



OULUN YLIOPISTO  
UNIVERSITY of OULU

# **Käyttäjäkeskeisen suunnittelun sovittaminen ketteriin menetelmiin**

Oulun yliopisto  
Tietojenkäsittelytieteen laitos  
Luonnontieteiden kandidaatin tutkielma  
Miia-Maria Mikkola  
10.12.2015

## Abstrakti

Jokaisen yrityksen tarkoitus on olla mahdollisimman kilpailukykyinen kilpailijoihinsa verrattuna. Kilpailukykyä parantaakseen yritykset ja organisaatiot usein yrittävät parantaa tuotantoaan ja tuotteitaan, jotta ne olisivat parempia kuin kilpailijoilla. Sekä ketterät menetelmät että käyttäjäkeskeinen suunnittelu tarjoavat työkalut kilpailukyvyn parantamiseen. Tämän tutkielman tarkoitus on esitellä, miten ja miksi ketterät menetelmät ja käyttäjäkeskeinen suunnittelu ovat nousseet niin suosituiksi kehitysmetodeiksi ja miten ne on omaksuttu yrityksissä ja organisaatioissa. Lisäksi tarkastellaan, mitä hyötyjä ja haasteita nämä menetelmät tuovat yrityksille ja organisaatioille sekä yksinään että yhdistettynä.

### *Avainsanat*

Käyttäjäkeskeinen suunnittelu, ketterät menetelmät, metodi, asiakas, käyttäjä, kehittäjä, jakso (sprint)

### *Ohjaaja*

Yliopisto-opettaja, Mikko Rajanen

# Sisällys

Abstrakti.....	2
Sisällys .....	3
1. Johdanto.....	4
2. Aiempi tutkimus .....	5
2.1 Ketterien menetelmien ja käyttäjakeskeisen suunnittelun kehitys yritysmaailmassa.....	5
2.1.1 Ketterien menetelmien nousu .....	5
2.1.2 Käyttäjakeskeisyys osaksi ketteriä menetelmiä.....	6
2.1.3 Ketterien menetelmien omaksuminen suuryrityksissä käyttäjakeskeistä suunnittelua käyttäen.....	9
2.2 Metodien hyödyt eriteltynä ja yhdistettynä.....	12
2.2.1 Ketterien menetelmien hyödyt .....	12
2.2.2 Käyttäjakeskeisen suunnittelun hyödyt .....	13
2.2.3 Yhdistämisen hyödyt .....	13
2.3 Metodien heikkoudet eriteltynä ja yhdistettynä.....	14
2.3.1 Ketterien kehitysmetodien heikkoudet .....	15
2.3.2 Käyttäjakeskeisen suunnittelun heikkoudet.....	15
2.3.3 Yhdistämisen heikkoudet .....	16
2.4 Tulevaisuuden näkymiä .....	17
3. Pohdinta.....	19
3.1 Ketterien menetelmien ja käyttäjakeskeisen suunnittelun kehitys ja yhdistyminen.....	19
3.2 Ketterien menetelmien ja käyttäjakeskeisen suunnittelun hyödyt ja haasteet.....	20
3.3 Mitä tulevaisuudessa kannattaa ottaa huomioon?.....	21
4. Yhteenveto.....	22
Lähdeluettelo.....	24
Liite1: Tutkimussuunnitelma: Käyttäjakeskeisen suunnittelun sovittaminen ketteriin menetelmiin.....	26

# 1. Johdanto

Joseph Barjisin (2008) mukaan kaksi viime vuosikymmentä ovat hyvin osoittaneet, miten ohjelmistoyritykset ovat epäonnistuneet huonon mallintamisen vuoksi. Jopa 18 % kaikista ohjelmistohankkeista epäonnistuvat täysin, eivätkä syyt ole esimerkiksi teknisissä epäonnistumisissa. Onneksi tähän epäkohtaan on ryhdytty puuttumaan yhä enemmän ja enemmän ja yrityksissä on otettu käyttöön eri työkaluja, joiden avulla lähdetään parantamaan suunnittelua.

Tänä päivänä ketterät menetelmät ovat nousseet suureen arvoon ohjelmistoyritysten keskuudesta. Suurin osa yrityksistä, jotka eivät vielä ole omaksuneet ketteriä menetelmiä, ovat kiinnostuneet ottamaan sen osaksi toimintatapaansa. On siis selvä, että yritykset ympäri maailmaa ovat havainneet ketterät menetelmät hyödylliseksi oman toimintansa kannalta. (Laanti et al., 2011) Ketterien menetelmien lisäksi myös toisenlainen suunnittelu on noussut hyvin tärkeäksi ja suosituksi ohjelmistoyritysten keskuudessa, nimittäin käyttäjäkeskeinen suunnittelu.

Käyttäjäkeskeisestä suunnittelusta tuli suosittua siinä vaiheessa, kun verkossa käytävän kaupankäynnin määrä lisääntyi. Tuolloin kuluttajista tuli käyttäjiä, kun he saattoivat hoitaa juoksevia asioitaan kotoaan käsin erilaisen ohjelmien avulla. (Mao et al., 2005) Tällöin oli otettava huomioon, että kaikki käyttäjät eivät ole yhtä tietoisia tai kiinnostuneita ohjelmista kuin ammattilaiset, joten heidät ja heidän osaamisensa tulee ottaa huomioon myös suunnittelussa. (de Ruyter, van de Sluis, 2006)

Tutkimusongelmanani tässä tutkimuksessa onkin aiempiin tutkimuksiin perustuen selvittää,

1. Miten nämä kaksi erilaista työskentelytapaa on sulautettu toisiinsa ja kuinka ne toimivat yhdessä?
2. Mitä hyötyjä ja heikkouksia puolia kummastakin löytyy sekä yksin että yhdessä?

Käsittelen myös sitä, kuinka nämä alat ovat kehittyneet, sulautuneet yhteen ja mitkä ovat niiden tulevaisuuden näkymät.

## 2. Aiempi tutkimus

Vuonna 2001 julkaistiin Ketterän ohjelmistokehityksen julistus, Agile Manifesto, ja sen seurauksena tutkijayhteisö lähti todenteolla kehittämään ketterää ohjelmistokehitystä. Tämä julistus esitteli ketterien menetelmien ydinajatuksia ja sai ne kuulostamaan uskomattoman hyviltä. (Dingsøyr et al., 2013) Ensimmäiseksi julistuksessa luvattiin asettaa ihmiset prosessien yläpuolelle, toiseksi vähentää turhaa työtä dokumentoimalla vain kaikkein tärkeimmät asiat, kolmanneksi ottaa asiakkaat ja muut sidosryhmät mukaan tuotekehitykseen ja neljänneksi kyetä kontrolloimaan muutosta ja ottaa se huomioon. (Beck et al., 2001) Tämä ei kuitenkaan täysin riittänyt kattavaan ohjelmistokehitykseen. Haluttiin suunnitella ja kehittää tuotteita tai ohjelmistoja, joita käyttäjät voisivat käyttää tehokkaasti ja olla täysin tyytyväisiä niihin. (Baek et al., 2008) Tässä tarkastellaan, miten ketterät menetelmät ja käyttäjäkeskeinen suunnittelu ovat kehittyneet toistensa rinnalle ja sulautuneet yhteen. Lisäksi tarkastellaan näiden metodien ja niiden yhdistelmän vahvuuksia ja heikkouksia. Loppujen lopuksi, jos nämä menetelmät eivät olisi missään vaiheessa osoittaneet, että todella toimivat ja parantavat tuotantokehitystä, ei niistä olisi tullut niin iso osa nykyistä ohjelmistokehitystä. (Laanti et al., 2011)

### 2.1 Ketterien menetelmien ja käyttäjäkeskeisen suunnittelun kehitys yritysmaailmassa

Koko 2000-luku on ollut todella hektistä ohjelmistokehityksen näkökulmasta. Jatkuvasti on esitelty uusia metodeja, työkaluja, tekniikoita ja harjoituksia, joilla parantaa ohjelmistokehitystä ja luoda parempia tuotteita. Ketterät menetelmät tarjosivat paljon sellaista, mitä yritykset ja organisaatiot halusivat kilpailukykyensä parantamiseen, joten se nousi hyvin suosituksi. Tässä metodissa asiakkaat liittyivät hyvin tiiviisti tähän prosessiin antamalla omaa palautettaan ja mielipiteitään, jotta tuote vastaisi paremmin tarkoitettua. (Dingsøyr et al., 2013) Myöhemmin asiakkaiden lisäksi tuotekehitykseen tulivat loppukäyttäjät eli henkilöt, jotka oikeasti tulisivat käyttämään tuotettuja tuotteita tai ohjelmistoja.

#### 2.1.1 Ketterien menetelmien nousu

Tarve menetelmäpohjaiselle lähestymistavalle ohjelmistokehityksessä tuli Lycettin ja muiden (2003) mukaan jo 1968–1969 ohjelmistokriisin aikaan, kun the North Atlantic Treaty Organization (NATO) kehotti standardisoimaan ohjelmistokehityksen ja parantavan laatua, mutta samalla vähentävän kulutusta ja lisäävän tuottavuutta. Ketterien menetelmien syntyminen onkin ollut yksi merkittävimmistä muutoksista ohjelmistokehityksessä. Monet katsovat ketterien menetelmien syntyneen 1970-luvulla, mutta ketterien menetelmien ideat ovat kuitenkin olleet käytössä jo ennen sitä. Tästä syystä harvat ketterien menetelmien sisältämistä ideoista ovat uusia. Ensimmäinen ketterien metodien todellinen malli oli vesiputousmalli, jota edelleen monet lähteet suosittelivat käytettäväksi (Abbas et al., 2008). Ideana vesiputousmalli ei kuitenkaan ole täysin toimiva. Nykyään sen pohjalta onkin suunniteltu useita uusia malleja, jotka

kuvaavat ohjelmistosuunnittelua paljon paremmin. Losadan ja muiden (2013) mukaan ketterien menetelmien kehitystä johtivat ne kehittäjät, jotka ensimmäisenä halusivat keskittyä luomaan parhaan mahdollisen lopputuloksen.

Ketterät menetelmät esitettiin yritysmaailmalle viimein 1990-luvulla, jolloin yritykset alkoivat kunnolla kiinnostua niistä. (Laanti et al., 2011). Tuolloin niiden esittely etenkin suurille yrityksille koettiin hankaliksi (Dingsøyr, Dybå, 2009). Suurissa yrityksissä oli monesti jo valmiina omat järjestelmänsä ja tapansa tehdä asioita, joten tämän uuden tekniikan omaksumisessa kuviteltiin olevan oma haasteensa. Dingsøyrin ja Dybånin (2009) mukaan ketterien menetelmien esittely oli kuitenkin helpompaa pienemmille yrityksille ja nämä kokivatkin ketterät menetelmät helpoiksi omaksua ja hyvin toimiviksi.

