

Lappeenrannan teknillinen yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma

Kandidaatintyö

Eetu Heimala

**KETTERÄN OHJELMISTOKEHITTÄMISEN HYÖDYNTÄMINEN
KÄYTTÖLIITTYMÄKEHITYKSESSÄ ASIAKKAIDEN
TODELLISEN TARPEEN YMMÄRTÄMISEKSI**

Työn tarkastaja(t): Tutkijatohtori Ari Happonen

Työn ohjaaja(t): Tutkijatohtori Ari Happonen

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
School of Business and Management
Tietotekniikan koulutusohjelma

Eetu Heimala

Ketterän ohjelmistokehittämisen hyödyntäminen käyttöliittymäkehityksessä asiakkaiden todellisen tarpeen ymmärtämiseksi

Kandidaatintyö

2016

32 sivua, 3 kuvaa, 1 liite

Työn tarkastajat: Tutkijatohtori Ari Happonen

Hakusanat: käyttöliittymä, prototyyppi, ketterä kehitys, toimintatutkimus

Työn tavoitteena oli toimintatutkimuksen kautta tutkia ketterän ohjelmistokehityksen keinoin toteutetun käyttöliittymäkehityksen kykyä vastata asiakkaiden todellisiin tarpeisiin. Työssä haettiin tapaustutkimusyritykselle olemassa olevan työkalun uuden version käyttöliittymän toteutusvaihtoehtoja ja toteutettiin korkean tarkkuuden prototyyppiä näitä hyödyntäen. Ketterän ohjelmistokehityksen arvot ja periaatteet soveltuivat kehitysprosessissa käytettäviksi erinomaisesti. Iteratiivinen lähestymistapa kehitykseen ja läheinen yhteistyö tapaustutkimusyrityksen ja kandidaatintyöntekijän kanssa mahdollistivat yrityksen odotusten täyttämisen. Työkalun käyttöliittymä saatiin tasolle, joka mahdollistaa jatkokehittämisen aloituksen. Kattavamman testauttamisen sisällyttäminen kehitysprosessiin olisi edesauttanut vielä paremman lopputuloksen saavuttamista.

ABSTRACT

Lappeenranta University of Technology
School of Business and Management
Degree Program in Computer Science

Eetu Heimala

Utilizing Agile Software Development in user interface development to understand the real needs of clients

Bachelor's Thesis

2016

32 pages, 3 figures, 1 appendix

Examiners: D.Sc. (Tech.) Ari Happonen

Keywords: user interface, prototyping, agile development, action research

The purpose of this project was to study how “Agile Software Development”-driven user interface development can answer to the real needs of clients. In this project different implementation possibilities for a new version of an existing software were researched and built into high-fidelity prototypes. Agile values and principles complemented the development process excellently. Iterative approach to development and close collaboration between the case-study company and the Bachelor's thesis worker made the fulfillment of the company's expectations possible. The software's user interface was brought to a level that allowed the start of further development. Including a more comprehensive user-testing approach would've aided in reaching an even better result.

ALKUSANAT

Tämä opinnäytetyö on tehty tutkimusyhteistyössä Tekes rahoitteen Digikaappaus tutkimushankkeen osallistujayrityksen ja Lappeenrannan teknillisen yliopiston kanssa keväällä 2016. Tahdon kiittää ohjaajaani Ari Haposta hyvin läheisestä ja aktiivisesta tuesta työn aikana, sekä erittäin innokkaasti työn toteutuksessa mukana ollutta yritystä.

SISÄLLYSLUETTELO

1 JOHDANTO	2
1.1 TUTKIMUKSEN TAUSTA	2
1.2 TAVOITTEET JA RAJAUKSET	3
1.3 TYÖN RAKENNE	4
2 TUTKIMUSMENETELMÄN TARKKA ESITTELY	5
2.1 TOIMINTATUTKIMUS	5
2.2 TOIMINTATUTKIMUKSEN VAIHEET JA TAVOITE	5
2.3 TOIMINTATUTKIMUKSEN SOVITTAMINEN KANDIDAATINTYÖHÖN	7
3 KETTERÄ OHJELMISTOKEHITYS	9
3.1 KETTERÄN OHJELMISTOKEHITYKSEN NELJÄ ARVOA	9
3.2 KETTERÄN OHJELMISTOKEHITYKSEN MERKITYS TÄLLE KANDIDAATINTYÖLLE.....	11
4 KÄYTTÖLIITTYMÄKEHITYS	13
4.1 PROTOTYPOINTI	13
4.2 PROTOTYYPPIEN HYÖDYNTÄMINEN KEHITYKSESSÄ.....	14
5 KEHITYSPROSESSI	15
5.1 KEHITYSPROSESSIN VALMISTELU	15
5.2 YHDEN KEHITYSJAKSON VAIHEET	16
5.3 KEHITYSTYÖN PÄÄTTYMINEN	19
6 TUTKIMUSTULOKSET	21
6.1 KEHITYSPROSESSISSA KOETUT HAASTEET.....	21
6.2 KEHITYSPROSESSIN ONNISTUMISEN ARVIOINTI	22
6.3 KEHITYSLISTA JATKOA VARTEN.....	24
7 JOHTOPÄÄTÖKSET.....	25
LÄHTEET.....	26
LIITTEET.....	28
LIITE 1. KETTERÄN OHJELMISTOKEHITYKSEN JULISTUKSEN TAKANA OLEVAT PERIAATTEET	28

1 JOHDANTO

Tämä kandidaatintyö on Lappeenrannan teknillisen yliopiston tietotekniikan koulutusohjelmaan tehty käytännönläheinen tutkimus. Tutkimuksessa tutustuttiin yrityksen toimialaan ja haettiin tapaustutkimusyrietykselle ratkaisuja siihen kuinka asiakasrajapintaan tarjottavan verkkopalvelun käytettävyys ja ulkoasu saadaan vastaamaan digitalisoituneen maailman palvelutasotarpeita. Tutkimus ja kehitystyö suoritettiin ketterän ohjelmistokehityksen keinoin.

1.1 Tutkimuksen tausta

Ketterät ohjelmistokehitysmenetelmät (Agile) ja -prosessit ovat kehitysmalleja, joiden tarkoituksena on yleensä vastata todellisen loppukäyttäjän tarpeisiin hakemalla jatkuvaa palautetta loppukäyttäjiltä ja vastaamalla näihin tarpeisiin joustavasti suoraan kehitystyön aikana (Stellman & Greene 2014). Ketterät ohjelmistokehitysmenetelmät luovat joustavan menettelytavan kehitystyölle, koska muutosten ja kehityksen määrä ei ole koskaan suuri kehitysvaiheiden välissä. Epäedulliset kehityssuunnat eivät käytä turhaan resursseja, koska huonon kehityssuunnan valinta mahdollistaa oletettujen ja todellisten tarpeiden välisten erojen ymmärtämisen.

Tapaustutkimusyritys on suomalainen yritys, joka haluaa tehdä esivaiheen tutkimusta elinkaarikustannusten laskentatyökalun verkkorajapintaan tarjottavaa versiota varten. Ohjelmistokehitys on aikaa ja rahaa vaativa prosessi, joten yrityksen asiakkaiden tarpeet halutaan kartoittaa mahdollisimman hyvin etukäteen. Kattavalla alustavalla tutkimuksella voidaan minimoida todellisen ohjelmistoprojektin kustannuksia. Tapaustutkimusyrietyksen tarpeeseen vastataan toteuttamalla erilaisia käyttöliittymäprototyyppejä ja kehittämällä näitä ketterän ohjelmistokehityksen keinoin. Käyttöliittymäprototyyppejä luovutetaan tapaustutkimusyrietyksen omille asiakkaille testattavaksi, jolloin yrityksen omien toiveiden lisäksi saadaan selville todellisten käyttäjien tarpeita ja toiveita. Tästä prosessista saadaan käytännönläheistä tutkimusmateriaalia, jota voidaan verrata jo olemassa olevaan teoriamateriaaliin. Vertailun tuloksia tulkitsemalla voidaan tuoda esiin eroja ja yhtäläisyyksiä yleisen teorian, ja tutkimustapauksessa toteutuvan käytännön välillä.

Valitsin tämän kandidaatintyöaiheen, koska opintojeni tuloksena mielenkiintoni eri ohjelmoinnin osa-alueisiin on kasvanut huomattavasti. Työn valitsemisen taustalla on siis vahvimpana ajavana voimana halu oppia uutta, ja kandidaatintyön yhteydessä pääsen vielä soveltamaan opittua käytännössä.

