

Henri Venäläinen

ASIAKASKESKEINEN OHJELMISTOKEHITYS



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2018

TIIVISTELMÄ

Venäläinen, Henri

Asiakaskeskeinen ohjelmistokehitys

Jyväskylä: Jyväskylän yliopisto, 2018, 30 s.

Tietojärjestelmätiede, kandidaatintutkielma

Ohjaaja: Halttunen, Veikko

Tämän tutkimuksen tarkoituksena on luoda käsitys asiakaskeskeisyydestä osana ohjelmistokehitystä. Tutkimuksessa tarkastellaan myös käytänteitä asiakaskeskeisyyden ja ohjelmistokehityksen välillä. Asiakaskeskeisyyteen panostaminen voi merkittävästi vaikuttaa asiakkaan ja ohjelmistoa kehittävän organisaation suhteeseen. Asiakaskeskeiset prosessit auttavat ohjelmistokehitysorganisaatiota kehittämään entistä laadukkaampia ohjelmistoja. Jatkuvasti muuttuvassa toimintaympäristössä asiakasyhteistyöllä voidaan hakea tukevampaa kasvua ja sellaista kilpailuetua, mitä kilpailijoiden on vaikea kopioida. Tässä tutkielmassa esitellään yleisesti ohjelmistokehitystä ja tarkemmin tarkastellaan ketteriä ohjelmistokehitysmenetelmiä. Ketterät menetelmät ovat 2000-luvun jälkeen nousseet suosituimmiksi ohjelmistokehitysmenetelmiksi. Ketteriä menetelmiä käytetään jopa 80%:ssa ohjelmistoprojekteista. Vaikka ketterät menetelmät ovat luonteeltaan asiakaslähtöisiä, voi ohjelmistoa kehittäville yrityksille puuttua syvempi ymmärrys asiakaskeskeisyyden tuomista eduista. Tämä tutkielma pyrkii tarjoamaan syvempää tietoa asiakaskeskeisyydestä, jotta sitä voidaan hyödyntää entistä paremmin myös ohjelmistoa valmistavissa organisaatioissa. Tässä tutkimuksessa käsitellään asiakaskeskeisyyttä, sen määritelmää sekä millaisia käytänteitä asiakaskeskeisillä organisaatioilla on. Lisäksi on esitelty ohjelmistokehitystä yleisesti, jotta lukijalla on mahdollisuus ymmärtää näitä molempia ajankohtaisia aiheita tarkemmin. Näin lukijan on myös helpompi muodostaa yhteyksiä näiden käsitteiden välille. Asiakaslähtöisyydestä on tullut tämän vuosikymmenen yksi kuumista puheenaiheista liiketoiminnan kehittämisessä. Tutkimuksessani olen huomannut sen periaatteiden tukevan hyvin myös ohjelmistokehittämisen prosesseja. Ohjelmistokehittämisestä voidaan asiakaskeskeisyyttä syvemmin ymmärtämällä valmistaa asiakasyhteistyössä entistä laadukkaampia ohjelmistoja, joilla yrityksiä on mahdollista luoda kilpailuetua kovasti kilpailluilla markkinoilla.

Asiasanat: ohjelmistokehitys, ohjelmistotuotanto, asiakaskeskeisyys, asiakaslähtöisyys, ketterät menetelmät

ABSTRACT

Venäläinen, Henri

Customer centric software development

Jyväskylä: University of Jyväskylä, 2018, 30 pp.

Information Systems, Bachelor's Thesis

Supervisor: Halttunen, Veikko

The purpose of this thesis is to create an understanding of customer centricity in software development and its practices regarding to it. In organizations investing to customer centricity can significantly make a difference in the customer relationship between the customer and software developers. Continuously changing environments where companies compete, customer collaboration is a way to gain a larger foothold in the markets. By means of customer centricity it is possible to gain a competitive advance that can not be easily copied by competitors. In my thesis I will introduce software development in general and focus a bit more on agile software development methods. Since they have become the most used software developing methods of the 21st century and about 80% of software projects are considered agile. Agile methods are customer centric by their principles. However, there are studies which show that customer centric approach is not so widely known around software developing organisations. Deeper knowledge in customer centricity could create a great benefit for these companies. This thesis covers customer centricity, the definition of it and the practices of customer centric organisations. This thesis also discusses software development in general. That way the reader can have a perception of both customer centricity and software development to allow understanding the similarities between these interesting topics. Customer-oriented approach has become a trend when speaking about business development. Customer centricity could be a way for software developer to produce software with higher quality.

Keywords: Software development, software engineering, customer centricity, customer oriented, customer focused, agile methods

TAULUKOT

Taulukko 1. Agile Manifeston arvot (mukaillen Beck ym., 2001)	17
Taulukko 2. Agile Manifeston periaatteet (mukaillen Beck ym., 2001)	17
Taulukko 3. Perinteisen ja ketterän menetelmän eroja (mukaillen Boehm, 2002).....	21

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
TAULUKOT	4
SISÄLLYS.....	5
1 JOHDANTO	6
2 ASIAKASKESKEISYYS	9
2.1 Asiakaskeskeisyyden määrittelyä	9
2.2 Asiakaskeskeisen organisaation piirteitä	12
2.3 Yhteenveto asiakaskeskeisyydestä	14
3 OHJELMISTOKEHITYS	15
3.1 Perinteinen ohjelmistokehitys.....	15
3.2 Ketterä ohjelmistokehitys	16
4 ASIAKASKESKEINEN OHJELMISTOKEHITTÄMINEN	20
4.1 Perinteisen ja ketterän ohjelmistokehitysmenetelmän vertailu.....	21
4.2 Asiakaskeskeinen liiketoiminta ohjelmistoalalla.....	22
5 YHTEENVETO	25
LÄHTEET	27

1 JOHDANTO

Tietojärjestelmien kehittämisen tutkiminen on ajankohtainen aihe, sillä maailma digitalisoituu kasvavaa vauhtia. Digitalisoitumisen tukemiseksi on myös järjestelmien oltava entistä laadukkaampia ja tätä varten tarvitaan laadukasta tutkimusta. Tutkimuksien mukaan vain 34% ohjelmistokehitysprojekteista onnistuu. Yhdeksi avaintekijäksi onnistumisen kannalta nähdään asiakkaiden aktiivinen osallistuminen ohjelmistokehitysprosessiin. Tämä nähdään toisaalta myös yhdeksi haastavimmiksi asioiksi toteuttaa. (Schwartz, 2004.) Asiakkaan osallistuttaminen ohjelmistokehitykseen vaatii organisaatiolta rakenteellisia muutoksia ja uudenlaisia menetelmiä työskennellä. Ketterät ohjelmistokehityksen menetelmät tuovat työkaluja organisaatioille rakentaa parempaa asiakasyhteistyötä. Näitä menetelmiä käytetään jo noin 80 prosentissa ohjelmistoprojekteja. Kun ymmärretään paremmin asiakaskeskeisyyttä ohjelmistokehityksessä, voivat ohjelmistoprojektit onnistua todennäköisemmin.

Useissa tutkimuksissa on esitetty asiakaslähtöisyyden merkittävästi vaikuttavan yritysten suorituskykyyn. Yritysten tulisi rakentaa entistä enemmän asiakkaalle kohdennettuja palveluita ja tuotteita. Asiakaskeskeisyyden on nähty olevan jopa seuraava trendi liiketoiminnan kehityksessä, jossa asiakkaan tarpeet on integroitu organisaation prosesseihin. (Hakala & Kohtamäki, 2011). Yritykset, jotka onnistuvat osallistuttamaan asiakkaan liiketoimintansa kehittämiseen todennäköisesti pystyvät luomaan parempaa asiakasarvoa.

Asiakaskeskeisyyttä ei ole kuitenkaan kirjallisuudessa tutkittu riittävän paljon ohjelmistokehityksen näkökulmasta. Aiemmin asiakaskeskeisyyttä ja asiakaslähtöisyyttä on tutkittu osana liiketoimintaa ja asiakaslähtöisyydellä on havaittu olevan kilpailuetua parantava vaikutus (Blocker, Flint, Myers, & Slater, 2011).

Asiakaskeskeisyys näkyy myös ohjelmistokehityksessä ja sillä on erilaisia ilmenemismuotoja riippuen käytetystä kehitysmenetelmästä. Esimerkiksi ketterät ohjelmistokehitysmenetelmät ovat perusajatukseltaan asiakaskeskeisempiä, kun taas perinteiset ohjelmistokehitysmenetelmät ovat prosessikeskeisiä. On tärkeää ymmärtää asiakaskeskeistä toimintaa laajemmassa kuvassa osana orga-

nisaation prosesseja, jotta asiakaskeskeisyys voidaan nähdä tarkemmin osana ohjelmistokehitystä. Näin ollen tämä tutkimus on tarpeellinen ja ajankohtainen. Nykypäivänä yhteiskunnassa asiakaslähtöisyys nähdään entistä suuremmassa roolissa ja useat yritykset ovat ottaneet asiakaskeskeiset käytännöt osaksi liiketoiminnan prosessejaan. Nykyään voidaan nähdä asiakaslähtöisyyden olevan jopa elinehto entistä kovemmin kilpailuilla markkinoilla (Shah, Rust, Parasuraman, Staelin, & Day, 2006). Asiakaskeskeisyydestä on tullut jopa kuin kulutustavaraa ja asiakkaat osaavat vaatia entistä asiakaslähtöisempää palvelua.

Tämän tutkimuksen tarkoituksena on kirjallisuuskatsauksen keinoin käsitellä asiakaskeskeisyyden teemaa ja tarkastella, miten se näkyy ohjelmistokehityksessä.

Tutkimustehtävään pyritään vastaamaan seuraavien tutkimuskysymysten avulla:

- Miten asiakaskeskeisyys ilmenee ohjelmistokehityksessä?
- Miksi asiakaskeskeinen kehittäminen on tärkeää?
- Millaisia piirteitä asiakaskeskeisellä ohjelmistokehitysjärjestelmällä on?

