

Riitta Takanen

**ASIAKASORGANISAATIOIDEN OSALLISTUMISEN
VAIKUTUS LAATUUN OHJELMISTOTUOTANNOSSA**



JYVÄSKYLÄN YLIOPISTO
TIETOJENKÄSITTELYTIEDEIDEN LAITOS
2012

TIIVISTELMÄ

Takanen, Riitta

Asiakasorganisaatioiden osallistumisen vaikutus laatuun ohjelmistotuotannossa

Jyväskylä: Jyväskylän yliopisto, 2012, 35 s.

Tietojärjestelmätiede, kandidaatintutkielma

Ohjaaja: Halttunen, Veikko

Asiayhteys ja ympäristö vaikuttavat siihen, miten määrittelemme laadun yleisesti sekä miten määrittelemme erityisesti ohjelmistojen laadun. Laatuäkemykset, jotka ovat ensisijaisia ohjelmistokehittäjille ja -yrityksille, eivät välttämättä vastaa asiakkaalle tärkeitä näkökulmia.

Uuden sovelluksen kehittäminen on usein ohjelmistoyrityksen ja asiakkaan yhteistyötä. Etenkin ketterässä ohjelmistokehityksessä asiakkaan edustajia tarvitaan projektiryhmässä koko ohjelmistokehitysprosessin ajan. Suunnitelmapohjaisissa ohjelmistokehitysmenettelyissä asiakkaan osallistuminen voi olla rajoitetumpaa, tyypillisesti siihen sisältyy vaatimusten määrittely ja tuotteen hyväksymismenettelyt.

Tämän tutkielman tavoitteena on selvittää asiakkaan roolin ja tehtävien ilmenemistä ohjelmistotuotannossa sekä sitä, miten asiakasorganisaatioiden panostus vaikuttaa tuotteen laatuun. Kiinnostuksen kohteena oli erityisesti asiakasorganisaatioissa koettu ohjelmiston laatu.

Asiakasorganisaatioiden tapoja osallistua ohjelmistokehitykseen on kuvattu useissa tutkimusraporteissa. Niissä on arvioitu myös osallistumisen vaikutusta ohjelmistojen laatuun ja nähty se pääsääntöisesti positiivisena. Etenkin vaatimusmäärittelyn laadulla näyttäisi raporttien mukaan olevan ratkaiseva merkitys projektin menestykselle ja sille miten tuote täyttää tarkoituksensa. Kuitenkin monet tämänkin tutkielman lähdeaineistona käytetyistä tutkimuksista perustuvat kyselyihin tai haastatteluihin, joiden kohteena ovat olleet ohjelmistokehittäjät ja muut ohjelmistoyritysten edustajat eivätkä asiakasorganisaatiot suoraan. Vastauksia tutkimuskysymyksiin voidaan siten pitää vain osittaisina.

Monet ohjelmiston elinkaarta kuvaavat mallit painottavat tuotekehityksen vaiheita ennen tuotteen julkaisua. Elinkaaren pisin ja asiakkaan kannalta oleellisin vaihe, ylläpitovaihe, alkaa kuitenkin tuotteen käyttöön hyväksymisen jälkeen. Asiakkaan tulee voida vaikuttaa ohjelmiston jatkokehitykseen myös ylläpitovaiheessa, ja on tarpeen tutkia ja kehittää menetelmiä tätä yhteistyötä varten.

Asiasanat: asiakasorganisaatio, laatu, ohjelmisto, ohjelmistotuotanto

ABSTRACT

Takanen, Riitta

The impact of participation of customer organizations on quality in software engineering

Jyväskylä: University of Jyväskylä, 2012, 35 p.

Information Systems, Bachelor's Thesis

Supervisor: Halttunen, Veikko

Context and environment influence on how we define quality in general and also how we determine software quality in particular. Quality aspects which are primary to software developers and the vendor companies may not be equally important to customer organizations.

Creating a new application is often a cooperative effort of a customer organization and a software company. Especially in the agile software development representatives of the customer organization are needed in the project group during the whole software development process. In plan-driven software development methods customer involvement is often more restricted, it typically includes requirements elicitation and acceptance procedures of the product.

The purpose of this bachelor's thesis is to study the role and tasks customers have in software engineering and what kind of an influence the participation of customer organizations has on the quality of the product. Interest was taken especially in the customer experience in software quality.

There are numerous research reports describing how customers participate in software engineering. The effects of customer involvement on software quality have also been assessed. In most of those reports customer involvement is considered to have a positive influence. The reports seem to show that particularly the quality of requirements engineering has a crucial effect on the success of the project and on how the product fulfills its intended functions. However many of the reports, also those used as sources for this bachelor's thesis, are based on surveys and interviews that were targeted to software developers and other representatives of software vendors and not directly to the customer organizations. Thus the research questions remained partially unanswered.

Software life cycle models often emphasize the phases of software development before releasing the product. However, the longest and for the customer the most relevant phase - the maintenance phase - starts when the product is accepted for use. The customer should be able to influence on further development of the software product even during the maintenance phase, and methods for this cooperation should be researched and developed.

Keywords: customer organization, quality, software, software engineering

TAULUKOT

TAULUKKO 1 Yhteenveto asiakkaan osallistumisen vaikutusta kuvaavista havainnoista	25
---	----

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
TAULUKOT	4
SISÄLLYS.....	5
1 JOHDANTO.....	6
2 OHJELMISTON LAATU.....	9
2.1 Käsitteitä.....	9
2.1.1 Ohjelmisto ja ohjelmistotuotanto	9
2.1.2 Laatu.....	10
2.1.3 Ohjelmiston laatuominaisuudet.....	10
2.1.4 Laadunhallinta.....	11
2.2 Ohjelmistotuotannon prosessien laadunvarmistus	11
2.3 Ohjelmistojen laadunvalvonta.....	12
2.4 Luvun yhteenvedo.....	14
3 ASIAKKAAN ROOLI JA TEHTÄVÄT OHJELMISTOTUOTANNOSSA ..	15
3.1 Käsitteitä ja taustatekijöitä.....	15
3.1.1 Asiakas.....	15
3.1.2 Ohjelmistotuotetyypit ja ohjelmistojen hankinta	16
3.1.3 Ohjelmistoprosessi ja vaihejakomallit.....	16
3.2 Tutkimuksia asiakkaan roolista ohjelmistotuotannossa	18
3.2.1 Asiakkaan osallistuminen uuden ohjelmiston kehitykseen	18
3.2.2 Asiakasyhteistyö ketterässä ohjelmistokehityksessä	20
3.2.3 Asiakas valmisohjelmiston käyttöönotossa ja räätälöinnissä	21
3.2.4 Asiakas ja ohjelmiston ylläpitovaihe	22
3.3 Luvun yhteenvedo.....	23
4 POHDINTA JA ARVIOINTI	24
4.1 Vaikutukset laatuun ja asiakkaan kokemaan hyötyyn	24
4.2 Osallistumisen hyödyt suhteessa ohjelmiston elinkaareen.....	27
4.3 Tutkielman arviointia ja jatkotutkimusehdotuksia	28
5 YHTEENVETO	30
LÄHTEET	31

1 JOHDANTO

Laatu ei ole yksiselitteinen käsite. Tilanne, ympäristö sekä arvioijan henkilökohtainen arvomaailma ja tietämys voivat vaikuttaa siihen, miten itse kukin laadun määrittelee, vaikka arvioinnin kohde olisi sama. Ohjelmistotuotannossa moniselitteisyyttä on pyritty vähentämään määrittelemällä kansainvälisiä ja kansallisia standardeja, ohjeita ja käytäntöjä, jotka ohjaavat etenkin ohjelmistokehityksen toiminnan laatua. Subjektiiivisen näkemyksen merkitys korostuu, kun puhutaan tuotteen tai palvelun laadusta, joka voidaan määrittelijästä riippuen kuvata esimerkiksi vaatimusten täyttymisenä ja asiakastyytyväisyytenä (Haikala & Märijärvi, 2006, s. 193).

Ohjelmistot ovat monimutkaisia tuotteita, joiden kaupallinen tuotanto yleensä perustuu olemassa olevaan tai ennakoituun tarpeeseen: ratkaiseeko ohjelmisto jonkin käytännön ongelman, onko sillä hyöty- tai viihdearvoa mahdollisille asiakkaille? (Evans, 2004, s. 31) Ennen kuin ohjelmisto voidaan suunnitella ja rakentaa, tarve tulee täsmentää vaatimuksiksi. Ohjelmiston elinkaaren katsotaan alkavan vaatimusten määrittelystä ja päättyvän käytöstä poistoon. Elinkaari jaetaan osiin vaihejakomalleilla, joista useimmat painottuvat kuvaamaan prosessia ohjelmiston kehitystyön näkökulmasta, jolloin vaiheita ovat esimerkiksi vaatimusten määrittely, suunnittelu, toteutus, testaus, käyttöönotto ja ylläpito. (Haikala & Märijärvi, 2006, 36-37)

Käyttäjän tai asiakkaan näkökulmasta olennainen vaihe on ohjelmiston toimitus ja sitä seuraava käyttövaihe. Viime mainittu, jota monissa vaihejakomalleissa kutsutaan ylläpidoksi, on yleensä myös ajallisesti pisin ohjelmiston elinkaaren vaihe. (Evans, 2004, s. 154) Käyttö- tai ylläpitovaihe ei kuitenkaan ole pelkkää rutiininomaista ohjelmiston käyttöä. Sen aikana ohjelmistoa korjataan, muunnetaan ja täydennetään (Haikala & Märijärvi, 2006, s. 41). Nämä ylläpitotoimet voivat näkyä asiakkaan organisaatiossa esimerkiksi käyttökatkoina, käyttöohjeistuksen muutostarpeena ja kustannuksina. Laajan toiminnanohjausjärjestelmän versiopäivityksen kulut voivat olla 25-33 % alkuperäisen käyttöönoton kuluista (See Pui Ng, 2001). Mahdollisilla laatuvaikutuksilla, joita asiakkaan osallistuminen ohjelmistotuotantoon aiheuttaa, voi siis olla huomattava ja pitkäkestoinen merkitys asiakkaan

toiminnalle. Evans (2004, s. 237) esittää ohjelmiston elinkaaresta mallin, jossa ohjelmiston tuotekehitysprosessin jatkumona kuvataan ohjelmiston ylläpito prosessi. Molemmat päävaiheet voidaan jakaa yksityiskohtaisempiin vaiheisiin ja havainnollistaa täten tarkemmin myös ylläpitovaiheen tapahtumia asiakkaan käyttöympäristössä.

Tutkielman tavoitteena on tarkastella kirjallisuuskatsauksen avulla, miten ohjelmistojen hankkivat ja käyttävät organisaatiot osallistuvat ohjelmistotuotantoon ja mikä vaikutus osallistumisella on ohjelmistojen laatuun. Laatu pyritään tarkastelemaan asiakasorganisaation näkökulmasta: Onko kirjallisuudessa raportoitu *asiakkaiden tai käyttäjien tunnistamaa* laadun parantumista tai muita hyötyjä asiakkaan osallistumisesta ohjelmistotuotantoon? Olisi kiinnostavaa lisätä ymmärrystä myös siitä, millainen ja missä ohjelmiston elinkaaren vaiheissa tapahtuva yhteistyö asiakkaan ja ohjelmistotoimittajan välillä on tuloksekasta, ja osallistuminen asiakasorganisaatiolta edellyttää.

Lähteitä tutkielmaa varten etsittiin pääsääntöisesti tietokannoista IEEE Xplore - IEEE/IEE Electronic Library, ACM Digital Library, Proquest Computing (ProQuest), SCOPUS (Elsevier) ja Google Scholar. Hakufraaseissa on yhdistetty ohjelmistotuotantoon liittyviä hakusanoja kuten

- software development,
- software engineering,
- software implementation,
- software quality,
- software requirements,
- software testing,
- software validation ja
- software maintenance.