Vaikka aluksi ketterät menetelmät koettiin vain pienten yritysten väliseksi, pian isommissakin yrityksissä alettiin huomata, että näiden menetelmien avulla tuotteet saataisiin nopeammin markkinoille, olemalla kuitenkin yhtäaikaisesti mukautuva ja joustava. Suurista yrityksistä esimerkiksi Yahoo! omaksui ketteristä menetelmistä Scrumin vuonna 2005. Ensin menetelmää testattiin vain neljällä tiimillä, jotka sitten jakoivat tulokset muun yhtiön kesken. Yhteensä 74 % koki menetelmän paremmaksi kuin entiset toimintatavat ja muutkin tiimit ja johtoporraskin alkoivat todella kiinnostua siitä. Kun oli aika ottaa menetelmä käyttöön koko yrityksessä, se omaksuttiin pilottiohjelman kautta. Tähän ohjelmaan sisältyi keskustelua, koulutusta sekä valmennusta. Tämä toteutettiin yksi tiimi kerrallaan ja vuoden kuluttua neljä tiimiä oli noussut jo neljäksikymmeneksi tiimiksi. (Benefield, 2008)

Lopullisesti ketterät menetelmät todettiin päteviksi, kun kyseltiin eri ryhmien mielipidettä ketteristä menetelmistä. Esimerkiksi asiakkailta kysyttäessä palautetta, he sanoivat olevansa oikein tyytyväisiä uusiin toimintatapoihin ja niiden saavuttamiin tuloksiin. Kuitenkin kaikkein tyytyväisimpiä olivat kehittäjät. Heistä uuden menetelmän toimintatapa oli paljon tehokkaampi ja teki työn teosta helpompaa ja tuotteesta paremman. Myös yliopiston opiskelijat kokivat ketterät menetelmät hyödyllisiksi tulevaisuuden kannalta ja uskoivat menetelmien parantavan tuottavuutta. He kuitenkin ajattelivat, että niin sanottu parikoodaus oli vaikeaa, jos parien osaamisen erot olivat erityisen suuria. (Dingsøyr, Dybå, 2009) Ketterät menetelmät tulivat siis jäädäkseen. Nykyisin monissa yrityksissä on luovuttu parikoodauksesta ja se on muutettu tiimien väliseksi, joissa jokainen tiimin jäsen työskentelee itsenäisesti osana tiiminä.

Ketteriin menetelmiin siirtyminen on kuitenkin iso askel mille tahansa yritykselle. Tätä siirtymää on kuvattu koko yrityksen kehityskulttuurin ja organisatorisen kulttuurin muutoksena. Tämä yksinään selittää jo sen, miksi siirtyminen on usein koettu niin vaikeaksi ja isoksi asiaksi. (Dingsøyr et al., 2010)

### 2.1.2 Käyttäjäkeskeisyys osaksi ketteriä menetelmiä

Ennen kuin ketterien menetelmien ja käyttäjäkeskeisen suunnittelun yhdistämiseen ryhdyttiin kunnolla panostamaan, tilanne oli hyvin sekava. Ketterät menetelmät olivat juuri tulleet tutuksi ja niitä alettiin omaksumaan yrityksissä ja mukaan yritettiin epätoivoisesti sulattaa hieman käyttäjäkeskeistä suunnittelua ottamalla asiakkaat mukaan suunnitteluun. Tulos oli katastrofi. Ohjelmistot toimitettiin myöhässä, niissä oli ohjelmointivirheitä ja usein ne eivät edes vastanneet toivottua lopputulosta. Tämä johtuu siitä, että käyttäjävaatimukset jätettiin usein vähemmälle huomiolle ja

pahimmillaan arvioitiin väärin. Usein kävikin ilmi, että tärkeiksi luullut ominaisuudet paljastuivat tarpeettomiksi ja tarpeettomiksi arvioidut oli jätetty kokonaan pois, vaikka niitä olisi ehkä tarvittu. (Patton 2002)

Ohjelmistokehityksessä onkin tärkeää muistaa, että vaikka asiakas tilaakin tuotteen, hän ei välttämättä ole tuotteen lopullinen ostaja tai käyttäjä. Loppukäyttäjien tärkeys on vähitellen alkanut selviämään myös ohjelmistotuottajille ja siksi nykyään onkin hyvin tyypillistä, että loppukäyttäjiä kuullaan jo hyvin varhaisessa vaiheessa ohjelmistokehitystä. Tämä helpottaa tuotekehityksen priorisointia, jaksottamista sekä julkaisua. (Sy, 2007) Loppukäyttäjien huomioiminen muutti ohjelmistokehitystä siinä määrin, että viimeistään nyt alettiin suunnitella ohjelmistoja käyttäjiä ajatellen, jotka oikeasti tulisivat lopullista tuotetta käyttämään ja alettiin painottamaan termiä: käyttäjakeskeisyys. Käyttäjakeskeisyys näkyy suunnitteluvaiheessa jo sillä, että loppukäyttäjiltä kysytään, mitä he haluavat, toteutusvaiheessa sillä, että loppukäyttäjät osallistuvat testaukseen ja julkaisuvaiheessa sillä, että uusi ohjelmisto vastaa loppukäyttäjän tarpeita mahdollisimman hyvin ja sen käyttöönotto on helpompaa. (Baek et al., 2008)

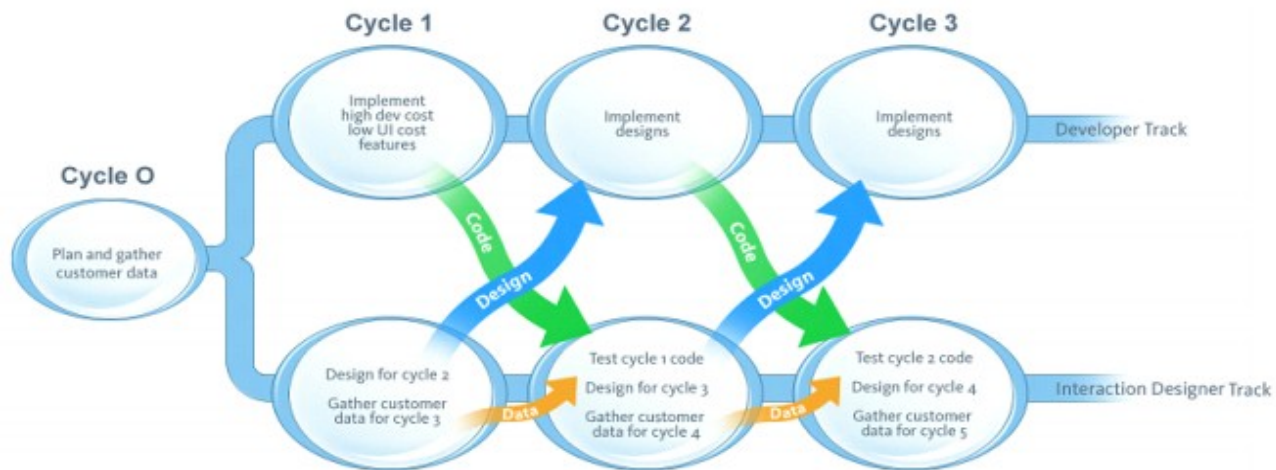
Ketterien menetelmien tarkoituksena on ollut aina paitsi parantaa yrityksen tuottavuutta myös miellyttää asiakasta. Jotta loppukäyttäjien olisi helpompi käyttää lopullisia ohjelmistoja ja laitteita, niihin tulee luoda toimiva käyttöliittymä. Ketterät menetelmät eivät kuitenkaan yksinään tee käyttöliittymästä helppokäyttöistä ja toimivaa. (Fox et al., 2008) Ketterissä menetelmissä keskitytään enemmän tuotteen toimituksen vaatimiin asioihin (Patton 2002), kun taas käyttäjakeskeisessä suunnittelussa pyritään korjaamaan käyttäjien huomaamat virheet prosessin aikana, esimerkiksi käyttäjätestausta käyttäen. (Barnum 2002) Käyttäjakeskeisen suunnittelun yhdistäminen ketteriin menetelmiin on lähinnä näiden kahden toimintatavan yhdistämistä.

Myös Losadan ja muiden (2013) mukaan sekä ketterät menetelmät että käyttäjakeskeinen suunnittelu on molemmat tarkoitettu olemaan käyttäjä-/asiakaskeskeisiä metodeja. Ero näiden välillä vain on se, kuinka ne toimivat päämääränsä hyväksi ja mihin keskitytään kehitysvaiheessa. Voisi kuvitella, että koska näissä metodeissa on kuitenkin kohtalaisesti samoja elementtejä, yhdistäminen ei olisi niin hankalaa. Ketterissä menetelmissä on kuitenkin ominaisuus, joka vaikuttaa huomattavasti käyttäjakeskeiseen suunnitteluun. Ketterissä menetelmissä koko toteutus on jaettu pienempiin jaksoihin (sprint), jotka pituudeltaan voivat olla viikosta kuukauteen organisaatiosta riippuen. (Sy, 2007)

Patton (2002) alkoi ensimmäisten joukossa kehittää ketterien menetelmien ja käyttäjakeskeisen suunnittelun yhdistämistä yrityksessään Tomax Technologies. Tuolloin testiryhmälle ensin esiteltiin ketteristä menetelmistä Extreme Programming (XP). Kun ryhmä oli sisäistänyt tämän toimintatavan, he alkoivat liittää mukaan käyttäjakeskeisyyttä. Tuolloin tämä käyttäjakeskeisyys kuitenkin ymmärrettiin eritavoin kuin nykyään. Silloin riitti, että kehittäjät sisäistivät sen kenelle he olivat sitä suunnittelemassa ja osasivat asettua käyttäjän rooliin käyttäjätestauksen ajaksi. Täydellistä yhdistelmää etsittäessä esille nousi mallipohjainen kehittäminen. Mallipohjaisessa kehityksessä luodaan malli eli prototyyppi lopullisesti tuotteesta ja annetaan se käyttäjätestaukseen. Aluksi nämä mallit annettiin asiakkaille testattavaksi, mutta lopulta ymmärrettiin ottaa myös loppukäyttäjiä mukaan testaukseen. (Losada et al., 2013)

Kuinka nämä kaksi metodologiaa on sitten yhdistetty toimivasti? Sy (2007) on keksinyt tähän ratkaisun (Kuva 1). Tässä mallissa, kuten myös molemmissa metodeissa, projekti alkaa

suunnittelulla ja tarvittavien tietojen keräämisellä asiakkaalta ja myös mahdollisuuksien mukaan loppukäyttäjältä, kuvassa Cycle 0. Tämän jälkeen alkaa itse tuotteen työstäminen. Seuraavat jaksot, Cycle 1-3, kuvastavat ketterien menetelmien mukaisia jaksoja. Ylempi haara kuvastaa kehittäjien linjaa, eli ketteriä menetelmiä ja alempi taas suunnittelijoiden eli käyttäjakeskeisen suunnittelun linjaa. Ensimmäisessä jaksossa, Cycle 1, luodaan ensimmäinen prototyyppi. Kehittäjät valmistavat prototyypin rakentamalla aluksi rungon ohjelmistolle ja suunnittelijat tekevät tuotteen ulkonäköön liittyvät suunnitelmat ja jatkavat neuvotteluja asiakkaan kanssa, millainen tuotteen tulee olla. Ensimmäisen jakson jälkeen kehittäjät luovuttavat suunnittelijoille koodin käyttäjätestausta varten ja suunnittelijat luovuttavat suunnitelmat kehittäjille. Toisessa jaksossa, Cycle 2, kehittävät luovat saamiensa tietojen pohjalta uuden prototyypin ja suunnittelijat suorivat käyttäjätestauksen ja luovat uudet suunnitelmat kolmatta jaksoa varten ja keräävät asiakkaalta ja loppukäyttäjältä tietoja neljättä jaksoa varten. Tällä tavalla suunnittelu on aina askeleen edellä toteutustimiä, joten muutosten sattuessa, suunnitelmat ehditään vielä korjata. Näin jatkuu aina kaikkien jaksosten loppuun asti, kunnes tuote on valmiina.



