

Timo Haapala

Ketterä ohjelmistokehitys ja asiakkuudet

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mediatekniikan koulutusohjelma

Insinööriytyö

12.4.2016

Tekijä Otsikko	Timo Haapala Ketterä ohjelmistokehitys ja asiakkuudet
Sivumäärä Aika	32 sivua + 1 liite 12.4.2016
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Mediatekniikka
Suuntautumisvaihtoehto	Digitaalinen media
Ohjaajat	Toimitusjohtaja Kristjan Laansalu Yliopettaja Kari Aaltonen
<p>Insinööriyön tarkoituksena oli selvittää ketterien kehitysmenetelmien käyttöä projektinhallinnan työkaluina sekä niiden eroja lähestymistavoissa projektinhallintaan verrattuna perinteisempiin metodeihin. Opinnäytetyö tehtiin muotoilutoimistolle tavoitteena tutkia yrityksen toimintaa projekteissa ja sen prosesseja sekä kehittää entistä tehokkaampia tapoja ja konkreettisia ehdotuksia siihen, kuinka ketterät kehitysmenetelmät käytännössä voisivat tehostaa työskentelyä.</p> <p>Insinööriyön tekemisen aikana ketterien kehitysmenetelmien tarkoitus selvisi aluksi teorian, ja myöhemmin käytännön kautta. Projekteissa työskentely voi olla erittäin haastavaa ja isojen kokonaisuuksien hallitsemiseksi on käytettävä erilaisia työkaluja, joita muun muassa ketterät kehitysmenetelmät tarjoavat. Kahden erilaisen projektin etenemistä tarkastelemalla saatiin käsitys sovelluskehitysprojektien työnkulusta yleisellä tasolla ja eroista projektien välillä. Toinen projekteista oli lyhyt ja tarkasti aikataulutettu, ja toinen laajempi pitkäkestoinen kokonaisuus.</p> <p>Vaikka opinnäytetyön tilaajan työskentelyn voi suurelta osin ajatella olevan ketterää ja muutoksiin pystytään reagoimaan todella tehokkaasti, ottamalla käyttöön jonkin määritellyn ketterän kehitysmenetelmän, prosesseja saadaan kehitettyä entistäkin tehokkaammiksi. Tämä parantaisi, ei pelkästään kyseessä olevan projektin tuotetta, mutta myös itse työprosessia ja täten kaikissa tulevilla projekteilla työskentelyä.</p>	
Avainsanat	projektinhallinta, ketterät kehitysmenetelmät, scrum, asiakkuudet, sovelluskehitys

Author Title	Timo Haapala Agile Software Development and Customership
Number of Pages Date	32 pages + 1 appendice 12 April 2016
Degree	Bachelor of Engineering
Degree Programme	Media Technology
Specialisation option	Digital Media
Instructors	Kristjan Laansalu, CEO Kari Aaltonen, Principal Lecturer
<p>The purpose of this study was to analyse the usage of agile development in project management and the differences in the approach compared to more traditional methods. The study was carried out for a design agency and the goal was to study the corporation's operations and processes in projects and to develop more efficient ways and concrete propositions about how agile development could improve working in practice.</p> <p>While carrying out the study, the goal of the agile development methods became clearer, first through theory, and later through practice. Working in projects can be very challenging, and to manage big complex wholes, the usage of different tools that agile development methods provide is compulsory. In the study, by observing two different projects, the development's working processes and the differences between projects, could be comprehended in general level. One of the project was short with a tight schedule and the other much larger and long-last.</p> <p>Despite the fact that the design agency's project work can be considered as agile for most parts and reactions to changes can be made efficiently, processes can be developed to be even more productive by implementing some of the official agile development methods. This would improve not only the product of the current project, but also the working process itself and therefore the work in all projects yet to come.</p>	
Keywords	Project management, agile development, Scrum, customership, software development

Sisällys

Lyhenteet

1	Johdanto	1
2	Projektinhallinta ja asiakkuudet	2
2.1	Projektiluontoinen työ	2
2.2	Projektinhallinnan perusasiat	3
2.3	Projektinhallinnan menetelmät ja työkalut	4
2.4	Asiakkuudet	5
3	Ketterä ohjelmistokehitys	7
3.1	Ketterän ohjelmistokehityksen historia	7
3.2	Periaatteet	8
3.3	Ketteriä kehitysmenetelmiä	9
3.4	Scrum-viitekehys	10
4	Projektit ja kehitysehdotukset	17
4.1	Asiakastiedon sijoittajasivusto	17
4.1.1	Lähtötilanne ja asiakkaan tarpeet	18
4.1.2	Tulosten esittely	19
4.2	Helsingin kaupungin perusopetuksen opetussuunnitelman julkaisualusta	22
4.2.1	Lähtötilanne ja asiakkaan tarpeet	22
4.2.2	Tulosten esittely	25
4.3	Kehitysehdotukset	27
5	Yhteenveto	30
	Lähteet	31

Liitteet

Liite 1. Asiakastiedon sijoittajasivuston rakenne listautumista varten

Lyhenteet

PHP	Hypertext Preprocessor. Verkkosivujen ohjelmointikieli, jolla palvelin luo verkkodokumentteja.
FDD	Feature Driven Development. Ketterä kehitysmenetelmä, jossa keskitytään ensisijaisesti tuotteeseen kehitettäviin ominaisuuksiin.
UML	Unified Modelling Language. Graafinen mallinnuskieli, jolla kuvataan ohjelmiston rakennetta, ominaisuuksia ja toimintoja.
DSDM	Dynamic Software Development Method. Ketterä kehitysmenetelmä, jossa tuotetta testataan koko kehitysprosessin ajan ja kehittämistä jatketaan saatujen tulosten perusteella.

1 Johdanto

Jotta sovelluskehitysprojektit onnistuisivat ja projektista valmistuva tuote vastaisi mahdollisimman hyvin työn tilaajan – asiakkaan – tarpeisiin, projektin kulun on oltava hallittavissa ja työskentelyn ja resurssien käytön tarkasti suunniteltua. Tätä varten on kehitetty lukuisia työkaluja ja toimintamalleja, joita käyttämällä päästään selvästi parempiin tuloksiin kuin toisilla. Projektinhallintaan liittyvät oleellisesti tuotteen parissa työskentelevät henkilöt, tuotteen tilanneen tahon edustajat ja henkilöt, jotka tuotteen näkökulmasta edustavat ominaisuuksiltaan loppukäyttäjiä. Muita resursseja ovat muun muassa aika, raha, osaaminen ja motivaatio. Koska muuttujia on näin paljon, kokonaisuuden hallinta vaatii tiettyjen prosessien noudattamista, empiirisen tiedon hyödyntämistä ja tätä kautta jatkuvaa projektinhallinnan kehittämistä.

Tässä opinnäytetyössä perehdytään ketteriin kehitysmenetelmiin projektinhallinnan työkaluina ja tarkastellaan asiakkuuksia osana sovelluskehitysprojektia. Aihe valittiin siitä syystä, että tekijällä ei ole aiempaa kokemusta tai tietoa ketteristä kehitysmenetelmistä ja rajallinen tieto myös projektinhallinnan menetelmistä yleisemmällä tasolla. Opinnäytetyö tehdään suunnittelutoimisto Agenda Helsingille tarkoituksena tutkia projektien kulkua käytännössä ja pohtia, kuinka työskentelyä projekteissa voisi kehittää niiden koosta ja luonteesta riippumatta.

Aluksi opinnäytetyössä perehdytään projektityöskentelyyn ja asiakkuuksiin yleisesti. Luvussa 3 käsitellään ketterien kehitysmenetelmien yhteisiä piirteitä ja esitellään osa niistä. Lisäksi luvussa käydään läpi tarkemmin Scrum-viitekehys ja sen perusperiaatteet. Luvuissa 4 ja 5 esitellään kaksi hyvin erilaista projektia, ja kehitysehdotukset sekä näihin kahteen projektiin että yleisemmällä tasolla sovelluskehitysprojekteissa työskentelemiseen esitellään luvussa 6. Palautetta projekteissa työskentelystä kerätään sekä Agenda Helsingin sisällä että asiakkailta. Tulokset esitetään projektikohtaisesti projektien esittelyjen yhteydessä, ja niitä käytetään myös kehitysehdotusten pohjana.

2 Projektinhallinta ja asiakkuudet

2.1 Projektiluontoinen työ

Jotta voidaan ymmärtää projektinhallintaa kokonaisuutena ja sen käsitteitä sekä osia, on ensin ymmärrettävä mitä projektilla tarkoitetaan. Työ voidaan luokitella karkeasti kahtia sen luonteen, jatkuvuuden perusteella: projektiluontoiseksi tai jatkuvaksi työksi. [1, s. 4–5.]