Nämä on yhdistetty käyttäjän tai asiakkaan osallistumista kuvaaviin hakusanoihin:

- (user or client or customer) involvement,
- (user or client or customer) participation,
- (customer or client) vendor cooperation ja
- (customer or client) supplier cooperation.

Relevanssijärjestykseen lajiteltujen hakutulosten alustava seulonta on tehty tiivistelmien perusteella. Tästä vaiheesta mukaan otetuista artikkeleista tehtiin vielä tutkielman suunnittelu- ja kirjoitusvaiheessa hieman karsintaa. Lähinnä sillä pyrittiin ylläpitämään tekstin tasapainoa välttämällä jonkin osa-alueen tarpeettoman yksityiskohtaista tai laajaa käsittelyä.

Opinnäytteiden ja kirjojen etsimiseen käytettiin myös tietokantoja JYK-DOK, Linda ja ebrary. Osa lähteistä löytyi tarkastellun aineiston lähdeluettelosta sekä selaamalla hakujen yhteydessä löytyneiden lehtien ja konferenssijulkaisujen muuta sisältöä. Minkään lehden osalta ei kuitenkaan tehty niin katta-

vaa sisällön läpikäyntiä, että voisi puhua systemaattisesta kirjallisuuskatsauksesta.

Tutkielman luvussa 2 määritellään ohjelmiston laatuun liittyviä käsitteitä sekä tehdään katsaus laadunhallintaan ohjelmistotuotannossa. Luku 3 keskittyy kuvaamaan asiakasta ohjelmistotuotannossa: keitä asiakkaat ovat, millaiset taustatekijät vaikuttavat osallistumiseen sekä laajempaa kokonaisuutena aiheeseen liittyvien tutkimusraporttien esittely. Luvussa 4 pohditaan kahden edeltävän luvun sisältöä tutkimuskysymysten näkökulmasta: saadaanko niiden pohjalta kuvaa asiakkaan kokemista laatuvaikutuksista suhteessa panostukseen, jota ohjelmistotuotantoon osallistuminen edellyttää. Lisäksi arvioidaan tutkielmaa sekä esitetään jatkotutkimusehdotuksia. Luku 5 sisältää lyhyen yhteenvedon tutkielmasta.

2 OHJELMISTON LAATU

Tässä luvussa määritellään ohjelmistotuotantoon ja laatuun liittyviä käsitteitä sekä kuvataan tiivistetysti menetelmiä ja toimintoja, joilla ohjelmistojen laatuun pyritään vaikuttamaan. Aihealue on laaja, eikä tavoitteena ole sen kattava kuvaaminen, vaan ydinkohtien esilletuonti tukemaan seuraavissa luvuissa käsiteltäviä aiheita.

Laadunhallinta jakautuu useaan osa-alueeseen, joista tässä tarkastellaan erityisesti laadunvarmistusta ja laadunvalvontaa. Laadunvarmistus on ensisijaisesti ohjelmistotuotannon prosesseihin kohdistuvaa toimintaa ja laadunvalvonnassa keskitytään tuotteen laatuun.

2.1 Käsitteitä

Tässä aliluvussa tarkastellaan muutamia ohjelmistotuotantoon ja laatuun liittyviä tämän tutkielman kannalta keskeisiä käsitteitä. Käsitteitä on voitu eri lähteissä määritellä ja tulkita hieman eri tavoin. Myös näitä eroja on pyritty tuomaan esille.

2.1.1 Ohjelmisto ja ohjelmistotuotanto

Ohjelmisto sisältää Sommervillen (2007, s. 5) mukaan suoritettavan ohjelmakoodin lisäksi dokumentaation ja konfiguraatitiedon, jolla ohjelmisto saadaan toimimaan toivotulla tavalla. Evans (2004, s. 194-195) käyttää termiä *ohjelmistotuote* kokonaisuudesta, johon kuuluu ohjelmisto dokumentaatioineen sekä lisäksi käyttö- ja tukiprosessit, käyttöönottoprosessit, liiketoimintaprosessit, koulutus ja mahdolliset käyttöönotossa tarvittavat konversio-ohjelmat.

Ohjelmistotuotanto on muun muassa Haikalan ja Märijärven (2006, s. 16) käyttämä suomennos englanninkieliselle termille software engineering. IEEE:n määritelmän mukaan se on "systemaattisen, kurinalaisen ja määrällistettävän lähestymistavan soveltamista ohjelmistojen kehittämiseen, käyttöön ja ylläpi-

toon” (IEEE 610.12, 1990). Sama käsite esiintyy suomenkielisessä kirjallisuudessa myös nimillä ohjelmistotekniikka ja systeemityö (Haikala & Märijärvi, 2006, s. 16-17).

2.1.2 Laatu

Laatua käsitteenä on vaikea määritellä yksiselitteisesti. Haikala ja Märijärvi (2006, s. 193) listaavat esimerkiksi nämä eri lähteistä kootut erilaiset näkemykset laadusta:

- erinomaisuuden aste (Oxford dictionary),
- se, missä määrin luontaiset ominaisuudet täyttävät vaatimukset (ISO),
- asiakkaalle tärkein tuote (Deming),
- asiakastytyväisyys (Ishikawa) ja
- laatu = objektiivisesti arvioitavissa oleva komponentti + subjektiivisesti arvioitavissa oleva komponentti + kokonaan arvioimattomissa oleva komponentti (Wesselius).

Evans korostaa asiayhteyden ja arvioijan näkökulman vaikutusta laatukäsitteeseen. *Tuotelähtöistä* laatua voidaan mitata tuotteelle asetettujen laatuattribuuttien avulla objektiivisesti ja määrällisesti. *Tuotantolähtöinen* laatu keskittyy tuotantotoimintoihin: määrittelyyn, suunnitteluun ja toteuttamiseen. *Käyttäjälähtöinen* laatu on soveltuvuutta käyttöön. *Arvolähtöisestä* laadusta on kyse, kun arvioidaan vaikutusta toiminnan kokonaisuuteen. *Transendentti* laatu on arvioijan käsityksistä, havainnoista, aistimuksista ja tunteista riippuvaa. (Evans, 2004, s. 20)

Tuotteen, esimerkiksi ohjelmiston, laadun voidaan siis nähdä koostuvan useista osatekijöistä. Niiden painotus, ja sitä kautta kokonaisnäkemys laadusta, voi olla hyvin erilainen eri tilanteissa.

2.1.3 Ohjelmiston laatuominaisuudet

Vaatimusmäärittelyssä ohjelmistolle asetetaan toiminnalliset ja ei-toiminnalliset vaatimukset. Ei-toiminnallisilla vaatimuksilla kuvataan usein laatuominaisuuksia, jotka ohjelmiston tulee täyttää. (Saleh, 2009, s. 71) Niitä ovat esimerkiksi:

- toiminnallinen sopivuus,
- luotettavuus,
- suoritustehokkuus,
- käytettävyys,
- yhteensopivuus,
- ylläpidettävyys ja
- siirrettävyys.

Yksittäisen toiminnallisen vaatimuksen toteutumattomuus on ehkä kierrettävissä muuttamalla toimintatapoja ohjelmiston käytössä. Sen sijaan puutteet laatuominaisuuksissa voivat aiheuttaa tuotteen kokonaisvaltaisempaa käyttökelpottomuutta. Laatuominaisuuksien ilmaisemiseen yksityiskohtaisesti ja määrällisesti tulisi pyrkiä vaatimusmäärittelyssä, mutta aina se ei ole mahdollista. Suoritustehokkuudelle ja luotettavuudelle on asetettavissa mittareita (kuten vasteaika ja virheiden esiintymistiheys), mutta ylläpidettävyys on esimerkiksi vaikeasti mitattavasta laatuominaisuudesta, jonka toteutumista on haastavaa todentaa hyväksymistestauksessa. (Sommerville, 2007, s. 124-125)

2.1.4 Laadunhallinta

Laadunhallinta johtamiseen liittyvä käsite. ISO 9000 -standardin mukaan laadunhallintajärjestelmä on ”johtamisjärjestelmä, jonka avulla suunnataan ja ohjataan organisaatiota laatuun liittyvissä asioissa”. *Laatujärjestelmä* on vastaava termi, joka on edelleen laajassa käytössä, vaikka se ISO 9000 -standardissa onkin korvattu laadunhallintajärjestelmä-käsitteellä. Laadunhallinta sisältää *laadunvalvonnan, laadunvarmistuksen, laadun suunnittelun ja laadunparantamisen*. Laadunvalvonta (josta käytetään myös termejä *laadunohjaus* tai *laaduntarkkailu*) tarkoittaa tekniikoita ja toimintoja, joiden avulla pyritään täyttämään laatuvaatimukset, kun taas laadunvarmistuksella tavoitellaan asiakkaiden ja viranomaisten luottamusta vaatimusten toteutumiseen. (Haikala & Märijärvi, 2006, s. 197)

Evans (2004, s. 102) kuvaa *laadunvarmistuksen prosessien ja toimintojen tarkistamisena* sekä *laadunvalvonnan tuotteeseen liittyvinä täydellisyys-, oikeellisuus- ja soveltuvuustarkistuksina*. Rajanveto ei ole yhtä tiukkaa kaikissa lähteissä. SWEBOK Guiden (2004, luku 10) mukaan ohjelmiston laadunvarmistusprosessit varmistavat tuotteen elinkaaren mittaisesti, että ohjelmistotuote ja prosessit vastaavat niille asetettuja vaatimuksia. Toisaalta Sommerville (2007, s. 653) katsoo laadunvalvonnan kohdistuvan myös prosesseihin, kun sen avulla tarkkaillaan, että laadunvarmistusmenettelyitä ja -standardeja noudatetaan.

2.2 Ohjelmistotuotannon prosessien laadunvarmistus

Ohjelmistotuotannon yhtenä erityispiirteenä on näkymättömyys. Keskenäisen ohjelmiston valmiusastetta tai sen tulevaa laatua on vaikea arvioida tuotteen itsensä perusteella. Siksi hyvät toimintakäytännöt ja laadukkaat prosessit ovat erityisen tärkeitä ohjelmistotuotannossa. Prosessin laadulla on yhteys tuotteen laatuun. (Haikala & Märijärvi, 2006, s. 29, 198)

Ohjelmistotuotannon laadun arviointiin, kehittämiseen ja hallintaan voi soveltaa yleiskäyttöisiä malleja, kuten Capability Maturity Model (CMM), European Foundation for Quality Management (EFQM) Excellence model, sekä standardeja, kuten ISO 9001. Malleja voi käyttää joko sellaisenaan, mukautettu-

na tai yhdistettynä. (Evans, 2004, 13-15; EFQM) Yksi esimerkki mukautuksesta on Ravichandranin ja Rain (2000) artikkelissa esitelty kokonaisvaltaiseen laatujohtamiseen (Total Quality Management, TQM) perustuva organisaatiolähtöinen laadunhallintamalli ohjelmistokehitykseen. Ohjelmistotuotantoon suunnatuista voisi mainita TickIT-mallin (TickIT, 2012) ja IEEE:n julkaiseman oppaan ISO 9001-standardin soveltamisesta ohjelmistotuotantoon (IEEE Guide 90003-2008, 2008; SWEBOK Guide, luku 10).