Kuva 1. Malli, kuinka ketterät menetelmät ja käyttäjakeskeinen suunnittelu saadaan yhdistettyä (Sy, 2007)

Tämä on toki vain yhden yrityksen versio metodien yhdistämisestä. Ketterien kehitysmetodien ja käyttäjakeskeisen suunnittelun yhdistäminen on hyvin organisaatiokohtaista. Se riippuu täysin siitä, kuinka nämä kaksi asiaa tulkitsee. Koska ketterissä menetelmissä on jo itsestään niin paljon elementtejä, jotka viittaavat käyttäjakeskeiseen suunnitteluun, on mahdollista ajatella, että ketterät menetelmät itsessään ovat jo näiden metodien yhdistelmä. Joissain organisaatioissa ne jopa ovat. (Gangjun et al., 2009)

Ideana ketterien menetelmien ja käyttäjakeskeisen suunnittelun yhdistäminen on suorastaan ideaalinen. Ketterät menetelmät edesauttavat saamaan markkinoille parempia tuotteita nopeammin ja käyttäjakeskeinen suunnittelu auttaa saavuttamaan loppukäyttäjien täydellisen tyytyväisyyden tuotteeseen. (Patton 2002) Tämä yhdistäminen on myös miltei pakollista, jos tavoitteena on luoda tuote, joka laadullisesti ja hinnaltaan vastaa täysin asiakkaan toiveita. Totuus on, että kumpikaan näistä metodeista ei yksinään pysty moiseen. Siinä missä ketterät menetelmät pyrkivät käyttäjien avulla selvittämään, että tuote toimii oikein, niin käyttäjakeskeinen suunnittelu katsoo, että koko tuote on halutunlainen ja tehokas. (Losada et al., 2013)



### 2.1.3 Ketterien menetelmien omaksuminen suuryrityksissä käyttäjäkeskeistä suunnittelua käyttäen

Kuten aiemmin on todettu, ketterät menetelmät koettiin enemmän pienten yritysten väliseksi, koska suuremmat yritykset olivat joutuneet luomaan jo valmiiksi omat systeeminsä, jolla saivat yrityksen toimimaan. (Laanti et al., 2011) Tämä ei kuitenkaan ole koko totuus. On myös suuria yrityksiä, joilla ei ole aiemmin ollut käytössä minkäänlaista metodia teolliseen suunnitteluun. Koska pienet ja keskisuuret yritykset olivat jo omaksuneet ketterät menetelmät, se antoi niille huomattavasti kilpailuetua, joten suurten yritysten oli myös lähdettävä kehittämään toimintaansa ketteräksi, jotta ne pysyisivät mukana kilpailussa. (Gangjun et al., 2009)

Ohjelmistokehityksessä loppukäyttäjistä puhuttaessa on tärkeää huomata, mitä nämä käyttäjät oikeastaan ovat. Usein heidät mielletään "tavalliseksi ihmisiksi", joilla ei ole teknillistä koulutusta tai tietotaitoa. (Sy, 2007) Seuraavissa tapauksissa loppukäyttäjät ovat kuitenkin kahden suuren ohjelmistokehitysyrityksen työntekijöitä, jotka omaksuvat täysin uuden menetelmän ja uudet työkalut työnsä tekemiseksi.

Suuryrityksissä ketterien menetelmien omaksuminen ei ole helppo, eikä nopea prosessi. Yleisesti tästä siirtymävaiheesta ei ole tehty niin paljon tutkimuksia (Laanti et al., 2011), mutta onneksi tutkimukset löytyivät kahden tunnetun suuryrityksen siirtymisestä siihen. Yahoo!:n ja Nokian tapauksissa ketterät menetelmät on otettu käyttöön erittäin käyttäjäystävällisesti ja loppukäyttäjät huomioiden.

#### 2.1.3.1 *Yahoo! ennen ketterien menetelmien omaksumista ja sen jälkeen*

Yahoo!:n kohdalla tarkastellaan, kuinka ketterien menetelmien omaksuminen alkoi. Alussa yrityksellä ei ole käytössä vielä mitään vastaavaa menetelmää ja kehittäjiä on esiteltävä kokonaan uusi metodi yrityksen työntekijöille ja ohjeistaa heitä sen käytössä. Yahoo! aloitti ketterien menetelmien puolesta puhumisen jo vuonna 2002 vesiputousmallilla, mutta silloin tämä ajatus ei saanut kovinkaan paljon kannatusta. Vuonna 2004 Tobias Mayer tuli puhumaan Yahoo!:lle eräästä ketterästä menetelmästä nimeltä Scrum ja tämä sai huomattavasti lämpimämmän vastaanoton. Helmikuussa 2005 Yahoo!:ssa otettiin käyttöön ensimmäinen Scrumin pilottiohjelma. (Benefield, 2008)

Aluksi yrityksestä valittiin vain neljä vapaaehtoista tiimiä, jotka ryhtyivät toteuttamaan uusia tuotteita Scrumin alaisuudessa. Aluksi nämä tiimit saivat todella kattavan koulutuksen alan asiantuntijoilta. Seuraavaksi se saivat ulkopuolisen Scrumvalmentajan, joka työskenteli heidän kanssaan usean ensimmäisen jakson aikana. Heiltä kerättiin palautetta heti ensimmäisen 30-päiväisen jakson jälkeen sekä kahden kuukauden kuluttua. (Benefield, 2008) Näiden palautteiden avulla kehittäjätiimi sai arvokasta tietoa siitä, miten heidän pilottiryhmänsä omaksui Scrumin ja kuinka hyödylliseksi he kokivat sen. Tämän perusteella on siten helppo tehdä päätös, kannattaako kyseinen menetelmä ottaa käyttöön koko yrityksessä.

Kahden kuukauden kokeilun jälkeen pilottiryhmä oli hyvin kiintynyt Scrumiin, eikä halunnut luopua siitä, joten se päätettiin ottaa osaksi koko yritystä. Aluksi etenkin vanhemmat johtajat olivat tätä ajatusta vastaan (Benefield, 2008), mikä oli ymmärrettävää. Yahoo!:n kokoisessa yrityksessä koko henkilökunnan kouluttaminen yksistään vaatii paljon aikaa ja rahaa. Jos tulos ei ole halutunlainen, koko yritys voisi olla isoissa ongelmissa. Vuonna 2008, kun Benefield julkaisi artikkelinsa ketterien menetelmien tuloksista, ei suuryritysten käytössä niitä ollut kovin paljon. Benefieldin

(2008) mukaan kehittäjätiimi sai kuitenkin johdon vakuuttuneeksi Scrumin hyväksymisestä ja se päätettiin lopullisesti omaksua koko yritykseen.

Kun Scrum päätettiin esitellä muille työntekijöille, sitä varten luotiin oma ohjelmansa. Tässä ohjelmassa oli tarkoitus aluksi vain keskustella ihmisten kanssa ja kertoa heille, kuinka Scrum toimii, laatia aikataulut ja organisoidusti kouluttaa sen käyttöön. Seuraavaksi tavattiin tuotteen omistaja ja luotiin yhdessä hänen kanssaan tuotteen kehitysjono. Tämän jälkeen pidettiin yhteinen kahdenpäivän mittainen koulutusjakso, jonka jälkeen oli tiimeille yksilöllinen valmennusjako, jossa jaksotettiin tuotekehitys. (Benefield, 2008) Tällä tavalla loppukäyttäjille annettiin kaikki mahdollinen tuki ja valmius työskennellä Scrumia käyttäen. Valmentajien ansiosta käyttäjien oli mahdollista kysyä neuvoa ja saada apua jopa jaksojen aikana. Äkkiseltään voisi kuvitella, että tämän kaltainen tuotteen käyttöönotto toisi paljon hyviä tuloksia nopeasti.

Kuin kuvitella saattaa, tämän kokoisessa projektissa kaikki ei kuitenkaan mennyt aivan suunnitelmien mukaan. Vastaan tulivat ensinnäkin rahalliset ongelmat. Tietysti tälle kaikelle oli laadittu budjetti, mutta se ei ollut kyllin pätevä kattamaan koko organisaation koulutusta. Tällöin kehitystiimin oli todistettava, että kaikki oli tuon rahan arvoista. Kehittäjät tutkivat tuottavuutta ja selvisi, että tuottavuus oli noussut 34 % vuodessa. Tuo ei kuitenkaan ollut tarpeeksi suuri luku vakuuttamaan johtajia. Vastaus kuitenkin löytyi talousosastolta. He loivat yksinkertaisen taulukkolaskenta mallin, joka todisti, että yksi valmentaja oli säästänyt yritykselle jopa 1.5 miljoonaa dollaria vuodessa. (Benefield, 2008)

Raha ei kuitenkaan ollut ainoa tekijä, joka aiheutti ongelmia. Tämä käyttöönotto osoitti, että ihmiset ovat inhimillisiä, mikä voi aiheuttaa vastoinkäymisiä. Kun Scrum oli esitelty työntekijöille, heti toisen jakson jälkeen moni oli jo unohtanut, kuinka uuden metodin pitäisi toimia. Oli jopa tiimejä, jotka ilmoittivat käyttävänsä Scrumia, mutta he olivat jättäneet pois kaikki tärkeät avaintekijät, kuten päivittäiset Scrumpalaverit. Menetelmä kuitenkin oli hyväksi havaittu useimpien työntekijöiden toimesta. Kuitenkin työntekijöiden joukosta löytyi ihmisiä, joiden mielestä siirtyminen oli huono ajatus ja entiset menetelmät olivat aivan hyviä. (Benefield, 2008) Yahoo!':n kokoisessa yrityksessä on toki mahdotonta, että kaikki tulisivat pitämään uusista ideoista ja kannattamaan niitä. Ihmisten tietämättömyys ja unohtaminen on helppo korjata pienellä kertauksella ja tärkeiden asioiden painotuksella. Suuressa yrityksessä ongelmana on, ettei yksilöllinen neuvominen ja opettaminen ole mahdollista ja kestää aikaa ennen kuin tulokset alkavat kunnolla näkyä.

Tässä kuitenkin on näkyvillä kaikki käyttäjäkeskeisen suunnittelun peruselementit, ensinnäkin jatkuva palautteen anto, haastattelut sekä käyttäjien tarkkailu. Lisäksi kehittäjien on ansaittava käyttäjiensä luottamus ja puhuttava kieltä, jota käyttäjät ymmärtävät. Kuitenkaan kaikesta tuosta huolimatta kaikki eivät ikinä tule pitämään muutoksesta. Yleisesti on todettu, että 10–15 % käyttäjistä vastustaa ajatusta. Jos kukaan ei vastusta, käyttäjille on todennäköisesti kerrottu vain, mitä he haluavat kuulla, eikä sitä mikä heidän pitäisi kuulla. (Benefield, 2008) Kun nämä käytännöt ja tiedot yhdistää ketterien menetelmien kanssa on todella helppo luoda uusi tuote nopeasti ja tehokkaasti ja siten, että se vastaa mahdollisimman hyvin käyttäjien tarpeita.

### *2.1.3.2 Nokia ketterien menetelmien omaksumisen aikana ja jälkeen*

Nokian tapaus on tavallaan jatkoa Yahoo!':n tapaukselle. Tässä ei siis niinkään kiinnitetä huomiota siihen, kuinka ketterät menetelmät otettiin käyttöön, vaan siihen, miten työntekijät ympäri maailmaa ottivat ketterät kehitysmenetelmät vastaan. Tiedot tähän

kerättiin kyselyillä seitsemästä eri maasta ja useilta eri sidosryhmiltä ympäri Eurooppaa, Pohjois-Amerikkaa ja Aasiaa. Tämä tutkimus suoritettiin vuonna 2008 heti sen jälkeen, kun Nokia oli ottanut käyttöönsä ketterät menetelmät. (Laanti et al., 2011) Tässä vaiheessa Nokia on siis jo tehnyt kaiken tuon, mistä puhuttiin Yahoo!n kohdalla ja nyt vertaillaan työntekijöiden tuntemuksia tämän suuren muutoksen jälkeen. Käyttäjakeskeisen suunnittelun näkökulmasta uusi metodi on nyt otettu käyttöön ja tarkastellaan, ovatko lopulliset käyttäjät tyytyväisiä uuteen tapaansa työskennellä. Toisin sanoen tämä kuvastaa todella hyvin käyttäjätestausta.