1.2 Tavoitteet ja rajaukset

Kandidaatintyön tavoitteena on selvittää, miten ketterien ohjelmistokehitysprosessien periaatteet soveltuvat käyttöliittymäsuunnitteluun ja prototypointiin olemassa olevan sovelluksen nykyaikastamiseen digitalisaation mukanaan tuomien asiakastarpeiden mukaiseksi. Tutkimus suoritetaan yhdessä tapaustutkimusyrityksen kanssa toimintatutkimuksena. Tutkimuksessa testataan ketterän prototypoinnin soveltuvuutta suomalaisen pk-yrityksen sovelluskehitystarpeisiin kehittämällä tapaustutkimusyritykselle asiakasrajapintaan tarjottavan verkkopalvelun käyttöliittymäprototyyppejä. Kehitystyössä hyödynnetään yrityksen ja yrityksen asiakkaiden toiveita sekä kehitysehdotuksia ja edetään niiden mukaisesti. Vaikka tutkimuskohteena on itse kehitysprosessi, työssä käsitellään myös käyttöliittymäsuunnitteluun liittyviä periaatteita ja ohjeita. Tutkimuksen tuloksien tulkinta vaatii myös kehitysprosessin kehityskohteen onnistumisen käsittelyä eli prosessin tuloksena syntyvien käyttöliittymäprototyyppien tulkintaa. Lopputulosten pääasiallinen analysointi tapahtuu vertaamalla käytännön kehitystyöstä saatua informaatiota teoriamateriaaleihin.

Tutkimuksessa toteutetut käyttöliittymäprototyypit ovat jossain määrin toiminnallisia, eli prototyypit eivät ole staattisia kuvia, vaan manipuloitavissa olevia käyttöliittymäkokonaisuuksia. Tiettyjä käyttöliittymän osia klikkaamalla saa siis aikaan muutoksen näkyvässä, esimerkiksi välilehtivalikko on toimiva, ja jokaisessa on oma sisältö. Työkalulla on hyvin spesifinen loppukäyttäjärühmä, eli tämän käyttäjärühmän voidaan olettaa haluavan tiettyjä asioita käyttöliittymältä. Tämän pohjalta voidaan tehdä jonkin verran oletuksia kehitystä ohjaamaan, ennen kuin kunnollista palautetta on kerätty.

1.3 Työn rakenne

Seuraavassa luvussa esitellään kandidaatintyön tutkimusmenetelmän eli toimintatutkimuksen teoriataustaa yleisesti, sekä sovitetaan kyseinen teoria tämän kandidaatintyön käytännön osuuteen sopivaksi. Sovituksen kautta selviää millä tavoin toteutetun kehitysprosessin soveltuvuutta käyttöliittymäkehitykseen tutkitaan tässä tapauksessa. Sen jälkeen käydään läpi ketterän ohjelmistokehityksen pääperiaatteita ja ideologiaa, joihin tässä tutkimuksessa käytetty kehitysmenetelmä perustuu. Kehitysmenetelmän esittelyn jälkeen tutustumme käyttöliittymäkehityksen olennaisimpiin osa-alueisiin opinnäytetyön kannalta.

Viimeiset kolme lukua käsittelevät itse tehtyä työtä. Ensimmäiseksi viidennessä luvussa käydään läpi kehitysprosessia edeltäviä toimenpiteitä ja prosessin, sekä työskentelytapojen suunnittelua. Tästä siirrytään prosessin etenemisen seuraamisen kautta sen päättymisvaiheeseen. Kehitysprosessin kokonaisuuden läpikäymisen jälkeen esitellään löydettyjä huomioita, kehitysprosessin onnistumista ja mahdollisia parannuskohtia. Viimeisessä luvussa esitellään koko tutkimuksen tulokset tiivistetysti johtopäätösten muodossa.

2 TUTKIMUSMENETELMÄN TARKKA ESITTELY

Tutkimusmenetelmänä kandidaatintyössä käytettiin toimintatutkimusta. Aluksi käydään läpi mitä toimintatutkimus on yleisesti, jonka jälkeen se sovitetaan kandidaatintyön puitteisiin sopivaksi. Esille tuodaan myös työn edellyttämiä rajoituksia tässä sovituksessa.

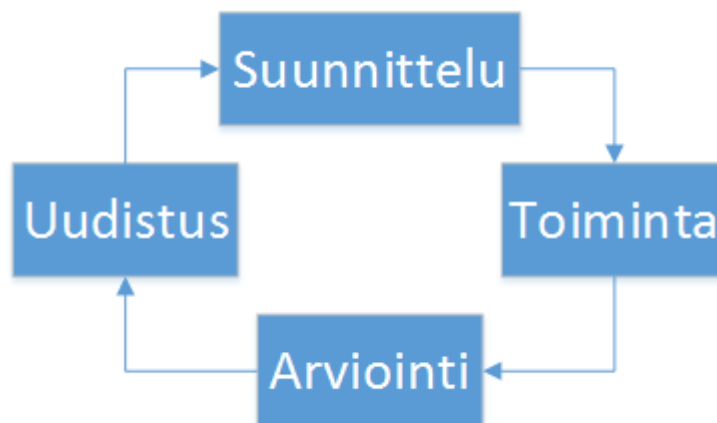
2.1 Toimintatutkimus

Toimintatutkimus on käytännönläheinen keino tutkia ja parantaa yksilöiden toimintaa. Tutkimuksen kohde voi olla yksittäisen henkilön toiminta, myös tutkijan oma, tai esimerkiksi jonkin prosessin osana olevien henkilöiden toiminta, kuten opetus tai kehitystyö. Pääasiallisesti toimintatutkimuksen suorittaa joku, joka on tavallisestikin osallisena tutkimuksen kohteessa, esimerkiksi opettaja joka tutkii toimintatutkimuksen keinoin omaa opettamistaan. Tällä tavoin tutkimus on tavoitteellista, koska tutkija hyötyy tutkimuksesta suoraan itse (Sagor 2000). Toimintatutkimusta hyödynnetäänkin usein juuri opetuksessa, mutta sen voi yhtäläillä sovittaa muihinkin ympäristöihin, kuten jonkin yrityksen tai organisaation osiin tai muihin tavoitteellisiin aktiviteetteihin. (Riel 2016)

Toimintatutkimus on usein jatkuva syklimäinen prosessi, jonka tavoitteena on saada aikaan jatkuvaa kehitystä (Riel 2016, Rigsby 2005). Toimintatutkimuksella kerätty data on yleensä kvalitatiivista, laadullista dataa, koska tutkimus perustuu tekemällä kerättyyn ymmärrykseen tapahtuvasta toiminnasta, eikä tätä tietoa yleensä voida, tai kannata esittää numeerisessa muodossa. Kvalitatiivisen datan alkuperä on juuri tällaista kenttätyössä saatua, yleensä sanallisessa muodossa esitettyä ja tulkittua (Patton 2014 s. 13-16).

2.2 Toimintatutkimuksen vaiheet ja tavoite

Eri lähteissä tarkka vaiheiden lukumäärä vaihtelee 3-7 päävaiheen välillä (Rigsby 2005, Riel 2016, Sagor 2000), mutta kaikki mallit kuitenkin pitävät sisällään saman prosessin ja päävaiheiden laaja määrä on saatettu vain tiivistää toisen vaiheen sisään. Toimintatutkimus neljävaiheisena prosessina on esitelty kuvassa 1. Tavallisesti prosessi alkaa suunnitteluvaiheesta, jossa arvioidaan nykytilanne ja kartoitetaan kehitysvaihtoehtoja.



Kuva 1. Toimintatutkimus prosessina

Tutkittava toiminta kannattaa valita siten, että odotettu hyöty olisi mahdollisimman suuri, ja useiden iteraatioiden kautta voidaan tarkastella tutkimuksen kohdetta tarkemmin. Kun soveltuva aihe on valittu, tulee suunnitella miten sen soveltuvuutta voidaan kokeilla käytännössä. Suunnitelman pohjalla tulee olla jonkinlainen teoriatausta, tietysti mitä kattavampi sitä parempi, jotta voidaan tehdä edes jonkinlaisia oletuksia tutkimuksen tuloksesta. Pohjamateriaalin tutkimisella ei tulisi olla tässä vaiheessa pääpainoa, vaan tarkoitus on löytää vastauksia sellaisiin tutkimuskysymyksiin joihin ei ole mahdollista löytää suoraa vastausta kirjallisuudesta. (Sagor 2000) Suunnitteluvaiheeseen kuuluu myös itse tutkimuskysymysten asettaminen. Tutkimuskysymysten tarkoitus on ohjata tutkimusta oikeaan suuntaan ja kohdistaa tutkimusmenetelmät ja tulkinta siten, että kysymys sisältäisi ehdotetun toiminnan ja odotetun lopputuloksen. Tällä tavoin kysymys ei ole liian laaja ja todennäköistä on, että tulosten tulkintojen määräkin on pienempi. (Riel 2016)

Toinen vaihe toimintatutkimuksessa on itse toiminta. Kun kokeiltavalle toiminnalle on tehty tarpeeksi tarkka suunnitelma, toteutetaan suunnitelma käytännössä ja kerätään tutkimusdataa toiminnasta. Datan tarpeeksi tarkka kerääminen on olennaista, jotta tutkimustuloksia voidaan myöhemmin tulkita, ja päätyä tulkinnan tuloksena perusteltuihin johtopäätöksiin. Optimaalisessa tilanteessa dataa kerää samanlaisesta toiminnasta myös toinen tai jopa useampi henkilö, jotta tuloksia voidaan tulkita kattavammin ja luotettavammin. Datan keräämisessä tulee pyrkiä tasapainoon sen määrällisen kattavuuden ja hyödyllisyyden välillä. Datan paljous on siis lähtökohtaisesti hyvä asia, kunhan kerätty data on oleellista ja merkitsevää tutkimuksen kannalta. (Sagor 2000)