CMM-mallissa organisaation prosessien kyvykkyyden kypsyttä arvioidaan viisitasoisella asteikolla. Mallissa kuvataan kuhunkin tasoon liittyvät vaatimukset, jotka täyttämällä organisaatio voi parantaa prosessiensa laatua seuraavalle tasolle. Mallista kehitettiin versio myös ohjelmistotuotantoon, SW-CMM. (Herbsleb, Zubrow, Goldenson, Hayes & Paulk, 1997) Sitten on luotu aikaisempien versioiden pohjalta yhdistetty ja uudistettu malli Capability Maturity Model Integration (CMMI). CMMI:stä puolestaan on kolme versiota: ohjelmistokehitystä, hankintoja ja palveluita varten. (Software Engineering Institute, 2012)

2.3 Ohjelmistojen laadunvalvonta

Toiminnot, joilla ohjelmiston laatua valvotaan, jakautuvat ohjelmiston koko elinkaaren ajalle. Tavoitteena on verifioida (todentaa) ja validoida (kelpoistaa) ohjelmistotuote. *Verifioinnilla* varmistetaan, että dokumentaatio ja ohjelmointi on toteutettu vaatimusten, standardien ja hyvien käytäntöjen mukaisesti, eli on tehty tuotetta oikein. Verifiointin läpäisy takaa kuitenkin tuotteen käyttökelpoisuuden vain, jos siinä vertailukohtana käytetyt vaatimukset ovat valideja. *Validoinnilla* varmistetaan, että ohjelmisto (tai vaatimusdokumentaatio) vastaa sille asetettua tarkoitusta, eli on tehty oikeaa tuotetta. (Sommerville, 2007, s. 516; Haikala & Märijärvi, 2006, s. 268)

Laadunvalvontamenetelmät voidaan jakaa staattisiin ja dynaamisiin menetelmiin, jotka täydentävät toisiaan (Sommerville, 2007, s. 518). Staattisia ovat esimerkiksi tarkastus, katselmointi, läpikäynti ja vertaisarviointi. Niiden välillä on eroja käytäntöjen muodollisuusasteessa ja osallistujaryhmän laajuudessa, mutta peruseriaate on käydä dokumentaatio tai ohjelmakoodi läpi systemaattisesti lukien ja vertaamalla tarkistuslistaan tai muihin olemassa oleviin kriteereihin. (Haikala & Märijärvi, 2006, s. 268-269)

Testaus on dynaaminen laadunvalvontamenetelmä. Testauksessa ohjelmaa tai sen osaa suoritetaan, jolloin on mahdollista löytää myös sen tyyppiset ongelmat, jotka eivät tule ilmi vain koodia lukemalla. (Sommerville, 2007, s. 518). Testausta luonnehtii SWEBOK Guiden (2004, luku 5) määritelmän mukaan dynaamisuuden lisäksi

- testitapausten joukon äärellisyys (mahdollisia tapauksia olisi tavallisesti äärettömän paljon),
- valikoidut testitapaukset kattamaan erilaiset syötet ja tilanteet ja

- odotettu tulos, johon testitapauksella saatua tulosta verrataan.

Lähestymistapa testaukseen voi olla virheitä etsivä tai validoiva. Viime mainitulla voidaan varmistaa, että ohjelmiston toiminta on riittävän lähellä oikeaa käymällä läpi suunnitellut testitapaukset ja vertaamalla tulosta vaatimuksiin. Täydellistä virheettömyyttä ei voida osoittaa äärellisellä määrällä testitapauksia. (Sommerville, 2007, 538-539)

Testaus nivoutuu kaikkiin ohjelmiston toteutuksen vaiheisiin, vaikka se vaihemalleissa toisinaan nähdään erillisenä osana prosessia. Virheiden löytäminen ja korjaaminen mahdollisimman varhaisessa vaiheessa on toivottavaa. Mitä myöhemmin se tapahtuu, sitä työläämpää ja kalliimpaa korjaus on. Yksikkö- tai moduulitestaus kohdistuu yksittäisiin funktioihin, luokkiin tai komponentteihin. Yksikkötestaus etenee rinnakkain ohjelmoinnin kanssa. Integrointitestauksen tarkoituksena on varmistaa kahden tai useamman moduulin yhteen toimivuus. (Haikala & Märijärvi, 2006, s. 288-290.) Järjestelmätestauksessa tarkastellaan kokonaista ohjelmistoa. Se on myös vaihe, jossa ohjelmiston eitoiminnallisten vaatimusten toteutumista voidaan testata. (Saleh, 2009, s. 245.)

Ohjelmiston julkaisuun liittyy usein järjestelmätestauksia, joihin osallistuu myös asiakkaan edustajia tai käyttäjiä. Alfa-testaus toteutuu ohjelmistotoimittajan tiloissa ja beta-testauksen asiakkaat suorittavat itsenäisesti omassa käyttöympäristössään (Haikala & Märijärvi, 2006, s. 291). Hyväksymistestaus on julkaisuun liittyvä järjestelmätestaus, jonka perusteella asiakas hyväksyy tuotteen käyttöön (Sommerville, 2007, s. 541).

Erityisesti ylläpitovaiheeseen liittyy regressiotestaus. Sen avulla pyritään varmistamaan, että ohjelmisto toimii edelleen hyväksyttävästi sen jälkeen, kun siihen on tehty muutos. (Saleh, 2009, s. 280)

Järjestelmän käytettävyyden testaaminen voidaan erottaa muusta järjestelmätestauksesta omaksi kokonaisuudekseen tai sitä voidaan tehdä käyttöliittymäprototyypin avulla jo ennen kuin toimiva järjestelmä on olemassa. Käytettävyydestestauksessa käyttäjät suorittavat valvotussa koetilanteessa erilaisia ennalta suunniteltuja tehtäviä. Erityisessä käytettävyydelaboratoriossa on laitteistoa käyttäjien toiminnan rekisteröintiin. (Haikala & Märijärvi, 2006, s. 291)

Testaukseen tarvitaan merkittävä määrä organisaatioiden henkilöresursseja ohjelmistotuotannon eri vaiheissa. Testausautomaatiolla pyritään vähentämään testauksen aiheuttamaa työmäärää. Etenkin yksikkötestaus ja regressiotestaus soveltuvat automatisoitaviksi. (Haikala & Märijärvi, 2006, s. 297)

Sommervillen (2007, s. 562) mukaan ohjelmiston automaattinen testaus työkalu voi koostua esimerkiksi seuraavanlaisista osista:

- testien ajon hallinta,
- testitapausten generointi,
- oraakkeli (odotettujen tulosten generointi),
- tiedostovertailija,
- raporttigeneraattori,

- dynaaminen analysaattori (esimerkiksi testitapausten suorituskertojen laskenta) ja
- simulaattori (esimerkiksi useiden yhtäaikaisten käyttäjäistuntojen tai muun kuormituksen simulointi).

Erilaisia vaihtoehtoja myös järjestelmä- ja hyväksymistestauksen automatisointiin on kuvattu kirjallisuudessa. Puolitaival ja Kanstrén (2010) esittelevät mallipohjaisesta testausta (MBT, model-based testing), jossa testigeneraattorisovellus tuottaa testitapaukset testattavan ohjelmiston toimintaa kuvaavien sovel-lusaluekohtaisen mallinnuskielen avulla tuotettujen mallien pohjalta. Tapaukset voidaan suorittaa automaattisesti tai manuaalisesti. Korkealla abstraktiotasolla mallinnuksen vaatima työpanos on heidän mukaansa kustannustehokasta, minkä vuoksi menetelmä soveltuisi juuri järjestelmätestaukseen. Testitapausten generointia on raportoitu myös UML-aktiviteettikaavioiden pohjalta (Heinecke, Brückmann, Griebe & Gruhn (2010) ja käyttämällä sopivaa merkkausta olemassa olevaan dokumentaatioon (Connolly, Keenan & McCaffery, 2009).

2.4 Luvun yhteenveto

Tässä luvussa käsiteltiin laatua, sen ilmenemistä ohjelmistotuotteiden yhteydessä ja ohjelmistotuotannossa. Todettiin, että laatu on asiayhteydestä riippuva ja moniselitteinen käsite, johon sisältyy sekä objektiivisesti arvioitavia että subjektiivisia ominaisuuksia. Laadunhallinta on organisaation laatuasioiden johtamista, johon sisältyy muun muassa laadunvarmistus ja laadunvalvonta. Niiden avulla huolehditaan tuotantoprosessien ja tuotteen laadun pitämisestä hyväksyttävällä tasolla. Prosessien laadun arviointia ja kehittämistä tukemaan on useita malleja ja standardeja, kuten CMMI ja ISO 9001. Tuotteen laatua valvotaan esimerkiksi tarkastuksilla, katselmoinneilla ja testauksella.

Seuraavassa luvussa tarkastellaan asiakasorganisaation osallistumista ohjelmistotuotantoon.

3 ASIAKKAAN ROOLI JA TEHTÄVÄT OHJELMISTOTUOTANNOSSA

Tässä luvussa tehdään katsaus siihen, miten ohjelmistoja hankkivan ja käyttävän organisaation sekä sen edustajien osallistumista ohjelmistotuotantoon on kuvattu kirjallisuudessa. Rooliin ja tehtäviin vaikuttavat esimerkiksi ohjelmiston hankinta- ja toteuttamistapa sekä ohjelmiston elinkaaren vaihe. Luvussa tarkastellaan tutkimustuloksia asiakasorganisaation osallistumisesta uuden sovelluksen tuotantoon ketterässä ja suunnitelmapohjaisessa mallissa. Lisäksi esitellään tutkimuksia asiakkaan roolista valmisohjelmiston räätälöinnissä ja ylläpitovaiheen tehtävissä.

3.1 Käsitteitä ja taustatekijöitä

Tässä aliluvussa kuvataan käsitteitä ja taustatekijöitä, jotka liittyvät asiakasorganisaation rooliin ohjelmistotuotannossa. Lisäksi esitellään, miten käsitteitä on tutkielmassa tulkittu sekä sen vaikutusta aihepiirin rajaukseen.

3.1.1 Asiakas

Ohjelmistotuotantokontekstissa asiakkaat ovat monimuotoinen ryhmä. Evansin (2003, s. 31-32) mukaan he voivat olla esimerkiksi

- ihmisiä, jotka maksavat ohjelmistosta,
- ihmisiä, jotka käyttävät toimitettua ohjelmistoa,
- ohjelmiston käyttäjien asiakkaita,
- julkisen hallinnon yhteisöjä,
- organisaation muita yksiköitä,
- tuotteiden jälleenmyyjiä tai
- ohjelmistotuotanto-organisaation sisäisiä asiakkaita.

Tässä tutkielmassa *asiakkaalla* viitataan asiakasorganisaatioon, eli yritykseen tai yhteisöön, joka hankkii ohjelmistoja tukemaan toimintaansa. Ohjelmistotuotantoon osallistuvilla henkilöillä, *asiakkaan edustajilla*, voi olla organisaatioissa erilaisia rooleja, kuten ohjelmiston käyttäjä, päätöksentekijä tai tekninen asiantuntija (Evans, 2004, 18-20).

Näkemyks laadusta ja *asiakkaan saavuttamasta hyödystä* voikin vaihdella riippuen siitä, tarkasteleeko asiaa organisaatiossa ohjelmiston käyttäjiä, tietohallintoa, teknistä tukea, talousosastoa tai johtoa edustava henkilö. Kokonaisvaltaisempaa arviota tukee asiakasorganisaation arviointimalli, laatujärjestelmä tai -strategia, jossa eri näkökulmat on otettu huomioon. (Ruohonen & Salmela, 2005, 181-186)

Asiakkaan osallistumisella (ohjelmistotuotantoon) tarkoitetaan tässä raportissa yhteistyötä, jota asiakkaan edustajat tekevät ohjelmistotoimittajan kanssa ohjelmiston määrittelyn, suunnittelun, toteutuksen, laadunvalvonnan sekä käyttöönoton ja ylläpidon edistämiseksi.