Syy, miksi Nokia halusi siirtyä ketterien menetelmien käyttöön, oli se, että yrityksessä oli jo käytössä useita elementtejä ketteränä työskentelystä, joten he halusivat ottaa ne käyttöön täysivaltaisesti ja kunnolla (Laanti et al., 2011). Kuten Abbas ja muut (2008) totesivat, ketterät menetelmät ovat loppujen lopuksi uudelleen käytettyjä ja uudelleen organisoituja vanhoja ideoita. Ei siis mikään ihme, että Nokian kokoisella, menestyneellä yrityksellä oli jo osa näistä ajatuksista käytössään ainakin jollain tasolla.

Kuten käyttäjätestauksessa yleensä, tässäkin kerättiin ensin tietoa ihmisten tauoista. Haluttiin tietää, kuinka paljon kokemusta käyttäjillä oli ketterien menetelmien ja perinteisten menetelmien käytöstä sekä heidän havainnoistaan ja tyytyväisyydestään ketteriin menetelmiin. Päälimmäisenä tutkimusongelmana haluttiin tietää, vaikuttiko käyttäjän kokemus hänen mielipiteeseensä siitä ja olivatko ketterät menetelmät parempia kuin perinteiset menetelmät. (Laanti et al., 2011)

Tuloksista voitiin nähdä, että 60 % oli sitä mieltä, että ketterät menetelmät olivat parempia ja niihin vaihtaminen todella kannatti. Osa näin vastanneista ei ollut vielä edes työskennellyt ketteriä menetelmiä käyttäen. Vain 9 % ketteriä menetelmiä käyttäneistä halusi palata vanhaan menetelmään. (Laanti et al., 2011) Kuten Benefield (2008) totesi, aina on joku, joka ei pidä muutoksesta.

Lisäksi kyselyssä oli kaksi avointa kysymystä selvittämään, mitä hyvää ja mitä huonoa ketterissä menetelmissä oli työntekijöiden mielestä. Tämä osio ei kuitenkaan ollut kovin suosittu. Useat vastaajista jättivät nämä kohdat kokonaan vastaamatta ja niissä, joihin näihin oli vastattu, osa vastauksista ei auttanut tutkimusta lainkaan. Eniten negatiivista palautetta kaikilta vastanneilta sai osakseen itse käyttöönotto ja prosessin parannus, mutta näkyvyys ja avoimuus saivat positiivista palautetta jopa siinä ryhmässä, joka halusi palata vanhoihin tapoihin. Käyttöönotossa käyttäjät eivät pitäneet siitä, että koko idea heitettiin koko yritykselle yhtäaikaaisesti. Näkyvyydessä ja avoimuudessa pidettiin siitä, että joka aamuisissa palavereissa saatiin aina tietää, mitä toinen oli tekemässä ja näin kaikilla oli selvä näkemys, mihin oltiin menossa ja mitä tahtia. Mieluiten takaisin vanhaan halusivat palata ne, joilla oli vähiten käytännön kokemusta työskentelystä ketterien menetelmien kanssa. Tutkijat pyrkivät myös selvittämään, miksi he halusivat palata vanhaan. Tähän vastauksena oli vain kokemuksen puute. (Laanti et al., 2011) On täysin selvää, että negatiivista kritiikkiä tulee kaikessa käyttäjätestauksessa ja juuri siksi täytyykin selvittää tarkoin, mistä tämä johtuu ja voiko sen eteen tehdä jotain. Vaikka kyse on kokemuksen puutteesta, se ei tarkoita etteikö organisaatio voisi tehdä asialle jotain.

Kaiken kaikkiaan on selvää, että ne joilla oli enemmän kokemusta jo aiemmin ketteränä työskentelystä, pitivät myös metodia parempana. Tällä tavalla Nokia sai kuitenkin empiiriset todisteet ketteryyden käytännöllisyydestä ja siirtymisen kannattavuudesta myös isommassa yrityksessä. Tämän tutkimuksen jälkeen Nokialla keskityttiin paljon enemmän itse ketterien kehitysmetodien käyttöönottoon niissä maissa ja kaupungeissa, joissa sitä ei oltu vielä omaksuttu ja pyrittiin selittämään, kuinka se käytännössä

tapahtuu. (Laanti et al., 2011) Käyttäjätestauksen ideana on selvittää heikot puolet uudesta ohjelmasta tai tässä tapauksessa metodista. Kun huonot puolet ja niiden syyt ovat selvinneet, on helppo lähteä parantamaan ja korjata puutteet. Nokian tapauksessakin virheistä opittiin ja korjattiin ne, mitkä suurin osa käyttäjistä totesi huonoiksi.

## 2.2 Metodien hyödyt eriteltynä ja yhdistettynä

Ketterät menetelmät ja käyttäjäkeskeinen suunnittelu ovat nousseet todella suosituiksi ja etenkin niiden yhdistelmä kuulostaa miltei utopistiselta sekä organisaation että käyttäjän näkökulmasta. Molemmissa metodeissa on omat hyvät puolensa, miksi ne on omaksuttu, mutta niiden yhdistelmä tarjoaa näiden lisäksi vielä jotain, mitään kumpikaan näistä metodeista ei voi yksin tarjota.

### 2.2.1 Ketterien menetelmien hyödyt

Nykyään on hyvin tavatonta, että joku ohjelmistokehitysyritys ei työskentele ketteränä. On siis oltava jokin syy, miksi tämä metodi valloitti koko ohjelmistokehitysmaailman ja tuli niin suosituksi. Schwaberin, Laganzan ja D'Silvan (2007) mukaan ketterät menetelmät muun muassa nostavat tuotteen laatua ja vähentävät resurssien tuhlausta. Lisäksi ne edesauttavat ennustettavuutta. Lycettin ja muiden (2003) mukaan ketterät menetelmät myös antavat organisaatiolle paremmat mahdollisuudet vastata tapahtuvaan muutokseen.

Yleisesti ketterät menetelmät parantavat yrityksen kilpailukykyä markkinaympäristössä, koska ne auttavat paremmin määrittämään projektin päämäärän ja varmistavat, että koko ajan pyritään samaa päämäärää kohti. (Gangjun et al., 2009) Jaksotuksensa myötä ketterät menetelmät myös antavat organisaatioille joustavuutta. Tämä mahdollistaa nopeamman ja paremman reagointivalmiuden muutokseen. (Losada et al., 2013) Ketterät menetelmät ovatkin enimmäkseen mukautuvaa suunnittelua. Terminä "ketterä" tarkoittaakin juuri sitä, että ymmärretään, mikä on päämäärä ja pystytään muutosten sattuessa muuttamaan suunnitelmia lyhyessä ajassa ja pienillä kustannuksilla. (Gangjun et al., 2009)

Menestyvät teolliset suunnittelumetodit vaativat monia tekijöitä, jotta ne pysyvät suosittuina (Gangjun et al., 2009). Ketterän kehityksen periaatteiden mukaan nopea reagoiminen muutokseen tuotannon jakamiselle pienempiin osiin ei ole sen ainoa hyvä puoli. Ketterät menetelmät pyrkivät olemaan ihmisläheisiä sillä, että ne haluavat miellyttää asiakasta, mutta myös motivoimaan kehittäjiä ja suunnittelijoita yksilöinä ja tukemaan heidän tarpeitaan. Ihmisläheisyyden lisäksi ketterät menetelmät pyrkivät parantamaan tuottavuutta useilla tavoilla. Olisi tarkoitus, että kehittäjät työstävät joka päivä projektia ja ohjelmistoa ja tekevät siitä niin hyvän kuin mahdollista, mutta pitävät sen yksinkertaisena. Lopuksi koko kehitystiimin tulisi tietää, kuinka tulla tehokkaammiksi ja työskennellä sen mukaisesti. (Beck et al., 2001)

Kaiken kaikkiaan ketteristä menetelmistä teki niin suosittu ja hyväksi havaittu menetelmän juuri sen mukautuvuus muutokseen, jaksottuminen ja ihmiskeskeisyys. Tarkoituksena on, että tuote valmistuu vähän kerrallaan jokaisen jakson aikana ja jakson lopulla tarkistetaan, että tämä osa toimii ja on sen mukainen kuin oli tarkoitus. (Abbas et al., 2008) Tällainen osissa kasaaminen jakaa työn selkeisiin komponentteihin, joiden

työstäminen on helppoa ja yksinkertaista ja antaa jokaiselle selkeän kuvan, mitä tiimin muut jäsenet ovat työstämässä.

## 2.2.2 Käyttäjäkeskeisen suunnittelun hyödyt

Käyttäjäkeskeisessä suunnittelussa käyttäjät ovat keskeisessä asemassa koko ohjelman suunnittelun, toteutuksen ja testauksen ajan. Tämän myötä tuotekehityksessä tullaan keskittymään enemmän tuotteen käytettävyyteen. Toisin sanoen osataan ottaa huomioon paitsi käyttäjän tarpeet, myös käyttäjän aiempi osaaminen sekä odotukset ohjelmalta ja luoda sen pohjalta toimiva käyttöliittymä. (Baek et al., 2008) Yleisesti käyttäjien mukaan ottamien tuotekehitykseen parantaa tuotteen suunnittelua, käyttöliittymää, toiminnallisuutta, käytettävyyttä ja laatua (Shas, 2007).

Käyttäjien ei kuitenkaan tarvitse olla jatkuvasti läsnä kehitysprosessissa. On nimittäin olemassa useita eri tasoja, joilla käyttäjät voi sisällyttää kehitykseen. Jokainen organisaatio voi löytää omalta kohdaltaan sen ratkaisun, mikä sopii parhaiten. Alimmalla tasolla kommunikointi käyttäjän kanssa tapahtuu usein epäsuorasti, esimerkiksi asiakkaan kautta, lisäksi tapaamissessioita on vähän ja ne ovat lyhyitä. Ylimmällä tasolla puolestaan kommunikointi on suoraa loppukäyttäjien kanssa ja tapaamissessioita on paljon ja ne ovat pitempiä. (Baek et al., 2008) Tämä suunnittelumalli antaa organisaatiolle oikeastaan täysin vapaat kädet toimia, kuten se parhaaksi sopii sekä organisaatiolle että loppukäyttäjälle. Harvoin kuitenkaan käytetään jompaakumpaa ääripäätä. Usein käyttäjät haluavat osallistua suurella panostuksella, mutta se ei aina ole mahdollista. Tästä syystä monesti käytetään useita eri vuorovaikutusvälineitä. Tällä tavalla tuetaan vuorovaikutusta ja käyttäjä on mukana kehityksessä. (Klann et al., 2006a) Tässä mallissa ei myöskään ole niin tiukkaan säädelyä, mitä tulee tehdä milloinkin, kuten ketterissä menetelmissä.