Arviointivaiheessa tulkitaan kerättyä dataa ja yritetään löytää selityksiä tapahtuneille ilmiöille. Olennaista tässä vaiheessa on jakaa löydettyjä tuloksia kanssatutkijoiden kanssa, jos mahdollista ja selvittää vastaavatko tulokset myös muiden tutkijoiden löytämiä havaintoja. Toimintatutkimusta voidaan soveltaa hyvin erilaisiin ja jopa henkilökohtaisiin tilanteisiin Tarkoituksena ei ole siis sovittaa jotain yleisen tason teoreettista mallia omaan tapaukseensa sellaisenaan. Tutkimuksen perusteella halutaan löytää nimenomaan juuri omaan tilanteeseen sopivia käytäntöjä, joten on täysin odotettavissa, että jokin olemassa oleva käytäntö tai malli ei vastaa tätä lainkaan. Arvioinnissa tarkoituksena onkin selittää miksi todettu ilmiö sopii tutkimustilanteeseen. Tässä vaiheessa yhteistyön hyödyt ovat suurimmillaan, koska se mahdollistaa omien löydösten soveltuvuuden arvioinnin muiden tutkimukseen. (Sagor 2000, Riel 2016)

Viimeinen vaihe toimintatutkimuksessa on uudistuksen tekeminen. Kun kerätty data on arvioitu ja tuloksille on esitetty syitä, voidaan uudistaa toimintatutkimuksen kohdetta. Kun uudelle löydetylle tiedolle on löydetty soveltuva käyttökohde, voidaan se lisätä toimintatutkimuksen kohteena olleeseen kokonaisuuteen. Tämän jälkeen tutkimussykli alkaa taas alusta suunnitteluvaiheella, jossa voidaan kohdistaa tutkimus joko tarkemmin aiempaan tai vaihtoehtoisesti valita uusi painopiste. (Sagor 2000)

Toimintatutkimuksen tarkoitus on ensisijaisesti kehittää tutkimuksen kohteena olevaa prosessia tai käytäntöä. Tämän kehittämisen myötä saavutettu kehitys ja osaaminen voidaan laajentaa sen jälkeen paikalliseen ympäristöön, esimerkiksi kollegoihin tai yhteisön muihin jäseniin. Tällä tavalla voidaan luoda suuria muutoksia jo verrattain laajalla alalla. Viimeisellä, laajimmalla tasolla, tutkija pyrkii luomaan tutkimuksensa perusteella yleistettyjä kokonaisuuksia ja selityksiä joita voitaisiin käyttää koko alalla, eikä pelkästään esimerkiksi omassa yrityksessä tai organisaatiossa. (Sagor 2000, Riel 2016)

2.3 Toimintatutkimuksen sovittaminen kandidaatintyöhön

Kandidaatintyössä kehitetään käyttöliittymäprototyyppiä tapaustutkimusyritykselle. Kandidaatintyössä tehtävä toimintatutkimus kohdistuu tähän käyttöliittymäprototyyppöinnin prosessiin ja sen koostaviin osiin ja ihmisiin. Kehitysprosessi aloitettiin helmikuun alussa ja sen suunniteltiin alustavasti kestävän heinäkuun loppuun saakka. Tästä aikavälistä

muodostuu yksi toimintatutkimuksen sykli, jonka kestoksi tulee noin kolme kuukautta. Useamman täyden prosessin sisällyttäminen kolmen kuukauden ajalle, jossa varsinaisia tapaamisia on alustavasti arvioiden alle kymmenen kappaletta, ei ole tarpeeksi kattava katsaus toiminnan tutkimiseen syventävästi. Tästä syystä toimintatutkimus aloitetaan hieman muutetusta suunnitteluvaiheesta, koska prosessi ja toimiala eivät ole vielä käytännössä koettuja. Suunnitteluvaiheen tarkoituksena on vain luoda toteutuskelpoinen malli jota noudattaa ja tutkia. Suunnitelman valmistuessa siirrytään toimintavaiheeseen eli itse käyttöliittymäkehitykseen.

Toiminnan aikana kerätään kanssakäymisten ja prosessin osien toimivuudesta mahdollisimman kattavasti dataa, jotta soveltuvuutta voidaan mahdollisimman perustellusti arvioida jälkeenpäin. Arviointivaiheessa reflektoidaan kehitysprosessin aikana koettua ja kirjataan ylös prosessin eri osien koettu onnistuneisuus, hyvät ja huonot puolet, sekä pohditaan mahdollisia kehityskohtia. Nämä seikat voidaan koostaa alustavaksi kehityslistaksi, jonka pohjalta voidaan suunnitella muutoksia tehdyn prosessin seuraavaan vaiheeseen tai toiseen samankaltaiseen kehitysprosessiin. Kandidaatintyön puitteissa näitä muutoksia ei kuitenkaan toteuteta, mutta ne tuodaan tapaustutkimusyrityksen tietoon.

3 KETTERÄ OHJELMISTOKEHITYS

Ketterän ohjelmistokehityksen varsinainen alkuperä on vuonna 2001 pidetyssä tapaamisessa, jossa yhteensä 17 eri taustoista tulevaa ohjelmistokehittäjää pohtivat nykyisten kehitysmallien puutteita. Beck et al. päätyivät yhteisymmärrykseen neljästä noudatettavasta arvosta, joihin ketterät ohjelmistokehitysmenetelmät perustuvat. Vastaavia prosesseja ja ajatuksia on ollut olemassa jo 1990-luvulta alkaen, mutta ne konkretisoituivat kun arvot liitettiin termiin Agile (Agile Software Development) ja ne koottiin yhteen julkaisuun nimeltä ”Agile Manifesto”. (Beck et al. 2001)

3.1 Ketterän ohjelmistokehityksen neljä arvoa

”Agile Manifesto”, eli ketterän ohjelmistokehityksen julistus sisältää neljä tärkeintä arvoa jotka ovat ketterän ohjelmistokehityksen perusta:

- **Yksilöitä ja kanssakäymistä** enemmän kuin menetelmiä ja työkaluja
- **Toimivaa ohjelmistoa** enemmän kuin kattavaa dokumentaatiota
- **Asiakasyhteistyötä** enemmän kuin sopimusneuvotteluja
- **Vastaamista muutokseen** enemmän kuin pitäytymistä suunnitelmassa

(Beck et al. 2001)

Mainitut neljä arvoa ovat julistuksen korostettu osa ja julistuksen tarkoitus onkin selventää, ettei oikealla mainittuja asioita pidä jättää kokonaan huomiotta, vaan vasemman puolen arvoja pitää vain korostaa enemmän. (Beck et al. 2001)

Ambler (2002) esittelee ensimmäisen arvon käytännön esimerkkinä, kahden erilaisen kehitystiimin vertailuna. Ensimmäisessä tiimissä on kokeneita kehittäjiä, jotka työskentelevät luontevasti yhdessä, mutta noudattavat omia menetelmiään. Toisessa tiimissä on keskinkertaisia kehittäjiä, joilla on käytössään hienostuneita menetelmiä ja työkaluja. Ajatuksena on, että hienointakaan prosessia ei päästä hyödyntämään täysin jos prosessin osana olevat ihmiset eivät toimi luontevasti yhdessä. Prosessien ja työkalujen merkitystä ei

tietenkään ole tarkoitus vähätellä, mutta kanssakäymiselle annetaan vain painoarvoa. (Ambler 2002, s. 6)

Julistuksen toinen kohta käsittelee toimivan ohjelmiston tärkeyttä. Amblerin (2002) mukaan suurin osa ohjelmistoyritysten asiakkaista haluaisi rahalleen mieluummin vastineeksi toimivan ohjelman kuin kattavan määrityksen siitä miten sen tulee toimia. Stellman ja Greene (2014) selittävät tämän myös samalla tavalla: toimiva ohjelma tuo arvoa enemmän kuin kattava määrittely siitä. Dokumentointi on edelleen tärkeää, mutta liian spesifinen dokumentointi etukäteen saattaa olla aivan turhaa työtä, jos kehitettyä osaa joudutaan muuttamaan tai sitä ei käytetäkään (Stellman & Greene 2014, s. 34-35).