Projektilla tarkoitetaan päämäärähakuista laajaa kokonaisuutta, joka on aina uniikki toiseen projektiin nähden, sillä on määritellyt alkamis- ja päättymisajankohta sekä asetetut tavoitteet. Projektit ovat siis väliaikaisia pyrkimyksiä, joilla yritetään päästä ennalta määriteltyihin tavoitteisiin määrätyillä resursseilla. Tällaisia resursseja ovat muun muassa käytettävissä oleva työaika, pääoma, henkilöstön määrä ja osaaminen sekä aiemmin kehitetyt menetelmät ja tekniikat. [1, s. 4–9.]

Jatkuva työ koostuu operaatioista, joille on mahdotonta asettaa tarkkaa päättymisajankohtaa ja sitä kautta myös kestoa. Yrityksen tai organisaation päivittäiset samankaltaisina toistuvat työt ovat operatiivisia, kuten esimerkiksi laskutus ja kirjanpito, työn valvonta sekä muut hallinnolliset ja ylläpidolliset toimet. Projektityö pyrkii keskittymään valmiisiin kokonaisuuksiin ennalta määrätyillä resursseilla. Operatiivisessa työssä pyritään toteuttamaan toistuva työtehtävä tehokkaammalla resurssien käytöllä. [1, s. 3–8; 2.]

Projektimuotoinen työskentely koetaan usein palkitsevammaksi ja mielekkäämmäksi työskentelymuodoksi kuin operatiivinen samankaltaisten työtehtävien toistaminen. Projektityö on helpompi jakaa pienempiin osiin, joiden valmiiksi saaminen on todennettavissa ja projekti etenee lähemmäs valmista tuotetta. Projektin päätyttyä onnistuneesti lopputuloksena on tuote, josta on nähtävissä tehdyn työn määrä ja laatu kokonaisuutena. [1, s. 7, 361–363.]

2.2 Projektinhallinnan perusasiat

Projekteissa työskennellään yhden tai useamman henkilön kokoisissa ryhmissä, joilla on yhteinen tavoite ja joiden työskentely tähtää yhteiseen päämäärään. Ryhmän jäsenellä on ennalta osoitetut työtehtävät, jotka määräytyvät henkilön osaamisalueen tai -alueiden perusteella. Ryhmässä on vastuuhenkilö, joka vastaa koko ryhmän toiminnasta. Projektissa voi olla yksi tai useampia ryhmiä, joiden tavoitteet pyrkivät kohti yhteistä päämäärää. [1, s. 10–12.]

Ryhmien työskentelyä ohjaa projektipäällikkö, jonka toiminta tähtää projektin eteenpäin viemiseen suunnitelmallisesti. Projektipäällikkö vastaa resurssienhallinnasta, kuten käytetystä ja vielä käytettävissä olevasta ajasta ja pääomasta, työn tuloksista sekä muista tarvittavista resursseista, kuten tiedon ja materiaalien sekä mahdollisen projektin ulkopuolisen tietotaidon hankkiminen. Projektipäällikön tehtäviin kuuluu myös ilmenevien ongelmien ratkaiseminen ryhmät ylittävällä tasolla ja tulevien ongelmien ennalta ehkäiseminen. Projektipäällikön toiminnan tulee olla myös muilta osin proaktiivista, jotta tuleviin tilanteisiin pystytään reagoimaan tehokkaasti mahdollisesti jo etukäteen. [1, s. 12–16.]

Ryhmien ja projektipäällikön lisäksi projektilla on yhteyshenkilö, joka vastaa yhteydenpidosta asiakkaan kanssa. Myös asiakkaalla on oltava yhteyshenkilö ja kommunikointi – tai sen organisointi – osapuolien välillä tapahtuu näiden henkilöiden toimesta. Pienemmissä projekteissa projektipäällikkö voi toimia myös yhteyshenkilönä. Organisaation sisäisissä projekteissa yhteyshenkilöä ei välttämättä tarvita, jos projekti ei vaadi yhteydenpitoa organisaation muiden osien kanssa. [1, s. 76–77; 3, s. 55–56.]

2.3 Projektinhallinnan menetelmät ja työkalut

Jotta projekti etenisi suunnitellusti ja työnkulku olisi hallittua, kaikki työ ja sen prosessit on suunniteltava etukäteen ja tässä suunnitelmassa on pyrittävä pysymään. Projektisuunnitelma on yksi keskeisimpiä työkaluja, ja sen tehtävänä on kuvata koko projektin etenemisen lisäksi myös sen pienempien osioiden suunniteltua etenemistä. Esimerkki projektisuunnitelmasta on esitetty kuvassa 1. [3, s. 3–6.]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Week of...	14-Jul	21-Jul	28-Jul	4-Aug	11-Aug	18-Aug	25-Aug	1-Sep	8-Sep	15-Sep	22-Sep	29-Sep	6-Oct	13-Oct
2	Research Design Phase														
3	Kickoff														
4	Define Scope														
5	Revise														
6															
7	Data Collection Phase														
8	Schedule Interviews														
9	Conduct Interviews														
10	Survey														
11	Analysis														
12															
13	Final Report Creation														
14	Writing														
15	Copy editing														
16	Layout														
17	Report Launch														
18															

Kuva 1. Projektisuunnitelma [9].

Projektisuunnitelmassa (kuva 1) esitetään ajanjakso, jonka aikana projekti on suunniteltu tehtävän. Projekti on jaettu pienempiin osioihin, virstanpylväisiin, jotka kuvaavat aikaan sidottuja kokonaisuuksia, joista koko projekti koostuu. Kullekin virstanpylväälle voidaan osoittaa vastuuhenkilö, jonka tehtävänä on varmistua tavoitteisiin pääsemisestä määräaikaan mennessä. Osa virstanpylväistä vaatii edellisten suorittamista loppuun ennen työskentelyn aloittamista, mutta osa voidaan suorittaa myös samanaikaisesti riippuen niiden sisältämistä tavoitteista ja työtehtävistä. [1, s. 156–158.]

Virstanpylväät koostuvat pienemmistä kokonaisuuksista, jotka ovat jaettavissa aina yksittäisen työtehtävän tasolle asti. Kuinka pieniin osiin projekti pilkotaan, riippuu projektin luonteesta ja koosta. Virstanpylväät ovat määräaikoja, joihin mennessä niiden sisältämät tavoitteet on saavutettava suorittamalla tavoitteiden saavuttamisen edellyttämät työtehtävät. [1, s. 156–158.]

Projektisuunnitelman lisäksi projektinhallinnan yleisesti käytettyjä työkaluja ovat muun muassa resurssienhallintasuunnitelma, riskienhallintasuunnitelma, viestintäsuunnitelma, asialista ja tehtävälista. Näiden lisäksi käytössä on myös lukuisia muita työkaluja, joiden käyttö vaihtelee organisaatiosta ja projektista riippuen. [4]

Valtaosassa projekteja on mahdotonta määrittää projektin alkuvaiheessa kaikkia tarvittavia resursseja, työtehtäviä ja toimenpiteitä sekä esille nousevia ongelmatilanteita ja niiden ratkaisuja. Myös vallitsevien olosuhteiden muuttuminen etenkin pitkissä projekteissa luo haasteita projektin onnistuneesti loppuun saattamiselle. [1, s. 135, 175–177.]

Jos resurssienhallintaan ei ole panostettu riittävästi, osaa työtehtävistä ei saada tehtyä määräaikaan mennessä tai ennalta määritellyn budjetin mukaisesti. Tämä voi aiheuttaa viivästymisen projektin valmistumisessa, kuluksen tarpeettoman kasvamisen tai pahimmillaan koko projektin lakkauttamisen. Resursseja voidaan myös ohjata väärin. Projektin toisiin osa-alueisiin voidaan käyttää tarpeettoman paljon voimavaroja, jolloin muihin osa-alueisiin ei välttämättä jää riittävästi aikaa, pääomaa tai osaamista. Ajankäytön ja pääoman käytön lisäksi myös henkilöstön hallinta on oleellinen osa resurssienhallintaa. [1, 176–178; 5, s. 2–4.]

Tekemättä jäävien töiden lisäksi uhkana on myös ylimääräinen turha työ, joka kuluttaa tarpeettomasti projektin resursseja. Kaikilla projektin työntekijöillä on oltava selkeä käsitys tavoitteista, joihin työtehtävillä pyritään, sekä käytettävissä olevista resursseista. Työtehtävät on myös pystyttävä priorisoimaan, koska osa on tärkeämpiä ja osan jäädessä vähemmän tärkeiksi. [5, s. 2–4.]