Tutkimustani varten etsin tietoa, millä tavalla asiakaskeskeisyyttä on käsitelty ohjelmistokehityksessä. Tutkimukseni lähteitä etsin valituilla asiasanoilla kansainvälisistä e-aineistoista ja englanninkielisistä tietokannoista, käyttämällä hakusanoja esimerkiksi customer centricity, customer orientation, software development, agile development methods ja näiden erilaisilla kombinaatioita.

Tutkimusprosessi vaati useiden eri lähteiden etsimistä, selaamista ja lukemista. Tätä tutkimusta varten luin ja selasin useita eri teoksia, sillä asiakaskeskeisestä ohjelmistokehityksestä löytyi mielestäni yllättävän vähän tietoa. Tutkimustani varten keräsin siis tietoa hyvin useista eri lähteistä. Tutkimuksessani käytetyt lähteet ovat vertaisarvioituja ja valideja tutkimuksen kannalta.

Tämä tutkimus on myös minulle henkilökohtaisesti tärkeä ja ajankohtainen, sillä työskentelen näiden asioiden parissa päivittäin omassa työssäni. Työssäni olen joutunut näkemään asiakkaan kannalta huonosti hoidettuja ohjelmistoprojekteja. Toisaalta olen myös nähnyt erittäin onnistuneita projekteja, joissa asiakaskeskeisyys on ollut avainasemassa.

Tämän tutkimuksen tavoitteena on lisätä tietoisuutta asiakaskeskeisyydestä ja sen vaikutuksista ohjelmistokehitykseen. Tarkoituksena on myös samalla syventää omaa osaamistani aiheen parissa.

Monet lähteet, jotka koin tärkeiksi tutkimukseni kannalta, ovat kirjoitettu 2000-luvun alkupuolella. Aluksi koin tämän uhkana tutkimuksen luotettavuudelle, mutta jälkeen päin löysin uudempia tutkimuksia, jotka tukivat näitä väitteitä. Toisaalta ketterät menetelmät ovat kehittyneet vasta vuosittuhannen vaihteessa ja tutkimukseni kannalta oleellimmat periaatteet ohjelmistokehityksessä niiden osalta eivät ole juuri muuttuneet. Myös asiakaslähtöisyys on ollut liiketoiminnassa nouseva trendi 2000-luvun alussa, jolloin sitä on tutkittu runsaasti. Ketterillä menetelmillä ja asiakaskeskeisyydellä onkin useita yhtymäkoh-

tia ja uskon asiakaskeskeisen ajattelun tukeneen ketterien menetelmien kehitystä.

Asiakaskeskeisyys on ajankohtainen aihe ja sitä on erittäin mielekästä tutkia, sillä sitä voidaan hyödyntää laajasti liiketoiminnassa. Aiemmin sitä ei ole vielä kovin kattavasti tutkittu ja on epäselvää mitä asiakaskeskeiset yritykset tekevät paremmin, kuin yritykset, jotka eivät ole niin asiakaskeskeisiä (Lamberti, 2013).

Ohjelmistojen ja niiden kehitysmenetelmien kehittyessä on tärkeää keskittyä myös ohjelmistokehittämisen pehmeisiin arvoihin, kuten asiakaslähtöisyyteen. Vaikka monet kehitysmenetelmät korostavat asiakkaan aktiivista osallistuttamista, puuttuu monilta yrityksiltä syvempi ymmärrys syistä, miksi organisaatio pyrkii toimimaan asiakaslähtöisesti. Tämän tutkielman tarkoituksena on tuoda esille, millaisia asiakaskeskeisiä ohjelmistokehityksen käytänteitä on olemassa ja selvittää syitä, miten asiakaskeskeisyydellä voidaan saavuttaa kilpailuetua kovasti kilpailluilla globaaleilla markkinoilla.

Tämän tutkimuksen ensimmäinen luku on johdanto, jossa lukijalle avataan, mitä aiheita tutkimus käsittelee ja mihin kysymyksiin se pyrkii vastaamaan. Tutkielman toisessa luvussa käsitellään asiakaskeskeisyyttä, sen määrittelmää ja asiakaskeskeisten organisaatioiden toimintoja. Kolmannessa luvussa tutustutaan ohjelmistokehitykseen, sekä käydään läpi perinteinen ja ketterä ohjelmistokehitys. Tässä luvussa esitellään myös Scrum, joka on ketterän kehityksen menetelmistä tunnetuin. Neljännessä luvussa esitellään tarkemmin ketterien ja perinteisten ohjelmistokehitysmenetelmien eroavaisuuksia. Luvussa kerrotaan myös, millaisia etuja asiakaskeskeisellä ohjelmistokehityksellä on liiketoiminnassa. Tutkimuksen lopussa on yhteenveto, joka esittelee mitä tutkimuksessa on käsitelty.

2 ASIAKASKESKEISYYS

Tämän luvun tarkoituksena on luoda perusta tutkimuksessa myöhemmin käsiteltäville aiheille asiakaskeskeisestä näkökulmasta. Ensimmäisenä esitellään asiakaskeskeisyyden määritelmä ja avataan sen historiaa osana liiketoiminnan strategiaa. Tämän jälkeen käsitellään asiakaskeskeisen organisaation keskeisiä piirteitä.

Asiakaskeskeisyyden ja asiakaslähtöisyyden välille on haastavaa tehdä eroa, sillä ne ovat käsitteellisesti kovin lähellä toisiaan (Gummesson, 2008). Myös lähdekirjallisuudessa näitä termejä on välillä käytetty sekaisin. Seuraavissa alaluvuissa määritellään mitä asiakaskeskeisyydellä tarkoitetaan tässä tutkielmassa.

Asiakaskeskeisyys on käsitteenä asiakaslähtöisyyttä vanhempi, se on noussut esiin jo 1930-luvulla. Käsitteen lähtökohtana on asiakkaiden tarpeiden parempi huomioiminen ja asiakkaan kunnioitus. Asiakaskeskeisyydellä tarkoitetaan asiakkaan asettamista palveluiden keskiöön huomioimalla tämän tarpeet. Vaikka asiakas on sijoitettu keskiöön, se ei välttämättä tarkoita suoraa dialogia asiakkaan kanssa. Asiakaslähtöisyys on asiakaskeskeisyydestä hieman kehittyneempi aste, jossa korostuu yhteistyö asiakkaan kanssa, ei pelkästään asiakkaan tarpeiden huomioiminen. (Ruotsalainen, 2000.)

2.1 Asiakaskeskeisyyden määrittelyä

Asiakaskeskeisyydellä tarkoitetaan organisaation kykyä tuntea ja ymmärtää asiakkaitaan. Tarkoituksena on kohdella asiakkaita kuten he odottavat heitä kohdeltavan. Pyrkimyksenä on täyttää heidän tunnistamattominkin tarpeensa. Asiakaskeskeisyys on vuorovaikutussuhde, joka perustuu suureen kunnioitukseen asiakasta kohtaan. Tässä suhteessa korostuu asiakkaan ja yrityksen välinen luottamus ja asiakasymmärrys. (Bolton, 2004.)

Kirjallisuudessa asiakaskeskeisyys esitetään monesti tuotokeskeisyyden vastakohtana. Tuotokeskeiset organisaatiot näkevät organisaation säilytyspaik-

kana resursseille ja kompetensseille, joilla uusia tuotteita tai palveluita tehdään. Tuotteet ja palvelut ovat siten organisaation ydin. Yrityksen tarkoituksena on pyrkiä tarjoamaan tuotteitaan ja palveluitaan mahdollisimman monelle asiakkaalle. Asiakaskeskeiset yritykset puolestaan keskittävät huomionsa ostajaan löytääkseen molempia tyydyttävän lopputuloksen. (Lamberti, 2013.)

Asiakaskeskeisyys nähdään organisaation kykyinä tuottaa, kerätä ja prosessoida asiakastietoa, jota hyödyntämällä paremmin ymmärretään asiakasta ja hänen vuorovaikutusta organisaation kanssa. Tämä mahdollistaa molemminpuolisen arvонуontiprosessin, jossa asiakas aktiivisesti osallistuu yhteistyöhön. Asiakaskeskeisyys keskittää huomion pelkästään asiakkaalle tarjotusta ratkaisusta koko prosessiin, jossa asiakaskokemus syntyy. (Lamberti, 2013.)

Lambertin (2013) mukaan asiakaskeskeisyys näkyy organisaation toimina, joita yritys toteuttaa myynti- ja palveluhenkilöstöään tukeakseen. Näiden toimien tavoitteena on pyrkiä täyttämään asiakkaan tarpeet ja vaatimukset mahdollisimman hyvin. Tällaisia liiketoimintastrategioita ovat asiakaslähtöiset toimet, kuten tuotteiden ja palveluiden kehittäminen yhdessä asiakkaan kanssa, asiakkaan pyyntöihin vastaaminen mahdollisimman nopeasti ja asiakasta ymmärtävällä tavalla. (Lamberti, 2013.)

Boltonin (2004) mukaan asiakaskeskeisyys korostaa, että kaikkien yksilöiden ja liiketoimien pitää olla keskittyneitä asiakkaaseen. Hänen mukaansa kaikkien organisaation prosessien tulisi olla asiakaskeskeisiä. Tämä saavutetaan ottamalla asiakkaan tarpeet huomioon yrityksen ydinprosesseissa. Kun asiakaskeskeinen ajattelu on osana yrityksen kaikkia prosesseja, koskettaa se koko organisaatiota ja siten heijastuu asiakkaalle hänen kaikissa kosketuspisteissään organisaation kanssa. (Bolton, 2004.)

Kaikista kohtaamisista asiakkaan kanssa on tärkeää tallentaa asiakastietoa. Kohtaamispisteitä asiakkaan ja organisaation välillä voi olla useita ja tiedon tallennus on tärkeää. Organisaatiolla olisi hyvä olla asiakkaasta ajantasaisin tieto kaikissa kohtaamispisteissä. Asiakastiedon avulla organisaatio voi paremmin optimoida toimintaansa. (Bolton, 2004.)