Ocambin (2013) mukaan ketterien kehitysmenetelmien kolmas arvo on yleensä suurin este sopimusten solmimiselle. Perusteena on ajatus asiakkaiden pyrkimyksestä siirtää vastuu suurimmaksi osaksi toteuttavalle osapuolelle, ja mahdollisimman tarkalla sopimuksella voidaan yrittää turvata oma tarve. Ongelmat kehityksessä syntyvätkin siinä vaiheessa kun asiakas haluaakin muuttaa jotain jälkeinpäin, mutta tarkasti määritellyt sopimusehdot estävät sen. Ocambin mukaan kolmas kohta korostaa yleistä ajattelutapaa siitä, että ketterä kehitys olisi pelkästään julistuksessa korostettuja arvoja, eikä sisältäisi ollenkaan niiden oikeanpuoleisia pareja. Tarpeeksi joustavalla sopimuksella on tarkoitus mahdollistaa lopullisen ohjelmiston mahdollisimman hyvä vastaavuus asiakkaan todellisten tarpeiden kanssa. (Ambler 2002, s. 7; Ocamb 2013)

Viimeinen peruseriaate on jatkoa kolmannelle arvolle. On kovin yleistä, että suunnitelmat muuttuvat omien intressien vaihtumisen johdosta tai jonkin ulkopuolisen, oman vaikutusalueen saavuttamattomissa olevan seikan takia. Myöskin oma ymmärrys projektista sen etenemisen myötä voi aikaansaada halun muuttaa asioita. Amblerin (2002) mukaan merkittävästi muuttuneessa tilanteessa vanhasta suunnitelmasta voi tulla kokonaan merkityksetön, jos sen tuottama tulos ei enää vastaakaan todelliseen tarpeeseen. Tästä syystä suunnitelman tulee olla kattava, mutta samanaikaisesti sen tulee olla muutettavissa. Muutoksen esteenä voi olla myös sisäinen elementti, sillä Stellmanin ja Greenen (2014) mukaan muutosta saattaa vastustaa esimerkiksi projektipäällikkö joka joutuisi laatimaan pahimmassa tapauksessa kokonaan uuden projektisuunnitelman eikä halua tehdä niin paljoa työtä.

3.2 Ketterän ohjelmistokehityksen merkitys tälle kandidaatintyölle

Ketterän ohjelmistokehityksen periaatteisiin ja arvoihin perustuvia kehitysprosesseja on useita, kuten SCRUM tai XP (eXtreme Programming), mutta kaikki näillä yhteinen piirre on se, että mallit on tarkoitettu toteutettavaksi yrityksissä, joissa prosessimallien useat eri roolit pystytään täyttämään, ja kehitysprojektit ovat lähtökohtaisesti paljon laajempia. Kandidaatintyön puitteissa ei voida toteuttaa eikä sitoutua näin laajasti määriteltyihin malleihin jo sen takia, että toteutuksesta vastaa loppukädessä vain yksi henkilö, kandidaatintyöntekijä. Tapaustutkimusyritys toimii projektissa ensikädessä asiakkaana, mutta projektiin osallistuvat yrityksen työntekijät täyttävät myös muita kehittävän osapuolen rooleja, kuten graafisen suunnittelijan. Opinnäytetyön puitteissa tehtävä kehitystyö on ketterä vain arvoiltaan ja periaatteiltaan, eikä noudata kokonaisuudessaan mitään olemassa olevaa yleisessä käytössä olevaa mallia. Bristow (2014) toteaaakin julkaisemassaan artikkelissa, ettei kehitysprosessin tarvitse noudattaa mitään valmista ketterää mallia, vaan jo arvojen ja periaatteiden noudattaminen riittää. Stelman ja Greene (2014) puoltavat myös samaa ajatusta kertoessaan projektiryhmästä joka sisällytti toimintaansa vain muutamia ketterän kehityksen periaatteita, muttei uudistanut prosessiaan kokonaan. Periaatteiden sisällyttämisen jälkeen kehittäjät kokivat saavuttaneensa parempia tuloksia aikaisempiin projekteihinsa verrattuna (Stelman & Greene 2014, s. 22-26). Luvussa 5 esiteltävän kehitysprosessin voidaan katsoa sisältävän kaikki ketterän ohjelmistokehityksen arvot sekä lähes kaikki sen periaatteet. Tästä syystä prosessia voidaan kuvailla siis ketteräksi.

Ketterän ohjelmistokehityksen mallien keskeisin ominaispiirre on inkrementaalinen ja iteratiivinen kehitystapa. Inkrementaalinen kehitys tarkoittaa osissa tapahtuvaa kehitystä, eikä ohjelmistoa toteuteta yhdellä kertaa määrittelyn pohjalta. Iteratiivinen kehitys taas tarkoittaa, että ohjelmiston kehitettyjä kohtia kehitetään vielä eteenpäin ensimmäisen version jälkeen. Tämä kehitystapa kuvastaa parhaiten ketterän ohjelmistokehityksen toista ja neljättä arvoa: toimivan ohjelmiston toimittamista asiakkaalle ja muutokseen vastaamista. Perinteisissä kehitysmalleissa, kuten vesiputousmallissa, suunnitelman pohjalta toteutetaan yksi suuri kokonaisuus yhdellä kertaa. Tällöin vaarana on resurssien käyttäminen sellaisiin kehityskohtiin mitä halutaan jälkeenpäin muuttaa tai ei halutakaan edes sisällyttää lopputuotteeseen. Mikolukin (2013) mukaan tämä on juurikin yksi ketterän ohjelmistokehityksen hyödyistä, koska ketteryys ohjelmistokehityksessä kiteytyy paljolti

tämän virheen estämiseen, eli vaiheittaisella kehityksellä voidaan jatkuvasti korjata kehityssuuntaa asiakkaan kannalta paremmaksi, etenkin kun haluttu lopullinen kehityssuunta ei ole täysin asiakkaan tiedossa. (Shore & Warden 2007, s. 323-325; Stellman & Greene 2014, s. 246-247)

Kandidaatintyössä tehtävä kehitystyö on rajattu kohdistumaan pelkästään kehitettävän työkalun käyttöliittymään. Käyttöliittymää pyritään kehittämään sellaisilla menetelmillä, joita voitaisiin mahdollisesti hyödyntää myös työkalun lopullisessa, toiminnallisuudet omaavassa versiossa. Kehitysvaiheet kohdistuvat yhdellä kerralla pieneen osaan käyttöliittymää ja tehdyt muutokset ovat verrattain pieniä. Kehitys on siis iteratiivista että inkrementaalista, eli työkalun käyttöliittymää rakennetaan ja kehitetään osa kerrallaan. Käyttöliittymän kehitystyö on suunnittelun ja prototypoinnin lisäksi myös mahdollisten toteutustapojen kartoittamista käytännön toteuttamisen kautta.

4 KÄYTTÖLIITTYMÄKEHITYS

Käyttöliittymäkehityksen sisältämät vaiheet ja osat voidaan jakaa useilla eri tavoilla moniin eri osiin. Tälle kandidaatintyölle olennaisimpia seikkoja käyttöliittymäkehityksen osalta ovat prototypoinnin selittäminen sekä prototyypin käytön hyöty kehityksen apuna.

4.1 Prototypointi

Prototyypit mahdollistavat kehitysideoiden ja konseptien konkretisoimisen. Siinä missä suunnitelmat antavat vain mielikuvan toiminnasta, prototyypit mahdollistavat todellisen toiminnan kokemisen (Cao et al. 2015, s.10). Prototyyppejä voidaan jakaa erilaisiin ryhmiin niiden visuaalisuuden tarkkuuden ja toiminnallisuuden tason perusteella. Matalan visuaalisuuden ja toiminnallisuuden omaava prototyyppi voi olla esimerkiksi paperille piirretty luonnos, kun taas toiminnallinen ja visualisuudeltaan tarkka prototyyppi voi olla lähes lopullisen toteutustavan mukainen ohjelmoitu kokonaisuus tai sen osa. (Cao et al. 2015, s. 32-36)

Tässä kandidaatintyössä pyritään prototypoimaan suoraan lopullisella toteutusmenetelmällä, eli ohjelmoimaan HTML5-kielellä verkkorajapintaan tarjottavan työkalun käyttöliittymää. Prototyypin taso alkaa ensin matalasta visuaalisuudesta ja vähäisestä toiminnallisuudesta, mutta useiden kehitysjaksojen myötä lähestyy lopullista muotoaan niin visuaalisuutensa kuin toiminnallisuutensa puolesta. Koska kandidaatintyössä ei toteuteta muuta kuin käyttöliittymäkehitystä, eikä työkalu siis sisällä muuta kuin käyttöliittymän visuaalisia toiminnallisuuksia, voidaan puhua koko ajan lopullisen työkalun prototyypistä.

Kehitysprosessin aikana tuotettujen prototyypin osat ovat hyvin vaihtelevin menetelmin toteutettuja. Pyrkimys on aina toteuttaa käyttöliittymän osia mahdollisimman teknisellä tavalla, jotta ne palvelisivat jatkokehityksen helpottamista, mutta vaihtoehtoisia toteutustapoja on käytetty pääasiassa kahdesta syystä. Ensinnäkin käyttöliittymän ulkoasuun liittyvästä valinnasta ei välttämättä olla täysin varmoja, joten on mahdollisesti turhaa työtä toteuttaa kehityskohta vaikeammalla tavalla. Tarkoituksenmukaisempaa on mallintaa

lopullista ulkonäköä helpommalla menetelmällä, ja jos ulkonäön katsotaan olevan hyvä, toteuttaa se vasta sen jälkeen kunnollisilla menetelmillä. (Warfel 2009, s. 85-88) Toinen syy mallintavien tapojen käyttöön tässä tapauksessa oli resurssitehokkuus. Menetelmän toteuttaminen tietyillä työkaluilla saattoi vaatia kohtuuttoman määrän työkalun käytön opettelua, jolloin tehty työmäärä olisi myös suhteellisen suuri verrattuna saavutettuun arvoon. Myöhemmällä ajalla, sekä mahdollisesti toisen kehityskohdan toteuttamisen myötä hankitulla osaamisella aiempaan kohtaan pystyttiin palaamaan ja kehittämään sitä uudestaan.