Se miten käyttäjäkeskeinen suunnittelu ehkä eniten rajoittaa organisaatiota on prototyyppien suuri määrä. Käyttäjäkeskeiseen suunnitteluun liittyy usein metodi, jossa pyritään nopeasti saamaan uusi prototyyppi ulos ja testaukseen. Tämä ei kuitenkaan ole huono asia. Yleisesti nopea prototyyppien valmistus on vain avuksi ihmisen ja tietokoneen välisessä vuorovaikutuksessa sekä ohjelmistosuunnittelussa. Ylipäätän tämän metodin tarkoitus on rakentaa nopeasti jotain konkreettista käyttäjälle nähtäväksi, jotta käyttäjä heti huomaa, onko suunnittelu menossa oikeaan suuntaan. (Baek et al., 2008)

Loppujen lopuksi menestyvä loppukäyttäjille suunnittelu on monen eri metodin, tekniikan ja työkalun yhdistelmä. Tällä tavalla ei-ammattilaiset voivat tehokkaasti osallistua tuotteen tai ohjelmiston luontiin, muokkaamiseen ja laajentamiseen ohjelmisto kehityksessä. (Klann et al., 2006a)

## 2.2.3 Yhdistämisen hyödyt

Paras asia, mitä käyttäjäkeskeisen suunnittelun sovittamien ketteriin menetelmin on tuonut, on että nyt käyttäjät oikeasti ovat läsnä koko suunnitteluprosessin ajan (Baek et al., 2008). Tapa, jolla ketterät menetelmät ja käyttäjäkeskeinen suunnittelu on yhdistetty, on hyvin pitkälle yrityskohtainen. Tämä johtuu siitä, että jokainen yritys haluaa panostaa eri suuruudella eri osa-alueisiin. Ketterät menetelmät on aina tarkoitettu olemaan ihmiskeskeinen, teollinen suunnitteluprosessi. Toki tässä "ihminen" kuitenkin tarkoittaa enemmän suunnittelijaa ja asiakasta kuin loppukäyttäjää. (Gangjun et al.,

2009) Yhdistämisellä on siis paljon hyviä puolia, mitä yksittäisellä metodilla ei niinkään ole. Ensinnäkin hyviä prototyyppejä syntyy hyvin nopeasti. Kun suunnitelmat ovat sellaiset, kuin käyttäjä toivoo, niiden pohjalta on helppo rakentaa prototyyppi, joka vastaa lopullista ohjelmaa joltain osa-alueelta. Toiseksi on helpompi toteuttaa käyttäjättestaus ja saada siitä heti enemmän tietoa irti. (Sy, 2007)

Vaikka ketterät menetelmät ovat myös tarkoitettu käyttäjä-/asiakaskeskeiseksi suunnittelumalliksi, siitä puuttuu muutama tärkeä asia, joita käyttäjakeskeinen suunnittelu painottaa. Ketterissä menetelmissä on jokaisen jakson lopuksi testausjakso, jossa testataan ohjelman toimivuutta ja vastaavuutta toivottuun. (Losada et al., 2013) Ei kuitenkaan vaadita, että loppukäyttäjät suorittaisivat nämä testit. On mahdollista, että yrityksen omat testaajat vain asettuvat käyttäjän rooliin ja suorittavat testit. (Patton 2002) Kun tähän metodiin lisätään käyttäjakeskeinen suunnittelu, on suotavaa, että vähintään asiakas on mukana testaustilanteessa tai saa sen tulokset. Ylipäätään on tärkeää saada jatkuvaa palautetta mielellään suoraan käyttäjältä. Tämä palaute varmistaa sen, että ohjelma vastaa mahdollisimman pitkälle sitä, mitä asiakas tai loppukäyttäjä haluaa tai tarvitsee. Kun tuote suunnitellaan vastaamaan mahdollisimman hyvin haluttua, se on suuri etu kaikille, erityisesti käyttäjälle ja asiakkaalle, mutta myös yritykselle. (Begier, 2010)

Tietenkään ainoat yhdistämisen hyödyt eivät ole organisaation tuottavuuteen tai kilpailukykyyn vaikuttavia. Käyttäjien liittäminen suunnitteluprosessiin kunnioittaa myös eettisiä ja sosiaalisia arvoja ja auttaa toteuttamaan asiakkaan toiveet tuotteen kannalta mahdollisimman hyvin. Tarve etiikalle ja sosiaalisuudelle tulevat väistämättä esiin, kun työskennellään ihmisten kanssa. Monesti nämä omaksutaan jo ihan tiedostamatta. Kaikilla on jokin käsitys tietyistä normeista, mitä esimerkiksi käyttäjätutkimuksessa saa kysyä tai tehdä. (Begier, 2010)

Vuonna 2014 Cajander ja muut osoittivat tutkimuksessaan, että niissä projekteissa, missä käytettiin sekä käyttäjakeskeistä suunnittelua että ketterää menetelmää sisälsivät vähiten virheitä. Positiivisia huomioita tuli myös siitä, että metodien yhdistäminen pienensi kuilua paljastaa käytettävyysongelmia ja helpotti niiden ongelmien korjaamista. Cajander ja muut (2014) myös osoittivat, että palautteen anto ei ole ainoa tapa kerätä käyttäjiltä tietoa tuotteesta. Käyttäjakeskeisen suunnittelun myötä pystyttiin seuraamaan ja tarkkailemaan oikeita loppukäyttäjiä käyttämässä laitteita ja ohjelmistoja ja kuulla heiltä suorat kommentit.

Käyttäjakeskeinen suunnittelu ja ketterät menetelmät yhdessä ovat mielenkiintoinen ja potentiaalinen yhdistelmä parantaa ohjelmistokehitystä ja käytettävyyttä yhtä aikaa. Tulevaisuudessa aiotaankin kiinnittää enemmän huomiota näiden yhdistämiseen ja räätälöidä ketteristä menetelmistä enemmän käyttäjakeskeisiä. Ajatuksena myös olisi oppia enemmän käyttäjien vaatimuksista ja suunnitella heille toimivat laitteet ja ohjelmistot ketteränä. (Cajander et al., 2014)

### 2.3 Metodien heikkoudet eriteltynä ja yhdistettynä

Kun lähdetään yhdistämään kahta eri suunnittelumetodia, ei voida välttyä haasteita. Varmasti suurin haaste on varmistaa, että koko henkilöstö on ymmärtänyt, kuinka valittu metodi tai niiden yhdistelmä toimii ja he osaavat hyödyntää sitä työssään. Tietysti tämä ymmärrys ei ole se ainoa haaste tai riski. Ohjelmistokehityksessä voi aina

tulla vastaan jotain yllättävää ja on mahdollista, että valittu metodi ei aina taivu uusiin käännteisiin.

### 2.3.1 Ketterien kehitysmetodien heikkoudet

Useissa yrityksissä etenkin ketterien kehitysmallien omaksuminen on osoittautunut haasteelliseksi. Varsinkin suuryrityksissä omaksuminen on erittäin hidasta. Yahoo!':n tapauksessa muutos laitettiin alulle vuonna 2005. Vuoteen 2008 mennessä 150 tiimiä oli jo sisäistänyt muutoksen, mutta silti jäljellä oli vielä useita tiimejä, jotka eivät olleet. (Benefield, 2008) On jopa väitetty, että omaksujat eivät todellisuudessa edes tiedä, mitä ketterät menetelmät todellisuudessa tarkoittavat (Laanti et al., 2011).

Omaksumisen haasteet eivät kuitenkaan ole ainoat syyt siihen, miksi ketterät menetelmät ovat haaste varsinkin suuryrityksille. Suurin osa suuryrityksistä työskentelee maailmanlaajuisesti ja tämä sotii ketterien menetelmien periaatteita vastaan, esimerkiksi usein maailmanlaajuinen kommunikaatio vaatii muodollisempaa kommunikointia, kuin ketterien menetelmien tarkoitus olisi. Ketterät menetelmät suosivat kasvotusten tapahtuvaa kommunikaatiota, mutta se on harvoin mahdollista, kun työskennellään eri maissa. Suurin haaste tässä onkin ketterien menetelmien soveltaminen. Ne on nimittäin luotu toimimaan parhaiten tiimin kesken, joka istuu samassa huoneessa. (Shrivastava, Rathod, 2015)

Lisäksi hyvin usein on todella haastavaa arvioida, mitä resursseja ja kuinka paljon niitä tähän muutokseen todella tarvitaan (Laanti et al., 2011). Päälimmäisiä vaadittavia resursseja ovat raha, aika ja henkilöstö. Rahaa kuluu, jos haluaa Yahoo!':n tavoin palkata ammattilaisia, mutta myös siksi, että osa arvokasta työaikaa kuluu metodin ymmärtämiseen ja opetteluun. Tällöin on hyvin mahdollista, että varsinkin pienemmässä yrityksessä tuottavuus mahdollisesti putoaa hetkellisesti. Koska opettelu vie aikaa, on hyvä varata tuo aika sille. Ei voi olettaa, että kaikki ymmärtävät asiat heti ja joillakin voi mennä paljonkin aikaa asian hyväksymiseen ja ymmärrykseen. Tästä syystä myös henkilöstö resurssit vaikuttavat suuresti. Ei vain se, kuinka paljon tai vähän henkilöstöä on, vaan myös se, kuinka suvaitsevaisia ja nopeasti oppivia he ovat. Lisäksi myös aiempi kokemus vaikuttaa huomattavasti.

Jotta ketterät menetelmät toimisivat, tarvitaan intensiivistä kommunikaatiota, koordinaatiota, yhteistyötä ja luottamusta tiimin jäsenten ja sidosryhmien välillä. (Shrivastava, Rathod, 2015) Jos yksikään näistä on heikko tai täysin olematon, se lisää automaattisesti riskiä epäonnistua koko ohjelmistokehityksessä.

### 2.3.2 Käyttäjakeskeisen suunnittelun heikkoudet

Käyttäjien ottaminen suunnitteluun ei kuitenkaan aina helpota ohjelman kehitysprosessia. Jopa loppukäyttäjien saaminen mukaan kehitysprosessiin voi olla ongelma, vaikka se olisikin heidän etujensa mukaista. Usein käyttäjillä on heidän oma täysipäiväinen työnsä, joten yhteisen ajan löytäminen voi olla haastavaa. (Baek et al., 2008) Kriittisimmät pisteet käyttäjakeskeisessä suunnittelussa ovatkin käyttäjien saatavuus, valmistautuminen, kouluttaminen, tukeminen ja yhteistyökyky. (Shas, 2007)

Yksi vaikeimmista haasteista onkin nimenomaan organisaation ja loppukäyttäjien tehokas yhteistyö suunnitteluprosessissa. Etenkin suurissa kaupallisissa projekteissa on suorastaan mahdotonta kuunnella kaikkia loppukäyttäjiä, jolloin kaikilla heillä on täysin erilaiset taustat ja lähtökohdat. Näissä tapauksissa tuleekin haasteeksi löytää käyttäjien testiryhmä, joka edustaa kaikkia loppukäyttäjiä. Tällöin käyttäjät ovat enemmänkin

mukana antamassa tietoa kuin mukana suunnittelussa. (Baek et al., 2008) Käyttäjäkeskeisessä suunnittelussa onkin ymmärrettävä, että yksi suunnittelu malli ei sovi kaikille yrityksille ja organisaatioille. Esimerkiksi lääketieteellisessä kehityksessä loppukäyttäjien ottaminen mukaan liian aikaisessa vaiheessa kehitystä voi olla hyvinkin haitallista kehityksen kannalta, koska käyttäjillä ei ole kylliksi tietoa asioista. (Shas, 2007)