Asiakaskeskeisyys ei tarkoita sitä, miten parhaiten myydä asiakkaalle tuotteita tai palveluita. Sen sijaan asiakaskeskeisyys tarkoittaa, miten asiakkaalle ja yritykselle tuotetaan arvoa. Toisin sanoen asiakaskeskeisyys keskittyy kahdenkeskisen arvon luomiseen asiakkaan ja yrityksen välillä. (Boulding, 2005.)

Asiakaskeskeisyyden määrittäminen on yhä haaste sekä kirjallisuudessa, että käytännössä. Gummesson (2008) sanoo, että asiakaskeskeisyydelle on vaikea löytää yleispätevää määritelmää, mutta silti sen piirteitä pystytään määrittämään tarkasti. Gummessonin ajatukseen yhtyy myös Lamberti (2013), joka määrittelee asiakaskeskeisyyden olevan edelleen monitulkintainen termi, vaikka sitä on käsitelty useista eri näkökulmista viimeisen 20 vuoden aikana.

Monet suuret yritykset ovat ottaneet käyttöönsä asiakaskeskeisen liiketoimintastrategian ja asettaneet sen keskeiseksi osaksi saadaakseen arvokasta kilpailuetua. Tunnettuja tällaisia yrityksiä ovat esimerkiksi IBM, JetBlue Airways ja Best Buy. Asiakaslähtöisyyden puolesta puhuvat myös monet suuret

konsulttiyritykset, kuten McKinsey, Forrester ja BCG, jotka korostavat asiakas-keskeisyyden etuja luotaessa kilpailuetua kovasti kilpailluilla maailmanmarkkinoilla. (Lamberti, 2013.) Asiakaskeskeisyyden käsite ja sen edut ovat olleet keskustelun aiheena jo yli 60 vuoden ajan. Ne ovat pääasiassa liittyneet yrityksen organisaatiokulttuuriin, rakenteeseen ja prosesseihin, sekä osaltaan myös kannattavuuteen. (Shah ym., 2006.)

Digitaalinen vallankumous on tuonut mukanaan yrityksille useita erilaisia keinoja kerätä, tallentaa ja jakaa asiakastietoa. Yksi näistä keinoista on CRM-järjestelmät (*CRM, Customer Relationship Management*), jotka mahdollistivat asiakastiedon keräämisen ja sen tehokkaan hyödyntämisen. CRM-järjestelmät teknologiana mahdollistivat yrityksen tarjota personoitua kohtelua asiakkailleen jokaisessa kohtaamispaikassaan. (Shah ym., 2006.) Shah ja kumppanit (2006) esittivät tutkimuksessaan, että suurimmalta osalta yrityksistä kuitenkin puuttui edellytykset ymmärtää täysin asiakaskeskeisyyden tarjoamat hyödyt.

Tässä tutkimuksessa keskitytään erityisesti asiakaskeskeisyyteen ja jätetään tarkemman tarkastelun ulkopuolelle asiakkuuksien hallinta. Kirjallisuudessa näiden kahden termin tarkka määrittäminen on nähty haastavaksi. Bolton (2004) kuitenkin määrittelee asiakkuuksien hallinnan toimiksi, joiden avulla asiakkaista kerättyä tietoa voidaan hyödyntää. Tietoa hyödynnetään esimerkiksi tarjoamalla asiakkaalle uusia tuotteita ja palveluita, perustuen hänen aikaisempaan käyttäytymiseensä. Asiakkuudenhallinnan keinoin yritys pyrkii palvelemaan asiakkaita aiempaa paremmin.

Asiakaskeskeisyyden periaate ei ole uusi, mutta vasta 1960-luvun vaihteessa tutkijat kiinnostuivat tästä aiheesta ja se sai jalansijaa (Shah ym., 2006). Drucker (1954) kertoo kirjassaan, että asiakas on se, joka päättää mitä yrityksen liiketoiminta on, mitä se tuottaa ja tuleeko se menestymään. Löytänä (2014) käsittelee kirjassaan sitä, miten asiakkaiden odotukset ja käyttäytyminen ovat oleellisesti muuttuneet maailman digitalisoituessa. Asiakkailta on yhä enemmän informaatiota käytettävissään ja he todella päättävät yritysten menestymisestä tulevaisuudessa.

Shah ja kumppanit (2006) esittävät Levitin (1960) näkemyksen, jonka mukaan yrityksen ei pitäisi keskittyä vain myymään tuotteitaan, vaan täyttämään asiakkaan tarpeet. Hän viittaa tilanteeseen, jossa yritykset keskittivät huomionsa tuotteiden valmistamiseen sen sijaan, että olisivat keskittyneitä näiden tuotteiden ostajiin ja käyttäjiin. Ilmiön on huomionut myös Bolton (2004). Hänen mukaansa tuotteita valmistavien yritysten strateginen fokus on aiemmin ollut työn tuottavuuden parantamisessa. Palveluita tarjoavien yritysten strateginen fokus on sen sijaan ollut tarjota entistä laadukkaampia palveluita.

Yrityksien siirtyessä massatuotannon aikakaudesta kohti nykyaikaa on tapahtunut paljon muutosta yrityksen ja asiakkaiden välisissä suhteissa. Esimerkiksi yrityksen tietoisuus asiakaskeskeisyyden tärkeydestä on kasvanut. Pärjätäkseen kovasti kilpailluilla markkinoilla on panostettava entistä enemmän asiakaslähtöisiin tekijöihin, kuten asiakastyytyväisyyteen. (Shah ym., 2006.) Kumar ja Shah (2004) esittävät, että asiakastyytyväisyyden lisäksi yrityksen tulee kiinnittää huomiota lisäksi asiakasuskollisuuteen.

Ohjelmistokehityksen kontekstissa asiakkaalla voidaan tarkoittaa useaa eri tahoa. Asiakkaita voi olla esimerkiksi ihmiset, jotka käyttävät tai maksavat ohjelmistoista, ohjelmiston käyttäjien asiakkaat, organisaation eri yksiköt, julkisen hallinnon yhteisöt tai organisaation sisäiset asiakkaat. (Evans, 2004.) Asiakkaaksi voidaan siis käsittää kaikki sellaiset henkilöt tai organisaatiot, jotka käyttävät tai olisivat valmiita käyttämään organisaation valmistamia tuotteita tai palveluita. Ketterässä kehittämisessä asiakas voi myös tarkoittaa kehittämisprojektiin liittyvää asiakasorganisaation edustajaa.

Tässä tutkielmassa asiakkaalla tarkoitetaan organisaatiota, yritystä tai yhteisöä, joka hankkii ohjelmistoja käyttöönsä. Asiakas voi olla myös yksittäinen organisaation jäsen, jolla on valtuudet päättää ohjelmistoprojektista (Mohammadi, Nikkahan, & Sohrabi, 2009).

2.2 Asiakaskeskeisen organisaation piirteitä

Asiakaskeskeisyys korostaa entistä läheisemmän asiakassuhteen kehittämistä, jossa asiakkaan kanssa yhdessä luodaan parempia asiakaskokemuksia. Nämä asiakaskokemukset ovat arvokkaita molemmille osapuolille (Antikainen, Mäkipää & Ahonen, 2010). Näin asiakaskeskeinen toiminta korostaa tuotteiden ja palveluiden olevan vain apuna arvontuontiprosessissa, joka yhdistää asiakkaan ja yrityksen resurssit. (Pralhad & Ramaswamy, 2004).

Asiakaskeskeisissä organisaatioissa korostetaan vuorovaikutusta ja arvontuontia yhdessä asiakkaan kanssa. Tämä vaatii organisaation kiinnittävän huomiota jokaiseen kohtaamispisteeseen asiakkaan ja organisaation toimitusketjun välillä. Toimintojen koordinoiminen vaatii organisaatioilta ennalta määritettyjä käytänteitä. Tarkoituksena on palvella asiakasta hänen haluamallaan tavalla koko toimitusketjussa. (Lamberti, 2013.)

Lambertin (2013) tekemässä tutkimuksessa haastateltava määritteli asiakaskeskeisyyden tarkoittavan hänen yrityksessään seuraavaa: ”Ymmärretään jokaista asiaa, joka voisi turhauttaa asiakasta hänen asioidessaan yrityksen kanssa”. Nämä asiat on huomioitava siitä hetkestä lähtien, kun asiakas ensimmäisen kerran kohtaa yrityksen, siihen hetkeen, kun hän päättää asioinnin. Tarkoituksena on, että jokaisella asiakkaan kanssa kosketuksissa olevalla työntekijällä on tavoitteena vähentää asiakkaan turhautumisen tunteita.

Puhuttaessa asiakaskeskeisistä yrityksistä, kirjallisuudessa nousee esille tärkeimpänä organisaatiokulttuuri, joka tukee asiakaskeskeisyyttä. Tärkeitä asioita ovat myös organisaation käytänteet ja rakenteet. (Shah ym., 2006). Asiakaskeskeisiä yrityksiä yhdistää arvot ja normit, joissa korostuu asiakaslähtöisyys. Organisaatiokulttuuriin tärkeänä osana on normit, jotka ovat jaettuina uskomuksia soveliaasta ja toivotusta käyttäytymisestä. Yleinen normi asiakaskeskeisissä organisaatioissa on, että työntekijät ovat asiakkaan puolesta puhujia. Toinen yleinen normi on, että työntekijät ovat valmiita jakamaan tietoa asiakkaasta organisaationsa sisällä kollegoilleen. Näin organisaatio kykenee paremmin vastaamaan asiakkaan tarpeisiin. Asiakaslähtöisissä yrityksissä ei siis tarvi-

ta kirjallisia sopimuksia, kuinka asiakasta tulisi kohdella. Kaikki pohjautuu organisaatiokulttuuriin, jossa jokaisella työntekijällä tavoitteena on antaa asiakkaalle nopein ja parhain vastaus tämän kysymyksiin ja pyyntöihin. (Lamberti, 2013.)