4.2 Prototyyppien hyödyntäminen kehityksessä

Korkean tarkkuuden omaavien prototyyppien avulla käytettävyyden arviointi mahdollistaa kattavan palautteen saamisen kehityksen tueksi. Kun prototyyppi on lähes lopullisen työkalun mukainen ulkoasultaan ja interaktiivisuudeltaan, voidaan arvioida käytettävyyttä ja käyttökokemusta käyttöliittymän ulkonäön lisäksi. Tämän perusteella voidaan tehdä perusteltuja muutoksia olemassa olevaan käyttöliittymäkokonaisuuteen, tai täydentää sitä testauksessa ilmenneiden puutteiden mukaisesti. Toisaalta Pichlerin (2012) mukaan aikaisemmassa vaiheessa karkealla prototyyppillä testauttamalla voidaan saada tärkeämpää palautetta, koska testaaja ei ole mielessään sitoutunut tietyn näköiseen käyttöliittymään, ja saattaa paremmin ilmaista mitä toivoisi käyttöliittymältä (Pichler 2012).

Testauttamisella saadun palautteen lisäksi, prototyypointi helpottaa myös tiedon välittämistä yritykselle. Interaktiivisella käyttöliittymäprototyyppillä voidaan välittää toteutettuja muutoksia myös asiakkaalle helpommin. Toimiva prototyyppi voi mahdollistaa myös omien ideoiden sisällyttämisen lopulliseen tuotteeseen silloin kun suunnitelman perusteella asiakas on varautunut. Prototyypin avulla asiakas voi valintansa perusteena käyttää toimivaa ominaisuutta ehdotetun suunnitelman sijaan. (Cao et al. 2015, s. 12)

5 KEHITYSPROSESSI

Tapaustutkimusyrietyksellä on olemassa oleva valmis toiminnallinen työkalu joka halutaan kääntää nykyisestä versiosta verkkorajapinnassa tarjottavaan muotoon. Tässä luvussa käydään vaiheittain läpi millaisia toimintatapoja prosessin suunnittelussa ja käyttöliittymäprototyypin toteutuksessa käytettiin.

5.1 Kehitysprosessin valmistelu

Tapaustutkimusyrietyksen kanssa sovittiin ensimmäisen tapaamisen tapahtuvan yrityksen tiloissa helmikuun alussa. Aivan ensimmäiseksi tutustuttiin hieman yrityksen toimitiloihin, työntekijöihin ja toimialaan, jotta kehitysprosessin tueksi saataisiin yhteinen käsitys siitä millaiselle alalle ja millaiseen tarkoitukseen työkalun verkkoversiota ollaan kehittämässä. Näiden jälkeen suunniteltiin toteutuksen yksityiskohtia sisältäen työn aikataulun, yrityksen tekniset odotukset ja rajaukset toteutettavalle työkalulle. Lopuksi edellä mainitut seikat sovittiin kandidaatintyön puitteissa toteutettavaksi.

Kandidaatintyössä toteutetaan vain käyttöliittymäsuunnittelua ja käyttöliittymän toiminnallisuuden toteuttamista. Ulkoasulliset toiminnallisuudet kehitetään mahdollisimman toimiviksi, jotta työkalun ulkoasun lisäksi voidaan arvioida sen käytettävyyttä. Arvioinnin mahdollistamisen lisäksi kehitystyön tavoitteena on luoda jatkokehittävää käyttöliittymäkokonaisuus. Toimiva käyttöliittymä mahdollistaa helpomman jatkokehityksen aloittamisen. Työkalun muut toiminnallisuudet, kuten laskentaa suorittavat taustafunktiot sekä yhteydet tietokantoihin jätetään kokonaan toteuttamatta. Työkalua kehitetään myös erillään lopullisesta ympäristöstään, mitä varten yritys loi omille palvelimilleen erillisen osion, jonne kehitetyt käyttöliittymäprototyypit voidaan lisätä. Erilliseltä palvelimelta niitä voidaan lisäämisen jälkeen tarkastella ja esitellä muille, esimerkiksi yrityksen omille asiakkaille.

Kehitysprosessin vastuut jaettiin yrityksen ja kandidaatintyöntekijän kesken seuraavalla tavalla. Yrityksen vastuulla oli toimittaa suunnittelua varten kuvankaappauksia olemassa olevasta työkalusta. Näitä kuvankaappauksia käytettiin ensimmäisen käyttöliittymäversion

alkukohtana. Ensimmäiseen kehitysjaksoon riitti käyttöliittymän elementtien sijainnin hahmottelu ja kuvankaappauksien sijoittaminen niihin. Ensimmäisen version toteuttamisen jälkeen yrityksen oma graafikko toi kehityksen tueksi omia näkemyksiään esille staattisten kuvien muodossa. Näitä kuvia voitiin käyttää kehityksen perustana, mutta niiden ohella oli mahdollista tuottaa visuaalisesti erilaisia prototyyppejä. Lopullisen kehityssuunnan ja kehityskohteen valinta pysyi kuitenkin aina yrityksen vastuulla, mutta kehitysideat molempien osapuolien toimesta olivat sallittuja ja jopa toivottuja. Oikean kehityssuunnan löytämiseksi haettiin myös palautetta käyttöliittymästä ja sen käytettävyydestä yrityksen omilta asiakkailta. Yritys vastasi testaaajien löytämisestä ja testauttamisen toteuttamisesta.

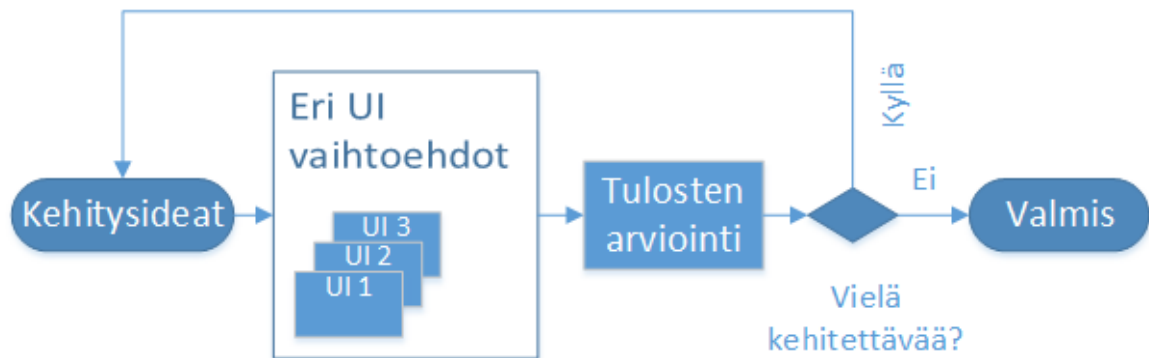
Kandidaatintyöntekijän päätehtävä oli käyttöliittymäprototyyppien toteuttaminen valitsemallaan ohjelmointitavalla. Toteutuksessa voi hyödyntää myös valmiita ohjelmointikehyksiä eli frameworkkejä. Näiden avulla oli mahdollista helpottaa tai parantaa tehtyä työtä. Toteutuskeinojen toiminnan selvittäminen ja käyttöliittymässä hyödyntäminen oli kandidaatintyöntekijän toinen pääasiallinen vastuu. Näiden ohella muiden kehitysideoiden esille tuomista kannustettiin. Työn tulosten suhteen yrityksen odotukset pidettiin realistisina, koska prototypoinnin oli tarkoitus olla käyttöliittymäkehittämisen lisäksi myös teknisten toteutustapojen kartoitusta.

Kehitysprosessista suunniteltiin kevyt, jotta se olisi yhden henkilön toteutettavissa. Huomioon tuli myös ottaa se, että tämä henkilö koko kehitysajanjakson aikana ensisijaisesti suorittaa opintojaan. Tästä syystä ketterän ohjelmistokehityksen mukainen prosessi ei sisällä monissa valmiissa malleissa käytettävää 40 tunnin työviikkoa, ja kehitysjaksot ovat myös hieman pidempiä. Kehitysjaksojen pituudeksi sovittiin noin kaksi viikkoa, ja jaksoja oli erottamassa palaveri, jossa käytiin toteutetut asiat läpi ja valittiin seuraavan jakson kehitystavoitteet. Yhteydenpitovälineinä käytettiin ensisijaisesti sähköpostia ja palaverien pitämiseen Skype-kommunikointiohjelmaa.

5.2 Yhden kehitysjakson vaiheet

Kuvassa 2 on esitelty tiivistetyssä muodossa koko käyttöliittymäkehitysprosessi. Yksi kehitysjakso sisälsi ensimmäisenä vaiheenaan kehitysideoiden esittelyn ja kehityssuunnan valitsemisen. Toisena vaiheena oli käyttöliittymäprototyyppien eri versioiden toteuttaminen

ja niiden toimittaminen yritykselle sähköpostilla. Jakson viimeisessä vaiheessa eli palaverissa arvioitiin tehtyjä muutoksia ja nykyisen käyttöliittymän tilaa jatkokehityksen tarvetta ajatellen. Jos päätetään uusi kehityskohde, alkaa siitä uusi kehitysjakso. Kun käyttöliittymän katsottiin olevan riittävällä tasolla, voitiin kehitysprojektin katsoa olevan valmis ja prosessi päättyy. Jakson keskimääräinen kesto oli noin kaksi viikkoa, mutta kumman tahansa osapuolen tarvittaessa, tarkkaa kestoä voitiin muuttaa.