2.4 Asiakkuudet

Projektin toteuttavan osapuolen lisäksi tuotteella on oltava myös tilaaja – asiakas. Yksinkertaistetusti asiakas pyytää organisaatiolta ratkaisua ongelmaansa ja täten täyttämään jonkin tarpeen. Asiakas voi olla organisaation ulkopuolinen taho tai osa organisaatiota, jolloin projekti on organisaation sisäinen.

Jotta organisaatio voi palvella asiakasta mahdollisimman tehokkaasti, kuuntelemisen lisäksi organisaation on ymmärrettävä asiakkaan tarvetta ja arvontuotantoa. Organisaatio

tion kilpailukyvyyn lähtökohtana on osaamisen ja asiantuntemuksen lisäksi asiakastuntemus, jonka perustana on asiakkaan tavoitteiden, toiveiden ja tarpeiden määrittäminen. Organisaation tehtävänä on myös tunnistaa ja käsitellä asiakkaan projektiin liittyvät huolet sekä pyrkiä muuttamaan omaa prosessiaan asiakkaan kanssa yhteensopivaksi. [6, s.15.]

Asiakassuhteet projekteissa

Asiakkaan ja projektin toteuttavan tahon välistä viestintää on hallittava ja ylläpidettävä koko projektin ajan. Asiakkaalla, projektiryhmän lisäksi, on oltava käsitys projektin kuluista, tarvittavista resursseista ja siitä, miksi ja miten projekti toteutetaan. Yleinen ongelmatilanne on asiakkaan puutteelliset tiedot vaadittavien työtehtävien määrästä ja laadusta, tarvittavista resursseista ja työnkulusta. Toisaalta ongelmatilanne syntyy myös, jos toteuttava osapuoli ei ymmärrä asiakkaan tarpeita ja toiveita eikä täten pysty lähtökohtaisesti niitä toteuttamaan. Näiden ongelmien välttämiseksi osapuolien on pystyttävä kommunikoimaan siten, että kumpikin osapuoli ymmärtää toista ja muuttaa toimintaansa tarvittaessa. Toteuttavan osapuolen työskentelyn on lisäksi oltava riittävän läpinäkyvää, jotta asiakas tietää, mihin resursseja käytetään ja kuinka projekti edistyy. [1, s. 207–210.]

Tarpeen vaatiessa asiakasta on myös autettava omien prosessiensa kehittämisessä siltä osin, kuin ne projektiin liittyvät. Asiakas ei myöskään välttämättä ole itse määritellyt tarpeitaan projektin alussa tarkasti, jolloin organisaation on yhdessä asiakkaan kanssa määriteltävä, mitkä ovat asiakkaan tavoitteet, mikä on asiakkaalle arvokasta ja mikä on asiakkaan oikea tarve. Asiakkaan toiveet ja tarpeet eivät aina kohtaa kaikilta osin, joten organisaation on varmistettava, että myös asiakas ymmärtää toiveen ja tarpeen eron ja että ainoastaan jälkimmäinen lisää tuotteen arvoa. [6, s. 15.]

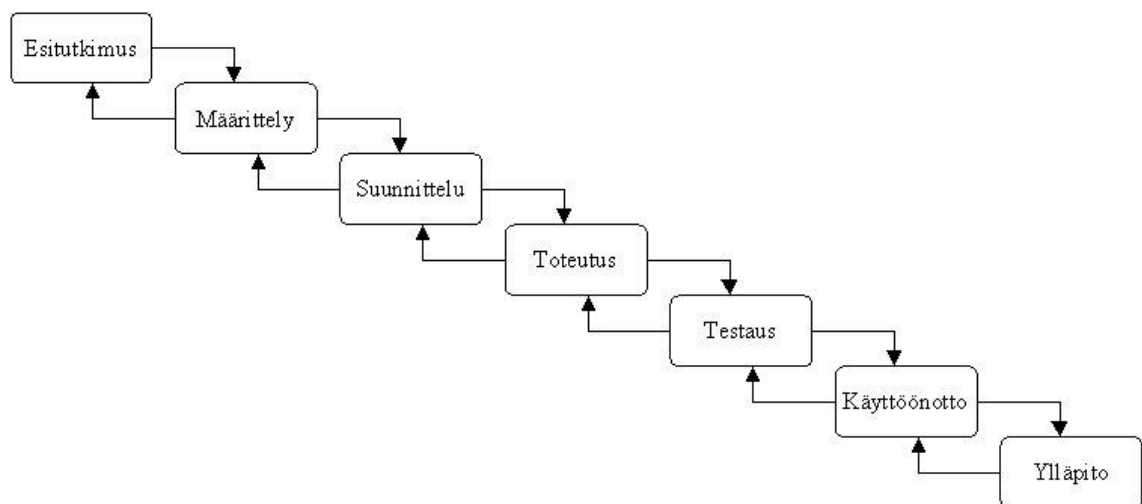
Asiakastyytyväisyyden edellytykset

Organisaatio voi toimillaan lisätä asiakastyytyväisyyttä käsittelemällä asiakkaan huolia ja ratkaisemalla ongelmia, jotka niitä aiheuttavat. Asiakas tulee myös pitää osana projektin kehitystyötä koko sen elinkaaren. Toisin sanoen asiakkaan tulee tietää tarkasti, missä vaiheessa projekti on, mitä on tehty ja mitä on vielä tekemättä. Myös tarpeen uudelleenarviointi ja sitä kautta syntyvien muutosten toteuttaminen on tehtävä yhteistyössä asiakkaan kanssa. Asiakastyytyväisyys on suoraan yhteydessä tuotteen arvon nostamiseen. [6, s. 23–24, 121–122.]

3 Ketterä ohjelmistokehitys

3.1 Ketterän ohjelmistokehityksen historia

Ensimmäiset sovelluskehityksen menetelmäopit kehitettiin 1970-luvulla vastaamaan aiemmin vallinneeseen kaoottiseen ja suunnittelemattomaan työskentelyyn, jota sovelluskehityksessä esiintyi. 1990-luvun loppupuoliskoon mennessä kehitettyjä raskaita sovelluskehitysprosesseja pidettiin hitaina ja byrokraattisina, ja osa niistä oli kehitetty jo 1980-luvulla. Nämä prosessit käyttivät vesiputousmallia, jossa projekti etenee asteittain ja seuraavan kokonaisuuden kehittäminen alkaa vasta kun edellinen on saatu valmiiksi (kuva 2). [7; 8.]



Kuva 2. Sovelluskehitysprojektin vesiputousmalli [10].

Vastineeksi näille raskaille sovelluskehitysprosesseille joukko alan toimijoita kannatti ajatusmaailmaa, jossa muutoksiin negatiivisesti suhtautumisen sijaan niihin mukaudutaan ja suhtaudutaan positiivisesti. He havaitsivat, että tällainen lähestymistapa mahdollistaa nopean reagoinnin muuttuviin olosuhteisiin, kuten teknologian muutokseen vaatimuksiin, ja johtaa tehokkaamman sovelluskehityksen strategian luomiseen [8]. Tämä toimijoiden joukko koostui 17 henkilöstä, jotka kokoontuivat Snowbirdissä, Yhdysvalloissa vuonna 2001 ja koostivat tilaisuutta varten julistuksen ketteristä kehitysmenetelmistä.

Julistuksen nimi on The Agile Software Development Manifesto ja siinä esitetään tiivistetysti ketterän sovelluskehityksen ajatusmaailma, joka on perustana nykyaikaisille ketterille kehitysmenetelmille. [11.]

3.2 Periaatteet

Ketterien kehitysmenetelmien tarkoituksena on vastata mahdollisimman tehokkaasti muutoksiin, joita projektin aikana esiintyy. Projektin aloitusvaiheessa ei ole selvää, mitkä olosuhteet ovat tuotteen aiotun julkaisun ajankohtana tai sen jatkokehitysvaiheessa. Lisäksi olosuhteet voivat muuttua jo kehitystyön aikana. Tuotteen lopullisia vaatimuksia on mahdotonta tietää tarkasti ennalta projektia käynnistettäessä, joten muutoksia on pysyttävä tekemään tarpeen vaatiessa. Teknologian kehittyminen ja asiakkaan tarpeiden ja markkinoiden muuttuminen projektin aikana luo selvän tarpeen muutosten tekemiselle, ja muutoksia on kyettävä tekemään suhteellisen lyhyessä ajassa hallitusti. [12, s. 222–225; 7.]