Asiakaskeskeisyys koskettaa koko organisaatiota, jolloin askel asiakaskeskeytyksen tuo mukanaan muutoksia organisaatiokulttuuriin. Se voi myös vaikuttaa organisaation rakenteisiin ja sen prosesseihin uusien toimintamallien myötä. (Bolton, 2004.) Asiakaskeskeisissä organisaatioissa jokaisen yksilön tavoitteena on ratkaista asiakkaan ongelmat parhaalla mahdollisella tavalla. Asiakaskeskeytyksen tulee olla päämääränä ja organisaatiolla tulee olla yhteinen näkemys siitä, miten se saavutetaan (Galbraith, 2002). Tällainen toiminta vaatii oikeanlaisen organisaatiokulttuurin kehittämistä, jossa kaikki toimet ovat integroitua palvelemaan asiakasta. Tämä vaatii työntekijöiden kouluttamista asiakaskeskeiseen ajattelutapaan ja tekoihin. Asiakkaalle ei ole välttämätöntä osoittaa tiettyä avainasiakashenkilöä silloin, kun organisaation jokaisen työntekijän päämäärä on palvella asiakasta mahdollisimman hyvin. (Lamberti, 2013.)

Organisaatiokulttuurin muutos seuraa yrityksen työntekijöiden käyttäytymisen muutosta. Tämä voidaan saavuttaa muuttamalla käyttäytymismalleja ja auttamalla työntekijöitä paremmin ymmärtämään, kuinka uudet toimintatavat parantavat organisaation tehokkuutta. (Shah ym., 2006.)

Jotta yritys olisi asiakaskeskeinen, se vaatii asiakkaan syvällistä ymmärtämistä. Tämä saavutetaan vain asiakkaan tarpeiden ja mieltymysten tuntemisella (Sun, Li, & Zhou, 2006.)

Lambertin (2013) mukaan parhaiten asiakkaista saadaan tietoa keräämällä sitä dialogin avulla asiakkaiden kanssa. Tämän tarkoituksena on tunnistaa asiakkaan tiedostamattomia tarpeita, joita perinteisin keinoin on vaikea tunnistaa.

Asiakaskeskeiset organisaatiot keräävät tietoa asiakkaan jokaisesta kosketuspisteestä organisaatioon. Tätä dataa varten tarvitaan järjestelmiä, jotta koko organisaatiolle voidaan jakaa samat tiedot asiakkaasta. Tällöin asiakas voidaan kohdata personoidusti ja tarjota hänelle parhainta mahdollista palvelua. Näitä asiakastietoja keräämällä ja analysoimalla voidaan muodostaa ryhmiä ja segmenttejä asiakkaista, jolloin asiakasryhmille voidaan tarjota lähtökohtaisesti heille mahdollisesti sopivampaa palvelua. (Lamberti 2013; Bolton, 2004.)

Asiakaskeskeytyksestä keskusteltaessa nousee esille asiakkaan aktiivinen osallistuminen arvonluontiprosessiin. Asiakkaan osallistuttaminen osaksi yrityksen prosesseja on myös keino lisätä organisaation ja asiakkaan välistä dialogia. (Lamberti, 2013.) Lambertin (2013) tekemässä tutkimuksessa asiakaskeskeytyksen nähtiin tarkoittavan kykyä tarjota asiakkaalle hänen haluamiaan tuotteita hänen haluamallaan tavalla.

Kirjallisuudessa toimitusketjun roolia korostetaan vain osittain asiakaskeskeytyksen osana. Lamberti (2013) toteaa, että puhuttaessa asiakaskeskeytyksestä, tarkastellaan yleensä organisaation sisäisiä prosesseja. Hänen tekemässään haastattelututkimuksessa haastateltavat kertoivat asiakkaan ja yrityksen välisiä kohtaamispisteitä olevan useita ja näistä suurin osa voi olla organisaation ulkopuolella, kuten internetissä tai jälleenmyyjillä. Tämä aiheuttaa sen, että

organisaation on tarkemmin suunniteltava sitä, miten toiminnasta saadaan asiakaslähtöistä, myös organisaation ulkopuolella.

Lambertin (2013) tutkimuksessa haastateltava kertoi, että jopa 90% asiakkaan kohtaamisista tapahtuu organisaation ulkopuolella. Näiden kohtaamisten pitää olla yhdenmukaisia yrityksen linjauksien kanssa. Toimitusketju huomioidulla on mahdollista palvella asiakasta hänelle parhaalla tavalla, hänen jokaisessa kohtaamispisteessään. Näistä kohtaamisista voidaan kerätä tietoa asiakkaasta, ja tätä tietoa voidaan käyttää toisaalla asiakkaan palvelemiseksi. Tietoa kulkee silloin toimitusketjussa molempiin suuntiin. (Lamberti, 2013.)

Asiakaskeskeinen toiminta vaatii organisaatiolta erilaisia käytäntöjä ja toimenpiteitä. Näiden asiakaskeskeisten ja asiakaslähtöisten prosessien toteuttamiseksi organisaatioissa tulee tehdä myös rakenteellisia muutoksia, sillä asiakaskeskeisyys koskettaa organisaation kaikkia osia. Näissä organisaatioissa kaikki toiminnot ovat integroitu tuottamaan parempaa asiakasarvoa. Asiakkaiden tarpeiden asettaminen kaiken edelle vaatii organisaatiolta uusia tapoja työskennellä, sillä se vaikuttaa kaikkiin työntekijöihin organisaatioissa. Työskentelytapojen muutos vaikuttaa organisaation rakenteisiin. (Bolton, 2004.)

Organisaation kyky pitää asiakkaat ainutlaatuisina ja tarjota heille yksilöityä palvelua jokaisessa kohtaamispisteessä on yksi asiakaskeskeisen organisaation merkkeistä. Sen lisäksi, että asiakkaiden toiveet ja tarpeet on integroitu osaksi organisaation prosesseja, asiakaslähtöisissä organisaatioissa asiakkaat on integroitu osaksi yrityksen prosesseja. Nämä ovat tunnistettavimpia esimerkkejä asiakaskeskeisen organisaation piirteistä. (Lamberti, 2013.)

2.3 Yhteenveto asiakaskeskeisyydestä

Asiakaskeskeisyys ei ole uusi asia, mutta yrityksillä on yhä hankaluuksia sen hallitsemisessa. Asiakaskeskeisyyden perusajatuksena on asiakkaan halujen ja tarpeiden ymmärtäminen, jotta nämä voidaan täyttää parhaalla mahdollisella tavalla. Kovasti kilpailluilla markkinoilla asiakaskeskeisyyden nähdään tuovan yritykselle kilpailuetua.

Asiakaskeskeisyydellä pyritään luomaan kestäviä ja molemmin puolisesti tuottavia suhteita, jotka perustuvat molemmin puoliseen luottamukseen. Asiakas nähdään osana tuotekehitystä ja organisaation prosesseja, jolloin jokainen organisaation päätös on aidosti asiakaslähtöinen. Asiakkaan tarpeet ja halut vaikuttavat yrityksen toimintaan, sillä niitä ajatellaan organisaation jokaisella tasolla. (Bolton, 2004.)

3 OHJELMISTOKEHITYS

Tässä luvussa on käsitelty ohjelmistokehitystä tarkemmin, jotta on helpompi ymmärtää asiakaskeskeisyyden roolia osana ohjelmistokehitysprosesseja. Luvussa esitellään perinteinen ja ketterä ohjelmistokehitys. Tarkemmin tässä tutkimuksessa on keskitytty ketteriin menetelmiin. Ketterät menetelmät ovat perusarvoiltaan asiakaslähtöisempiä, kuin perinteiset menetelmät ja siten sopivat paremmin yhteen asiakaskeskeisen toiminnan kanssa. Eräs esimerkki ketteristä menetelmistä on Scrum, jota on tarkemmin kuvattu alaluvussa 3.2.

Ohjelmistokehityksellä tai ohjelmistotuotannolla tarkoitetaan kaikkia niitä organisaation työnteon menetelmiä, tekniikoita, työkaluja ja teorioita, joiden tarkoituksena on tuottaa tietokoneohjelmia tai -ohjelmistoja. Laajasti ymmärrettyinä ohjelmistokehitys sisältää kaiken organisaation toiminnan näiden ohjelmien valmistamiseen. Ohjelmistoja on valmistettu jo 1940-luvulta saakka, jolloin ensimmäiset nykyaikaiset tietokoneet kehitettiin. Nykyään tarve entistä laadukkaimmille ohjelmistoille on kova. Teknologian kehittyminen asettaa haasteita sekä mahdollisuuksia entistä parempien ohjelmiston tuottamiselle. Ympäristön jatkuvasti muuttuessa on tärkeää tutkia erilaisia tapoja, joilla voidaan tuottaa entistä parempia ohjelmistoja.

Ohjelmistokehityksessä käytettävät menetelmät voidaan jakaa kahteen ryhmään niiden piirteiden mukaan. Näitä ovat perinteiset Plan-Driven menetelmät ja uudemmat ketterät menetelmät. Menetelmien on tarkoituksena ohjata ohjelmistokehittäjiä organisoimaan ohjelmistokehitykseen liittyvää työtä. Ne sisältävät samat ohjelmistokehityksen vaiheet, mutta niiden toteuttamisjärjestys ja toteuttamistapa poikkeavat toisistaan.

3.1 Perinteinen ohjelmistokehitys

Perinteiset ohjelmistokehitysmenetelmät ovat yksiä varhaisimmista ohjelmistokehityksen menetelmistä. Niiden tarkoituksena oli hallita ensimmäisten tietokoneohjelmien kehitysprosesseja, vähentää ohjelmointivirheitä ja parantaa oh-

jelmistojen laatua. Näiden prosessien perusajatukset ovat edelleen käytössä, eivätkä ne ole juuri muuttuneet. Perinteisessä ohjelmistokehityksessä ohjelmistoprojekteilla nähdään olevan määrätynlainen elinkaari, joka on vaiheittain etenevä prosessi. Kaikkien perinteisten ohjelmistokehitysmallien toimintavaiheet ovat samankaltaisia, vaikka niiden toteutustapa voikin vaihdella. (Haikala & Märijärvi, 2004.)