Kuva 2. Käyttöliittymäkehitysprosessi

Kehitysideavaiheeseen kuului olennaisimpana kollektiivinen ideointi kehitettävästä kohteesta ja sen kehitystavasta. Yleensä keskityttiin pääosin yhteen elementtiin tai alueeseen käyttöliittymässä kerrallaan, esimerkiksi sivun jakamiseen osiin, lasketun tiedon esittämismuotoon tai syötekenttien sijaintiin ja ulkoasuun. Kehityskohdat valittiin lähes aina yrityksen graafikon tekemästä käyttöliittymävedoksesta suurin piirtein samalla ulkoasulla. Tämä oli niin sanottu perusprototyyppi, joka toteutettiin lähes sellaisenaan. Perusprototyypin rinnalla toteutettiin myös muita vaihtoehtoja, jotka erosivat joko pelkästään väritykseltään tai hieman muodoltaan. Ylimääräisiin kehitysversioihin prototyyppien toteuttajalla oli vapaat kädet, jos toteuttamisen aikana tulee mieleen joku kehitysidea, jota haluaa kokeilla, se on sallittua, kunhan se on vain erillinen versio muiden vaihtoehtojen rinnalla.

Kehityssuunnan valitsemisen jälkeen alkoi kandidaatintyöntekijällä tutkimustyö, jonka tavoitteena oli vastata seuraaviin kysymyksiin:

- Miten sovitut kehityskohdat voidaan toteuttaa käytännössä?
- Onko toteutukseen olemassa jo valmiita kehyksiä joita voisi hyödyntää?

- Jos kehyksiä on useita, niiden erojen vertailu ja soveltuvuuden arviointi, sekä valitsemisen perustelun esittely yritykselle.

Kun näihin kysymyksiin oli saatu vastaus, pystyttiin aloittamaan itse ohjelmointityö. Oli hyvin yleistä, että kehityskohdan toteuttaminen vaati uusien menetelmien opettelua. Oppimisen myötä kandidaatintyöntekijä saavutti paremman kyvyn arvioida erilaisten toteutustapojen soveltuvuutta käyttöliittymäprototyyppien työkaluiksi. Tämän lisäksi vaihtoehtojen kartoittamisen myötä voitiin myös arvioida menetelmien soveltuvuutta lopullisiksi toteutusvaihtoehdoiksi. Jonkin kehyksen käytön syynä saattoi myös olla tulevaisuuteen varautumisen sijaan sen koettu helppous suhteutettuna saavutettavissa olevaan lopputulokseen. Ketterän käyttöliittymäkehityksen tarkoitus ei tässä tapauksessa välttämättä ollut heti toteuttaa prototyyppiä parhaaksi katsotuin ohjelmointikäytäntein, vaan yksinkertaiset, helposti toteutettavat ratkaisut olivat jopa haluttuja (Pichler 2012). Helposti tehtävillä kehitysmuutoksilla voitiin pienellä työmäärällä arvioida ratkaisun soveltuvuus, jonka hyväksymisen jälkeen sen toteuttamistapaan voitiin palata ja toteuttaa se asianmukaisemmilla ohjelmointimenetelmillä.

Joka kehitysjaksossa oli useampi kuin yksi käyttöliittymäprototyyppi samanaikaisesti, koska jo prosessissa mukana olevan tiimin mielipiteet ”parhaasta” vaihtoehdosta erosivat paljon. Kehitystiimin koko ei missään vaiheessa yltänyt edes kymmeneen henkilön määrään ja siltikin eroavia mielipiteitä oli aina. Tästä syystä useiden toimivien, eli interaktiivisten, prototyyppien toteuttaminen oli merkittävä tiedonlähde seuraavien kehityskohtien valinnassa. Samasta syystä testaaminen yrityksen omilla asiakkailla olisi ehdottoman tärkeää, koska näin saadaan kehitysryhmän tietoon lopullisen käyttäjän mieltymyksiä käyttökokemuksen ja työkalun näkymän suhteen.

Tulokset arvioitiin noin kahden viikon välein järjestettävissä kehitysjaksojen lopussa järjestettävissä palavereissa. Palaverit oli toteutettu maantieteellisistä rajoitteista johtuen etänä, hyödyntäen Skype-kommunikointiohjelmaa. Toisesta palaverista alkaen, yrityksen luvalla, palaverit nauhoitettiin, jotta läpikäytyihin asioihin saattoi palata myöhemmin. Esimerkiksi jokin kehitysehdotuksen aikaansaaman reaktion tarkistaminen nauhoitteelta auttoi kehitysvaihtoehtojen valinnassa. Palaverissa osallistujina olivat tapaustutkimusyrityksen edustajat tai osa heistä ja kandidaatintyöntekijä, sekä mahdollisuuksien mukaan myös kandidaatintyön

ohjaaja. Ohjaajan osallistuminen ei kuitenkaan ollut edellytys kehitysprosessin puolesta. Tarkoitus tässä tapauksessa oli ainoastaan selvittää ohjaajalle kandidaatintyön etenemistä. Palaverit alkoivat poikkeuksetta aina esittelemällä sen kertaiset osallistujat ja kertomalla estyneiden henkilöiden poissaolosta. Näin kaikille osallistujille oli selvää mitä palaverissa on, ja pystyttiin heti tekemään oletuksia läpikäytävistä asioista, koska yrityksen edustajilla on jokaisella hieman eroavaa erikoisosaamista. Esittelyjen jälkeen käytiin edellisen palaverin kehityslista läpi, ja suoraan toteuttajalta kysymällä merkattiin toteutetut kohdat siltä erää tehdyiksi. Tämän jälkeen kehityskohdat käytiin yksitellen tarkemmin läpi, ja yritettiin herättää keskustelua erikseen jokaisen seikan kohdalla. Merkittäväksi todetut huomiot, sekä huonot, että hyvät, merkattiin kehityskohtien yhteyteen jatkokäsittelyä odottamaan. Jos tässä kohtaa tehdyt huomiot katsottiin jatkokehityksen perusteeksi, sovittiin heti muutoksen laajuus ja suunta, yleensä pieniä asioita, kuten tekstin lisääminen painikkeen yhteyteen, tai jonkin tekstin sanallisen muodon muuttaminen.

Kuluneessa kehitysjaksossa tehtyjen muutosten läpikäymisen jälkeen pohdittiin seuraavaa kehityskohtaa. Yleensä kehityskohta poimittiin graafikon tekemästä käyttöliittymäkuvasta, ja kyseinen elementti tulisi olemaan seuraavan jakson pääkehityskohta. Tässä vaiheessa kaikki palaveriin osallistujat saivat ideoida vapaasti ja hyvältä kuulostavat ehdotukset kirjattiin mahdollisiin jatkokehityskohtiin. Jos kehityskohta oli riittävän yksinkertainen, saatettiin se suoraan määrätä toteutettavaksi. Palaverissa keskusteltiin myös käyttöliittymäkehityksen ulkopuolisista seikoista, pääasiassa alustavista suunnitelmista työkalun varsinaiseen toteutustapaan. Tämä suunnittelu oli avointa vaikka keskustelun kohde jää aiheensa puolesta jo kandidaatintyön ulkopuolelle. Kandidaatintyöntekijältä saatettiin myös kysyä mielipiteitä ja kehitysehdotuksia, mutta näitä ei kuitenkaan alettu käyttöliittymäkehitysjakson aikana vielä toteuttamaan. Viimeisenä palaverissa sovittiin seuraavan kehitysjakson pituus, eli seuraavan palaverin ajankohta.

5.3 Kehitystyön päättymisen

Kehitysprojektin oli alustavasti suunniteltu kestävän huhtikuun loppuun saakka. Projekti päätettiin kuitenkin jo huhtikuun puolessavälissä, vaikka suunnitelman mukaan aikaa olisi ollut vielä yhdelle kehitysjaksolle. Viimeisessä pidetyssä palaverissa käyttöliittymän katsottiin vastaavan projektin alussa asetettuja yrityksen odotuksia. Tähän viimeiseen

palaveriin mennessä yritys oli saanut ensimmäistä kertaa kunnollista palautetta omilta asiakkailtaan ja palautteen perusteella päätös projektin päättämisestä voitiin tehdä. Kehityslistaan jääneet, toteuttamattomat ideat, jätettiin vielä ylös kirjatuiksi muistioon, jotta käyttöliittymäkehitystä voidaan jatkaa myöhemmin, jos sille on tarvetta. Käyttöliittymän sisältö täytti tässä vaiheessa vähimmäisvaatimukset toiminnallisuuksien toteuttamiselle, eli syötteille ja tulosteille oli omat paikkansa. Myös muut työkalun tarvittavat toiminnot, kuten erillinen käyttöohje, voidaan sovittaa käyttöliittymään koska niillekin oli paikat valmiina.

6 TUTKIMUSTULOKSET

Tutkimustuloksissa esitellään ensin kehitysprosessissa koetut haasteet, sekä arvioidaan prosessin onnistumisen taustalla olevia syitä. Lopuksi tuodaan esille toimintatutkimuksen tuloksena saadut mahdolliset kehityskohdat.