3.1.2 Ohjelmistotuotetyypit ja ohjelmistojen hankinta

Ohjelmiston hankinta on vaihtelevanmittainen prosessi, johon sisältyy tarpeen määrittäminen, vaihtoehtojen kartoitus, tarjouspyynnön tekeminen, tarjousten käsittely ja sopimuksen tekeminen. Toisinaan myös käyttöönotto ja jatkokehittäminen katsotaan osaksi hankintaa. (Tietotekniikan liitto, 2005, s. 158)

Tarpeen ja vaihtoehtojen kartoituksen perusteella ratkaistaan hankittavan ohjelmistotuotteen tyyppi. Paketti- eli valmisohjelmisto on tuote, jonka oikeudet omistaa sen tuottanut ohjelmistoyritys. Ohjelmisto on yleensä tarkoitettu sellaisenaan käyttöönotettavaksi, mutta sen sovittaminen asiakkaan prosesseihin voi olla myös mahdollista. Räättälöinnin tarvetta on etenkin laajoissa toiminnanohjausjärjestelmissä (ERP, Enterprise Resource Planning). Ohjelmisto voidaan myös toteuttaa kokonaan tilaustyönä asiakkaan asettamien vaatimusten perusteella. (Sommerville 2007, 6-7; Brehm, Heinzl & Markus, 2001)

Ohjelmistotuotteen ostamisen sijaan organisaatiot voivat ulkoistaa toiminnot, joissa ohjelmistoa käytettäisiin (Ruohonen & Salmela, 2005, s. 55-56). Vaihtoehtona on myös hankkia ohjelmisto palveluna. Tällöin palveluntarjoaja ylläpitää sovelluksia sekä niiden käyttöympäristöä, ja asiakkaalla on tilausperusteinen käyttöoikeus. Tietotekniikan liiton hankintaoppaassa (2005, s. 19) käytetään ohjelmistopalvelusta käsitettä ASP, Application Service Provider. Samaan hankintatapaan liittyvä termi SaaS, Software as a Service, joka kuvaa jakelumallia eikä niinkään palveluntarjoajaa. SaaS-palveluna tarjottaviin ohjelmistoihin voi jossain määrin sisältyä myös asiakaskohtaisen räätälöinnin mahdollisuuksia (Li, Liu & Pan, 2012).

3.1.3 Ohjelmistoprosessi ja vaihejakomallit

Sommervillen (2007, s. 8) mukaan ohjelmistotuotannon keskeisiä prosessiaktiiviteetteja ovat ohjelmiston

- määrittely (vaatimustiedon hankinta ja analysointi)
- kehitys (suunnittelu ja toteutus),
- validointi (sen varmistaminen, että ohjelmisto täyttää tarkoituksensa)
- sekä evoluutio (vastaaminen ylläpidon aikana ilmeneviin muutostarpeisiin).

Ohjelmiston elinkaarta on perinteisesti kuvattu vesiputousmallilla, jossa vaiheet seuraavat toisiaan, niin että edellisen vaiheen dokumentaatio on hyväksymisensä jälkeen seuraavan vaiheen syöte. Vaatimusten määrittelystä edetään suunnittelun kautta toteutukseen ja sen jälkeen käyttöönottoon ja ylläpitoon. Jokaiseen vaiheeseen liittyy laadunvarmistustoimia, kuten katselmointia ja testausta. (Haikala & Märijärvi, 2006, s. 36-37.)

Evoluutio- ja spiraalimalleissa ohjelmistosta kehitetään nopeasti alustava versio, joka myöhemmin viimeistellään asiakkaalta saadun palautteen perusteella vastaamaan asiakkaan tarpeita. Lähtötilanteessa vaatimukset voivat olla hyvinkin abstraktilla tasolla ja ne täsmentyvät tuotekehitysprosessin aikana. (Sommerville, 2007, s. 65.) Kehittäminen voi edetä lisäysten kautta (inkrementaalinen malli) tai sykleissä, joiden jälkeen julkaistaan uusi versio (iteratiivinen malli).

Sommerville (2007, s. 65) listaa lisäksi komponenttipohjaisen ohjelmistotuotannon kolmantena prosessimallina. Tämän mallin mukaan toimittaessa uudelleen käytetään olemassa olevia komponentteja sen sijaan, että sovellus ohjelmoitaisiin alusta lähtien puhtaalta pöydältä. Merkittävä osa sovelluskehityksestä koostuu tällöin komponenttien integroinnista.

Malleista on käytössä useita muunnelmia ja niitä voidaan myös yhdistää samassa ohjelmistotuotantoprojektissa. Esimerkiksi RUP (Rational Unified Process) -prosessi perustuu iteraatioihin, joiden sisällä ohjelmistokehitys etenee vesiputousmaisesti. (Haikala & Märijärvi, 2006, s. 45-46).

Ketterissä menetelmissä sovelletaan iteratiivista mallia niin, että syklit ovat hyvin lyhyitä. (Haikala & Märijärvi, 2006, s. 42-47) Menetelmissä korostetaan myös perinteisestä, suunnitelmapohjaisesta ohjelmistokehityksestä poikkeavia periaatteita ja käytäntöjä: vuorovaikutus ja yksilöt ovat tärkeämpiä kuin prosessit ja työkalut, toimiva ohjelmisto on tärkeämpi kuin perusteellinen dokumentaatio, yhteistyö asiakkaan kanssa on tärkeämpää kuin sopimusneuvottelut ja muutokseen vastaaminen on tärkeämpää kuin suunnitelman noudattaminen. (Koch, 2004, s. 4)

Boehmin ja Turnerin (2004, s. 51) mukaan sekä suunnitelmapohjaisella että ketterällä mallilla on paikkansa. Suurten hankkeiden läpivienti ilman kattavaa dokumentaatiota aiheuttaa jo viestinnällisiäkin haasteita. Usein niissä tarvitaan ainakin joitakin suunnitelmapohjaisuuden piirteitä mukaan projektiin. Ketterät menetelmät taas soveltuvat hyvin silloin, kun projekti on suhteellisen pieni ja muutoksia vaatimukseen on odotettavissa.

3.2 Tutkimuksia asiakkaan roolista ohjelmistotuotannossa

Tässä aliluvussa tarkastellaan tutkimuksia, joissa asiakkaan ja käyttäjän osallistumista ohjelmistotuotantoon on raportoitu eri tyyppisissä tilanteissa ja ohjelmiston elinkaaren vaiheissa.

Asiakkaan osallistumisessa tuotekehitykseen yleisesti on Kaulion (1998) mukaan eri tasoja:

- kehitetty asiakasta varten (*designed for*), jolloin asiakkaalta kerätään tietoa suunnittelun pohjaksi,
- kehitetty asiakkaan kanssa (*designed with*), jolloin tiedon keräämisen lisäksi kerätään myös palautetta ja reagoidaan siihen sekä
- kehitetty asiakkaan toimesta (*designed by*), jolloin asiakas osallistuu aktiivisesti tuotekehityksen toteutukseen.

Kujala (2008) kuvaa vastaavat käyttäjien roolit ohjelmistotuotannossa siten, että heidän panoksensa ohjelmistotuotantoon voi olla informoivaa, konsultoivaa tai osallistuvaa.

3.2.1 Asiakkaan osallistuminen uuden ohjelmiston kehitykseen

Ohjelmisto kehitetään joko räätälöiden tietylle asiakkaalle tai valmisohjelmistona tuntemattomalle joukolla käyttäjiä. Tuotekehityksen lähtökohtana on tunnistettu tarve, johon ohjelmistolla pyritään vastaamaan. Tarve täsmennetään vaatimuksiksi. Vaatimusmäärittelyssä käytettävän tiedon lähteinä SWEBOK Guide (2004, luku 2) luettelee tavoitteet, sovellusalueen tietämyksen, osapuolet sekä käyttöympäristön teknisessä ja organisatorisessa mielessä. Haikalan ja Märijärven (2006, s. 94-95) kirjassa mainitaan markkinointi, oma organisaatio, asiakaspalaute, aikaisempien versioiden käyttäjät, ideointiaivoriihet, kilpailijoiden tuotteiden tutkiminen ja prototyypin rakentaminen keinoina hankkia asiakasvaatimuksia.

Asiakkaan tai käyttäjien rooli vaatimuslähteenä ja vaatimusten validoijana nousee usein esille myös tutkimusraporteissa. Toinen paljon tutkimusmielenkiintoa herättänyt alue on asiakkaan osallistuminen ketterien menetelmien projekteissa. Coramin ja Bohnerin (2005) mukaan suunnitelmapohjaisessa ohjelmistotuotannossa asiakkaan osallistuminen ajoittuu lähinnä prosessin alku- ja loppuvaiheisiin. Alussa asiakas on mukana vaatimusmäärittelyssä ja tuotteen käyttöönottoa edeltäen mahdollisesti alfa-, beta- ja hyväksymistestauksissa. Sen sijaan ketterään ohjelmistokehitykseen osallistuminen on asiakkaalle kokonaisvaltaisempi, pitkäkestoisempi ja myös vaativa tehtävä.

Kujala, Kauppinen, Lehtola ja Kojo (2005) tutkivat asiakaskontaktin vaikutusta ohjelmistoprojekteissa. Tutkimus toteutettiin kyselyinä ja haastatteluina ohjelmistoyritysten projektihenkilöille. Projektit toteutettiin useissa erikokoisissa yrityksissä ja ne olivat joko versioiteja tai kokonaan uuden sovelluksen

tuottamista. Kysymykset kartoittivat vaatimusmäärittelyn laatua ja projektin onnistumista, kun projektissa on suora kontakti käyttäjiin tai kontakti puuttuu. Tuloksena todettiin tilastollisesti merkittävä parannus sekä vaatimusmäärittelyn laadussa että projektin onnistumisessa, kun käyttäjät osallistuivat projektiin. Käyttäjien osallistuminen auttoi erityisesti tunnistamaan eroja erilaisten käyttäjä- ja asiakasryhmien välillä. Projekteissa, joissa käyttäjät eivät olleet mukana, tuotekehittäjät näkivät helpommin käyttäjäjoukon homogeenisena massana. Osallistuminen vaatimusmäärittelyyn on myös Luomansuun (2009) haastattelututkimuksen perusteella asiakkaiden tehokkain tapa vaikuttaa ohjelmistojen laatuun.

Majchrzakin (2010) tutkimus tehtiin saksalaisissa ohjelmistoalan yrityksissä puolirakenteisena haastatteluna ja sen tavoitteena oli löytää ohjelmistotestauksen hyviä käytäntöjä. Raportissa suositellaan esi- ja beta-julkaisujen testaamista pilottiasiakkailta ja ehdotetaan myös ohjelmistokehittäjien läsnäoloa asiakkaan tuotantoympäristössä pilottikäyttövaiheessa. Kujala (2008) sai samansuuntaisia tuloksia, kun hän selvitti ohjelmiston käyttöympäristössä kerätyn havaintomateriaalin hyödynnettävyyttä ohjelmistokehityksen tutkimuksessa: havaintona oli, että jalkautumalla kentälle tuotekehittäjät saavat tietoa tuotteen käyttökontekstista ja lisäksi oppivat ymmärtämään käyttäjien arvoja ja asenteita ja sitä kautta heidän käyttäytymismallejaan tuotteiden tulevana käyttäjinä. Kentätutkimusta hän arvioi lupaavaksi menetelmäksi, jossa haasteena on kuitenkin datan analysoinnin työläys. Tutkimuksessa keskityttiin prosessin varhaiseen vaiheeseen, jolloin informaatiota hankitaan lähinnä vaatimusten määrittelymiseksi.

Poralin ym. (2004) raporttia varten tehtiin kysely suomalaisille terveydenhuollon tietojärjestelmiä toimittaville yrityksille ja terveydenhuollon organisaatioille. Asiakasorganisaatiot osallistuivat selvityksen tekoaikana vastaajien mukaan usein vaatimusmäärittelyyn ja tulevaisuudessa sen katsottiin olevan yhä tärkeämpää. Ongelmaksi koettiin yhteisen kielen ja yhteisesti käytettävien työvälineiden löytäminen (s. 19). Käyttöönoton sujuvuutta katsottiin edistävän käyttäjien osallistuminen suunnitteluun jo varhaisessa vaiheessa. Käyttäjien todellisessa ympäristössä suorittamaa testausta ennen käyttöönottoa pidettiin myös toivottavana (s. 56).