Perinteisessä ohjelmistokehityksessä suunnittelu tehdään yksityiskohtaisesti ja hyvin aikaisessa vaiheessa kehitysprojektia (Leffingwell, 2007). Näissä menetelmissä kehitysprosessi on etukäteen suunniteltu ja hallinnoitu. Se jakaa kehitysprojektin vaiheisiin ja niihin liittyviin osiin. (Overhage, 2011.)

Tunnetuin perinteisen ohjelmistokehityksen menetelmistä on vesiputousmalli, jossa suunnittelu ja toteutus etenee vaiheittain alaspäin, kuin vesiputouksessa. Se alkaa esitutkimuksesta, jossa määritellään ratkaistava ongelma. Tätä seuraa määrittely, jonka tehtävänä on selvittää, millainen järjestelmä ratkaisee esitutkimuksessa määritetyn ongelman. Määrittelyn jälkeen alkaa järjestelmän suunnittelu, miten järjestelmä toteutetaan ja mitkä sen osat ovat. Suunnittelun jälkeen järjestelmä toteutetaan ja sen osat ohjelmoidaan. Viimeinen vaihe ennen käyttöönottoa ja ylläpitoa on järjestelmän integrointi ja testaus, jossa osat liitetään toisiinsa ja niiden toimivuus tarkastetaan. Jokaisen vaiheen lopussa pidetään katselmus, jossa tarkastellaan, onko kyseisen vaiheen vaatimukset täytetty ja voidaanko jatkaa seuraavaan vaiheeseen. Vesiputousmallissa, kuten muissakin perinteisissä menetelmissä, aikaisempiin vaiheisiin ei palata. Sen sijaan kehitysprojekti pysyy paikoillaan niin kauan, kunnes kyseinen vaihe on valmis. (McConnell, Toikkanen, & Arola, 2002.)

3.2 Ketterä ohjelmistokehitys

Ketteryys määritellään yleisesti ominaisuutena tai kykynä olla nopea ja vikkelä. Tietojärjestelmäkehityksessä ketteryydellä tarkoitetaan organisaation kykyä aistia ja vastata nopeasti odottamattomiin ympäristön muutoksiin. (Lyytinen & Rose, 2006.) Ketterien menetelmien syntymiseen vaikutti aikaisempien menetelmien raskaat byrokraattiset prosessit, jotka eivät taipuneet ympäristön muutoksiin tarpeeksi nopeasti ja joustavasti (Highsmith & Cockburn, 2001). Näissä perinteisissä menetelmissä nähtiin, että projektisuunnitelmat ja vaatimusmäärittelyt pystyttiin tekemään riittävällä tarkkuudella jo projektin alkuvaiheessa. Näin tarkasti etukäteen määritettyjen prosessien noudattaminen vaati kuitenkin valtaisesti resursseja aina muutoksien sattuessa. Kaikkiin muutoksiin ei kuitenkaan ole mahdollista etukäteen valmistua ja käytännössä ohjelmistoprojektit harvoin valmistuivat niille esitetystä budjetista tai ajassa. 1990-luvulla ohjelmistokehittäjät muodostivatkin erilaisia kevyempiä työtapoja, joissa muutokset hyväksytään niiden välttämisen sijaan. (Williams & Cockburn, 2003.)

Vuonna 2001 kevyiden menetelmien edustajia kokoontui yhteen työpajassa, jonka tuloksena syntyi ketterän ohjelmistokehityksen julistus eli *Manifesto for*

Agile software development (Williams & Cockburn, 2003). Tutkimuksessani käytän julistuksesta sen lyhempää englanninkielistä nimeä *Agile Manifesto*.

Qumer ja Henderson-Sellers (2006) määrittelevät ketteryyden sellaiseksi käyttäytymiseksi, jossa odotettuihin tai odottamattomiin muutoksiin reagoidaan joustavasti ja nopeasti. Toimenpiteet tehdään yksinkertaisilla, taloudellisilla ja laadukkailla välineillä joustavassa ympäristössä.

Abrahamsson, Salo, Ronkainen ja Warsta (2002) näkevät myös yksinkertaisuuden ja nopeuden ketterien menetelmien tärkeimpinä ominaisuuksina. Ketterät menetelmät syntyivät siis tarpeeseen, jossa perinteisillä menetelmillä ei pystytty vastaamaan muutoksiin riittävällä nopeudella.

Agile Manifesto esittää neljä arvoa, joissa yhdistyy parempi tapa tehdä ohjelmistoa. Arvot ovat jaettu 12 tarkemmaksi periaatteeksi, jotka esittävät miten niitä toteutetaan käytännössä. Arvot ja periaatteet ovat esitettyinä taulukoissa 1 ja 2, jotta lukija saa tarvittavan käsityksen mitä ketterällä menetelmällä tarkoitetaan tässä tutkielmassa. Manifeston arvot ovat niin laajasti levinneet ja tunnetut, että Agile Manifesto voidaan nähdä määritelmäksi ketterälle ohjelmistokehitykselle. Taulukossa 1 esitellään Agile Manifeston arvot ja taulukossa 2 sen periaatteet. (Beck, Beedle, Van Bennekum, Cockburn, Cunningham, Fowler & Kern, 2001.)

Taulukko 1. Agile Manifeston arvot (mukaillen Beck ym., 2001)

1. Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja.
2. Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota.
3. Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja.
4. Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa.

Taulukko 2. Agile Manifeston periaatteet (mukaillen Beck ym., 2001)

1. Tärkeintä on täyttää asiakkaan vaatimukset julkaisemalla jatkuvasti ja aikaisin uusia hyödyllisiä versioita ohjelmistosta.
2. Hyväksytään ja otetaan vastaan muuttuvat vaatimukset, jopa kehityksen loppuvaiheessa. Ketterät menetelmät valjastavat muutoksen asiakkaan kilpailueduksi.
3. Luovutetaan toimivia versioita kehitettävästä ohjelmistosta säännöllisesti, mielellään lyhyin väliajoin muutamasta viikosta muutamaan kuukauteen.
4. Liiketoiminnan ammattilaisten ja kehittäjien täytyy työskennellä päivittäin yhdessä koko projektin ajan.
5. Rakennetaan projektit motivoituneiden yksilöiden ympärille ja annetaan heille ympäristö ja tuki, jota he tarvitsevat, sekä luotetaan, että he saavat työn tehtyä.
6. Kaikkein tehokkain tapa välittää tietoa kehitystiimille ja kehitystiimissä, on kasvokkain tapahtuva keskustelu.
7. Toimiva ohjelmisto on ensisijainen edistymisen mitta.
8. Ketterät menetelmät suosivat kestävästä kehitystä. Rahoittajien, kehittäjien ja käyttäjien tulisi kyetä pitämään jatkuvasti yllä tasainen työtahti.
9. Jatkuva huomion kiinnittäminen tekniseen laatuun, sekä hyvään rakenteeseen ja suunnitteluun, lisää ketteryyttä. (jatkuu)

Taulukko 2 (jatkuu)
10. Yksinkertaisuus - taito maksimoida työn määrä, jota ei tarvitse tehdä - on olennaista.
11. Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat nousevat itseorganisoituvista tiimeistä.
12. Tiimi pysähtyy säännöllisesti miettimään kuinka tehostaa toimintaansa.

Agile Manifeston mukaisissa ohjelmistokehitysmenetelmissä ohjelmistoa tuotetaan inkrementaalisesti ja iteratiivisesti. Ohjelmistoprojektit koostuvat lyhyistä sykleistä, joissa toteutetaan toimivaa ohjelmistoa käymällä läpi kaikki ohjelmistokehityksen vaiheet vaatimusmäärittelystä valmiin ohjelman luovutukseen. Jokaisella syklillä ohjelmistoon lisätään uusia ominaisuuksia. Lyhyiden syklien ansiosta asiakas on jatkuvasti osana kehitystyötä, asiakas arvioi syklin aikana saadut tulokset ja priorisoi järjestelmän uusia vaatimuksia. Asiakkaan ja kehittäjän välinen kommunikointi on tärkeää, jolloin muutoksien sattuessa niihin voidaan reagoida tarpeen vaatiessa. (Abrahamsson ym., 2002.) Conboy (2009) esittää ketterien menetelmien hyödyntävän muutoksista oppimista ja aikaisemmista projekteista opitun tiedon hyödyntämistä.

Agile Manifesto on julistus paremmasta tavasta tehdä ohjelmistoja. Se itsessään ei sisällä työkaluja tai menetelmiä ohjelmistoprojektien tekoon. Siksi sen toteuttamiseen tarvitaan jokin menetelmä. Ketteriä ohjelmistokehitysmenetelmiä on olemassa paljon, näistä tunnetuimmat ovat Scrum ja Extreme Programming (XP). Tässä alaluvussa esitellään Scrum, joka on eniten tutkittu ja yleisemmin käytetyin ketterän kehittämisen menetelmä. (Schwaber & Sutherland, 2011.)

Sutherland on esitellyt Scrumin perusteet jo vuonna 1995. Scrumia kuvailaan menetelmäksi, jossa kehityksen ennustettavuus sekä odottamattomat muutokset hyväksytään ja niiden ratkaisemiseksi tehdään kaikki, mitä se vain vaatii. Schwaber & Sutherland (2011) mukaan ”Scrum on viitekehys, jonka puitteissa ihmiset voivat ratkaista monimutkaisia ongelmia kehittäessään tuotteita tuottavasti ja luovasti pyrkien mahdollisimman korkeaan lisäarvoon”. Scrum perustuu empiiriselle prosessinhallinnan teorialle. Scrumissa ketterien menetelmien tapaan työskennellään iteratiivisesti ja inkrementaalisesti, ennustettavuuden optimoimiseksi ja riskien kontrolloimiseksi. (Schwaber & Sutherland 2011.)

Scrumissa asiakkaan tehtäviä on esimerkiksi tehdä käyttäjätarinoita, joiden pohjalta kehittäjät voivat suunnitellaan ohjelmiston toteutusta. Asiakkaan tehtäviä on myös vaatimuksien priorisointi sekä niistä päättäminen, testaamiseen osallistuminen ja tulosten hyväksyminen. (Abrahamsson ym., 2002.)