Vesiputousmallia noudattavat sovelluskehitysmenetelmät mahdollistavat edelliseen kehitysvaiheeseen palaamisen, mutta ei sitä aiempiin (kuva putouksesta). Jos muutoksia tulee edellistä aiempiin vaiheisiin, on koko kehitystyö aloitettava alusta. Tämä kuluttaa tarpeettoman paljon resursseja ja suuri osa työstä on tehty turhaan. Näissä menetelmissä koko projektin kulku suunnitellaan projektin alkaessa, ja tässä suunnitelmassa pysytään koko projektin ajan riippumatta olosuhteiden muutoksesta. Jos edelliseen vaiheeseen palaaminen ei ole riittävä toimenpide tarvittavien muutosten aikaansaamiseksi, projekti keskeytetään ja aloitetaan uudelleen, mikäli resursseja on riittävästi ja asiakkaalla on edelleen tarve tuotteelle. [13.]

Jos tuote suunnitellaan kokonaan projektin alussa, se vastaa parhaimmillaan ennusteita tuotteen julkaisuajankohdan vallitsevasta tarpeesta. Jos tarve ei ole sama tuotteen julkaisuajankohtana, kuin mitä ennustettiin, tuote ei täysin vastaa asiakkaan tarpeisiin. Lisäksi tuote on mahdollista julkaista vasta kehitystyön päätyttyä eikä siitä ole aiempia julkaistuja versioita. Asiakkaan on siis odotettava koko kehityksen päättymistä, ennen kuin tuote on julkaisemiseen soveltuva. [13.]

Käyttämällä ketteriä kehitysmenetelmiä sovelluskehityksessä tuotteesta on mahdollista julkaista versioita koko kehitysprosessin ajan, eikä prosessin päättymistä tarvitse odottaa. Näissä menetelmissä projekti jaetaan pienempiin osiin, iteraatioihin, siten että tuotteen ominaisuuksia kehitetään omina kokonaisuuksinaan suhteellisen lyhyin ajanjaksoin. Suunnittelu tehdään muun kehitystyön kanssa samaan aikaan, ja suunnitelmia muutetaan asiakkaan tarpeen tai muiden olosuhteiden muuttuessa. Tuotteen testaaminen tehdään ohjelmointityön yhteydessä, eikä erillistä testausvaihetta ole. [12, s. 223–224.]

Ketteriä kehitysmenetelmiä käytettäessä työskentely tapahtuu ryhmissä, joissa on riittävä määrä osaamista iteraation toteuttamiseen ja tuoteversion julkaisemiseen. Muutoksiin pystytään reagoimaan nopeasti ja tehokkaasti, ja suunnittelu on mahdollista mukauttaa näihin muutoksiin. Kehittämisprosessiin suhtaudutaan innovatiivisesti, ja työskentely tapahtuu evolutiivisesti. Tuotetta myös pyritään parantamaan jatkuvasti ja tuoteversioita julkaisemaan aina, kun se on mahdollista, periaatteessa jokaisen iteraation päätteeksi. [12, s. 222–225.]

3.3 Ketteriä kehitysmenetelmiä

Kehitysmenetelmiä, joita voidaan kutsua ketteriksi, on useita. Kaikille niille on yhteistä mukautuvuus ja pyrkimys tuotteen arvon enimmäistämiseen. Tällaisia menetelmiä ovat muun muassa Crystal-menetelmäoppi, dynaaminen sovelluskehitys (DSDM), ominaisuuskeskeinen sovelluskehitys (FDD), Lean-sovelluskehitys, Extreme-ohjelmointi (XP) ja Scrum, joka käsitellään yksityiskohtaisemmin luvussa 3.4. [14, s. 3.]

Crystal-menetelmäoppi sisältää joukon metodeja, jotka on luokiteltu projektin työntekijöiden määrän ja tuotteen tärkeyden perusteella. Näitä metodeja ovat kirkas, keltainen, oranssi, punainen ja sininen. Kirkas-metodissa on vain yksi pieni ryhmä, joka työskentelee samassa tilassa. Keltainen-metodissa voi olla 20 työntekijää, oranssissa 40 ja punaisessa 80. Metodeilla on myös eri tuotteen kriittisyys, jolla tarkoitetaan sitä, kuinka paljon vahinkoa sovellus voi aiheuttaa mennessään virhetilaan. [12, s. 338] Esimerkiksi sairaalassa potilaan elintoimintojen ylläpitämiseen kehitetty sovellus voi aiheuttaa potilaan kuoleman mennessään epäkuuntoon [12, s. 337–338].

Dynaamisen sovelluskehityksen (DSDM) keskeisimpiä periaatteita ovat käyttäjien osallistuminen kehitysprosessiin koko sen keston ajan, kehitysryhmien valtuuttaminen tekemään päätöksiä, jatkuva tuoteversioiden julkaiseminen ja niiden ominaisuuksien keskittäminen kohti arvoa nostavaa tuotetta sekä pakollinen toistava ja inkrementaalinen kehitystyö. Muita periaatteita ovat ominaisuuksien palauttaminen muutosta edeltäneeseen versioon tarvittaessa, tärkeimpien ominaisuuksien määrittelemine ja niiden muutosten rajoittaminen, testaaminen koko projektin ajan sekä kannustaminen yhteistyöhön ja yhteisöllisyyteen työntekijöiden kesken. [15, s. 2–3.]

Ominaisuuskeskeinen sovelluskehitys (FDD) keskittyy ensisijaisesti tuotteen ominaisuuksiin. Kehitystyö on jaettu viiteen toimintoon: yleismallin kehittämiseen, ominaisuuksien listaamiseen, suunnittelemiseen ominaisuuden mukaan, muotoiluun ominaisuuden mukaan ja rakentamiseen ominaisuuden mukaan. Tuotteen suunnittelussa käytetään UML-malleja, joilla ominaisuuksia ja niiden rakenteellisia yhteyksiä voidaan kuvata. [16.]

Lean-sovelluskehitys on johdettu Toyotan autonvalmistuksessa käyttämistä prosesseista, joiden keskeisimpänä periaatteena on turhan työn minimointi ja tuotteen arvon maksimointi. Toisin sanoen asiakkaalle pyritään kehittämään mahdollisimman arvokas tuote mahdollisimman pienillä resursseilla. Tämä pyritään toteuttamaan keskittymällä tuotteen arvon optimointiin koko arvoketjussa sen sijaan, että keskityttäisiin erillisten resurssien ja teknologioiden kehittämiseen. [17.]

Extreme-ohjelmoinnin tärkeimpänä periaatteena on tuoteversioiden toimittaminen asiakkaalle sitä mukaa, kuin niitä tarvitaan. Sen sijaan, että tuote julkaistaan kaikkine haluttuine ominaisuuksineen tietyssä ajankohtana, siitä julkaistaan versio, kun tarvittava ominaisuus on lisätty. Kehittäjät kommunikoivat jatkuvasti asiakkaan sekä toistensa kanssa, ja testaaminen ja sitä kautta palautteen saaminen on samanaikaista kehitystyön kanssa. Suunnittelu pyritään Extreme-ohjelmoinnissa pitämään yksinkertaisena ja selkeänä. [18.]

3.4 Scrum-viitekehys

Scrum-viitekehys pyrkii vastaamaan projektissa esiintyviin haasteisiin mahdollistamalla joustavan ja nopean reagoimisen muutoksiin, helpon käyttöönoton sekä tuottavuuden

parantamisen ja tuotteen arvon maksimoimisen vähentämällä riskejä ja jatkuvalla työskentelyn prosessien kehittämisellä. Scrum on empiirinen lähestymistapa projektinhallintaan, ja se mahdollistaa innovatiivisen ja yksilöllisen työskentelyn sekä kannustaa yhteistyöhön ja jatkuvaan oman työn tarkasteluun ja kehittämiseen, mikä johtaa työn tulosten paranemiseen sekä yksilö- että ryhmätasolla. Scrum on skaalautuva ratkaisu, jossa sovelluskehityksen monimutkaiset vaatimukset saadaan pilkottua pienemmiksi kokonaisuuksiksi, mikä mahdollistaa niiden yksinkertaisemman käsittelyn. [19, s. 3; 5, s. 1–2.]

Scrumin historia

Scrumin kehittivät Jeff Sutherland ja Ken Schwaber 1990-luvun alkupuoliskolla. He käyttivät työnsä pohjana Hirotaka Takeuchin ja Ikujiro Nonakan julkaisemaa tutkielmaa ”The New New Product Development Game”, jossa myös termi scrum esiintyy ensimmäistä kertaa. Tutkielmassa viitataan rugbyyn ja muihin sen kaltaisiin joukkuelajeihin, jossa joukkueet toimivat pieninä itseohjautuvina yksikköinä, jotka keskittyvät tavoitteisiin tehtävien sijasta. Sutherland ja Schaber esittelivät määrittelemänsä viitekehityksen Oopsla-konferenssissa vuonna 1995. [20.]

