksissä käytössä olevat testauksen työkalut

Työkalujen kehitystarve

Pääsääntöisesti yritykset kertoivat olevansa tyytyväisiä tämänhetkisiin työkaluihinsa. Yritys, jolla on käytössä Zen Desk, toivoi tekemättä/tehty-listausmahdollisuutta, jotta sieltä näkisi paremmin, missä kukakin on menossa. Yksi yrityksistä on siirtymässä Lotus Notes -työkalusta Jiraan ja yksi kaipasi Jiran rinnalle jotain, joka integroituisi sen kanssa paremmin, koska tällä hetkellä tehdään osittain dublikaattityötä monen eri työkalun ollessa käytössä. Vastaukset saatiin seitsemältä yritykseltä.

Testiympäristöjen järjestäminen

Testiympäristöjen järjestämisestä kysyttäessä saatiin vastauksia kerättyä seitsemältä yritykseltä. Kaikilla vastanneilla on useampia erilaisia ympäristöjä, joissa testausta suoritetaan. Yleisimpinä mainittakoon kehitysympäristö, testausympäristö ja asiakkaan testausympäristö. Yksi yrityksistä suorittaa hyväksymistestausta ja kenttätestausta asiakkaiden luona, kun muu testaus suoritetaan omassa laboratoriossa. Testausympäristöjä käytetään pääsääntöisesti niin, että kehitysympäristössä testaavat kehittäjät, testausympäristössä testaajat ja asiakkaan testausympäristössä asiakkaat.

5.7 Virhekäsittely

11 yritykseltä saatiin selvitys virhekäsittelystä, joista yhdeksän käyttää hyväkseen jotain aiemmin luetelluista työkaluista raportoimalla virheet kyseisen työkalun kautta. Yhdessä yrityksessä on käytössä tarkastuslista, joka käydään projektin päätteeksi läpi. Kyseisessä tapauksessa virheet korjataan projektin aikana sitä mukaa, kun niitä löytyy.

Kolmella yrityksellä on käytäntö, jossa yksi henkilö on vastuussa löytyneiden virheiden korjauksen priorisoinnista. Kyseinen henkilö käy läpi löytyneet virheet ja resursoi niiden korjauksen sprintteihin sen mukaan, kuinka kiireellisesti ne pitää korjata.

Yhdessä yrityksessä raportointi hoidetaan alkuvaiheessa kasvotusten, mutta projektin edetessä siirrytään virallisempaan muotoon. Kahdessa muussa yrityksessä löytyneet virheet kirjataan käytössä olevaan työkaluun. Toisessa näistä yrityksistä kaikki kehitykseen osallistuvat seuraavat ja korjaavat virheitä, kun taas toisessa havainnot

ohjataan tuotteen omistajalle, joka huolehtii virheiden korjaamisesta, minkä jälkeen testaajat verifioivat korjauksen.

Yhden haastatellun yrityksen virheet raportoidaan hylättyinä testitapauksina, joista avataan uusi testitapaus. Tämän jälkeen suunnittelija varmentaa virheen ja määrittelee siihen tehtävän korjauksen, ja korjauksen jälkeen testitapaus ajetaan uudelleen. Toisessa taas sprintin aikana toteutettavat tehtävät testataan suoraan, ja virheitä löydettyä palautetaan takaisin kehitykseen kommentein varustettuna.

5.8 Kehitystarpeet

Kaikki 13 haastatteluihin osallistunutta yritystä löysivät kehitettävää omassa testauksessaan. Yrityksillä on hyvin erilaiset kehitystarpeet, mutta muutamalla nousi kuitenkin testauksen automatisointi esiin.

Kehitystarpeet yrityksittäin:

1. Yrityksellä on erinäisiä ongelmia projekteissa joita ei ole toteutettu Scrumilla, koska käytännöt eroavat siitä, mihin on totuttu. Näihin olisi saatava selkeämmät testausprosessit tai siirryttävä niidenkin osalta käyttämään Scrumia. Tämän lisäksi on pyrittävä painottamaan testauksen osuutta projektin alusta lähtien eikä niin, että testaus tulee mukaan vasta myöhemmin. Tietoturva-testaus koetaan yrityksessä heikkoudeksi, jonka opiskeluun on tarkoitus panostaa tulevaisuudessa. Yritys haluaisi käyttää opiskelijoita esimerkiksi ruuhka-apuna, mutta tämä vaatii lisää käytännön asioiden opettamisen lisäämistä kouluihin, sillä ruuhka-avusta ei ole hyötyä, jos ruuhka on ohi opiskelijan opitua käyttämään tarvittavia sovelluksia.
2. Yritys ei ole vielä löytänyt toimivaa työkalua testauksen automatisointiin, ja se tulisi ehdottomasti tarpeeseen.
3. Tämän hetkinen tilanne on yrityksessä testaamisen osalta hallinnassa, mutta tulevaisuuden osalta mietityttää uusien työntekijöiden pohjatietämys testaamisesta ja koulutustarpeesta testaamiselle. Yrityksillä on usein personoituja ratkaisuja, ja testausmenetelmiä kehitetään omanlaisiksi, minkä oppiminen vie aikaa. Kouluissa ja koulutuksissa saadut perustiedot kuitenkin helpottavat

oppimista uudessa työpaikassa. Kehitystarpeeksi yritys kokee, että dokumentointia voisi yleisesti jättää vähemmälle ja lisätä tekemistä.