Scrumissa ohjelmistokehitys alkaa asiakkaan konkretisoidessa visionsa tuotteesta vaatimusten muotoon. Näitä vaatimuksia priorisoidaan kehitysprosessin ajan. Vaatimukset määritellään yksittäisiksi suoritettaviksi tehtäviksi. Scrumissa kehitettävä ohjelmisto kehittyy pikkuhiljaa yhä täydellisemmäksi useiden kehitysjaksojen aikana. Kehitysjaksolle valitaan tuotteen työlistalta (*product backlog*) suoritettavat tehtävät. Kehitysjaksoja kutsutaan sprinteiksi ja ne ovat Scrumin tunnistavimpia piirteitä. Sprintit ovat 1-4 viikon mittaisia syklejä.

Scrumissa korostuu kehittäjien ja asiakkaiden välinen vuorovaikutus ja tähän on olemassa erilaisia työkaluja, kuten päivittäiset kokoontumiset (*daily scrum*). Sprinteissä työskennellään toistavasti ja lisäävästi, tavoitteena on saada kaikki työlistalla olevat tehtävät tehtyä. Sprintin lopussa järjestetään sprinttikatselmus (*sprint review*), jossa esitellään sprintin saavutukset ja arvioidaan sidosryhmien edustajien kanssa, täyttääkö tuote sille asetetut vaatimukset. Ennen seuraavan sprintin aloittamista kehitystiimi arvioi aikaisemman sprintin ja esittää tulevalle sprintille parannettavat kohdat. Tätä kutsutaan sprintin retrospektiiviksi. (Kautz, Johansen, & Uldahl, 2014.)

Vaikka kehittäjien ja asiakkaiden välinen vuorovaikutus on tärkeää, täytyy muistaa, ettei asiakkaalla välttämättä ole tarvittavaa tietotaitoa päättää ohjelmistoprojektin suunnasta. Martin, Biddle ja Noble (2004) toteavat tutkimuksessaan, ettei asiakkaan rooli ole kirjoittaa koodia tai korjata sitä. Se on kehittäjien rooli, kuten myös kehitysmenetelmän valinta. Ketterien menetelmien valinta voi olla hyväksi esimerkiksi nopeasti muuttuvassa ympäristössä, jossa muutoksiin tulee vastata nopeasti, innovatiivisesti ja kustannustehokkaalla tavalla.

4 ASIAKASKESKEINEN OHJELMISTOKEHITTÄMINEN

Tutkielman toisessa ja kolmannessa luvussa on määritelty asiakaskeskeisyyttä ja ohjelmistokehitystä. Tässä luvussa käsitellään perinteisen ja ketterän lähestymistavan eroja ohjelmistokehityssä. Taulukossa 3 esitetyt menetelmien eroavaisuudet tuodaan esille asiakaskeskeisestä näkökulmasta. Luvun lopussa esitellään millaisia piirteitä asiakaskeskeinen ohjelmistokehitys saa ohjelmistoalalla.

Yrityksen menestyksen loppukädessä päättävät aina sen asiakkaat. Ohjelmistoalan yritykset ovat tässäkin suhteessa täysin samanlaisia, kuin muillakin markkinoilla. Yrityksien on tärkeä keskittyä yhä enemmän asiakkaisiinsa ja hakea kilpailuetua liiketoiminnalleen yhdessä asiakkaiden ja heiltä saadun palautteen mukaan.

Ohjelmistokehitystä tehdään karkeasti jaettuna kahdella erilaisella lähestymistavalla, perinteisellä ja ketterällä. Molemmilla menetelmillä on omat vahvuutensa ja ne soveltuvat erilaisiin tarpeisiin. Luvussa onkin tarkemmin esitelty näiden menetelmien eroavaisuuksia. Tuntemalla menetelmien eroavaisuuksia voidaan projektiin käytetty menetelmä valita paremmin. Oikein valittu ja toteutettu menetelmä helpottaa projektin valmistumista sekä parantaa asiakastytyväisyyttä.

Organisaation kestävä kasvun perustana tulisi olla asiakkaiden tilanteiden ja tarpeiden ymmärtäminen. Tuntemalla asiakas ja hänen tarpeensa voidaan kehitysmenetelmäkin katsoa asiakkaan kykyjen ja tarpeiden mukaan. Yrityksien lähtökohtana ei voi olla pelkästään tarve tehdä laadultaan ensiluokkaisia ohjelmaa, vaan ohjelmien pitää olla tehty aidosti palvelemaan asiakkaita mahdollisimman hyvin. Asiakkaat onkin nähtävä yksilöinä, joilla on omat uniikit tarpeensa. Asiakkaat eivät ole samanlaisia, vaikka he kuuluisivat samaan segmenttiin; sama ohjelmisto ei välttämättä palvele kaikkia asiakkaita yhtä täydellisesti.

4.1 Perinteisen ja ketterän ohjelmistokehitysmenetelmän vertailu

Tässä alaluvussa käsitellään taulukko 3, jossa perinteisen ja ketterän ohjelmistoksen eroja on esitetty Boehmin (2002) mukaan. Taulukossa on seitsemän tarkastelun kohdetta, joita tarkastellaan valitun kehitysmenetelmän mukaan. Jäljempänä nämä seitsemän tarkastelun kohdetta on avattu sanallisesti auki ja esitetty lyhyesti, millaisia piirteitä ne saavat käytännössä.

Taulukko 3. Perinteisen ja ketterän menetelmän eroja (mukaillen Boehm, 2002)

Tarkastelun kohde	Ketterät menetelmät	Perinteiset menetelmät
Kehittäjät	Ketteriä, asioista perillä olevia, samassa tilassa ja yhteistyössä toimivia	Suunnittelupainoitteinen, riittävät taidot, pääsy ulkoiseen tietoon
Asiakkaat	Omistautuneita, asiantuntevia, samassa tilassa ja yhteistyössä toimivia, edustavia, valtuutettuja	Pääsy tietoon, yhteistyökykyisiä, edustavia, valtuutettuja
Vaatimukset	Muodostuvat projektin edetessä; Nopeasti muuttuvia	Tiedossa projektin alussa; Muuttumattomia
Arkkitehtuuri	Suunniteltu nykyisille vaatimuksille	Suunniteltu nykyisille ja ennakoituille vaatimuksille
Refaktorointi	Edullista	Kallista
Koko	Pienempiä projekteja ja tiimejä	Suuremmat tiimit ja projektit
Ensisijainen tavoite	Nopea arvo	Korkea varmuus

Kehittäjien nähdään ketterissä menetelmissä olevan ketterämpiä ja osaavampia, kuin perinteisissä menetelmissä. Ketterät tiimit ovat kooltaan huomattavasti pienempiä ja muutoksille avoimempia eli ketteriä. Tiimit ovat itseohjautuvia ja pystyvät päätöksentekoon itsenäisesti. Ne toimivat samassa tilassa jakaen tietoa keskenään ja auttaen toisiaan. Pienissä tiimeissä kehittäjät toimivat useammassa roolissa yhtä aikaa. (Conboyn, Coyle & Pikkarainen., 2011.) Perinteisessä ohjelmistokehityksessä kehittäjät ovat osa suurempaa kokonaisuutta, jossa he hoitavat oman etukäteen suunnitellun osansa. (Boehm, 2002)

Boehmin (2002) mukaan ketterä kehittäminen vaatii merkittävästi enemmän asiakkaan läsnäoloa ja osaamista kehitysprojektissa. Perinteisessä ohjelmistokehityksessä asiakas on aktiivisesti mukana vain projektin alussa, määrittelemässä ohjelmiston vaatimukset, ja lopussa hyväksymässä valmiin tuotteen. Ketterässä kehityksessä asiakas voi olla osa kehitystiimiä, työskennellen kehittäjien kanssa samassa tilassa jatkuvasti antaen palautetta.

Conboyn ja kumppaneiden (2011), mukaan asiakkaan rooli ketterissä menetelmissä on merkittävä. Asiakasta osallistavilla menetelmillä on vahvuutensa ja

heikkoutensa. Jatkuva palautteen ja uusien ominaisuuksien toivominen tuotteen voi hankaloittaa sen valmistumista (Kautz, 2011).

Vaatimukset ohjelmistolle on perinteisessä kehitysprojektissa tiedossa etukäteen ja ne eivät juurikaan muutu tuotteen elinkaaren aikana. Jos muutoksia tulee, on niihin myös valmistauduttu. Ketterissä menetelmissä vaatimukset muodostuvat projektin edetessä ja ne ovat luonteeltaan nopeasti muuttuvia. Nopeasti muuttuvissa olosuhteissa perinteiset menetelmät eivät pysy perässä, sillä etukäteen suunniteltuja vaiheita on vaikeaa tai jopa mahdotonta muuttaa projektin aikana. Ketterät menetelmät taas loistavat muuttuvissa olosuhteissa, sillä ohjelmistoa rakennetaan inkrementaalisesti ja iteroiden.

Ohjelmistoarkkitehtuurin tarkoituksena on mahdollistaa ohjelmiston jatkuva toiminta, se yhdistää tekniset vaatimukset ja liiketoiminnalliset tarpeet. Perinteisissä menetelmissä se on suunniteltu nykyisille ja ennakoiduille vaatimuksille. Boehmin (2002), mukaan ketterissä menetelmissä arkkitehtuuri on suunniteltu nykyisille tiedossa olevissa vaatimuksille.

Hadar ja Sherman (2012) käsittelevät tutkimuksessaan perinteisten ja ketterien menetelmien arkkitehtuurin eroja. Heidän mukaansa ketterissä menetelmissä ohjelmistoarkkitehdit joutuvat olemaan enemmän yhteydessä sidosryhmiin läpi ohjelmiston elinkaaren. Tämä voi jopa parantaa ohjelmiston laatua.