Roolit ja käsitteet

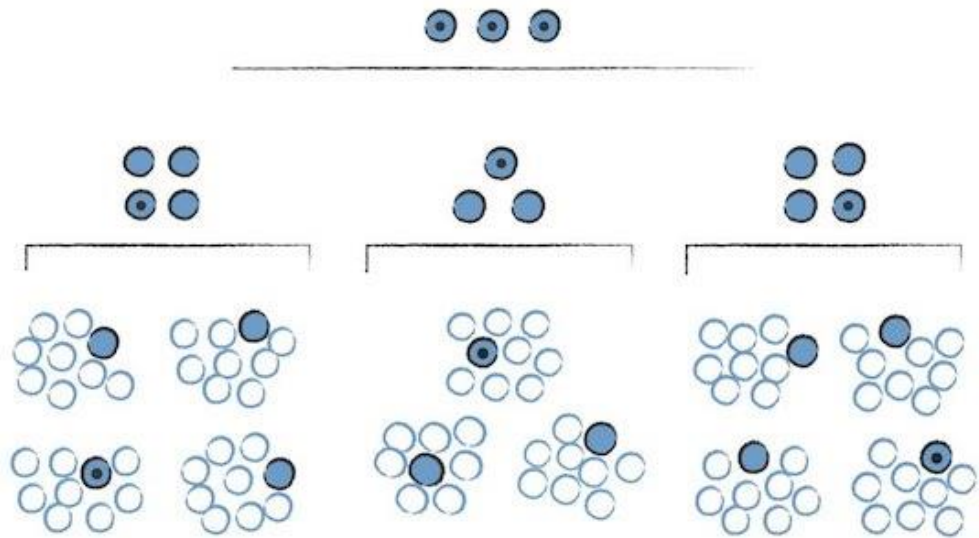
Scrum-viitekehystä noudattavassa kehitysmenetelmässä työntekijöillä on yksi kolmesta roolista: tuoteomistaja, scrummaster tai kehitystiimin jäsen. Kullakin roolilla on omat vastualueet ja työtehtävät. Lisäksi scrum sisältää useita käsitteitä, joilla työnkulkua ja eri työvaiheita kuvataan. Näitä käsitteitä ovat sprintti, päivittäinen sprintti, tuotteen kehitysjono ja edistymiskäyrä, joilla on kullakin lisäksi alakäsitteitä. [19, s. 4–14.] Kaikki roolit ja käsitteet kuvataan seuraavissa alaluvuissa.

Scrum-tiimi

Scrum-viitekehyksen toiminnallista yksikköä kutsutaan scrum-tiimiksi. Jäsenten määrä on tyypillisesti viidestä yhdeksään henkilöä, ja ryhmän jäsenen rooli on joko tuoteomistaja, scrummaster tai kehitystiimin jäsen [21]. Scrum-tiimit työskentelevät itsenäisesti ja ohjaavat itse omaa toimintaansa. Ryhmällä tulee olla riittävästi osaamista tehtävän suorittamiseksi ilman ulkopuolista apua. Kaikki jäsenet toimivat yhteisen päämäärän saavuttamiseksi, eikä scrum-tiimissä ole perinteisiä ohjelmistokehityksen rooleja, kuten esimerkiksi suunnittelija, ohjelmoija tai sovellusarkkitehti, mutta jäsenten työtehtävät painottuvat osaamisalueen mukaan. [21; 19, s. 4.]

Projektissa voi olla yksi tai useampia scrum-tiimejä, jotka suorittavat tehtävänsä itsenäisesti toisistaan riippumatta. Jos ryhmiä on enemmän kuin yksi, jokaisessa scrum-tiimissä tulee olla vastuhenkilö, joka osallistuu tapaamisiin, joissa scrum-tiimien työskentelyä ohjataan kohti ryhmien yhteistä tavoitetta. Tämän henkilön rooli on scrummaster. [21.]

Jos projektissa on lukuisia scrum-tiimejä, scrum skaalautuu luontevasti esimerkiksi jakamalla scrum-tiimit kokonaisuuksiin. Jokaisen kokonaisuuden scrum-tiimien scrummastereit tapaavat säännöllisin väliajoin ja sopivat kokonaisuuden seuraavasta yhteisestä tavoitteesta. Näiden tapaamisten osallistujista valitaan vastuhenkilö, joka osallistuu muiden kokonaisuuksien vastuhenkilöiden tapaamisiin, joissa kokonaisuuksien työskentelyä ohjataan kohti yhteistä päämäärää. [21.] Tämänkaltainen esimerkkutilanne on esitetty kuvassa 3.



Kuva 3. Projektin scrum-tiimit ja kehitystiimit [21].

Tuoteomistaja

Tuoteomistaja vastaa ensisijaisesti projektin pääoman tuottoasteesta (ROI). Toisin sanoen hän vastaa kehitystiimin työn kannattavuudesta ja sen pysymisestä kannattavana koko projektin ajan. Tehtävän suorittamiseksi tuoteomistajalla on oltava näkemys, millainen valmiin tuotteen tulisi olla ja miksi projekti toteutetaan. Hänellä on siis oltava visio, joka välitetään muille scrum-tiimin jäsenille. Tätä varten tuoteomistajalla on suora yhteys ryhmiin. Lisäksi tuoteomistaja priorisoi tuotteen ominaisuudet ja vastaa myös muilta osin tuotteen kehitysjonosta. [19, s. 4–5; 5, s. 34.]

Tuoteomistaja pyrkii olemaan vaikuttamatta ryhmän sisäiseen toimintaan, mutta asettaa ryhmille tavoitteet, jotka pyritään saavuttamaan seuraavan sprintin aikana. sprintin aikana tuoteomistaja ei puutu sen työnsuoritukseen. Hän voi kuitenkin keskeyttää sprintin tarvittaessa. [5, s. 34.]

Scrummaster

Scrummasterin tehtävänä on ohjata kehitystiimin ja tuoteomistajan sekä organisaation ja scrum-tiimin välistä kanssakäymistä ja varmistaa, että kaikki scrum-tiimin jäsenet ymmärtävät ja käyttävät scrumia. Tämä tapahtuu valvomalla ryhmän toimintaa ja varmistamalla, että scrumin sääntöjä, teoriaa ja käytäntöjä noudatetaan. Scrummaster ei puutu siihen, kuinka ryhmä työskentelee. Hän toimii myös puskurina kehitystiimille ja estää täten häiriötekijöiden vaikuttamisen kehitystiimin työhön. Myös kehitystiimin valmentaminen uudessa ympäristössä, kehitystiimin työskentelyä haittaavien esteiden poistaminen ja kehitystiimin tuottavuuden maksimoiminen kuuluvat scrummasterin työtehtäviin. [19, s. 6; 5, s. 31–32.]

Tekniikoiden ehdottamisella kehitysjonon hallintaan, varmistamalla ymmärryksen kehitysjonon järjestämisestä arvon maksimoimiseksi sekä auttamalla scrum-tiimiä ymmärtämään tarpeet kehitysjonon kohdille scrummaster palvelee tuoteomistajaa. Scrummasterin tulee lisäksi ymmärtää empiirisen ympäristön suunnittelua ja ketteryyttä sekä helpottaa muiden työntekijöiden työskentelyä tarpeen mukaan. [19, s. 6.]

Organisaation tarpeisiin scrummaster vastaa suunnittelemalla scrumin toteutusta, johtamalla ja valmentamalla scrumin käyttämisessä sekä työskentelemällä muiden scrum-

mastereiden kanssa tuottavuuden kasvattamiseksi. Scrummaster myös vastaa muutoksista scrum-tiimin tehokkuuden kasvattamiseksi ja avustaa työntekijöitä ja muita projektin osallisia ymmärtämään sekä käyttämään scrumia ja kokemuspohjaista tuotekehitystä. [19, s. 6.]

Kehitystiimi

Tuoteomistajan ja scrummasterin lisäksi scrum-tiimiin kuuluu kehitystiimi, jonka tehtävänä tuottaa valmis tuoteversio jokaisen sprintin päätteeksi. Tuoteversio luodaan kehitysjonon perusteella. Kehitystiimin ulkopuoliset työntekijät eivät osallistu tuotteen kehittämiseen, joten ryhmästä on löydettävä kaikki vaadittava osaaminen tuoteversion valmiiksi saamiseksi. Kuten scrum-tiimi, myös kehitystiimi vastaa itse omasta toiminnastaan. Kehitystiimi organisoii työnsä ja työtapansa sisäisesti eli ryhmän ulkopuoliset tahot eivät vaikuta ryhmän sisäiseen työkulkuun. [5, s. 35–38; 19, s. 5–6.]