4. Yrityksen mielestä heidän testauksessaan olisi monenlaisia kehitystarpeita tuluvia projekteja ajatellen. Suurimpana tarpeena nousee esille kuitenkin automatisoinnin kehittyminen, joka on ketterän ohjelmistotuotannon edellytys tänä päivänä. Eritoten sitä kaivataan web-maailmassa, jossa muutokset ovat toiminnallisuuksiltaan isoja, ja niitä tulee usein. Regressioita ei tänä päivänä pysty kukaan käymään läpi, kun tehdään responsiivistadesignia. Jos pystytään, niin testaukseen joutuu varaamaan niin paljon resursseja, että se ei ole enää kustannustehokasta.
5. Yritys haluaa lisätä testauksen automatisointia, kehittää TDD (TestDrivenDevelopment) -käytäntöä ja saada enemmän sekä parempaa mittaritietoa ulkoiselta testaajalta.
6. Yrityksen tulevaisuudesta puhuttaessa esille nousi kypsemmässä elinkaaressa olevien tuotteiden testauksen automatisointi.
7. Yrityksen kehitystarpeista ensimmäisenä selkeästi tulivat esille testauksen suunnitelmällisyys ja testaussuunnitelman tekeminen. Koko projektia koskevia testaussuunnitelmia on tehty todella harvoin. Seuraavaksi pyrkimyksenä olisikin tuottaa jonkinlainen dokumenttipohja, johon voisi yksinkertaisuudessaan dokumentoida, että tässä projektissa ei tehdä testaussuunnitelmaa. Suunnitelmällisuuden puuttuminen testauksessa johtuu osittain siitä, ettei yrityksessä ole varsinaisia testauksen ammattilaisia tai heitä on tällä hetkellä projekteihin nähden liian vähän. Testaajista osa on vuokratyöntekijöitä, jotka toimivat asiakkaan tiloissa. Kehityskohteenä on myös testaajien määrän lisääminen tai kouluttaminen niin, että jokaiselle osastolle riittäisi osaamista. Yrityksen työntekijöistä jotkut saattavat tehdä työajasta 50 % integraatiotyötä ja 50 % testausta. Iso osa kehitystarpeesta kohdistuu jo aiemmin mainittuun testaajien resurssipulaan. Kehityskohteenä nähdään myös testauksen myyminen osana projektia eikä erillisenä työnä, koska useimmat asiakkaat eivät tajua, että testaus on projektin olennainen osa. Asiakkaiden itse suorittama testaaminen toimii joskus, mutta joskus se menee täysin pieleen. Siinä tapauksessa vastuu siirtyy asiakkaalle, eikä asiakas voi syyttää siitä yritystä, koska testauksesta ei ole haluttu maksaa.

8. Yritys haluaa kehittää testausta työkalujen osalta ja kertoi ensimmäisenä, että Lotus Notes on vaihtumassa Jiraan. Testitapauksille yritys haluaa paremman työkalun, ja se olikin jo haussa. Testauksen automatisointia halutaan myös edistää, sillä tällä hetkellä sitä käytetään ainoastaan yhden tuotteen kanssa, mutta loputkin haluttaisiin mukaan. Yritys haluaa lisätä suorituskykytestausta ja totesi, että kasvupaineet lisäävät henkilöstön tarvetta.
9. Yritys tulee kehittämään koko testauksen prosessia kypsempään suuntaan työkalujen ja toimintatapojen osalta, mutta tärkeimpänä on löytää oikea taso testaukselle.
10. Yritys haluaa kehittää laatumittareita, testauskattavuutta ja muita testauskäytäntöjä.
11. Yritys kaipaa toista henkilöä tekemään testausta. Tekemistä heidän mukaansa riittää, ja kaksi ajattelevaa henkilöä on parempi kuin yksi. Testauksen automatisoinnissa yritys näkee olevan aina kehitettävää. Pienen yrityksen ollessa kyseessä mietinnässä on, miten testaus tulisi hoitaa, jotta sen saisi riittävän hyvälle tasolle, koska ei ole mitään järkeä pitää 10 hengen testaustiimiä, jos työtä ei ole kaikille. Yritys mietti, miten sellainen toimisi, että olisi yksi testmanager, joka organisoisi testausta ulkoiselle työryhmälle.
12. Yrityksen kasvu on tasaista, joten testaukseen tarvitaan koko ajan enemmän resursseja. Suurimpana kehitystarpeena yritys kertoi täyspäiväisen testaajan mukaan saamisen. Testauksen automatisointia yritys haluaisi hyödyntää, ja testaajienkoulutus kaipaa parantamista.
13. Suurimpana kehitystarpeena yritys mainitsi testauskäytänteiden yhtenäistämisen.