Käyttäjien mukaan ottaminen lisää myös huomattavasti muutosten mahdollisuutta, kun ihmisten myötä mukaan tulee kaikki henkilökohtaiset muutokset. Näihin liittyy muun muassa terveysriskit, epäeettiset suhtautumiset ja teot sekä ikävät muutokset ihmissuhteissa. Eettisten ja sosiaalisten arvojen vaaliminen suunnittelussa ei ole kovinkaan monen organisaation päällimmäinen tarkoitus. Ohjelmistoyrityksissä, kuten muissakin yrityksissä, huolehditaan usein enemmän tuottavuudesta. Tämä voi olla haitaksi käyttäjäkeskeisyyden kannalta. (Begier, 2010)

On myös tärkeä muistaa, että vaikka käyttäjät huomioidaan ja otetaan mukaan suunnitteluun, he eivät ole suunnittelijoita, eivätkä ohjelmistoalan ammattilaisia. Tästä syystä käyttäjät eivät aina edes tiedä, mitä tarkalleen ottaen he edes haluavat. Tietysti käyttäjien mielipiteet ovat tärkeitä, mutta ne eivät välttämättä ole aina oikeita. Onkin siis suunnittelijoiden vastuulla ymmärtää, mitä käyttäjä oikeasti haluaa ja tarvitsee ja ohjata käyttäjää tähän suuntaan. (Sugar, 2001)

Suurin haaste käyttäjäkeskeisessä suunnittelussa varmasti on se, että ei käytetä oikeita loppukäyttäjiä. Toteutus voidaan tehdä Pattonin (2002) tapaan, jossa vain yritetään asettua käyttäjän rooliin tai sitten käytetään asiakkaita. Suuressa kaupallisessa projektissa tämä on miltei välttämätöntä, mutta on vaikea yrittää suunnitella käyttäjille, jos ei kuulla yhtään käyttäjää. Tällöin riskinä on, että kehittäjä ja asiakas eivät oikeasti ole ymmärtäneet käyttäjää oikein ja toteutus ei ole lainkaan haluttu.

### 2.3.3 Yhdistämisen heikkoudet

Ketterät menetelmät ja käyttäjäkeskeinen suunnittelu ovat molemmat täysin oma suunnittelumetodinsa, jotka on tarkoitettu toimimaan itsenäisesti. On tietysti selvää, että nämä kaksi eivät sulaudu yhdeksi ilman ongelmia. Ketterien menetelmien jaksotus kuitenkin omalta osaltaan häiritsee käyttäjäkeskeistä suunnittelua. Jaksotuksen myötä käyttäjätestaukselle ja tulosten analyysille jää huomattavasti vähemmän aikaa kuin olisi tarkoitus ja tämä tarkoittaa, että osa vaadittavista ominaisuuksista saattaa jäädä kokonaan huomaamatta tai testaamatta. (Fox et al., 2008)

Aika on ongelma myös muullakin kuin käyttäjätestauksen saralla. Sy (2007) totesi, että vaikka ketterä käyttäjäkeskeinen suunnittelu olikin paljon nopeampaa ja tehokkaampaa, kuin he olivat odottaneet, jaksotuksen takia suunnitteluosaston tuli tehdä vain pieniä suunnitelmia jaksoittain. Jaksojen aikaväli oli aivan liian suppea tekemään kokonaissuunnitelmia, joiden tekeminen olisi vaatinut huomattavasti kauemmin aikaa. Tästä syystä suunnitteleminen on pakko jakaa pienempiin osiin.

Siinä missä ketterä kehitys häiritsee käyttäjäkeskeistä suunnittelua, myös käyttäjäkeskeinen suunnittelu häiritsee ketteriä menetelmiä. Käyttäjäkeskeisessä suunnittelussa ollaan kuitenkin tekemisissä ihmisten kanssa ja ihmiset ovat usein oikukkaita olentoja. On hyvin mahdollista, että tulevat käyttäjät eivät täysin ymmärrä, millainen uudesta tuotteesta on tulossa, kunnes on liian myöhäistä tai kehittäjä ja suunnittelijat eivät ymmärrä, että ohjelma ei ole lainkaan sellainen kuin käyttäjä halusi. (Begier, 2010) Vastaava tapahtui muun muassa Yahoo!n tapauksessa, kun yrityksen



työntekijät luulivat käyttävänsä ketteriä menetelmiä työnteossaan vaikka todellisuudessa, eivät olleet kunnolla ymmärtäneet sen pääkohtia. (Benefield, 2008)

Vaikka puhutaankin ketterien menetelmien ja käyttäjakeskeisen suunnittelun yhdistämisestä, se ei kuitenkaan ratkaise yhtä suurinta ohjelmistoalan ongelmista: kuilu kehittäjiä eli koodaajien ja käyttäjien välillä on edelleen suuri. Vaikka nämä menetelmät saataisiin sulautettua täydellisesti toisiinsa, on mahdollista, että kehittäjät eivät oikeasti koskaan tapaa käyttäjiä. (Begier, 2010) Tämän näkee selvästi myös Syn (2007) mallissa. Siinäkin käyttäjät ja asiakkaat asioivat vain suunnittelijoiden kanssa ja suunnittelijat toimittajat tiedot asiakkailtaan kehittäjille.

Lisäksi haasteena tulevat aina vastaan resurssit. Tässä tapauksessa tärkeimmät esille tulevat resurssit ovat raha, aika, työkalut ja tila. Huolimatta siitä, kuinka suuresta tai pienestä organisaatiosta on kyse, lähes aina tulee jokin näistä resursseista vastaan haasteena. (Baek et al., 2008) Kuitenkin suurin osa kaikista uhkista ja haasteista, mitä näiden metodien yhdistyessä ilmenee, voidaan pienentää tai poistaa kokonaan, jos ne todella otetaan huomioon edes jollain tasolla kehitysvaiheessa. (Begier, 2010)

## 2.4 Tulevaisuuden näkymiä

Maailma muuttuu jatkuvasti ja sekä ihmisten että organisaatioiden on pysyttävä siinä mukana. Kuten Lycett ja muut (2003) totesivat, että ketterät menetelmät on luotu sitä varten, että ne pystyvät hyvin reagoimaan muutokseen. Tämä kuitenkin tarkoittaa vain ohjelmiston kehityksessä tapahtuviin muutoksiin. Shrivastava ja Rathod (2015) osoittivat tutkimuksissaan, että ketterät menetelmät eivät välttämättä niin hyvin sovellukaan nykyisiin suuryrityksiin, jotka ovat jakautuneet ympäri maailmaa. Myös käyttäjakeskeinen suunnittelu kärsii monikulttuurisesta ympäristöstä, erilaiset kulttuurilliset ja kielelliset erot vaikeuttavat tuotekehitystä. Tämä voi tulevaisuudessa osoittautua suureksi ongelmaksi, koska maailma globalisoituu koko ajan enemmän ja enemmän.

Vuonna 2015 on juuri pyritty löytämään näitä erilaisia riskejä ja riskitekijöitä ketterien kehitysmetodien käytössä sekä tutkimaan ja kategorioimaan niitä. Näin on mahdollista löytää erilaisia tekniikoita riskien hallintaan ja oppia huomioimaan niitä. Lähitulevaisuudessa pyritään identifioimaan kaikkein kriittisimmät riskit ja vastaamaan niihin. (Shrivastava, Rathod, 2015) Tällä tavalla pyritään edelleen kehittämään ketteriä menetelmiä.

Loppukäyttäjien tärkeys ohjelmistokehityksessä vain jatkaa kasvamistaan ja siksi siihenkin pitää yrittää tulevaisuudessa panostaa. On kuitenkin mahdotonta täysin valmistautua tulevaisuuteen tällä saralla, koska käyttäjien tarpeet muuttuvat koko ajan. Siksi onkin tärkeää löytää oikeat prosessit ja organisatoriset rakenteet toteuttaa alati muuttuvaa loppukäyttäjähitystä. Vuonna 2006 ryhdyttiin kehittämään tätä ajatusta ja tutkijoiden tavoitteena oli, että vuonna 2020 sopeutumiskyky olisi noussut koko ohjelmistokehityksen uudeksi tilaksi. Ajatuksena olisi, että vuonna 2020 suurimmalla osalla ihmisistä olisi jonkinlaiset taidot käyttäjakeskeiseen suunnitteluun, ei vain ohjelmistosuunnittelijoilla, tällöin käyttäjakeskeinen suunnittelu olisi tullut osaksi jokaisen jokapäiväistä elämää, taustasta ja ammatista riippumatta. Tämä tarkoittaa sitä, että jokainen omalta osaltaan osallistuisi jollain tavalla uusien tuotteiden ja ohjelmistojen kehitykseen ja tuottamiseen, että loppukäyttäjä ei olisi enää näkymätön. (Klann et al., 2006b)

Tapa, jolla pyritään kehittämään suunnittelumetodeja suuntaan, että ne palvelisivat kaikkein parhaiten sekä kehittäjiä että käyttäjiä, on tehdä tutkimusta aiheesta (Klann et al., 2006b). Täytyy kyetä havaitsemaan riskit ja tutkia erilaisia ratkaisua niiden vähentämiseksi ja jopa poistamiseksi.

### 3. Pohdinta

Kun ketteriä menetelmiä kehitettiin, ajatuksena oli, että ne tekisivät ohjelmistosuunnittelusta huomattavasti tehokkaampaa ja asiakaskeskeisempää. Kuitenkaan pelkkä asiakaskeskeisyys ei enää riittänyt, kun loppukäyttäjät alkoivat käyttää enemmän erilaisia laitteita ja ohjelmistoja maailman teknologisoitumisen myötä. Ohjelmistokehitystä onkin paranneltu sillä, että asiakaskeskeisyys on muutettu käyttäjakeskeiseksi.

#### 3.1 Ketterien menetelmien ja käyttäjakeskeisen suunnittelun kehitys ja yhdistyminen

Organisaatioille ja etenkin kaupallisille yrityksille hyvän kilpailukyvyn saavutus ja sen jatkuva parantaminen on aina ollut erityisen tärkeää. Ketterät menetelmät parantavat yrityksen kilpailukykyä, koska ne jäsentävät työskentelyä ja tekevät siitä tehokkaampaa. Tehokkuus johtuu siitä, että kaikki tiimin jäsenet tietävät, mitä muut ovat tekemässä, lisäksi joka aamuisissa palavereissa on helppo kertoa, jos joku asia ei luonnistu tai tarvitsee apua.

Ketterien menetelmien omaksuminen vaatii yritykseltä sitä enemmän aikaa ja vaivaa mitä suurempi yritys on kyseessä. Etenkin, koska suuret yritykset usein ovat jo menestyneet omalla alallaan, heillä on jo valmiiksi toimivat menetelmät ja työkalut työnsä tukemiseksi. Tästä syystä suuret yritykset eivät ehkä aluksi edes halunneet omaksua ketteriä menetelmiä. Kuitenkin, kun pienet ja keskisuuret yritykset alkoivat menestyä ja jopa kilpailla samoilla markkinoilla suurten yritysten kanssa, se herätti suuret yritykset pohtimaan, mitä pienemmät tekivät eritavalla.

Kuten aiemmin todettiin, kaikki ketterien menetelmien ideat eivät ole uusia. Nokian tapauksessakin yrityksellä oli jo valmiiksi käytössä jotain elementtejä ketteristä menetelmistä. Suurimmat muutokset olivatkin joka aamuiset, seisten pidetyt, lyhyet palaverit ja tuotannon jakaminen pienempiin jaksoihin. Tämä auttoi hieman suurempia yrityksiä ketterien menetelmien omaksumisessa. Nykyään miltei kaikki yritykset käyttävät ketteriä menetelmiä.