Kehitystiimin jäsenten ammattinimike on kehittäjä riippumatta työntekijän työtehtävistä. Vaikka kehittäjällä on oma osaamisalueensa, koko kehitystiimi on yhteisesti vastuussa tavoitteeseen pääsemisestä eikä sen sisällä ole erillisiä pienempiä ryhmiä, jotka vastaavat eri osa-alueista. Kehitystiimin koon tulee olla riittävän suuri, jotta ryhmästä löytyy kaikki tarvittava osaaminen ja tavoitteen vaatima työmäärä saadaan tehtyä, mutta liian suuri ryhmäkoko heikentää ketteryyttä merkittävästi. Tyypillisesti kehitystiimi koostuu 3–9 henkilöstä, mutta suositeltu ryhmäkoko ei saisi ylittää 7:ää henkilöä. [19, s. 5–6.]

Sprintti

Projektin kehitystyö on jaettu säännöllisen pituisiin ajanjaksoihin, sprintteihin, jotka kestävät kuukauden tai vähemmän aikaa, ja niiden kesto pysyy samana koko projektin ajan. Tässä aikamääreessä scrum-tiimin on pystyttävä kehittämään julkaisukelpoinen ja käytettävä tuoteversio, jossa on ennalta määrätyt ominaisuudet. Sprinttien välissä ei ole taukoa, eli edellisen sprintin päätyttyä seuraava alkaa välittömästi. Sprintti sisältää suunnittelupalaverin, päiväpalaverit, katselmuksen ja retrospektiivin. [19, s. 7.]

Suunnittelupalaverissa scrum-tiimi suunnittelee juuri käynnistyneen sprintin aikana suoritettavien työtehtävien toteutuksen. Scrum-tiimi määrittelee, mitä on mahdollista toteuttaa sprintin aikana ja kuinka työ tullaan käytännössä toteuttamaan. Suunnittelun toteuttamisesta ja tarpeiden ymmärtämisestä vastaa scrummaster. [19, s. 8.]

Päiväpalaverissa kehitystiimi suunnittelee seuraavan vuorokauden työnsä tarkastelemalla edellisen päiväpalaverin jälkeen tehtyä työtä ja arvioimalla, mitä seuraavaan palaveriin mennessä saadaan tehtyä. Kehitystiimin jäsenet kertovat päiväpalaverissa yksittellen edellisen vuorokauden tehdyn työn tulokset, kuluvan vuorokauden aikana suunnittelemansa työn ja mahdolliset esteet tulevalle työlle. [5, s. 40–41.]

Sprintin katselmuksessa scrum-tiimi ja sidosryhmät tarkastelevat sprintin aikana tuotettua tuoteversiota ja tuotteen kehitysjono päivitetään tarpeen vaatiessa. Katselmoinnin tarkoituksena on luoda pohja tulevan sprintin suunnittelupalaverille. Osallistujat pohtivat, kuinka tuotteen arvo saadaan nostettua seuraavassa sprintissä, tarkastelevat käytössä olevien resurssien tilanteen ja keskustelevat vastaanulleista ongelmista, ja siitä, kuinka ne ratkaistiin, sekä tavoitteiden onnistumisesta. [5, s. 54–56; 19, s. 10–11.]

Retrospektiivissä scrum-tiimi tarkastelee omaa toimintaansa sprintin aikana kiinnittämällä huomiota ongelmakohtiin ja käymällä läpi onnistumiset sekä luo suunnitelma seuraavaan sprinttiin tuottavuuden parantamiseksi. Toisin sanoen, scrum-tiimi pyrkii määrittelemään keinot, joilla ryhmä pystyisi kehittämään omaa työskentelyään, välttämään todettuja ongelmakohtia ja luomaan ratkaisuja ennalta tunnistettaviin ongelmiin. [19, s. 11.]

Tuotteen ja sprintin kehitysjonot

Tuotteen kehitysjono on lista vaatimuksista, ominaisuuksista ja ideoista, joita tuotteen tulee sisältää, ja se edustaa kaikkea sitä, mitä projektiin osallistujat pitävät tärkeänä. Projektin edetessä kehitysjonoa päivitetään tarpeen mukaan. Kehitysjono on ainoa lähde, jonka perusteella tuotetta kehitetään. Se on järjestetty tärkeimpien vaatimusten perusteella siten, että arvokkaimmat ja kiireellisimmät ominaisuudet ovat etusijalla eli korkeammalla listassa. Tuoteomistaja on yksin vastuussa kehitysjonon ylläpitämisestä ja hallinnoinnista. [5, s. 32–34; 19, s. 12.]

Kehitysjonon kehitettävien kohtien lisäksi siihen listataan kaikki mahdolliset ongelmat, jotka ovat esteenä ominaisuuksien toteuttamiselle. Vaikka ongelma ei ole varsinainen kehitysjonon kohta, se on ratkaistava, ennen kuin jonkin ominaisuuden kehittäminen voidaan aloittaa. Se voi myös muuntautua omaksi kehitysjonon kohdaksi. Myös ongelmat on järjestetty kehitysjonossa tärkeyden perusteella. [5, s. 33–34.]

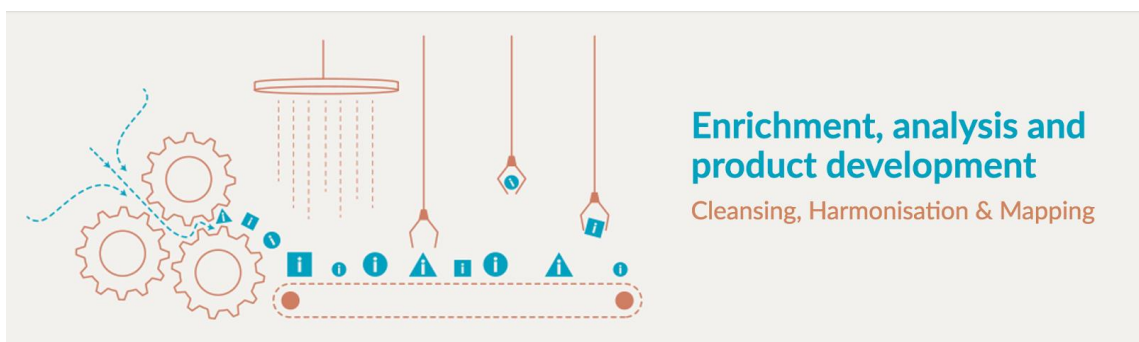
Sprintin kehitysjonoon valitaan ne tuotteen kehitysjonon kohdat, jotka aiotaan toteuttaa seuraavan sprintin aikana. Siihen kirjataan myös suunnitelma siitä, miten kehitys toteutetaan. Sprintin kehitysjono on yksityiskohtainen, jotta sen asettamien tehtävien edistyminen voidaan todentaa päivittäin. Sprintin kehitysjonoa päivitetään koko sprintin ajan, ja sen rooli on dokumentoida ja kuvastaa tehtyä työtä sprintin aikana. Kehitystiimi vastaa itse sprintin kehitysjonosta, ja se voi tarvittaessa todentaa valmiin tuoteversioon vielä tarvittava työmäärä ja täten hallita omaa työskentelyään. [19, s. 13.]

4 Projektit ja kehitysehdotukset

Tässä luvussa esitellään kaksi muotoilutoimisto Agenda Helsingin projektia. Ensimmäinen projekti on jatkokehitysvaiheessa ja tuote julkaistiin keväällä 2015. Toinen projekti on vielä kesken ja tuotteen lopullinen julkaisu tapahtuu kesällä 2016. Lisäksi luvussa esitellään kehitysehdotuksia sekä projektikohtaisesti että yleisemmällä tasolla.

4.1 Asiakastiedon sijoittajasivusto

Asiakastieto listautui pörssiin keväällä 2015. Listautumista varten Asiakastieto tilasi Agenda Helsingiltä sijoittajasivuston suunnittelun ja toteutuksen. Projekti tehtiin yhteistyössä viestintätoimisto Tekirin kanssa. Tuotteen julkaisun jälkeen projekti on jatkunut ja sivustoa on jatkokehitetty edelleen. Sijoittajasivuston lisäksi Agenda Helsinki suunnitteli ja taittoi markkinointiesitteen ja suunnitteli kuosin huiville. Kuva 4 on yksi sijoittajasivuston kuvituskuvista.