Subramanyamin ym. (2010) tutkimuksessa selvitettiin käyttäjien osallistumisen vaikutusta sekä asiakkaiden että sovelluskehittäjien tyytyväisyyteen. Tutkimus kohdistui 117 projektiin, joissa toteutettiin sekä uusia sovelluksia että olemassa olevien ylläpitoa. Tulokseksi uuden tuotteen kehitysprojekteissa saatiin, että käyttäjien tyytyväisyys projektiin oli suurinta, kun osallistuminen oli vähäistä ja se laski loivasti, kun osallistumisen määrä kasvoi kohtalaiseksi ja suureksi. Sovelluskehittäjien tyytyväisyys projektiin oli päinvastainen: suuri käyttäjien osallistuminen korreloi korkean tyytyväisyyden kanssa tässä ryhmässä. Ylläpitoprojekteissa sekä käyttäjät että sovelluskehittäjät olivat tyytyväisimpiä kohtalaiseen käyttäjien osallistumiseen.

Tutkimuksissa tuli esille asiakkaan merkittävä rooli vaatimusten asettajana uuden ohjelmistotuotteen kehittämisessä. Yhteistyöstä kuvattiin

hyötyjä myös testaus- ja käyttöönottovaiheissa. Saatiin myös viitteitä siitä, että osallistumisen ja hyödyn suhde voidaan nähdä erilaisena ohjelmistoja tuottavassa ja niitä hankkivassa organisaatiossa.

3.2.2 Asiakasyhteistyö ketterässä ohjelmistokehityksessä

Ketterissä menetelmissä, etenkin Extreme Programmingissa (XP), korostetaan jatkuvaa kontaktia asiakkaaseen. Tyypillisiä ovat nopeat julkaisusykli ja joustava muutuskäsittely. Ketterään projektiin osallistuvan asiakkaan tulee tietää käyttäjien tarpeet, olla sitoutunut ja päätösvaltainen. (Coram & Bohner, 2005)

Wang, Wu ja Zhao (2008) kuvaavat asiakkaan tehtäviä XP-projektissa: asiakasryhmän kokoaminen, tilajärjestelyt, projektisuunnitelman katselmointi, palautteenanto, vaatimusten hallinta, konsulttina toimiminen, käyttöttestaukseen osallistuminen, toimivan ohjelmiston hyväksyminen jokaisen syklin yhteydessä, projektin edistymisen seuraaminen ja valmiin tuotteen hyväksyminen. Asiakkaan tulee lisäksi perehtyä projektin toimintatapoihin ja käytäntöihin, tarvittaessa sovelluskehittäjän opastamana. Raportin mukaan ketterässä projektissa asiakkaalla on hyvät mahdollisuudet vaikuttaa tuotteen ominaisuuksiin ja laatuun, mutta onnistuminen edellyttää sitoumista ja merkittävää ajankäyttöä asiakkaalta.

Wojciechowski, Wesolowski ja Complak (2010) tutkivat kokeellisesti paikalla ja etäällä olevan asiakkaan vaikutusta ketterässä ohjelmistokehityksessä. Kolmelletoista tiimille annettiin viisi ohjelmointitehtävää rajoitettuna aikana suoritettavaksi laboratorio-oloissa. Kullakin tiimillä oli asiakas. Ohjelmoijille annettiin ennalta tehtäviin liittyvää teknistä tietoa ja asiakkaina toimivat koehenkilöt saivat tietoa tehtävien liiketoiminta-alueista. Tehtävien ratkaiseminen edellytti tiedon jakamista. Kuuden tiimin asiakas oli samassa laboratorioissa ohjelmoijien kanssa ja loput seitsemän tiimiä pitivät yhteyttä asiakkaaseensa Skypeen ja sähköpostin välityksellä. Kokeessa mitattiin ohjelmien laatua vaatimusten täyttymismielessä sekä tehtävien ratkaisunopeutta ja määrää. Sekä tehtävien ratkaisutehokkuudessa että hyväksymistestien läpäisyssä selkeästi parempia tuloksia saivat tiimit, joilla oli läsnäoleva asiakas.

Korkalan, Abrahamssonin ja Kyllösen (2006) tapaustutkimuksessa havaittiin asiakkaan läsnäololla samansuuntainen vaikutus kuin Wojciechowskin ym. kokeessa. Vaikutusta mitattiin ohjelmavirheiden määrällä. Kun asiakas oli tiiviisti mukana projektissa ja viestintä oli vuorovaikutteista, virheitä muodostui vähiten. Tehokkainta oli keskustelu kasvokkain, mutta myös esimerkiksi videoneuvottelulla saatiin hyviä tuloksia.

Cao, Ramesh ja Abdel-Hamid (2010) mallinsivat tutkimuksessaan ketterää ohjelmistotuotantoa System Dynamics (SD) -simulaatiomallin avulla. Yhtenä osa-alueena heidän esittelemässään mallissa on asiakkaan osallistuminen. Alkuvaiheen suunnittelu ja jatkuva prosessinaikainen osallistuminen vaikuttavat asiakkaan luottamukseen. Osallistuminen edellyttää projektin käytänteiden ja sisällön oppimista, joka osaltaan edistää luottamusta.

Asiakkaan luottamus lisää tuotekehitystiimin asiakkaalta saaman palautteen laatua ja nopeutta sekä vaikuttaa sitä kautta asiakastyytyväisyyteen.

Bakalovan ja Danevan (2011) tutkimuksessa verrattiin asiakkaan osallistumista kahden yrityksen ohjelmistoprojekteissa, joista toinen noudatti ketterää menetelmää ja toinen perinteistä suunnitelmapohjaista ja laajoihin kertajulkaisuihin perustuvaa mallia. Asiakas osallistui molempien yritysten projekteihin tiiviisti, mutta eroa todettiin työkäytännöissä ja etenkin kontaktien suunnitelmallisuudessa. Kummankin yrityksen projekteissa asiakkaan osallistumisen todettiin lisäävän molemminpuolista luottamusta ja asiakastyytyväisyyttä.

Ketterän ja suunnitelmapohjaisen ohjelmistotuotannon erot asiakkaan roolissa näkyvät osallistumisen ajoituksessa, määrässä ja laadussa. Ketterässä ohjelmistotuotannossa asiakkaalta yleensä edellytetään tiivistä sitoutumista projektiin. Toisaalta asiakkaalla on myös laajat mahdollisuudet vaikuttaa koko tuotekehitysprosessin ajan myös tarkentamalla ja muuttamalla vaatimuksia.

3.2.3 Asiakas valmisohjelmiston käyttöönotossa ja räätälöinnissä

Toiminnanohjausjärjestelmän (ERP) tehtävänä on liiketoimintaprosessien automatisointi, yhteisen tiedon ja käytänteiden jakaminen, tiedon tuottaminen ja käyttö reaaliaikaisessa järjestelmässä (Nah, Lau & Kuang, 2001). ERP:n katsotaan usein sijoittuvan valmisohjelmistojen ja tilauksesta valmistettavien ohjelmistotuotteiden välimaastoon, koska se voi vaatia räätälöintiä soveltuakseen erityyppisten organisaatioiden toimintaprosesseihin. Räätälöinnit voidaan toteuttaa eri tavoin. Ohjelmistossa on voitu jo varautua erilaisten konfiguraatioiden tarpeeseen mahdollistamalla vaihtoehtoiset ominaisuudet parametroinnilla. Tällöin asiakas voi itse ottaa käyttöön tarvitsemansa asetukset. Osassa tapauksia räätälöinti voi kuitenkin edellyttää ohjelmointia, joko itse ERP:n koodiin puuttumista tai erillisen lisäosan rakentamista. Jälkimmäisen voi tehdä myös kolmas osapuoli. Räätälöinnit lisäävät ongelmien riskiä ylläpitovaiheessa. Versiopäivitysten yhteydessä ne vaativat lisähuomiota laadunvarmistuksessa tai lakkaavat mahdollisesti kokonaan toimimasta. (Brehm ym., 2001)

Nah ym. (2001) etsivät tutkimuksessaan onnistuneen ERP-käyttöönoton kriittisiä menestystekijöitä. Niitä tunnistettiin kirjallisuuskatsauksen perusteella 11, joista tärkeimmiksi tutkijat katsoivat laaja-alaisen ja hyvin yhteentoimivan käyttöönottotiimin sekä muutoksenhallinnan. ERP-tiimin tulisi koostua sekä asiakasorganisaation omasta henkilöstöstä että konsulteista ja sisältää sekä teknistä että toimialan asiantuntemusta. Tiimillä tulee olla ylimmän johdon tuki. Liiketoimintaprosessien uudelleentarkastelu ja sovittaminen vähentää ohjelmiston räätälöinnin tarvetta. ERP:n arkkitehtuurin huolellinen etukäteissuunnittelu ja sen sovittaminen yrityksen tietojärjestelmien kokonaisarkkitehtuuriin on tärkeää. Vaatimusmäärittelyä ja testausta voidaan tarvita tässä sovittamisessa, aikaisemman datan migraatiossa sekä ERP:n mahdollisessa integroinnissa yrityksen muihin ohjelmistoihin, vaikka

räätälöinnin tarve pyrittäisiinkin minimoimaan. Tuotekehitys, testaus ja vianetsintä olikin yksi tunnistetuista menestystekijöistä käyttöönottoprojektivaiheessa.

Kumarin, Maheshwarin ja Kumarin (2002) tutkimus tehtiin kyselynä ERP-projektista tai -ohjelmasta vastaavalle henkilölle kymmenessä kanadalaisessa julkisen sektorin organisaatiossa. Käyttöönottoimien koostumus mukaili Nahin ym. (2001) kuvaamaa. Osassa organisaatioita tiimeihin osallistui myös laitetoimittajien edustajia. Ylin johto oli tiiviisti mukana puolessa tiimeistä. Ohjelmiston tuotantoonottokelpoisuuden varmistamiseen organisaatioissa käytettiin erillistä testausympäristöä, keskeisten käyttäjäryhmien suorittamaa pilotointia, testauskriptien ajamista sekä datan validointia ja verifiointia vanhan ja uuden järjestelmän välillä.

Laajan valmisohjelmiston sovittaminen asiakkaan toimintaympäristöön voi edellyttää räätälöintiä. Se toteutetaan usein asiakasorganisaation ja ohjelmiston toimittajan tai tämän alihankkijana toimivan yrityksen yhteistyönä, johon asiakas osallistuu vähintään päätöksenteko- ja sovellusalueen asiantuntijaroolissa. Asiakkaan teknisen infrastruktuurin tuntemusta tarvitaan myös joko asiakasorganisaatiolta itseltään tai sen mahdollisesti käyttämältä palveluntarjoajalta. Prosessiin sisältyy samoja vaiheita kuin uuden ohjelmistotuotteen kehittämisessä: vaatimusten määrittelyä, suunnittelua, toteutusta ja testausta. Lisähaasteena on usein integrointi organisaation muihin järjestelmiin sekä mahdolliseen aikaisempaan järjestelmään tallennetun datan käsittely.