6 Pohdinta

Tutkimuksen tavoitteena oli selvittää, millä tavoin ja millä tasolla testausta suoritetaan Jyväskylän seudun yrityksissä. Tavoitteena oli myös saada pohjatietoa siitä, millä tavoin testauskoulutukselle on tarvetta.

Tuloksina saatiin kohtuullisen hyvä kuva testauksen tavoista ja tasosta Jyväskylän seudun yrityksissä. Haastateltavina olleilta yrityksiltä nousi myös ideoita testauksen kouluttamisen parantamiseen kouluissa, ja muun muassa teorian opettamista kaivattiin lisää. Tuloksissa näkyy myös selkeästi, mitä osa-alueita testauksen koulutuksissa kannattaa pitää vahvoina ja mitä voidaan jättää hieman enemmän taka-alalle. Hyvänä esimerkkinä mainittakoon ketterän kehityksen menetelmä Scrum, joka oli käytössä lähes kaikissa haastatelluista yrityksistä.

Materiaalin kokoamisessa yhdeksi kokonaisuudeksi onnistuttiin kohtuullisesti. Tekemistä vaikeutti kuitenkin haastattelumateriaalien huima keskinäinen ero. Haastattelut suoritettiin nopealla aikataululla ja monen eri ihmisen toimesta, jolloin niistä kirjoitetut raportit olivat hyvin erilaisia rakenteeltaan. Yrityksiä oli myös todella hankala saada osallistumaan nopealla aikataululla, minkä vuoksi haastateltujen yritysten määrä jäi melko pieneksi.

Tuloksia voidaan hyödyntää testauksen koulutuksen parantamiseen. Lisäksi yritykset saavat vinkkejä siihen, miten heidän testaustaan voitaisiin parantaa. Haastatteluihin osallistuneet yritykset näkevät myös, mille tasolle heidän testauksensa sijoittuu muihin verrattuna.

Jatkokehityksenä voitaisiin suorittaa suurempi vastaavanlainen tutkimus. Tulokset olisivat luotettavampia, jos haastateltavia yrityksiä olisi enemmän. Haastattelut olisi kuitenkin hyvä suorittaa niin, että haastattelijana toimii sama tai samat henkilöt, jotta haastattelun runko pysyy koko ajan samana, jolloin luotettavuus kasvaa entisestään.

Kokonaisuudessaan tutkimus onnistui hyvin, ja tutkimuksen tekijäkin sai paljon lisätietoa testauksesta. Tutkimuksen tekeminen oli erittäin mielenkiintoista, vaikka tekijällä ei ennestään ollut erityisen palavaa kiinnostusta testausta kohtaan. Työtä tehtäessä ja materiaaliin tutustuttaessa kiinnostusta heräsi, mikä teki työskentelystä mieluista.

Testauksesta on ajan saatossa tullut tarpeellinen osa ohjelmistoprojekteja, ja se näkyykin yrityksissä testaukseen panostamisena. Kaikkialla se ei luonnollisesti näy samalla tavoin eikä yhtä vahvasti, mutta myös asiakkaan roolissa olevat ovat pikkuhiljaa

alkaneet ymmärtämään testauksen tärkeyden. Testauksessa ja sen toteuttamisessa on silti edelleen paljon parannettavaa, ja siihen tulee kaikkien, niin koulujen, koulutuksia järjestävien tahojen, tuotteita myyvien yritysten kuin asiakkaan roolissa olevienkin, pyrkiä.

Lähteet

Graham, D. 2008. Foundations of Software Testing: ISTQB Certification. Cengage Learning.

Halili, E. H. 2008. ApacheJMeter. Packt Publishing.

Kananen, J. 2008 Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylä: Jyväskylän ammattikorkeakoulun julkaisuja -sarja.

Myers, G. J., Sandler, C. &Badgett, T. 2012. The Art of Software Testing, Third Edition. John Wiley&Sons.

Mykkänen, T. 2009. Testausjärjestelmä – Testauksen apuväline.Jyväskylä: Jyväskylän ammattikorkeakoulu.

Müller, T., Beer, A., Klöckner, M. &Verma, R. 2010. Certified Tester – Sertifioitutestaaja. ISTQB – International Software Testing Qualifications Board.

Watkins, J. 2009. Agile Testing: How to Succeed in an Extreme Testing Environment. Cambridge University Press.