Kuva 4. Asiakastiedon sijoittajasivuston kuvituskuva [22].

4.1.1 Lähtötilanne ja asiakkaan tarpeet

Projektin tavoitteena oli suunnitella ja toteuttaa Asiakastiedolle responsiivinen sijoittajasivusto asiakkaan tarpeiden mukaan ja lainsäädännön asettamien vaatimusten perusteella annetussa määräajassa. Sivuston tarkoitus oli palvella Asiakastiedon potentiaalisia ja jo olemassa olevia sijoittajia sekä tiedottaa listautumiseen liittyvistä asioista. Julkaisu tapahtui vaiheistetuksi ennalta määrättyin välitavoittein. Projektin merkittävimmät virstanpylväät olivat ensimmäinen julkaisuajankohta, jolloin Asiakastieto ilmoitti julkisesti aikovansa listautua Helsingin pörssin päälistalle, sekä ajankohta, jolloin listautuminen tapahtui. Sivuston rakenne muuttui listautumisen alkaessa ja sen päätyttyä. Sijoittajasivusto toteutettiin kahdella kielellä, suomeksi ja englanniksi.

Tarvittavat resurssit

Sijoittajasivustoa varten tarvittiin sisällönhallintajärjestelmä, jolla asiakas pystyy helposti itse lisäämään ja muokkaamaan sisältöä tarvittaessa ilman syvempää tietämystä ohjelmoinnista tai siihen liittyvistä asioista. Sisällönhallintajärjestelmäksi valittiin WordPressin ominaisuuksien ja avoimen lähdekoodiin perusteella. WordPressiin on myös saatavilla lisäosia, joilla sisällönhallintajärjestelmään on helposti lisättävissä eri ominaisuuksia. Wordpress on ohjelmoitu PHP-ohjelmointikielellä, jonka järjestelmä käyttää JavaScriptiä ja sen jQuery-kirjastoa. Sivuston toteuttamisessa käytettiin lisäksi Bootstrap-ohjelmistokehystä, joka tarjoaa muun muassa työkaluja responsiivisuuteen, JavaScript-komponentteja ja kuvakekirjaston.

Lisäosien tarjoamia tarvittavia ominaisuuksia olivat muun muassa sisällön julkaiseminen suomeksi ja englanniksi, artikkelityyppien luominen, artikkelien lisäkentät, tietoturva, artikkelien järjestyksen muuttaminen ja Googlen analytiikkatyökalut. Lisäksi sivustolle toteutettiin rajapinta, jolla Nasdaqin järjestelmässä julkaistut tiedotteet ja osaketiedot, kuten osakkeen kurssi, hinta ja suurimmat osakkeenomistajat, esitetään sivustolla. Nasdaq on maailman suurin pörssitoimialan yritys, ja se tarjoaa kaupankäynti-, pörssiteknologia- ja listayhtiöpalveluita kuudella mantereella [23].

Toteutus ja vaiheiden esittely

Projektin voi jakaa karkeasti kolmeen osaan: kehitystyöhön ennen Asiakastiedon julkistamaa aikomusta listautua Helsingin pörssin päälisalle, kehitystyöhön listautumisen aikana ja sivuston jatkokehitysvaiheeseen. Kaikissa vaiheissa sivuston rakennetta muutettiin ja sisältöjä päivitettiin tarpeen mukaan.

Ensimmäisen vaiheen päätteeksi julkaistiin lain määäämät tiedot listautumiseen liittyen, joihin pääsyä rajattiin kaksivaiheisella lomakkeella. Ensimmäisessä vaiheessa käyttäjän oli kerrottava kansalaisuutensa ja missä maassa hän oleskeli sillä hetkellä. Jos kumpi tahansa vastauksista oli jokin lain määrittelemistä maista, joihin listautumiseen liittyviä tietoja ei saa luovuttaa, käyttäjän pääsy sivustolle estettiin. Toisessa vaiheessa käyttäjälle esitettiin vastuuvapauslauseke, jonka hyväksymällä käyttäjä pääsi tarkastelemaan listautumista koskevaa informaatiota. Ensimmäisen julkaistun version rakenne on esitetty liitteessä 1.

Toisessa vaiheessa listautuminen oli käynnissä, ja sen päätteeksi sivuston rakennetta muutettiin siten, että listautumiseen liittyvät materiaalit siirrettiin omalta sivultaan muun sisällön yhteyteen. Kolmannen eli jatkokehitysvaiheen aikana sivustolle tehtiin muutamia rakenteellisia muutoksia ja sisältöjä on päivitetty aika ajoin. Jatkokehitysvaihe on edelleen käynnissä.

4.1.2 Tulosten esittely

Sivusto kehitettiin tiiviissä yhteistyössä asiakkaan kanssa. Kehitystyöstä keskusteltiin lähes päivittäin, ja asiakas seurasi projektin etenemistä jatkuvasti, sillä aikataulu oli erittäin tiukka ja tuoteversiot oli julkaistava määräaikoihin mennessä, eikä niissä ollut varaa joutua. Kaikkiin tavoitteisiin päästiin määräaikoihin mennessä, ja Asiakastieto listautui onnistuneesti pörssiin.

Jatkokehitysvaiheessa kehitystarpeita on tullut kuukausittain. Suurin niistä on ollut pienkehitystä ja osin uudenlaisten sisältöjen julkaisemista. Myös suuressa osassa edellä mainituista asioista on ollut tarkat julkaisuajankohdat, sillä ne ovat olleet sidoksissa Asiakastiedon julkistamiin pörssitiedotteisiin.

Asiakas on ollut tyytyväinen Agenda Helsingin työskentelyyn projektissa, ja palaute on ollut positiivista. Etenkin kehitystyön joustavuuteen ja tarpeiden ennakoimiseen oltiin tyytyväisiä. Asiakkaan antamat arvosanat on esitetty taulukossa 1 aihealueittain.

Taulukko 1. Asiakkaan antama arviointi projektin osa-alueittain Asiakastiedon sijoittajasivustosta. Arviointiasteikko on yhdestä viiteen.

Osa-alue	Arvosana
Yhteistyö	5
Kommunikointi	4
Joustavuus kehityksessä	5
Tuote vastaa tarpeisiin	4
Tuotteen laatu	4
Tuote vastaa odotuksia	4
Yleisarvosana	4

Kehitystiimi oli tyytyväinen tuotteeseen etenkin, kun ottaa huomioon projektin aikataulullisen haastavuuden ja salassapitovelvollisuudesta johtuvat kehitystarpeet. Myös asiakkaan toimintaan oltiin todella tyytyväisiä ja yhteistyö sujui ongelmitta. Kehitystiimi, kuten asiakaskin, piti erityisen onnistuneena projektin osa-alueena aikataulussa pysymisen ja tarpeiden muuttumiseen reagoimisen. Kehitystiimin projektin osa-alueille antamien arvosanojen keskiarvo on esitetty taulukossa 2.

Taulukko 2. Kehitystiimin antama arviointi projektin osa-alueittain Asiakastiedon sijoittajasivustosta. Arviointiasteikko on yhdestä viiteen.

Osa-alue	Arvosana
Yhteistyö asiakkaan kanssa	4,5
Yhteistyö kehitystiimissä	4,5
Kommunikointi asiakkaan kanssa	4,5
Kommunikointi kehitystiimissä	4,5
Joustavuus kehityksessä	5
Tuote vastaa tarpeisiin	5
Tuotteen laatu	4,5
Tuote vastaa odotuksia	5
Yleisarvosana	5

Asiakkaan ja kehitystiimin antamista arvioinneista voi päätellä, että projekti on sujunut tavoitteiden mukaisesti ja työskentelyyn on oltu tyytyväisiä. Kaikkien kolmen vaiheen tavoitteisiin päästiin määräaikoihin mennessä, ja tuoteversiot ovat vastanneet asiakkaan tarpeita.

4.2 Helsingin kaupungin perusopetuksen opetussuunnitelman julkaisualusta

Helsingin kaupungin opetusvirasto tilasi Agenda Helsingiltä Helsingin koulujen perusopetuksen uuden opetussuunnitelman julkaisualustan keväällä 2015. Agenda Helsinkiä pyydettiin suunnittelemaan palvelu, sen ilme ja tekninen toteutus. Projekti on vielä käynnissä tämän opinnäytetyön julkaisun aikana. Kuva 5 on yksi opetussuunnitelman kuvituskuvista.



Kuva 5. Opetussuunnitelman kuvituskuva [24].