3.2.4 Asiakas ja ohjelmiston ylläpitovaihe

Ohjelmistokehityksessä ylläpitotoimilla mahdollistetaan ohjelmiston elinkaaren jatkuminen korjaamalla virheitä, täydentämällä ohjelmistoa uusilla ominaisuuksilla tai sopeuttamalla sitä ympäristön muuttuneisiin vaatimuksiin. (Haikala & Märijärvi, 2006, s. 41). Ylläpitovaiheessa ohjelmistokehittäjän näkökulmasta on ohitettu intensiivisimmät työvaiheet ja projektiryhmä on ehkä hajaantunut. Käyttöönotto ja sen jälkeinen aika kuitenkin antavat mahdollisuuksia arvioida projektin onnistumista kaikkien osapuolten näkökulmasta, myös ohjelmistokehittäjien ja testaajien. Asiakkaan kannalta taas toimituksen jälkeinen tila on se, johon koko hanke tähtäsi. (Evans, 2004, s.229-233)

Asiakkaan osallistumisessa ohjelmistokehitykseen ylläpitovaiheessa on nähtävissä samoja piirteitä kuin uuden tuotteen kehittämisessä. Tarpeen korjaukselle, täydennykselle tai muutokselle havaitsee ja ilmaisee usein asiakkaan edustaja, eli asiakas on vaatimustiedon lähde myös käyttöönoton jälkeen. Hyväksytty muutospyyntö käynnistää ylläpitoprosessin, johon sisältyy analyysi, suunnittelu, toteutus, testaus ja uuden version jakelu (IEEE 1219, 1998).

Tuotekehitysehdotuksia voidaan kerätä myös aktiivisesti asiakkailta. Kabbedijk, Brinkkemper, Jansen ja van der Veldt (2009) raportoivat tapaustutkimuksen suuren ohjelmistoyrityksen tavoista hankkia ja hallita

asiakkailta saatua vaatimustietoa. Asiakkaat ovat yrityksiä ja yhteisöjä. Tutkitussa ohjelmistoyrityksessä kirjataan kaikki käyttäjätukeen tulevat yhteydenotot tapausraporteiksi, joista noin 15 % johtaa tuotekehityspyyntöön. Lisäksi yritys järjestää asiakastilaisuuksia, joissa asiakkaiden edustajat voivat ottaa äänestämällä kantaa olemassa oleviin tuotekehitysehdotuksiin tai tehdä uusia ehdotuksia.

On myös menetelmiä, joilla kerätään tietoa asiakkaan käyttöympäristön tapahtumista tavanomaisen käytön yhteydessä ilman aktiivisia palautemenettelyitä. In vivo -testauksella tarkoitetaan ohjelmiston elinkaarenlaajuista yksikkötestien ajamista, niin että niitä suoritetaan myös käyttöönoton jälkeen tuotantoympäristössä. Tavoitteena on saada tietoa ohjelmiston toiminnasta erilaisissa käyttöympäristöissä. Ongelmana menetelmässä on sen kuormittavuus. Hajautetussa in vivo -testauksessa testit ajetaan usean eri asennuspaikan sovellusryhmälle, niin että niiden määrä pysyy alkuperäisenä, mutta kuormitus ei kohdistu vain yhteen sovelluksen asennuspaikkaan. (Chu, Murphy & Kaiser, 2009) Käytönaikaista tietoa voidaan kerätä myös analysoimalla lokeja (Thakkar, Zhen, Hassan, Hamann. & Flora 2008) ja välittämällä tietoa sovellusten käytöstä ja käyttöympäristöstä automaattisesti ohjelmistotoimittajalle, kuten Microsoftin Käyttömukavuuden kehittämishohjelmassa (Microsoft CEIP, 2012).

Hyvinkin määritely ja toteutettu ohjelmisto tarvitsee käyttöönoton jälkeen ylläpitotoimia. Ohjelmistoa voidaan joutua mukauttamaan organisaation toiminnan tai esimerkiksi lainsäädännön muutoksiin. Sitä voidaan täydentää suunnitellusti tai korjata havaittuja virheitä. Muutostarpeita voidaan havaita keräämällä asiakkailta palautetta ja käytönaikaista informaatiota. Muutostarpeen määrittelyssä ja uusien versioiden testauksessa asiakkaan rooli usein muistuttaa uuden tuotteen kehittämiseen osallistumista, tosin pienemmässä mittakaavassa.

3.3 Luvun yhteenveto

Tässä luvussa käsiteltiin asiakkaiden ja ohjelmistotoimittajien yhteistyön muotoja, kun kehitetään uusia ohjelmistoja, otetaan käyttöön valmisohjelmistoja tai ylläpidetään asiakkaan hankkimia sovelluksia. Todettiin, että asiakkaan edustajia osallistuu sekä suunnitelmapohjaisiin että ketteriin projekteihin. Ketterät menetelmät ovat kuitenkin muuttaneet roolia kokonaisvaltaisemmin osallistuvan projektiryhmän jäsenen suuntaan sekä siten, että tehtävät jakautuvat tasaisesti koko tuotekehitysprosessin ajalle. Asiakkaiden osallistumisesta on raportoitu hyötyjä, mutta näkemykset asiakkaan resursoinnin määrän ja hyödyn suhteesta eivät välttämättä ole asiakas- ja ohjelmistotuotanto-organisaatioissa aina yhtenevät.

Seuraavassa luvussa pohditaan, löytyikö käsitellyistä artikkeleista vastauksia tutkimuskysymyksiin sekä arvioidaan tutkielmaa ja esitetään jatkotutkimusehdotuksia.

4 POHDINTA JA ARVIOINTI

Tässä luvussa etsitään yhteyksiä ohjelmiston laadun ja asiakkaan osallistumisen väliltä. Toisaalta pohditaan tutkimuskysymyksen mukaisesti, millainen osallistuminen on asiakkaan näkökulmasta hyödyllistä. Lisäksi arvioidaan tutkielmaa ja esitellään jatkotutkimusehdotuksia, joita sen kirjoittamisen aikana heräsi.

Motiiveja käyttäjien tai asiakkaan edustajien ohjelmistotuotantoon osallistumiselle on Kujalan (2008) mukaan kuvattu useissa tutkimuksissa, muun muassa Muller ym. (1997) listaavat demokratian, tehokkuuden ja laadun parantamisen sekä sitouttamisen. Damodaranin (1996) mukaan hyötyjä ovat järjestelmän parempi laatu asianmukaisen vaatimusmäärittelyn ansiosta, mahdollisuus säästää tunnistamalla tarpeettomat ominaisuudet, järjestelmän hyväksymisen parempi taso, sekä tehokkaampi käyttö, kun käyttäjät ymmärtävät järjestelmää paremmin.

4.1 Vaikutukset laatuun ja asiakkaan kokemaan hyötyyn

Asiakkaiden osallistumisella ohjelmistotuotantoon raportoitiin pääsääntöisesti positiivisiksi tulkittavia tuloksia. Suuntaus on sama myös raporteissa esiteltyjen kirjallisuuskatsausten perusteella. Negatiivisiakin vaikutuksia on tosin esitetty, esimerkiksi Heinbokelin, Sonnetagin, Fresen, Stolten ja Brodbeckin (1996) mukaan käyttäjien osallistuminen lisää tuotekehitysprojektin vaikeutta, pituutta ja tehottomuutta. Hawk ja Dos Santos (1991) raportoivat tuloksia, joiden mukaan käyttäjien osallistumisen hyödyllisyys oli osoitettavissa vain joissakin tilanteissa riippuen esimerkiksi sovelluksen tyypistä ja osallistujan asemasta organisaatiossa. Edellä kuvatut tutkimukset ovat tosin melko vanhoja.

Taulukkoon 1 on koottu tätä tutkielmaa varten tarkastelluista raporteista ne, joissa esiteltiin käyttäjien osallistumista ja sen vaikutusta tai merkitystä selvittävän empiirisen tutkimuksen tuloksia.

TAULUKKO 1 Yhteenveto asiakkaan osallistumisen vaikutusta kuvaavista havainnoista

Tutkimus	Menetelmä	Kohde	Malli	Vaihe	Havainnot
Bakalova & Daneva (2011)	tapaus-tutkimus	o	ketterä ja suunnitelmapohjainen	projekti	Yhteistyöllä oli positiivinen vaikutus asiakastyytyväisyyteen molemmissa.
Cao ym. (2010)	simulatio	o, a	ketterä	projekti	Projektin käytänteiden oppimisella, luottamuksella ja asiakastyytyväisyydellä on yhteys.
Korkala ym., (2006)	tapaus-tutkimus	o	ketterä	projekti	Asiakkaan läsnäolo ja kasvokkaisviestintä vähentävät ohjelma- virheitä.
Kujala (2008)	kysely ja haastattelu	o	ei määritelty	vaatimusm.	Asiakkaan osallistumisen paransi vaatimusten laatua ja edisti projektin onnistumista.
Luomansuu (2009)	haastattelu	o	useita, painottuen ketterään	useita	Asiakkaalla on suurin mahdollisuus vaikuttaa laatuun vaatimusmäärittelyssä (69 % vastaajista).
Lynch & Gregor (2004)	haastattelu	o, a	useita	useita	Ohjelmistot, joiden kehittämiseen käyttäjien osallistumisella oli vaikutusta, saivat suuremman markkina- osuuden.
Majchrzak (2010)	haastattelu	o	ei määritelty	testaus	Asiakas voi vaikuttaa tuotekehitykseen ja pääsee tutustumaan alan uusimpiin tuot-teisiin osallistuessaan pilottikäyttöön.
Majid ym. (2010)	kysely	o	ei määritelty	useita	Elinkaarenlaajuinen osallistuminen tärkeää
Porali ym. (2004)	kysely	o, a	useita	useita	Asiakkaalla oli parhaat vaikutusmahdollisuudet vaatimusten ja testauksen kautta.
Subramanyam ym. (2010)	kysely	o, a	ei määritelty	useita	Kohtalainen asiakkaan osallistuminen oli hyödyllistä projektin lopputuloksen kannalta.
Wojciechowski ym. (2010)	kokeellinen tutkimus	o, a	ketterä	koe-tilanne/ projekti	Koetiimit, joilla oli läsnä oleva asiakas, suorittivat enemmän tehtäviä ja saivat parempia tuloksia hyväksymistesteissä.

Kohde: tutkimuksen kohderyhmä, o= ohjelmistoyritys, a= asiakasorganisaatio

Taulukosta 1 voidaan havaita, että useimmissa tutkimuksissa kohderyhmänä on ollut ohjelmistoyritysten henkilöstö, ja voidaankin aiheellisesti kysyä, ovatko raportoidut positiiviset löydökset silloin myös asiakkaan kokemia hyötyjä. Kuten luvussa 2 esiteltiin, laatua voidaan arvioida eri tavoin asiayhteydestä riippuen. Kun mittarina on asiakastyytyväisyys, on todennäköistä, että hyöty on ainakin jossakin määrin tunnustettu myös asiakasorganisaatiossa. Tuotteen kaupallinen menestys voi kertoa samaa, etenkin, jos kyseessä on tuote, jolla on melko laajat markkinat ja ostajalle on realistisia vaihtoehtoja tarjolla. Kun ohjelmistokehittäjä toteaa vaatimusten laadun parantuneen, voidaan vain olettaa, että hyöty tulee näkyviin myös asiakkaalla esimerkiksi nopeammin valmistuneena ja paremmin tarkoitustaan vastaavana tuotteena.

Kyselyjen, haastattelujen ja tapaustutkimusten ohella mukana on kaksi artikkelia, joissa kysymyksiä lähestytään kokeen ja simulaation kautta. Kohde on niissä hieman vaikeampi tulkita, mutta tässä on katsottu sen kuvaavan molempia osapuolia. Koeasetelmassa oli sekä ohjelmointi- että hyväksymistestausmittarit. Simulaatiota varten taas dataa oli koottu muun muassa ohjelmistoprojek-teista, joissa oli mukana asiakkaan edustajia.