6.1 Kehitysprosessissa koetut haasteet

Merkittävimmäksi puutteeksi kehitysprosessin aikana muodostui prototyyppien puutteellinen testauttaminen. Yrityksen edustajat itse testasivat sekä arvioivat työkalua kattavasti, mutta työkalun todellisten loppukäyttäjien mielipiteitä ja arvioita saatiin kerättyä vasta kehitysprosessin loppuvaiheessa. Litchfieldin (2014) mukaan kaikista hyödyllisintä olisi kerätä käyttäjäpalautetta mahdollisimman aikaisin ja tarkastuttaa tehdyt muutokset uusilla testeillä, jottei kehityssuunnassa tehdä liikaa oletuksia, varsinkaan virheellisiä. Cao, Ellis ja Zieba (2015) kehottavat käyttämään mahdollisimman todenmukaisia käyttöliittymäprototyyppijä käytettävyydestestauksessa, jotta testaajat osaavat keskittyä oikeisiin asioihin (Cao et. al 2015, s. 73-75). Tässä mielessä testauksen myöhäinen ajankohta sopi kehitysprosessiin hyvin, koska vasta loppuvaiheessa prototyyppi sisälsi eniten käyttöliittymän toiminnallisuuksia. Toisaalta siihen asti tehtyjä kehitysvalintoja ei oltu validoitu käyttäjätesteillä, vaan pelkästään yrityksen edustajien mieltymyksillä, joten toteutetut kehityskohdat eivät välttämättä vastanneet todellista tarvetta (Greever 2015).

Toissijaisina haasteina olivat toteutuneiden testien hallitsemattomuus ja palaveriinhin osallistuvien ihmisten vaihteleva paikallaolo. Toteutettuja testejä ei suoritettu hallitussa testiympäristössä. Asiakkaille esiteltiin prototyypin uusin versio ja pyydettiin kuvausta ensivaikutelmasta, eli testaus oli vain pinnallista. Tämä voidaan lukea vain toissijaiseksi haasteeksi, koska pinnallinenkin testaus suuremmissa määrissä olisi antanut kattavampaa palautetta. Palaveriinhin osallistuvien henkilöiden vaihtelevuus vaikutti tiettyjen kehityskohtien käsittelyn siirtymiseen myöhemmäksi. Tämä oli kuitenkin vain pienehkö hidaste, koska puuttuvat tiedot saatettiin koko kehitysryhmän tietoon myöhemmin sähköpostilla.

6.2 Kehitysprosessin onnistumisen arviointi

Suurin vaikuttava tekijä kehitysprosessin onnistumisessa oli tapaustutkimusyrityksen aktiivinen ja laaja osallistuminen kehitysprosessin kaikkiin vaiheisiin. Amblerin (2002-2012) mukaan kaikkien sidosryhmien osallistuminen kehitysprosessiin on tärkeää koska ihmiset eivät tavallisesti osaa määritellä haluamaansa tavoitetta täydellisesti, mutta osaavat kuitenkin suuntaa antavasti kertoa haluamansa. Myös ketterän ohjelmistokehityksen julistuksessa (2001) olevat arvot painottavat yhteistyön ja muutosten tekemiseen kykenemistä arvoissaan, mikä tässä kehitysprosessissa toteutui. Kandidaatintyön aikana jatkuva yhteistyö yrityksen kanssa mahdollisti useiden prototyyppien tekemisen ja niiden jatkuvan arvioinnin. Jatkuva osallistuminen mahdollisti myös kehityksen suunnan validoinnin tai puutteellisiin valintoihin puuttumisen.

Kommunikointi tapaustutkimusyrityksen kanssa oli avointa ja luottavaista, mikä edesauttoi päätösten syntymistä, kuin myös tehtyjen päätösten laatua (Ambler 2002, s. 42-43). Yritys luotti kandidaatintyöntekijän löytävän sopivia teknisiä toteutusratkaisuja valituille kehityskohdille sekä toteuttavan niitä. Yritys ei myöskään salannut palavereissa käytyjä asioita, jotka liittyivät työkalun kehityksen tulevaisuuteen eivätkä suoraan kandidaatintyössä tehtävään kehitystyöhön. Näiden asioiden jakaminen auttoi ymmärtämään työkalun kehityksen todellisia odotuksia kandidaatintyön jälkeiseltä ajalta, joka puolestaan vaikutti valittuihin toteutustekniikoihin. Cerejon (2010) ja Amblerin (2004) mukaan on turha prototypoida ominaisuuksia joita on vaikea toteuttaa todellisuudessa, ja on paljon hyödyllisempää keskittyä toteuttamiskelpoisiin vaihtoehtoihin (Ambler 2004, s. 197; Cerejo 2010). Opinnäytetyössä suoritettavan kehitystyön tapauksessa tällä oli vielä suurempi merkitys, koska yritys haki prototypoinnilla myös tekniseen toteutukseen vaihtoehtoja. Toteutetut prototyypit kallistuivat myös korkean tarkkuuden puolelle, eli prototyypit ovat lähes lopullisen toteutustavan mukaisia.

Kuvassa 3 on esiteltyä työkalun kehityksen jakaminen viiteen lopulliseen käyttöliittymäelementtiin. Kussakin kehitysvaiheessa annettiin pääpainoa enimmäkseen yhdelle kuvassa esitellylle elementille tai näiden osien sijoittelulle käyttöliittymän sisällä. Tällä tavalla etenemällä työkalun kehitys oli vaiheittaista, ketterän ohjelmistokehityksen mukaista. Työkalun käyttöliittymän kehittäminen osa kerrallaan mahdollisti

tapaustutkimusyrityksen väliintulon huonojen kehitystoimenpiteiden jälkeen ja nämä kohdat voitiin käydä läpi uudestaan ja valita erilainen lähestymistapa kehitykseen. Amblerin (2004) mukaan sidosryhmien kanssa keskustelu on ensisijainen tapa löytää nämä kehityskohdat, jonka jälkeen niistä voidaan toteuttaa prototyyppi tai useita (Ambler 2004, s.194-196). Samalla vaiheittainen kehitys myös kasvatti kehitysvaihe kerrallaan työkalun arvoa yrityksen silmissä ja yritys sai konkreettista vastinetta kehitysprosessiin osallistumiselle koko prosessin aikana.



Kuva 3. Käyttöliittymän jako elementteihin

Vaiheittainen kehitys ja yhteen elementtiin kerrallaan keskittyminen edesauttoivat myös kandidaatintyöntekijän suoriutumista työtehtävästä sekä kannusti opettelemaan uutta. Jokaista elementtiä varten tehty toteutusvaihtoehtojen kartoittaminen pakotti oppimaan uusia tekniikoita, mikä puolestaan tässä tapauksessa kasvatti myös kehityskohtien toteuttajan arvoa yritykselle. Sitä mukaa kun kandidaatintyöntekijän osaaminen kasvoi jokaisen kehitysvaiheen myötä, voitiin tätä hankittua taitoa hyödyntää aiempien kehityskohtien iteratiiviseen parantamiseen, sekä uusien kehitysmahdollisuuksien avaamiseen.

Ketterän kehityksen arvojen ja periaatteiden hyödyntäminen edesauttoi suuresti työkalun kehitystä, ja prosessin lopussa yrityksen odotukset saatiin täytettyä tyydyttävästi. Kehitysprosessin tuotoksena syntynyt työkalun käyttöliittymä täytti vaatimukset työkalun jatkokehittämiseksi, koska prototypoinnissa käytettyjä toteutusmenetelmiä voidaan suoraan käyttää työkalun toiminnallisuuksien, kuten laskennan tulosten esittämisen, toteuttamiseen.

6.3 Kehityslista jatkoa varten

Tutkimuksen tuloksien arvioinnin perusteella voidaan nostaa kaksi mahdollista kehityskohtaa esille myöhempiä projekteja varten. Ensinnäkin luvussa 6.1 käsitelty testauksen puutteellisuus tulisi korjata. Kunnollisella ulkopuolisella testauksella, varsinkin kehityksen alkuvaiheeseen sijoittuvalla, voitaisiin saada konkreettista tietoa varsinaisten käyttäjien mieltymyksistä, eikä kehityksen tarvitse nojata vain ajatukseen siitä minkä luullaan olevan hyvä valinta. Tässä kehitysprosessissa käyttöliittymää oli kehitetty jo pitkälle ennen kunnollista testauttamista, jolloin huomattiin että käyttäjät olivat jo tyytyväisiä käyttöliittymään, mutta halusivat enemmänkin muutoksia sen sisältöön. Aikaisempi testauttaminen olisi voinut kohdistaa kehitystä niihin kohtiin joilla on arvoa työkalun käyttäjille.