Ketterät menetelmät sopivat todella hyvin suuriin kaupallisiin projekteihin. Ne auttavat tyydyttämään asiakkaan tarpeita ja samalla edesauttavat laadukkaamman työtuloksen saavuttamista pienemmillä kustannuksilla. Suurissa kaupallisissa projekteissa on vaikea lähteä etsimään loppukäyttäjiä ja 2000-luvun alussa varmaan ajateltiin, ettei se ole niin tarpeellista, koska asiakkaat ovat läsnä. Kuitenkin nykyään kaiken ikäiset ihmiset käyttävät Internetiä, omistavat älypuhelimien ja on muutenkin ympäröity monenlaisella teknologialla. Näiden kaikkien käyttäjien teknologinen osaaminen on erilainen ja se tulee ottaa huomioon myös suunnittelussa.

Käyttäjakeskeinen suunnittelu on helppoa, kun pystytään nimeämään kaikki loppukäyttäjät. He voivat esimerkiksi olla asiakasyrityksen työntekijöitä tai kyseessä on tietylle käyttäjäryhmälle suunniteltu tuote, esimerkiksi kirurgeille. Vaikeaksi tämä metodi tulee siinä vaiheessa, kun suunnitellaan suuria kaupallisia projekteja, esimerkiksi

Internet-sivuja. Tällöin jokaisen painikkeen paikka, väritys ja jopa kieli saavat täysin uuden merkityksen. On todella haastavaa ottaa useat erilaiset ihmisryhmät huomioon suunnittelussa ja myös löytää sopivat testihenkilöt tästä joukosta käyttäjätestausta varten.

Kun käyttäjäkeskeinen suunnittelu ja ketterät menetelmät saadaan sovitettua yhteen, tuloksena todella on ihmiskeskeinen metodi, joka samalla parantaa tuotteen laatua ja tehokkuutta. Kuitenkin tässä yhdistämisessä on olemassa heikkouksia, joihin kannattaa kiinnittää huomiota tulevaisuudessa.

### 3.2 Ketterien menetelmien ja käyttäjäkeskeisen suunnittelun hyödyt ja haasteet

Ketterät menetelmät eivät nykyään yksinään riitä, kun halutaan luoda toimivia ohjelmistoja. Kun ihmiset käyttävät teknologiaa joka päivä, he myös paremmin tiedostavat sen virheet. Ennen saattoi olla yksi laite, jossa oli yksi ohjelma ja sen huono käytettävyyttä oli vain osa sitä. Nykyään ihmisillä on varaa valita laitteensa ja ohjelmansa itse ja monet ovat hyvinkin tarkkoja tämän suhteen.

Yleisen käsityksen mukaan asiakas ja etenkin käyttäjä ovat aina oikeassa ja etenkin käyttäjäkeskeisessä suunnittelussa loppukäyttäjien kuuleminen on erittäin tärkeää. Vuonna 2002 Patton totesi, että vaikka tekisi kaiken juuri niin kuin käyttäjä haluaisi, se ei tarkoita, että lopullinen tuote olisi käytettävyydeltään paras mahdollinen. Kun käyttäjät tiedostavat paremmin, mitä haluavat, suunnittelijoiden täytyy ottaa se huomioon. Paras tapa tähän on ottaa pelkkien asiakkaiden rinnalle oikeat loppukäyttäjät suunnittelemaan ja testaamaan tuotteita ja antamaan palautetta. Täytyy kuitenkin muistaa, että nämä loppukäyttäjät eivät ole ammattilaisia, eivätkä välttämättä edes tiedä, mitä haluta. Tämä johtuu siitä, että he eivät välttämättä tiedä kaikkia mahdollisuuksia, mitä teknologia tai ohjelmistokehitys voi tarjota. Kun oikeasti halutaan parantaa ohjelmaa käyttäjien palautteen vaikutuksesta, on tärkeä olla tarpeeksi paljon palautteen antajia. Mitä enemmän palautteen antajia on, sitä varmemmin saadaan oikea kuva siitä, mitä suurin osa käyttäjistä haluaa, eikä tulos perustu vain muutaman ihmisen mielipyyksiin. Lisäksi henkilökohtaiset mielipiteet luonteenpiirteet vaikuttavat hyvin raskaasti ihmisten kanssa työskennellessä. Ihmisillä myös on tapana muuttaa mieltään asioiden suhteen. Joustavuutensa takia ketterät menetelmät kuitenkin sallivat sen ja pystyvät hyvin reagoimaan siihen.

Käyttäjäkeskeinen suunnittelu metodina perustuu siihen, että kaikki tehdään rauhassa ja käydään käyttäjän kanssa tarkkaan läpi, mitä tuotteeseen halutaan. Ensin suunnitellaan kokonaiskuva ja sitten vähitellen yksityiskohtaisemmin ominaisuus ominaisuudelta. Ketterät menetelmät eivät kuitenkaan täysin tue tätä. Totta kai kaikki riippuu jaksojen pituudesta ja siitä, kuinka usein ja kuinka pitkiä jaksoja kerrallaan käyttäjät ja suunnittelijat voivat tavata. Ylipäätään jotta tämä yhdistelmä toimisi, käyttäjäkeskeisen suunnittelun on sopeuduttava ketterien menetelmien aikatauluun ja tapaan työskennellä. Tämä tarkoittaa muun muassa jaksotusta. Metodien yhdistelmässä suunnittelu on myös jaksotettava pienempiin osiin ja on mahdollista, että kokonaissuunnitelmaa ei ehditä tekemään ensimmäisen jakson aikana.

Myös maailman kansainvälistyminen tuo omat haasteensa molempiin metodeihin. Ketterät menetelmät eivät välttämättä toimi niin tehokkaasti kuin olisi tarkoitus, jos tiimit on hajautettu ympäri maailmaa. On myös mahdollista, että yksi tiimi voi koostua

erimaiden kansalaisista. Koska ketterät menetelmät vaativat intensiivistä kommunikointia tässä voi tulla vastaan myös muun muassa kommunikaatio-ongelmat, esimerkiksi kommunikaatiovälineillä on suuri merkitys kommunikaation sujumiseen. Toiseksi kommunikaatiokieli asettaa omat haasteensa. Ohjelmistoalalla puhutaan lähes poikkeuksetta englantia, mutta se ei ole kaikkien työntekijöiden äidinkieli ja siksi voi tulla väärinkäsityksiä. Monikulttuurisuus vaikuttaa myös käyttäjakeskeisyyteen. Kun tehdään käyttäjätestausta, minkä maalaiset testaavat tuotteen? Jos tuote on tarkoitettu kansainvälisille markkinoille, pitää ottaa huomioon sen eri kielivaihtoehdot ja ulkoasu, mutta myös se, ovatko esimerkiksi kiinalaisten käyttäjätottumukset samaa tasoa suomalaisten kanssa.

Ketterien menetelmien ja käyttäjakeskeisen suunnittelun yhdistelmää kyllä voi hyvinkin kuvata ihmisläheiseksi. Siinä huolehditaan käyttäjän tarpeista ja katsotaan, että haluttu tuote vastaa mahdollisimman hyvin tätä, mutta tämä helpottaa myös kehittäjiä. Kehittäjät tietävät pystyvät työskentelemään tehokkaammin, mutta silti inhimillisesti saaden tiiminsä jäseniltä sekä omaa tilaa että tukea sitä tarvitessa. Kun käyttäjät ja kehittäjät ovat tyytyväisiä, myös asiakas on tyytyväinen. Asiakas saa tuotteen nopeasti valmiiksi ja käyttöön ja se vastaa haluttua mahdollisimman hyvin. Voisi siis sanoa, että tämä yhdistelmä on luotu palvelemaan kaikkia sidosryhmiä mahdollisimman hyvin. Siinä missä ketterät menetelmät oli luotu palvelemaan asiakkaiden ja kehittäjien tarpeita, käyttäjakeskeinen suunnittelu tuo sen lähemmäs tuotteen loppukäyttäjiä.

### 3.3 Mitä tulevaisuudessa kannattaa ottaa huomioon?

Ketterät menetelmät ovat valloittaneet yritysmaailman hyvin pitkälle ja ovatkin tulleet lähinnä jäädäkseen. Parhaimpiakin ideoita täytyy tietysti kehittää, jotta ne sopivat paremmin ympäröivään maailmaan. Siksi ketteriin menetelmiin onkin keksitty lisätä käyttäjakeskeisyyttä ja tavalla luoda uusi voittoisa suunnittelu metodi.

Klann ja muut (2006) puhuivat, että vuoteen 2020 mennessä kaikki käyttäjät olisivat käyttäjätestaajia. Tavallaan sen näkee jo nykyään; yhä useamman palvelun lopussa pyydetään arvostelemaan äskettäin käytettyä ohjelmaa tai palvelua. Se kuitenkin vaikuttaa aika utopistiselta, että kaikki käyttäjät kykenisivät käyttäjakeskeiseen suunnitteluun. Maailmassa on paljon vanhoja ihmisiä, jotka ovat eläneet suurimman osan elämästään ilman tietokoneita ja suuren luokan ohjelmistoja ja on hyvin mahdollista, että nämä henkilöt ovat pudonneet kehityksen matkasta. Toisaalta, jokainen ihminen osaa varmasti ilmaista mielipiteensä, kuinka miellyttävä jotain ohjelmaa oli käyttää. Tässä kuitenkin kohtaamme taas ihmisten henkilökohtaiset mieltymykset.

Käyttäjakeskeistä suunnittelua ja ketteriä menetelmiä yhdessä ei ole tutkittu vielä kovin paljon. Muutenkin tämä yhdistelmä tuntuu olevan vielä kovin nuori. Se toimii hyvin tietyissä tapauksissa, mutta siinä löytyy haasteita. Kaikkein paras ratkaisu onkin nyt keskittyä noiden haasteiden ratkaisemiseen ja tutkia aihetta enemmän.

## 4. Yhteenvedo

Ketterät menetelmät tulivat tunnetuksi viimeistään vuonna 2001, kun julkaistiin Ketterän ohjelmistokehityksen julistus. Tällöin ketterät menetelmät olivat uusin ja hienoin metodi toteuttaa ohjelmistokehitystä ja ne omaksuttiinkin yrityksissä ja organisaatioissa todella innokkaasti. Myöhemmin tuli kuitenkin tarve metodille, joka huomioisi käyttäjiä paremmin. Tällöin syntyi idea yhdistää ketterät menetelmät ja käyttäjakeskeinen suunnittelu.

Koska käyttäjakeskeisen suunnittelun ja ketterien menetelmien yhdistäminen on kohtalaisen tuore aihe, sen toteuttamiseen ei löydy selkeitä ohjeita ja yritykset ovat soveltaneet tätä hyvin erilaisin keinoin. Kaikille on kuitenkin tyypillistä, että työskennellään ketterien menetelmien periaatteiden mukaan, mutta otetaan loppukäyttäjät mukaan jo suunnitteluvaiheesta alkaen. Lisäksi toinen tärkeä osa on, että palaute itse tuotteesta kerätään myös loppukäyttäjiltä.

Valitettavasti molemmat metodit joutuvat hieman sopeutumaan, jotta yhdistelmä toimisi. Käyttäjakeskeisessä suunnittelussa joudutaan jaksottamaan suunnittelu ja ketterissä menetelmissä ottamaan huomioon käyttäjien tietämättömyyden ja inhimillisyyden.