Refaktoroinnilla tarkoitetaan toimivan ohjelmistokoodin muuttamista, ilman uusien ominaisuuksien lisäämistä. Sitä tarvitaan esimerkiksi silloin, kun ohjelman lähdekoodia halutaan muokata laadukkaammaksi tai helpommin luettavaan muotoon. Sellaisia muutoksia, joiden tarkoituksena on parantaa ohjelmiston rakennetta, kutsutaan refaktoroinneksi. Ketterissä menetelmissä kaikkien muutoksien tekeminen on joustavaa ja siten edullista. Perinteisissä menetelmissä muutoksien teko jälkeenpäin on aina haastavaa ja siten kallista. (Boehm, 2002.)

Yleisesti perinteisillä menetelmillä tehdyt ohjelmistokehitysprojektit ovat suurempia ja vaativat suurempia tiimejä. Ketterät menetelmät loistavat pienemmissä projekteissa ja niitä tehdään pienemmillä tiimeillä. Ketterien menetelmien tarkoituksena on saavuttaa nopeita tuloksia, tuottamalla asiakkaalle nopeasti vaatimukset täyttävä tuote. Perinteiset menetelmät taas pyrkivät korkeaan luotettavuuteen ja varmuuteen, joka mahdollistetaan tarkalla suunnittelulla ja aukottomalla dokumentoinnilla.

4.2 Asiakaskeskeinen liiketoiminta ohjelmistoalalla

Agile Manifesto korostaa asiakasyhteistyötä yhtenä ketterän ohjelmistokehittämisen keskeisimmistä arvoista (Beck ym., 2001). Highsmithin ja Cockburnin (2001) mukaan tämä näkyy asiakkaan ja kehittäjän välisenä läheisenä asiakassuhteena. Kun asiakassuhde on läheinen, se mahdollistaa dialogin asiakkaan kanssa ja jatkuvan palautteen saamisen asiakkaalta. Tällainen keskusteleva asiakassuhde on huomattavasti läheisempi, kuin perinteisten menetelmien dokumentaation kautta tapahtuva vuoropuhelu.

Asiakaskeskeisen toiminnan periaate ei välttämättä ole sopiva kaikille organisaatioille, sillä sen implementoimiseen vaikuttavat myös monet organisaation ulkopuoliset asiat. Kaikissa organisaation prosesseissa ei välttämättä ole tarpeellista olla täysin asiakaslähtöinen. Lamberti (2013) esittää, että asiakaskeskeisiä periaatteita voidaan ottaa yrityksissä käyttöön myös vain osittain.

Tutkimukset ovat osoittaneet, että ketterissä menetelmissä asiakkaat ovat olleet tyytyväisiä mahdollisuudesta antaa palautetta ja olla mukana organisaation prosesseissa (Dybå & Dingsøyr, 2008). Asiakkaat nähdään erittäin tärkeänä tietolähteenä. Ohjelmistokehittäjät voivat käyttää tätä tietolähdettä avukseen ohjelmistoja tehdessään. Tämän vuoksi on tärkeää pitää asiakkaan ja kehittäjän välinen asiakassuhde läheisenä (Hope & Amdahl, 2011).

Pitämällä asiakkaat osana ohjelmistokehitystä, ohjelmistosta todennäköisemmin valmistuu sellainen, jota asiakkaat ja käyttäjät käyttävät myös jatkossa. Vaikka ohjelma olisi kuinka laadukkaasti toteutettu, se ei ole onnistunut, mikäli sillä ei ole käyttäjiä. (Subramanyam, Weisstein & Krishnan, 2010.)

Grönroos ja Helle (2010) toteavat, ettei asiakkaat aina välttämättä edes huomaa tuotteen laadun parantumista, eivätkä siten ole valmiita maksamaan siitä korkeampaa hintaa. Organisaatioilla onkin tärkeää olla käsitys siitä, millaiset ominaisuudet palvelussa asiakas näkee tärkeinä. Näin voidaan säästyä turhan työn tekemiseltä. Bolton (2004) toteaa, että ilman asiakaskeskeistä ajattelua pahimmillaan organisaatioissa päädytään tekemään vääristä prosesseista entistä tehokkaampia, implementoidaan väärää teknologiaa ja hukataan merkittäviä investointeja.

Yleisesti pidetään hyvänä tapana, että asiakas on läheisesti mukana ohjelmistokehitysprojekteissa. Ketterissä menetelmissä tämä onkin yleinen toimintatapa. Toisaalta Kautz (2011) esittää, että asiakaslähtöisyys voi myös haitata ohjelmistokehitystä. Tämä saattaa ilmetä varsinkin silloin, kun asiakas on paikan päällä mukana kehityksessä. Joissain tapauksissa asiakkaalta tulevat jatkuvat muutosehdotukset voivat olla liikaa tai niin sekavia, ettei niitä voi kehittäjät suoraan ymmärtää. (Hope & Amdahl, 2011.) Asiakkaan pyytämiin uusiin ominaisuuksiin keskittyminen monesti johtaa virheherkkään ja tietoturvattomaan ohjelmistoon (Mead, 2004).

Ketterät menetelmät vaativat myös asiakkaalta huomattavasti suurempaa osaamista, koska myös heillä tulisi olla riittävä ymmärrys ohjelmistonsuunnittelusta (Boehm, 2002). Vaikka jatkuvan asiakaspalautteen saaminen voi olla ohjelmiston kehittämiselle riski, on se kuitenkin onnistumisen kannalta ratkaisevassa roolissa.

Ries (2011) ehdottaa, että yrityksiä pitäisi omaksua ketteristä menetelmistä käytäntöjä liiketoimintansa kehittämiseksi. Näitä käytäntöjä voi olla esimerkiksi kokeilukulttuuri tarkan suunnittelun sijasta ja asiakaspalautteen parempi hyödyntäminen. Ohjelmistokehityksen ketteryys ei itsessään tarkoita, että organisaatio olisi ketterä. Ketteriä arvoja ja periaatteita voidaan kuitenkin soveltaa koko organisaatioon, jotta organisaation prosessit ja työntekijät ovat olosuhteisiin mukautuvia. (Pikkarainen, Haikara, Salo, Abrahamsson, & Still, 2008.)

Kasvavissa ohjelmisto-organisaatioissa käytetään asiakaslähtöistä suuntautumista, jolloin yritykset ymmärtävät asiakkaitaan ja heidän tarpeitaan paremmin (Hakala & Kohtamäki, 2011). Aktiivisen asiakasyhteistyön ja asiakkailta saadun palautteen avulla organisaatiot voivat tehdä kilpailuetua ja löytää uudenlaisia liiketoimintamahdollisuuksia. Menestyminen ei ole ominaisuuksien toimittamista, vaan menestys tulee asiakkaan ongelmien ratkaisemista (Ries, 2011). Tämä on organisaatiolle mahdollista vain tuntemalla asiakkaansa (Bolton, 2004). Shah ja kumppanit (2006) kertoivat tutkimuksessaan organisaation empatian määrän asiakkaitaan kohtaan parantavan organisaation asemaa markkinoilla.

Pelkästään asiakkaaseen ja asiakastyytyväisyyteen keskittymien ei ole mahdollista käytännössä. Liiketoiminnan tulee tasapainotella monien yrityksen sidosryhmien tarpeiden kanssa. Yrityksien tulee siis yhtä lailla keskittyä asiakkaidensa ja yrityksen omistajien tarpeisiin. (Gummesson, 2008.)

Asiakaslähtöisyydestä voidaan sanoa tulleen jopa kulutustavaraa, sillä kun jokainen toimittaja panostaa asiakassuhteeseensa, niin siitä tulee yleinen tapa ja samalla sen tuoma kilpailuetu pienenee. Asiakkaiden odotusten noustessa yrityksillä ei ole varaa olla olematta asiakaslähtöisiä. Kun asiakkaat ovat yhä vaativampia saamastaan palvelusta, voi se aiheuttaa yrityksissä kustannusten nousua. (Gummesson, 2008.)

5 YHTEENVETO

Tässä kandidaatintutkielmassa on tutkittu kirjallisuuskatsauksen keinoin asiakaskeskeisyyttä ja ohjelmistokehitystä. Tavoitteena on yleisellä tasolla määritellä asiakaskeskeisyys ja ohjelmistokehitys. Ohjelmistokehityksestä on esitelty perinteinen ohjelmistokehitys ja ketterät menetelmät. Tutkimus pyrkii vastaamaan siihen, millaista asiakaskeskeisyys on ohjelmistokehityksessä ja miksi asiakaskeskeinen ohjelmistokehitys on tärkeää.

Tutkimus selvittää, mitä asiakaskeskeisyydellä tarkoitetaan osana ohjelmistokehitystä sekä millaisia käytänteitä sillä on. Näiden ymmärtäminen on tärkeää, koska se tarjoaa tietämystä, miksi ketterissä menetelmissä asiakaslähtöinen toiminta on edellytys ohjelmistoprojektien onnistumiselle.

Asiakaskeskeisyys on käsitteenä laaja ja se pitää sisällään useita alakäsitteitä, kuten asiakaslähtöisyyden. Itsessään asiakaskeskeisyyttä on ollut merkityksellistä tutkia, sillä sen nähdään olevan yksi tärkeimmistä keinoista yrityksen hakiessa vaikeasti kopioitavaa kilpailuetua. (Lamberti, 2013). Myös empatian määrä asiakkaitaan kohtaan on nähty parantavan yrityksen asemaa markkinoilla (Shah ym., 2006).

Highsmith ja Cockburn (2001) esittävät läheisen asiakassuhteen olevan molempien osapuolten etu. Läheinen dialoginen vuorovaikutussuhde mahdollistaa asiakkaalle ketterämmän tavan antaa palautetta ja kehittäjälle mahdollisuuden palautteen avulla tuottaa asiakasta miellyttävä ohjelmisto. Asiakkaat ovatkin erittäin tärkeä tietolähde ja tämän vuoksi on tärkeää pitää asiakkaan ja kehittäjän välinen asiakassuhde läheisenä. (Hope & Amdahl, 2011.) Asiakkaat ovat tyytyväisiä mahdollisuudesta antaa palautetta ja olla mukana organisaation prosesseissa (Dybå & Dingsøyr, 2008).