Subramanyamin ym. (2010) tutkimuksessa saatujen tulosten valossa on myös syytä varoa vetämästä sen suuntaisia johtopäätöksiä, että ohjelmistokehittäjän organisaatiosta raportoitu hyöty olisi aina myös asiakkaan hyöty. Heidän tutkimuksessaan asiakkaan tyytyväisyys projektiin heikkeni osallistumisen kasvaessa, mutta ohjelmistokehittäjien tyytyväisyys samassa tilanteessa kasvoi. Tutkijat pohtivat, voisiko tuloksen taustalla olla konfliktien riski eri henkilöryhmien välillä, kun yhteistyö on tiivistä. Toisaalta käyttäjien odotukset tuotekehitysprojekteissa voivat kasvaa suhteessa osallistumiseen ja myös tämä selittäisi heikompaa tyytyväisyyttä runsaan osallistumisen yhteydessä, jos projektin tuotokset eivät vastaakaan lisääntyneitä odotuksia.

Lynch ja Gregor (2004) huomauttavat, että tutkittaessa käyttäjien osuutta ohjelmistotuotannossa pitäisi mitata vaikutusta eikä pelkkää osallistumista. Käyttäjien osallistumisella on vain vähän arvoa, jos heidän näkemyksensä eivät lopulta vaikuta siihen, millainen ohjelmistosta muotoutuu.

Bakalovan ja Danevan (2011) vertailututkimuksen perusteella asiakastyytyväisyys lisääntyi sekä suunnitelmapohjaisen että ketterän mallin mukaan toimittaessa, kun asiakkaat osallistuivat tiiviisti ohjelmistokehitysprojektiin. Ainakin tämän tutkimuksen perusteella voitaisiin päätellä, että osallistumisella sinänsä on mallia suurempi merkitys silloin, kun asiakkaan osallistuminen on riittävää. Sen sijaan ketterät menetelmät ovat haavoittuvampia tilanteessa, jossa asiakkaalla ei ole osoittaa tarpeeksi tai sopivia resursseja ketterään projektiin. Asiakkaan edustajan tulee olla paitsi tavoitettavissa tai mielellään kokopäiväisesti paikalla projektiryhmän jäsenenä (Koch, 2004, s. 18-19) lisäksi olla henkilö, joka aidosti pystyy toimimaan tiedonvälittäjänä oman organisaationsa käyttäjien ja sovelluskehittäjien välillä (Boehm & Turner, 2004, s. 32). Ramesh, Cao ja Baskerville (2010) tunnustivat myös asiakkaan heikon osallistumisen aiheuttamat puutteet vaatimusten laadussa yhdeksi vaikeimmin hallittavista riskeistä

ketterässä vaatimusmäärittelyssä. Toinen vakava riski oli ei-toiminnallisten vaatimusten huomiotta jättäminen, joka havaitsivat myös Majid, Noor, Adnan ja Mansor (2010).

4.2 Osallistumisen hyödyt suhteessa ohjelmiston elinkaareen

Vaatimusten määrittely esiintyy useassa tutkimuksessa vaiheena, jossa asiakkaan osallistumisesta on hyötyä. Luomansuun (2009) kyselyn vastaajista jopa 69 % oli sitä mieltä, että vaatimusmäärittely on asiakkaan paras mahdollisuus vaikuttaa ohjelmiston laatuun. Samassa tutkimuksessa vaikutusmahdollisuuksina nousivat esiin palaute, hyväksymistestaus ja projektinjohdollinen osallistuminen, jotka laajentavat osallistumisen koko tuotekehityksen ajalle, ja ainakin palautteen osalta myös ylläpitovaiheeseen. Vastaajat eivät kuitenkaan olleet asiakkaan edustajia.

Kuten edellä on todettu, ketterässä ohjelmistokehityksessä asiakkaan osallistuminen ajoittuu koko projektin ajalle, eikä vastaavia vaiheita kuin suunnitelmepohjaisessa prosessissa ole aina edes iteraatiosykliden sisältä tunnistettavissa. Asiakkaalle voidaan määritellä tehtäviä, kuten käyttötapausten laatiminen ja hyväksymistestaus (Wang, Wu & Zhao 2008), mutta voi olla vaikea osoittaa laatuvaikutusten yhteyttä tehtävien tasolle.

Perinteistä vaihejakomallia vasten tarkasteltuna suunnittelu- ja toteutusvaiheiden ajalle tutkimuksissa ei juurikaan ole raportoitu asiakasyhteistyötä. Tähän lienee luonnollisena syynä se, että nämä tyypillisesti sisältävät eniten ohjelmistoteknistä osaamista vaativia tehtäviä, ja toisaalta kaupallisten ohjelmistojen lähdekoodit voidaan haluta pitää salassa asiakkailta.

Grahamin (2002) artikkelin mukaan jo vaatimusten määrittelyvaiheessa tulee suunnitella testausta. Hän esittää, että tarve viime hetken muutoksiin vähenee, kun testaajat ja myös käyttäjät ovat mukana valmistelemissa hyväksymistestejä prosessin varhaisessa vaiheessa. Kun ohjelmistoprosessi mukailee vesiputousmallia, muutosten käsittely on työlästä. Asiakkaan edustajan rooli testauksessa voisikin olla enemmän suunnitteleva kuin testausta toteuttava, etenkin jos testausautomaatio edelleen kehittyy.

Etenkin toiminnanohjausjärjestelmien käyttöönottoihin sisältyy paljon tehtäviä asiakkaan organisaatiossa, kuten testi- ja tuotantoympäristöjen asennuksia, konversioita, integraatioita organisaation muihin järjestelmiin ja käyttäjien koulutusta (See Pui Ng, 2001). Tilanteet ovat ehkä asiakaskohtaisia ja kertaluonteisia, mutta jos niitä tehdään yhteistyössä ohjelmistotoimittajan kanssa, on ehkä silti mahdollista kerätä myös yleistettävää tietoa ohjelmiston jatkokehittämistä varten. Ylläpidon aikana yhteistyö on edelleen tärkeää. Virheilmoitusten, kehittämissuositusten ja palautteen avulla ohjelmistotoimittaja voi suunnitella ylläpitotoimia.

4.3 Tutkielman arviointia ja jatkotutkimusehdotuksia

Tavoitteena oli hakea tutkielmaa varten tutkimustietoa asiakasorganisaation roolista ohjelmistotuotannossa ja osallistumisen vaikutuksesta asiakkaan kokemaan ohjelmiston laatuun. Tutkimusraportteja, joissa olisi ollut nämä molemmat näkökulmat, löytyi lopulta varsin vähän. On joko niin, että asiakkaan näkökulmaa ohjelmiston laatuun ja siihen vaikuttamiseen on tutkittu vähän tai haku ei ollut riittävän kattava. Vaihtoehtona olisi ollut laajentaa hakua (mitä jonkin verran tehtiinkin) tai päätyä tekemään systemaattinen kirjallisuuskatsaus rajallisesta valikoimasta julkaisuja. Ongelmaksi systemaattisessa katsauksessa olisi kuitenkin muodostunut julkaisujen valinta ja työn laajuus. Riittävän kattavan katsauksen tekeminen tähän työhön varatuilla resursseilla ei ehkä olisi ollut mahdollista.

Tästäkin aineistosta voitiin silti löytää ainakin osavastauksia tutkimuskysymyksiin: onnistuttiin kartoittamaan osallistumisen muotoja ja niiden asettumista ohjelmiston elinkaaren ajalle. Asiakkaan roolin merkittävyys vaatimuskäytännönä ei yllättänyt. Osallistumisesta testaukseen oli raportoitu varsin vähän, poikkeuksena tästä tosin ketterien menetelmien hyväksymiskäytännöt. Asiakkaan roolista käyttöönotossa ja ylläpidossa materiaalia oli myös niukasti, vaikka asiakkaan kannalta nämä ovat merkittäviä vaiheita käyttäjä- ja talousnäkökulmasta sekä myös teknisen ympäristön ylläpidon ja tuen kannalta, jos ohjelmistot asennetaan asiakkaan omaan palvelin- ja työasemainfrastruktuuriin.

Aihepiiriä voitaisiin tutkia lisää tekemällä kysely-, haastattelu- tai tapaus- tutkimuksia, joiden kohteena on sekä ohjelmiston toimittaja että erityisesti asiakasorganisaatio. Laatua voitaisiin asiakasorganisaation sisälläkin tarkastella eri ryhmien kannalta: esimerkiksi käyttäjien, teknisten asiantuntijoiden ja johdon edustajien. Tätä kautta olisi mahdollista tunnistaa erilaisia tapoja mitata hyötyä sekä laadun suhdetta tuottavuuteen. Kirjallisuuskatsauksen laajentaminen käyttäytymis- ja taloustieteiden tietokantoihin voisi olla yksi mahdollisuus löytää aikaisempia tutkimuksia käyttäjä- ja asiakasnäkökulmasta.

Asiakkaan ja ohjelmistotoimittajan yhteistyötä käyttöönotto- ja ylläpito- vaiheessa sekä sen vaikutusta ohjelmistoon voitaisiin valottaa. Erityisesti laajojen toiminnanohjausjärjestelmien käyttöönottoon liittyy asiakkaan suorittamaa testausta, jolla varmistetaan ohjelmiston soveltuvuutta käyttöympäristöönsä ja organisaation prosesseihin (See Pui Ng, 2001). Havaitaanko näissä testauksissa oheislöydöksenä ohjelmavirheitä, joiden korjaamisesta hyöty on laajempi kuin yhteen asiakasorganisaatioon kohdistuva? Mäntylä, Itkonen ja Iivonen (2012) tutkivat testaaajien ammatillista taustaa ohjelmistoyritysten sisällä. He havaitsivat, että asiakkaan toimintaa tuntevat testaaajat (kuten sovellustukihenkilöt ja konsultit) löytävät enemmän ohjelmiston käytön kannalta merkittäviä virheitä, vaikka ammattimaiset testaaajat löytävätkin virheitä lukumääräisesti eniten. Tähän tulokseen peilaten voisi esittää hypoteesin, että asiakkaiden omilla löydöksillä on vastaava käytännöllinen merkitys.

Olisi myös kiinnostavaa kartoittaa yhteistyön välineitä ja menetelmiä. Millaisilla prosesseilla ja informaatioteknisillä apuvälineillä ohjelmistotoimittaja ja eri asiakasorganisaatiot voisivat yhdessä hyödyntää esimerkiksi toistensa testauksissa kerättyä tietoa tai muuten jakaa eri osapuolia hyödyttävää tietämystä? Lisäksi, kun pilvipalveluina tarjottavat hyötyohjelmat ovat yleistyneet, myös tältä alueelta voisi löytyä jatkotutkimusaiheita: miten asiakas-toimittaja-yhteistyö ja sen vaikutus laatuun ilmenee ohjelmistojen SaaS-jakelutavassa? Toisaalta, voisiko pilvipalveluna toteuttaa yllä mainittuja yhteistyön apuvälineitä, esimerkiksi eri organisaatioiden yhteisiä testaus- ja tietämyksenhallintaympäristöjä?

5 YHTEENVETO

Tässä tutkielmassa käsiteltiin ohjelmiston laatua asiakasorganisaation näkökulmasta. Tutkielman tavoitteena oli kartoittaa asiakasorganisaation roolia ohjelmistotuotannossa ja selvittää, onko kirjallisuudessa raportoitu asiakkaan osallistumiseen liittyviä vaikutuksia tuotteen laatuun.

Laatu-käsitteen tulkintaan vaikuttavat tilanne ja asiayhteys. Laadunhallinta on usein ohjelmistotoimittajan ja asiakkaan yhteistyötä. Ohjelmistoyritykset ylläpitävät tuotantoprosessiensa laatua (laadunvarmistus). Yhteisissä projekteissa voidaan noudattaa myös asiakkaan laatujärjestelmää. Laadunvalvonnalla varmistetaan etenkin tuotteen laatua suorittamalla tarkastuksia, katselmoitteja ja testausta.