Toisena kehitysehdotuksena on dokumentoinnin lisääminen prosessin osaksi. Vaikka ketterän ohjelmistokehityksen toinen arvo painottaa toimivaa ohjelmistoa enemmän kuin dokumentaatiota, ei sitä tulisi jättää kokonaan pois. Kehitystyön nopeasta etenemisestä huolimatta työkalun käyttöliittymän toiminnallisuutta olisi voinut jälkeempään dokumentoida. Prosessin alussa ei määritelty dokumentointikäytäntöjä millään lailla, joten sitä ei tässä tapauksessa tehty. Stellmanin ja Greenen (2014) mukaan ketterissä kehitysprojekteissa dokumentaation riittäväksi määräksi määritellään kehitystyötä tekevän ryhmän omasta mielestä riittävä määrä (Stellman & Greene 2014, s. 64-65). Toisaalta tilanteessa jossa kandidaatintyön tekeminen olisi keskeytynyt ennen valmistumista, olisi yritys joutunut omatoimisesti käymään käyttöliittymän toteutusta kooditasolla läpi päästäkseen jatkamaan kehitystä. Normaalisti tätä ongelmaa ei ole, koska kehitystyön taustalla on kehittäjän ja asiakkaan välinen kirjallinen sopimus, mutta kandidaatintyön taustalla ei ollut tällaisia velvoitteita.

7 JOHTOPÄÄTÖKSET

Tässä kandidaatintyössä tehtiin toimintatutkimusta ketterän ohjelmistokehityksen periaatteiden ja prototyypoinnin soveltuvuudesta erään suomalaisen pk-yrityksen käyttöliittymäkehitystarpeisiin. Yrityksen kanssa yhteistyössä suunniteltiin ketterän kehityksen piirteitä sisältävä kehitysprosessi, jossa toteutettiin jatkuvasti tarkemmiksi eteneviä käyttöliittymäprototyyppejä mahdollisimman todenmukaisin toteutuskeinoin. Kehitystyön tavoitteena oli saada käyttöliittymäprosessin päätteeksi kehitettävän työkalun käyttöliittymä jatkokehityksen mahdollistavalle tasolle.

Kehitysprosessi saatettiin hyväksytysti päätökseen aikataulua edellä, joten prosessin itsessään voidaan sanoa olleen erittäin onnistunut. Prototyyppien käyttö helpotti ja edisti kehitettävän käyttöliittymän arviointia. Prototyyppien jatkuva visuaalinen ja toiminnallinen tarkentuminen mahdollistivat viimeisen käyttöliittymäprototyypin käyttämistä sellaisenaan jatkokehityksen aloittamisessa. Ketterän ohjelmistokehityksen arvot ja periaatteet soveltuivat prosessissa käytettäväksi erittäin hyvin. Näistä arvoista ja periaatteista prosessin onnistumiseen eniten vaikuttaneita olivat toimivan ohjelmiston käyttäminen edistymisen mittarina, yksilöiden ja kanssakäymisen painottaminen, sekä tiivis ja avoin asiakasyhteistyö.

Toimintatutkimuksen tuloksena nostettiin esille kaksi kehityskohtaa, joiden sisällyttämistä vastaaviin prosesseihin yritys voisi harkita jatkossa. Todellisilla asiakkailla testaaminen aikaisemmassa vaiheessa prosessia helpottaisi kehityksen kohdistamista suuresti. Kohdistamisen lisäksi yritys voisi välttyä ylimääräiseltä työltä, koska toissijaiset kehityskohdat olisivat testaamalla jo aiemmin tiedossa. Toisena kehityskohtana oli dokumentaation sisällyttäminen prosessiin. Vaikka suoritettu prosessi toimikin hyvin ilman erillistä dokumentaatiota toteutetusta käyttöliittymästä, sen sisällyttäminen vähentäisi yrityksen kokemia riskejä kehitysprosessin keskeytyessä. Tiivistäen koko työn tulokset voidaan esittää seuraavasti: ketterän ohjelmistokehityksen hyödyntäminen paransi kehitysprosessin vaiheita, mutta suuremman hyödyn saavuttaminen olisi edellyttänyt kattavampaa testaamista lopullisilla käyttäjillä.

LÄHTEET

Ambler, S. 2002, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*, New York: John Wiley & Sons [Viitattu 10.5.2016] Saatavilla: https://books.google.fi/books?hl=en&lr=&id=uh_jSk2FSa0C&oi=fnd&pg=PR5#v=onepage&q&f=false

Ambler, S. 2004, *The Object Primer*, 3. painos, New York: Cambridge University Press [Viitattu 10.5.2016] Saatavilla: <https://books.google.fi/books?id=Zdv55a4cIWYC&printsec=frontcover#v=onepage&q&f=false>

Ambler, S. 2002-2012, *Active Stakeholder Participation: An Agile Best Practice*, [Verkkodokumentti] [Viitattu 10.5.2016] Saatavilla: <http://agilemodeling.com/essays/activeStakeholderParticipation.htm>

Beck, K. et al. 2001, *Agile Manifesto* [Verkkodokumentti] [Viitattu 27.4.2016] Saatavilla: <http://agilemanifesto.org/>

Beck, K. et al. 2001, *Agile Manifesto: Julistuksen takana olevat periaatteet* [Verkkodokumentti] [Viitattu 10.5.2016] Saatavilla: <http://agilemanifesto.org/iso/fi/principles.html>

Bristow, E. 2014, *9 Myths about agile* [Verkkoartikkeli] [Viitattu 10.5.2016] Saatavilla: <http://deloitte.wsj.com/cio/2014/03/25/9-myths-about-agile/>

Cao, J., Ellis, M. Zieba, K., 2015, *The Ultimate Guide to Prototyping* [E-Kirja] [Viitattu 10.5.2016] Saatavilla: <https://studio.uxpin.com/ebooks/guide-to-prototyping/>

Cerejo, L. 2010, *Design Better And Faster With Rapid Prototyping* [Verkkoartikkeli] [Viitattu 10.5.2016] Saatavilla: <https://www.smashingmagazine.com/2010/06/design-better-faster-with-rapid-prototyping/>

Greever, T. 2015, *Articulating Design Decisions*, Sebastopol: O'Reilly Media [Viitattu 10.5.2016] Saatavilla: <https://books.google.fi/books?id=XT-gCgAAQBAJ&pg=PT199#v=onepage&q&f=false>

- Litchfield, T. 2014, *The Value of Iterative User Research and Rapid Prototyping* [Verkkoartikkeli] [Viitattu 10.5.2016] Saatavilla: <http://www.effectiveui.com/blog/2014/07/09/the-value-of-iterative-user-research-and-rapid-prototyping/>
- Mikoluk, K. 2013, *Agile Vs. Waterfall: Evaluating The Pros And Cons* [Verkkoartikkeli] [Viitattu 11.5.2016] Saatavilla: <https://blog.udemy.com/agile-vs-waterfall/>
- Ocamb, S. 2013, *What Does the Agile Manifesto Mean?* [Verkkoartikkeli] [Viitattu 27.4.2016] Saatavilla: <https://www.scrumalliance.org/community/articles/2013/2013-april/what-does-the-agile-manifesto-mean>
- Patton, M. 2014, *Qualitative Research & Evaluation Methods: Integrating Theory and Practice*, 4. painos, SAGE
- Pichler, R. 2012, *Agile User Interface Design* [Verkkodokumentti] [Viitattu 27.4.2016] Saatavilla: <http://www.romanpichler.com/blog/agile-user-interface-design/>
- Riel, M. 2016, *Understanding Action Research*. Center for Collaborative Action Research, Pepperdine University [Verkkodokumentti] [Viitattu 26.4.2016] Saatavilla: <http://cadres.pepperdine.edu/ccar/define.html>
- Rigsby, L. 2005, *Action Research, How Is It Defined?* , George Mason University [Verkkodokumentti] [Viitattu 23.4.2016] Saatavilla: <https://gse.gmu.edu/assets/media/tr/ARRigsbyppt.htm>
- Shore, J. Warden, S. 2007, *The Art of Agile Development*, Sebastopol: O'Reilly Media
- Stellman, A. Greene, J. 2014, *Learning Agile*, Sebastopol: O'Reilly Media
- Warfel, T. 2009, *Prototyping: A Practitioner's Guide*, New York: Rosenfeld Media [Viitattu 11.5.2016] Saatavilla: <https://books.google.fi/books?id=aieWBrFeRtUC&printsec=frontcover#v=onepage&q&f=false>

LIITTEET

Liite 1. Ketterän ohjelmistokehityksen julistuksen takana olevat periaatteet

1. Tärkein tavoitteemme on tyydyttää asiakas toimittamalla tämän tarpeet täyttäviä versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti.
2. Otamme vastaan muuttuvat vaatimukset myös kehityksen myöhäisessä vaiheessa. Ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyyn edistämiseksi.
3. Toimitamme versioita toimivasta ohjelmistosta säännöllisesti, parin viikon tai kuukauden välein, ja suosimme lyhyempää aikaväliä.
4. Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan.
5. Rakennamme projektit motivoituneiden yksilöiden ympärille. Annamme heille puitteet ja tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä.
6. Tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu.
7. Toimiva ohjelmisto on edistymisen ensisijainen mittari.
8. Ketterät menetelmät kannustavat kestävään toimintatapaan. Hankkeen omistajien, kehittäjien ja ohjelmiston käyttäjien tulisi pystyä ylläpitämään työtahtinsa hamaan tulevaisuuteen.
9. Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi edesauttaa ketteryyttä.
10. Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on oleellista.
11. Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoiduissa tiimeissä.
12. Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan, ja mukauttaa toimintaansa sen mukaisesti.

(Beck et al. 2001)