Palatakseni tutkimusongelmiini:

1. Miten nämä kaksi erilaista työskentelytapaa on sulautettu toisiinsa ja kuinka ne toimivat yhdessä?

Nämä metodit toimivat lähinnä ketterien menetelmien ehdoilla ja niiden periaatteita mukailen, kuitenkin omaksuen käyttäjakeskeisen suunnittelun parhaat puolet. Pienemmissä projekteissa, joissa loppukäyttäjät on helppo nimetä, tämä toimii todella hyvin. Suurissa kaupallisissa projekteissa käyttäjät ovat lähinnä vain antavat tietoa, mutta eivät kunnolla voi osallistua suunnitteluun.

2. Mitä hyötyjä ja haasteita puolia kummastakin löytyy sekä yksin että yhdessä?

Ketterät menetelmät on luotu yrityksiä varten. Ne tuovat yrityksille tehokkuutta ja parempaa kilpailukykyä samalla, kun parantavat tuotteen laatua ja laskevat kustannuksia. Ne myös antavat yrityksille paremman mahdollisuuden reagoida muutokseen ja tekevät ohjelmistokehityksestä ihmiskeskeisempää. Käyttäjakeskeisessä suunnittelussa tuote vastaa paremmin asiakkaan toiveita ja lisäksi parantaa tuotteen suunnittelua, käyttöliittymää, toiminnallisuutta, käytettävyyttä ja laatua. Näiden yhdistelmässä yhdistyy molempien toiminta mallien hyödyt. Näiden metodien yhdistelmä tarjoaa yritykselle mahdollisuutta työskennellä tehokkaasti ja luoda laadukkaampia tuotteita, jotka vastaavat käyttäjän tarpeita. Käyttäjille tämä taas tarjoaa paljon suurempaa tyytyväisyyttä ja helppokäyttöisyyttä itse tuotteeseen.

Haasteena ketterissä menetelmissä on, että ne vaativat intensiivistä kommunikaatiota sekä tiimin sisällä että tiimien kesken. Tämä ei välttämättä

onnistu täydellisesti kaikissa yritysmalleissa. Käyttäjakeskeisessä suunnittelussa puolestaan haasteena ovat itse käyttäjät. Käyttäjät eivät ole ammattilaisia, eivätkä välttävät ole siksi tietoisia, mitä ohjelmaan voidaan toteuttaa ja mitä ei. Myös kansainvälisyys sekä yrityksen sisällä että käyttäjien keskuudessa tuo omat haasteensa. Yhdistäessä metodeiden omien haasteiden lisäksi suurimmat haasteet tulevat siinä, kun täytyy sovittaa erilaiset aikataulut toisiinsa. Ketterissä menetelmissä ei ole totuttu tekemään laajoja suunnitelmia heti aluksi ja käyttäjakeskeisessä suunnittelussa ei ole ennen tarvinnut työskennellä jaksojen mukaan.

Tulevaisuudessa olisi hyvä keskittyä enemmän tutkimaan ja poistamaan heikkouksia näiden suunnittelumallien yhdistelmästä ja luomaan sujuva, uusi metodi, joka rakentuisi kätevästi vanhojen päälle. Tällöin uuden toimintatavan omaksuminen olisi kaiken kokoisille yrityksille helppoa ja nopeaa.

## Lähdeluettelo

Abbas N, Gravell A M, Wills G B (2008) Historical Roots of Agile Methods: Where did “Agile Thinking” Come from?, *Agile Processes in Software Engineering and Extreme Programming, Lecture Notes in Business Information Processing Volume 9*, pp 94-103

Baek E, Cagiltay K, Boling E, Frick T (2008) *Handbook of Research for Educational Communications and Technology, User-Centered Design and Development*, 659-670

Barjis J (2008) The importance of business process modeling in software systems design, *Science of Computer Programming* 71 (2008) 73–87

Barnum C (2002) What is usability and what is usability testing, *Usability Testing and Research*, Longman, New York, NY USA, pp. 7-30.

Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin R, Mellor S, Schwaber K, Sutherland J & Thomas D (2001) *Manifesto for Agile Software Development*, Manifesto for Agile Software Development, Saatavilla: <http://agilemanifesto.org/> (9.12.2015)

Benefield G (2008) Rolling Out Agile in a Large Enterprise, *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*

Begier B (2010) Users' involvement may help respect social and ethical values and improve software quality. *Inf Syst Front* 12(4): 389-397.

Cajander Å, Gulliksen J & Lárusdóttir M (2014) Informal feedback rather than performance measurements – user-centred evaluation in Scrum projects, *Behaviour & Information Technology* 33(11)

de Ruyter B, van de Sluis R (2006) Challenges for End-User Development in Intelligent Environments. Lieberman, Paternó, Wulf (Eds.) *End User Development. Human-Computer Interaction series Volume 9*. (p. 243-250) Dordrecht, The Netherlands, Springer

Dingsøy T, Dybå T, Moe N, (2010) *Agile Software Development: Current Research and Future Directions*, Springer

Dingsøy T, Dybå T (2009) What Do We Know about Agile Software Development?, *The IEEE Computer Society*, Volume:26 , Issue: 5, p. 6-9

Dingsøy T, Nerur S, Balijepally V, Moe N B (2012) A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, Volume 85, Issue 6, June 2012, Pages 1213–1221



Fox, D., Sillito, J., & Maurer, F. (2008) Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry. Agile, 2008. AGILE '08. Conference. : 63-72.

Gangjun Yang, Guoding Chen, Suihuai Yu & Jianjie Chu (2009) The method of agile industrial design. Computer-Aided Industrial Design & Conceptual Design, 2009. CAID & CD 2009. IEEE 10th International Conference on. : 53-56.

Klann M, Lieberman H, Paternó F, Wulf V (2006a) End-User Development: An Emerging Paradigm. Lieberman, Paternó, Wulf (Eds.) End User Development. Human-Computer Interaction series Volume 9. (p. 1-8) Dordrecht, The Netherlands, Springer

Klann M, Lieberman H, Paternó F, Wulf V (2006b) Future Perspectives in End-User Development. Lieberman, Paternó, Wulf (Eds.) End User Development. Human-Computer Interaction series Volume 9. (p. 475-486) Dordrecht, The Netherlands, Springer

Laanti M, Salo O & Abrahamsson P (2011) Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. Information and Software Technology 53(3): 276-290

Losada B, Urretavizcaya M & Fernández-Castro I (2013) A guide to agile development of interactive software with a “User Objectives”-driven methodology. Science of Computer Programming. 78 (11): 2268-2281

Lycett M, Macredie R, Patel C, Paul R, (2003) Migrating agile methods to standardized development practice, Computer 36, 79–85.

Mao J-Y, Vredenburg K, Smith P W, and Carey T. (2005) The state of user-centered design practice, Communications of the ACM, Volume 48, 3/2005, p. 105-109.

Patton J (2002) Hitting the Target: Adding Interaction Design to Agile Software Development, Proceedings of OOPSLA , pp. 1-7.

Schwaber K, Laganza G & D’Silva D, (2007) The Truth about Agile Processes: Frank Answers to Frequently Asked Questions, Forrester Report

Shas S (2007) Benefits of and barriers to involving users in medical device technology development and evaluation. International Journal of Technology Assessment in Health Care. Volume: 23, Issue: 1 Pages: 131- 137

Shrivastava S & Rathod U (2015) Categorization of risk factors for distributed agile projects. Information and Software Technology 58(2): 373-387

Sugar W (2001). What is a good about user-centered design? Documenting the effect of usability sessions on novice software designers. J. Res. Comput. Educ., 3(3), 235–250.

Sy D (2007) Adapting Usability Investigations for Agile User-centered Design. Journals of Usability studies 2 (3): 112-132

# Liite1: Tutkimussuunnitelma: Käyttäjäkeskeisen suunnittelun sovittaminen ketteriin menetelmiin

## Johdanto

Nykyään yhä useammassa yrityksessä on hyvin tavallista, että halutaan tehdä töitä ketteränä. Toinen asia, mihin halutaan kiinnittää huomiota, on itse loppukäyttäjien huomioiminen jo suunnittelu- ja työstövaiheessa (Laanti, Salo, Abrahamsson, 2011). Yleensä, kun lähdetään suunnittelemaan ketteränä, käyttäjäkeskeisyys jää vähemmälle huomiolle. Tästä syystä haluan seurata, kuinka näiden kahden yhteen sovittaminen on yleensä toteutettu ja kuinka tämä on kehittynyt.

## Tutkimusongelma ja tutkimusmenetelmät

Tutkimusongelmanani aion analysoida, kuinka nämä kaksi yleensä sovitetaan yhteen, miten ne tulisi sovittaa yhteen ja mitä hyötyä tästä on kummallekin osapuolelle. Ajattelin myös kiinnittää huomiota, miten tämä on kehittynyt vuosien aikana vertaamalla vanhempia ja uudempia lähteitä ja niiden sisältöä. Kyseessä on siis lähteisiin perustuva kirjallisuuskatsaus, jossa ei ole tarkoitus tehdä omaa tutkimusta.

## Rajoitukset

Rajaan tutkielmani siihen, että tarkistelen nimenomaan käyttäjäkeskeisen suunnittelun ja ketterien menetelmien yhteen sovittamista yleisellä tasolla. Tarkoitukseni ei ole esitellä kaikkia ketteriä menetelmiä tai verrata niitä. En myöskään aio eritellä, millaisia erilaisia käyttäjäkeskeisen suunnittelun tyyppisiä on olemassa.

## Aiempi tutkimus

Aihe on hyvin käytännönläheinen, joten tieteellistä tutkimusta on tehty hyvin vähän. Useimmat lähteet ovat konferenssijulkaisuja tai opetuskäyttöön tarkoitettuja kirjoja. Tästä voisi päätellä, että ollaan pyrkimässä sitä kohti, että nämä käyttäjäkeskeisyys saataisiin saumattomaksi osaksi ketteriä menetelmiä.

Ketterät menetelmät esiteltiin 1990-luvun puolella välissä ja siitä alkaen yritykset ovat olleet kiinnostuneita niistä. Nykyään suuri osa yrityksistä onkin ketteriä (Laanti, Salo, Abrahamsson, 2011). Vaikka ketterät menetelmät ja käyttäjäkeskeinen suunnittelu ovat toiminnaltaan erilaisia, niissä esiintyy silti samankaltaisuutta. Ketterissä menetelmissä pyritään tekemään toimiva ohjelmisto, joka sitten toimitetaan asiakkaalle, mutta käyttäjäkeskeisessä suunnittelussa käyttäjät pyritään pitämään mukana alusta asti (Fox, Maurer, Sillito, 2008). Näiden kahden pohjalta on tullut termi ketterä suunnittelu, joka on enemmänkin joustavaa suunnittelua kuin ennustavaa suunnittelua (Gangjun, Guoding, Suihuai, Jianjie, 2009). On huomattu, että yksi tärkeimpiä strategisia tapoja vaikuttaa ohjelmiston laatuun on juuri loppukäyttäjien kuunteleminen ja jatkuvan palautteen kerääminen. On siis hyvin luonnollista, että käyttäjät liitetään ketteriin menetelmiin (Begier, 2010).

**Lähteet**

Begier B (2010) Users' involvement may help respect social and ethical values and improve software quality. *Inf Syst Front* 12(4): 389-397.

Fox, D., Sillito, J., & Maurer, F. (2008) Agile Methods and User-Centered Design: How These Two Methodologies are Being Successfully Integrated in Industry. *Agile, 2008. AGILE '08. Conference.* : 63-72.

Gangjun Yang, Guoding Chen, Suihuai Yu & Jianjie Chu (2009) The method of agile industrial design. *Computer-Aided Industrial Design & Conceptual Design, 2009. CAID & CD 2009. IEEE 10th International Conference on.* : 53-56.

Laanti M, Salo O & Abrahamsson P (2011) Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology* 53(3): 276-290