Etenkin ketterässä ohjelmistokehityksessä pidetään tarpeellisena, että asiakas on tiiviisti mukana koko tuotekehitysvaiheen ajan. Asiakkaan osallistumisesta on raportoitu hyötyjä myös suunnitelmapohjaista mallia sovellettaessa.

Vaatumusten määrittelyä pidetään yleisimmin vaiheena, jossa asiakkaan vähintään tulisi osallistua, mutta yhteistyö jakautuu laajemminkin ja siitä on todettu hyötyjä myös muissa ohjelmiston elinkaaren vaiheissa. Asiakkaan edustajat voivat testata järjestelmiä omassa tai ohjelmistotoimittajan ympäristössä. Asiakas testaa tavallisimmin kokonaista ohjelmistoa järjestelmä- tai hyväksymistestauksena. Käyttöönotto- ja ylläpitovaiheiden työmäärät ja kustannukset ovat merkittäviä asiakasorganisaatioissa (See Pui Ng, 2001). On tärkeää, että yhteistyö jatkuu myös näihin vaiheisiin saakka. Vaiheen tarkempi tutkiminen esimerkiksi yhteistyön muotojen ja välineiden kannalta olisi myös tarpeellista.

Kirjallisuuskatsauksesta oli pääteltävissä, että asiakkaan osallistumisella on pääsääntöisesti positiivinen vaikutus ohjelmiston laatuun. Suuri osa kyselyistä, haastatteluista ja tapaustutkimuksista oli kuitenkin tehty ohjelmistoyrityksissä. Tuloksia tulkittaessa on huomioitava, että laatu-käsitteessä on myös subjektiivinen komponentti, eivätkä eri tilanteissa ja asiayhteyksissä tehdyt laatuarviot ole aina suoraan vertailtavissa. Lisätutkimuksena olisikin hyvä tehdä vastaavaa kartoitusta painottuen enemmän asiakasorganisaation näkemyksiin.

LÄHTEET

- Bakalova Z. & Daneva M. (2011) A comparative case study on clients participation in a 'traditional' and in an agile software company. *12th International Conference on Product Focused Software Development and Process Improvement, June 20* (s. 74-80). Torre Canne: ACM.
- Boehm B. & Turner R. (2003). *Balancing agility and discipline: a guide for the perplexed*. Boston: Addison-Wesley
- Brehm L., Heinzl A. & Markus M. L. (2001). Tailoring ERP Systems: A Spectrum of Choices and their Implications. Teoksessa *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, January 3-6*, IEEE Computer Society
- Cao, L., Ramesh, B. & Abdel-Hamid, T. (2010). Modeling dynamics in agile software development. *ACM Transactions on Management Information Systems, 1(1)*, artikkeli 5, 26 sivua
- Chu M., Murphy C. & Kaiser G. (2008). Distributed In Vivo Testing of Software Applications. *2008 International Conference on Software Testing, Verification, and Validation, April 9-11* (s. 509-512). Lillehammer: IEEE Computer Society
- Connolly D., Keenan F. & McCaffery F. (2009). Developing acceptance tests from existing documentation using annotations: An experiment. *ICSE Workshop on Automation of Software Test, AST'09, May 18-19* (s. 123-129). Vancouver: IEEE Computer Society
- Coram M. & Bohner S. (2005). The Impact of Agile Methods on Software Project Management. *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, April 4-7* (s. 363-370).
- Damodaran, L. (1996). User involvement in the systems design process - a practical guide for users. *Behaviour & Information Technology, 15*, 363-377.
- EFQM (2010). European Foundation for Quality Management Haettu 6.8.2012 osoitteesta
<http://www.efqm.org/en/PdfResources/Overview%20EFQM%202010.pdf>
- Evans, I. (2004). *Achieving Software Quality Through Teamwork*. Norwood, MA: Artech House. Haettu 31.7.2012 osoitteesta
<http://site.ebrary.com.ezproxy.jyu.fi/lib/jyvaskyla/docDetail.action?docID=10081986>
- Graham D. (2002). Requirements and Testing: Seven Missing-Link Myths. *IEEE Software 19(5)*, 15-17
- Haikala I. & Märijärvi J. (2006). Ohjelmistotuotanto (11. painos). Helsinki: Talentum
- Hawk S. R. & Dos Santos B.L. (1991). Successful System Development: The Effect of Situational Factors on Alternate User Roles. *IEEE Transactions on Engineering Management, 38*, 316-327

- Heinbokel T., Sonnentag S., Frese M., Stolte W. & Brodbeck F.C. (1996). Don't underestimate the problems of user centredness in software development projects there are many! *Behaviour & Information Technology* 15, 226-236
- Heinecke A., Brückmann T., Griebe T. & Gruhn V. (2010). Generating Test Plans for Acceptance Tests from UML Activity Diagrams. *17th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS) March 20-22*, (s. 58-66). Oxford: IEEE Computer Society
- Herbsleb J., Zubrow D., Goldenson D., Hayes W. & Paulk M. (1997). Software Quality and the Capability Maturity Model. *Communications of the ACM*, 40(6), 30-40
- IEEE 610.12 Standard Glossary of Software Engineering Terminology
- IEEE Guide 90003-2008 (2008) Adoption of ISO/IEC 90003:2004 Software Engineering Guidelines for the Application of ISO 9001:2000 to Computer Software
- IEEE1219-1998 (1998) IEEE Standard for Software Maintenance
- Kabbedijk J., Brinkkemper S., Jansen S. & van der Veldt B. (2009). Customer Involvement in Requirements Management: Lessons from Mass Market Software Development. *Proceedings of the 17th IEEE International Requirements Engineering Conference, August 31 - September 4* (s. 281-286). Atlanta: IEEE Computer Society
- Kaulio M. A. (1998). Customer, consumer and user involvement in product development: A framework and a review of selected methods. *Total Quality Management*, 9(1), 141-149
- Koch A. (2005). *Agile Software Development: Evaluating The Methods For Your Organization*. Norwood, MA: Artech House
- Korkala M., Abrahamsson P. & Kyllönen P. (2006). A case study on the impact of customer communication on defects in agile software development. *Teoksessa Proceedings of AGILE 2006 Conference, July 23-28* (s. 77-88). Minneapolis: IEEE Computer Society
- Kujala S. (2008). Effective user involvement in product development by improving the analysis of user needs. *Behaviour & Information Technology*, 27(6), 457-473
- Kujala S., Kauppinen M., Lehtola L. & Kojo T. (2005). The Role of User Involvement in Requirements Quality and Project Success. *Teoksessa Proceedings of the 2005 13th IEEE International Conference on Requirements Engineering, August 29-September 2* (s. 75-84). La Sorbonne: IEEE Computer Society
- Kumar V., Maheshwari B. & Kumar U. (2002). ERP systems implementation: Best practices in Canadian government organizations. *Government Information Quarterly*, 19, 147-172
- Li Q., Liu S. & Pan Y. (2012). A Cooperative Construction Approach for SaaS Applications. *Teoksessa Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, May 23-25* (s. 398-403). Wuhan: IEEE Computer Society

- Luomansuu R. (2009). *Tilastollinen tutkimus ohjelmiston laatuun vaikuttavista tekijöistä*. Tuotantotalouden diplomityö, Lappeenrannan teknillinen yliopisto.
- Lynch T. & Gregor S. (2004). User participation in decision support systems development: influencing system outcomes. *European Journal of Information Systems*, 13(4), 286-301.
- Majchrzak T.A. (2010). Best Practices for the Organizational Implementation of Software Testing. Teoksessa *Proceedings of the 43rd Hawaii International Conference on System Sciences - 2010, January 5-8* (s. 1-10). IEEE Computer Society
- Majid R.A., Noor N.L.M., Adnan W.A.W. & Mansor S. (2010). A survey on user involvement in software Development Life Cycle from practitioner's perspectives. Teoksessa *Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT), November 30-December 2* (s. 240-243). Seoul: IEEE Computer Society
- Microsoft CEIP. (2012). Käyttömukavuuden kehittämisohjelma. Haettu 6.8.2012 osoitteesta <http://www.microsoft.com/products/ceip/fi-fi/default.msp>
- Muller, M.J., Hallowell Haslwanter, J., and Dayton, T. (1997). Participatory practices in the software lifecycle. Teoksessa: M. Helander, T.K. Landauer, and P. Prabhu (toim.) *Handbook of human-computer interaction*. 2nd ed. Amsterdam:Elsevier, 255-297.
- Mäntylä M. V., Itkonen J. & Iivonen J. (2012). Who tested my software? Testing as an organizationally cross-cutting activity. *Software Quality Journal*, 20, 145-172
- Nah F., Lau J. & Kuang J. (2001). Critical factors for successful implementation of enterprise systems. *Business Process Management Journal*, 7, 285-296
- Porali M., Riekkinen A., Pohjolainen P., Mykkänen J., Toroi T., Kärkkäinen T. & Eerola A. (2004). *Ohjelmistotuotannon nykytilaselvitys 2003 - kohderyhmänä terveydenhuollon ohjelmistoyritykset ja organisaatiot. PlugIT-hankkeen selvityksiä ja raportteja 14*. Kuopion yliopisto ja Savonia-ammattikorkeakoulu.
- Puolitaival O. & Kanstrén T. (2010). Mallipohjainen testaus ennen, nyt ja tulevaisuudessa. <http://www.vtt.fi/inf/julkaisut/muut/2010/mallipohjaintestaus.pdf> haettu 20.7.2012
- Ramesh B., Cao L. & Baskerville R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, 20, 449-480
- Ravichandran T. & Rai A. (2000). Quality Management in Systems Development: An Organizational System Perspective, *MIS Quarterly* 24 (3), 381-415
- Ruohonen M. J. & Salmela H. (2005). *Yrityksen tietohallinto* (1.-3. painos). Helsinki: Edita Prima
- Saleh K. A. (2009) *Software Engineering*. Ft. Lauderdale, FL: J. Ross Publishing

- See Pui Ng C. (2001). A decision framework for enterprise resource planning maintenance and upgrade: A client perspective. *Journal of Software Maintenance and Evolution*, 13, 431–468
- Software Engineering Institute. (2012). CMMI Overview. Haettu 6.8.2012 osoitteesta <http://www.sei.cmu.edu/cmmi/>
- Sommerville I. (2007). *Software Engineering* (8. painos). Essex: Pearson Education Limited
- Subramanyam R., Weisstein F. L. & Krishnan M. S. (2010). User Participation in Software Development Projects. *Communications of the ACM*, 53, 137-141
- SWEBOK Guide. (2004). Guide to the Software Engineering Body of Knowledge. IEEE. Haettu 6.8.2012 osoitteesta <http://www.computer.org/portal/web/swebok/html/contents>
- Thakkar D., Zhen M. J, Hassan, A.E., Hamann, G. & Flora, P. (2008). Retrieving Relevant Reports from a Customer Engagement Repository, Teoksessa *Proceedings of the IEEE International Conference on Software Maintenance, September 28 – October 4* (s. 117-126). Beijing: IEEE Computer Society
- TickIT. (2012). Introduction - TickITplus haettu 6.8.2012 osoitteesta <http://www.tickitplus.org/information/the-tickitplus-scheme.aspx>
- Tietotekniikan liitto (2005). *Tietojärjestelmän hankinta ohjelmistotoimittajan ja -ratkaisun valinta* (2. uudistettu painos). Helsinki: Talentum
- Wang X., Wu Z. & Zhao M. (2008). The Relationship between Developers and Customers in Agile Methodology. *International Conference on Computer Science and Information Technology 2008 August 28-September 2* (s. 566-572). Singapore: IEEE Computer Society
- Wojciechowski A., Wesolowski M. & Complak W. (2010). Experimental Evaluation of 'On-Site Customer' XPPractice on Quality of Software and Team Effectiveness. Teoksessa R. Meersman et al. (toim.) *OTM 2010 Workshops, LNCS 6428* (s. 269–278). Berlin: Springer-Verlag