Asiakaslähtöisyys voi myös haitata ohjelmistokehitystä, varsinkin kun asiakas on paikan päällä mukana kehityksessä (Kautz, 2011). Asiakkailta tulevat jatkuvat muutosehdotukset ja uusien ominaisuuksien vaatiminen, asiakkaiden kaikkiin pyyntöihin vastaaminen voi johtaa virheherkkään ja tietoturvattomaan ohjelmistoon (Hope & Amdahl, 2011; Mead, 2004).

Tässä tutkielmassa on keskitytty ketteriin menetelmiin, sillä ne ovat arvoiltaan ja periaatteiltaan lähempänä asiakaslähtöisiä menetelmiä. Ketterät me-

netelmät vaativat myös asiakkaalta huomattavasti suurempaa osaamista, sillä heilläkin tulisi olla riittävä ymmärrys ohjelmistonsuunnittelusta (Boehm, 2002).

Ketterät menetelmät ovat perusarvoiltaan asiakaslähtöisiä, mutta niiden asiakaskeskeisiä käytänteitä ei ole täysin sisäistetty ohjelmistokehitysorganisaatioissa. Pikkarainen ja kumppanit (2008) ehdottavat, että ketteriä arvoja ja periaatteita voidaan soveltaa myös koko organisaatioon, jotta myös muut organisaation prosessit ja työntekijät olisivat olosuhteisiin mukautuvia.

Hadar ja Sherman (2012) toteavat ketterien menetelmien tuoneen paljon arvoa ohjelmistokehitykseen, parantaen ohjelmiston laatua ja nopeutta ohjelmistotuotteiden markkinoille pääsyä. Tutkimuksessa esitellyiden kehitysmenetelmien välillä on suuriakin eroavaisuuksia, vaikka molemmat, sekä ketterät että perinteiset menetelmät tähtäävät samaan lopputulokseen; asiakkaan vaatimuksia vastaavan ohjelmiston valmistumiseen. Menetelmien väliltä on kehittäjien pystyttävä valitsemaan asiakkaan tilanteeseen parhain menetelmä.

Ohjelmistokehittäjien ja heidän asiakkaidensa välinen vuorovaikutus on tärkeää. On kuitenkin muistettava, ettei asiakkaan rooli ole kirjoittaa koodia tai korjata sitä. Tämä ja kehitysmenetelmien valinta ovat aina kehittäjien tehtävä.

Tämä tutkimus on esitelty asiakaskeskeisyyttä ja ohjelmistokehitystä. Tutkimuksen tarkoituksena on ollut määrittellä asiakaskeskeisyys ja tuoda esille perinteisen ja ketterän ohjelmistokehityksen eroavaisuuksia. Näiden teemojen pohjalta on esitelty, millaisia piirteitä asiakaskeskeisyys saa ohjelmistokehityksessä ja miten sitä voidaan hyödyntää liiketoiminnassa. Tutkimuksen ja oman kehittymiseni kannalta on ollut tärkeää käsitellä laajasti asiakaskeskeisyyttä, jotta voidaan paremmin ymmärtää, miten asiakaskeskeisyys liittyy ohjelmistokehitykseen.

Jatkotutkimuksen kannalta olisi mielenkiintoista tarkastella lähemmin asiakkaan ja kehittäjien välistä yhteistyötä, sen hyötyjä ja mahdollisia haasteita. Ohjelmistokehitysmenetelmät ovat kehittyneet valtavasti ketterien menetelmien tulon jälkeen ja tutkimusta voisi jatkossa laajentaa koskemaan tuoreempia ketterien menetelmien suuntauksia, kuten DevOpsia.

LÄHTEET

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and Analysis. Espoo: VTT Publications 478.
- Antikainen, M., Mäkipää, M., & Ahonen, M. (2010). Motivating and supporting collaboration in open innovation. *European Journal of Innovation Management*, 13(1), 100-119.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M. & Kern, J. (2001). Manifesto for agile software development.
- Blocker, C. P., Flint, D. J., Myers, M. B., & Slater, S. F. (2011). Proactive customer orientation and its role for creating customer value in global markets. *Journal of the Academy of Marketing Science*, 39(2), 216-233.
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64-69.
- Bolton, M. (2004). Customer centric business processing. *International Journal of Productivity and Performance Management*, 53(1), 44-51.
- Boulding, W. (2005). A Customer Relationship Management Roadmap: What Is Known, Potential Pitfalls, and Where to Go. *Journal of Marketing*, 69(4), 155-166.
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329-354.
- Conboy, K., Coyle, S., Xiaofeng, W. & Pikkarainen, M. (2011). People over Process: Key Challenges in Agile Development. *Software IEEE*, 28(4), 48-57. Los Alamitos: IEEE Computer Society.
- Drucker, P. 1954. *The Practice of Management*. New York: Harper & Brothers.
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9), 833- 859.
- Evans, I. (2004). *Achieving software quality through teamwork*. Norwood: Artech House. Haettu 10.5.2018 osoitteesta <http://ebookcentral.proquest.com/lib/jyvaskyla-ebooks/detail.action?docID=227672>

- Grönroos, C., & Helle, P. (2010). Adopting a service logic in manufacturing: Conceptual foundation and metrics for mutual value creation. *Journal of Service Management*, 21(5), 564-590.
- Gummesson, E. (2008). Customer centricity: Reality or a wild goose chase? *European Business Review*, 20(4), 315-330.
- Haikala, I., & Märijärvi, J. (2004). *Ohjelmistotuotanto* (10. uud. p. ed.). Helsinki: Talentum.
- Hakala, H., & Kohtamäki, M. (2011). Configurations of entrepreneurial-customer-and technology orientation: Differences in learning and performance of software companies. *International Journal of Entrepreneurial Behavior & Research*, 17(1), 64-81.
- Hadar, I., & Sherman, S. (2012, June). Agile vs. plan-driven perceptions of software architecture. In *Proceedings of the 5th International Workshop on Co-operative and Human Aspects of Software Engineering* (pp. 50-55). IEEE Press.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-122.
- Hope, K. L., & Amdahl, E. (2011). Configuring designers? Using one agile project management methodology to achieve user participation. *New Technology, Work and Employment*, 26(1), 54-67.
- Kautz, K. (2011). Investigating the design process: Participatory design in agile software development. *Information Technology & People*, 24(3), 217-235.
- Kautz, K., Johansen, T. H., & Uldahl, A. (2014). Creating business value through agile project management and information systems development: The perceived impact of scrum. Paper presented at the International Working Conference on Transfer and Diffusion of IT, 150-165.
- Kumar, V., & Shah, D. (2004). Building and sustaining profitable customer loyalty for the 21st century. *Journal of Retailing*, 80(4), 317-329.
- Lamberti, Lucio. (2013). Customer centricity: the construct and the operational antecedents. *Journal of Strategic Marketing*, august, 1-25.
- Leffingwell, D. (2007). *Scaling software agility: best practices for large enterprises*. Pearson Education.
- Lyytinen, K., & Rose, G. M. (2006). Information system development agility as organizational learning. *European Journal of Information Systems*, 15(2), 183-199.

- Löytänä, J. (2014). *Asiakkaan aikakausi: Rohkeus + rakkaus = raha*. Helsinki: Talentum.
- Martin, A., Biddle, R., & Noble, J. (2004). The XP customer role in practice: Three studies. In *Agile Development Conference, 2004* (pp. 42-54). IEEE.
- McConnell, S., Toikkanen, T., & Arola, J. (2002). *Ohjelmistotuotannon hallinta*. Helsinki: Edita, IT Press.
- Mead, N. R. (2004). Who is liable for insecure systems? *Computer*, 37(7), 27-34.
- Mohammadi, S., Nikkahan, B., & Sohrabi, S. (2009). Challenges of user involvement in extreme programming projects. *International Journal of Software Engineering and its Applications*, 3(1), 19-32.
- Overhage, S. (2011). What Makes IT Personnel Adopt Scrum? A Framework of Drivers and Inhibitors to Developer Acceptance. *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, 1-10.
- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3), 303-337.
- Prahalad, C. K., & Ramaswamy, V. (2004). Co-creation experiences: The next practice in value creation. *Journal of Interactive Marketing*, 18(3), 5-14.
- Qumer, A., & Henderson-Sellers, B. (2006). Comparative evaluation of XP and scrum using the 4D analytical tool (4-DAT). Paper presented at the *Proceedings of the European and Mediterranean Conference on Information Systems*, 1-8.
- Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Random House LLC.
- Ruotsalainen, P. (2000). *Asiakaslähtöinen palveluketju ja tietoteknologia*. Teoksessa S.Nouko-Juvonen, P.Ruotsalainen & I.Kiikkala (Toim.) *Hyvinvointivaltion Palveluketjut*. Helsinki: Tammi, 7-32.
- Schwaber, K., & Sutherland, J. (2011). *The scrum guide*. Scrum Alliance, 21.
- Schwartz, E. (2004). IT myth 5: Most IT projects fail. Haettu 11.5.2018 osoitteesta: <http://www.infoworld.com/article/2667144/technology-business/itmyth-5--most-it-projects-fail.html>
- Shah, D., Rust, R. T., Parasuraman, A., Staelin, R., & Day, G. S. (2006). The path to customer centricity. *Journal of Service Research*, 9(2), 113-124.

- Subramanyam, R., Weisstein, F. L., & Krishnan, M. S. (2010). User participation in software development projects. *Communications of the ACM*, 53(3), 137- 141.
- Sun, B., Li, S., & Zhou, C. (2006). "Adaptive" learning and "proactive" customer relationship management. *Journal of Interactive Marketing*, 20(3-4), 82-96.
- Sun, Li, J. R. (2002). Organizing to deliver solutions. *Organizational Dynamics*, 31(2), 194-207.
- Williams, L., & Cockburn, A. (2003). Guest editors' introduction: Agile software development: It's about feedback and change. *Computer*, 36(6), 39-43.