4.2.1 Lähtötilanne ja asiakkaan tarpeet

Aiemmin Helsingin kaupungin ja sen peruskoulujen opetussuunnitelmat oli julkaistu PDF-dokumentteina, joten useiden dokumenttien lukeminen ja selailu oli vaivalloista. Tilalle haluttiin moderni verkkopalvelu, jossa opetussuunnitelmien julkaisu ja lukeminen olisi mahdollisimman käyttäjäystävällistä. Kunnan oli pystyttävä vaivattomasti syöttämään oman opetussuunnitelmansa sisältö sivustolle, ja koulujen oli pystyttävä tekemään lisäyksiä ja muutoksia kunnan luomiin sisältöihin. Vaatimuksena oli sivuston toimiminen kaikilla moderneilla laitteilla ja selaimilla. Osa Helsingin yli sadasta peruskouluista on ruotsinkielisiä, ja ruotsinkielinen opetussuunnitelma poikkeaa hieman suomenkielisestä,

joten palvelusta tuli tehdä myös ruotsinkielinen versio. Lisäksi Helsingin kaupunki julkaisee kaiken sisältönsä molemmilla virallisilla kielillä. Opetussuunnitelmien lisäksi sivustolla on myös koulujen yhteystiedot ja kaupungin julkaisemia artikkeleita opetuksesta.

Opetussuunnitelma on jaettu kolmeen osaan: yhteiseen osuuteen, vuosiluokkakokonaisuuksiin ja oppiaineisiin. Yhteisessä osuudessa kuvataan opetusta Helsingin peruskouluissa yleisellä tasolla. Lukujen sisältö on kunnan tuottamaa. Näihin lukuihin koulut voivat tehdä omat lisäyksensä, joissa kuvataan luvun käsittelemän aiheen toteutusta kyseisessä koulussa. Vuosiluokkakokonaisuuksissa opetusta kuvataan vuosiluokkakokonaisuuksittain, jotka ovat vuosiluokat 1–2, 3–6 ja 7–9. Myös vuosiluokkakokonaisuuksiin koulut voivat tehdä samanlaiset tarkennuksensa kuin yhteiseen osuuteen. Oppiaineissa kuvataan oppiainekohtainen opetus. Sekä kaupunki että koulut voivat luoda, muokata ja poistaa oppiaineita. Lisäksi koulut voivat poistaa kaupungin tekemiä oppiaineita omasta opetussuunnitelmastaan.

Tarvittavat resurssit

Opetussuunnitelman julkaisualustan sisällönhallintajärjestelmäksi valittiin Wordpressin ominaisuuksien ja laajennettavuuden perusteella. Lisäosilla toteutettavia laajennuksia olivat kaksikielisyys, tietoturva, käyttäjähallinta, rekisteröityminen ja käyttäjätietojen laajennus, lisäkentät, julkaisujen uudelleen järjestäminen ja tietokantatyökalut. Lisäksi järjestelmään toteutettiin koulujen lisäysten muotoilu, dokumenttien lataaminen palveluun, oppiaineiden luominen, poistaminen, kopioiminen ja piilottaminen lukunäkymästä sekä koulujen omien tietojen hallinta. Alustan sisältö syötetään koulujen osalta kokonaan palvelun käyttöliittymästä, joten kaikkien sisällön syöttöön liittyvien Wordpressin omien toimintojen rinnalle toteutettiin omat ratkaisut.

Wordpressin lisäosien ohella palvelussa otettiin käyttöön muita työkaluja. Niitä ovat muun muassa Bootstrap, jQuery ja sen laajennus, Trumbowyg-Javascript-muokkain, Javascript-tavutuskirjasto, fonttikirjasto, Autocomplete-Javascript-kirjasto hakutoiminnon ennakoivaan hakuun, Slick-kuvakaruselli ja Lazyload-kirjasto kuvien asteittaiseen lataamiseen.

Toteutus ja vaiheiden esittely

Julkaisualustaa lähdettiin kehittämään keväällä 2015, jolloin päämääränä oli luoda intuitiivinen ja helppokäyttöinen alusta Helsingin kaupungin ja sen koulujen perusopetuksen tarpeisiin. Palvelun kohderyhmäksi määriteltiin Opetusviraston työntekijät, koulujen opettajat ja oppilaiden vanhemmat. Palvelun julkaisu päätettiin tehdä kolmessa vaiheessa. Ensimmäinen julkaisu tapahtui Helsingin kaupungin tehtyä oman opetussuunnitelmansa ja kehitystyön ollessa siinä vaiheessa, että kaupunki pääsi aloittamaan omien sisältöjensä syöttämisen palveluun. Toisen julkaisun jälkeen, kaupungin syötettyä kaiken oman opetussuunnitelmansa suomenkielisen sisällön, koulut rekisteröityivät palveluun vaiheittain. Koulut jaettiin ryhmiin, jotka vuorollaan rekisteröityivät palveluun ja aloittivat omien sisältöjensä syöttämisen. Tämän opinnäytetyön kirjoittamisen aikaan koulujen sisällön-syöttövaihe on vielä käynnissä. Kolmas julkaisu tapahtuu, kun kaupunki ja koulut ovat lisänneet kaiken sisältönsä molemmilla kielillä ja julkaisualusta on optimoitu sekä pienkehitys on päättynyt.

Koulut suunnittelivat ja kirjoittivat omia opetussuunnitelmiaan samanaikaisesti kehitystyön kanssa ennen ensimmäistä julkaisua. Sovelluskehitys aloitettiin mahdollisimman varhaisessa vaiheessa, kun ensimmäiset versiot sivuston rakennesuunnitelmasta ja käyttöliittymästä olivat valmiit. Myös palvelun graafinen ilme kehitettiin samanaikaisesti.

Projektin aikana asiakkaan tarpeet ovat muuttuneet useasti, ja palvelu on määritelty uudelleen kahdesti. Koska koulut ovat saaneet työstää omia sisältöjään haluamallaan tavalla, tuki erilaisille sisällöille ja niiden rakenteelle on tullut välttämättömäksi. Lisäksi palveluun on lisätty lukuisia tarpeellisia toimintoja, joita ei alkuperäisessä määrittelyssä ollut. Muutoksia on tullut muun muassa koulujen lisäysten sisällön tyypeihin, opetussuunnitelman rakenteeseen, oppiaineiden rakenteeseen ja oppiaineiden käyttäjäkohtaiseen hallinnointiin. Koska opetussuunnitelmasta tuli huomattavasti laajempi, kuin projektin alussa arvioitiin, myös sisältöjen hakemiseen palvelimelta on tullut muutoksia.

Koska tarvittavien ominaisuuksien määrä kasvoi todella paljon, muutosten toteuttamisen vaatiman kehitystyön määrä kasvoi moninkertaiseksi. Myös projektin aikataulu muuttui ja jouduttiin määrittelemään uudestaan. Osa projektin alussa testaukseen ja siinä ilme-neviin kehitystarpeisiin varatusta ajasta jouduttiin täten käyttämään sovelluskehitykseen.

4.2.2 Tulosten esittely

Huolimatta tuotteen ominaisuuksien laajentumisesta ja asiakkaan tarpeen muuttumisesta sekä sen myötä muuttuneesta ja tiukentuneesta aikataulusta, projekti on pääpiirteittäin sujunut hyvin ja muutostarpeisiin on pystytty reagoimaan uuden aikataulun mukaisesti. Ongelmia kuitenkin esiintyi, kun koulujen sisällön syöttäminen palveluun alkoi. Koulujen käyttämän tuoteversion testaukseen varattu aika jäi liian lyhyeksi, eikä käyttäjä ja palvelun kuormitustestejä ehditty tekemään riittävästi. Tämän vuoksi palvelun käyttäminen hidastui huomattavasti. Koulujen sisällön syöttämiseen liittyvät ominaisuudet toimivat kuitenkin nopeammin kuin muu palvelu, sillä nämä olivat tärkeimmän prioriteetin omaavia kehityskohteita ennen tuoteversion julkaisua, ja niiden kehittämiseen keskityttiin sen mukaisesti. Palvelua saatiin nopeutettua lisäämällä palvelimen resursseja ja optimoimalla kirjoitettua koodia tietokantahakujen tehostamiseksi. Koulut saivat syötettyä sisältönsä palveluun pääpiirteittäin hyvin ja sivusto toimi tältä osin riittävän tehokkaasti.

Asiakas oli tyytyväinen projektin osa-alueista etenkin kommunikointiin ja muutostarpeisiin vastaamiseen. Lisäksi erityistä kiitosta sai palvelun käyttöönottovaiheen eli koulujen sisältöjen syöttämisen aikainen tuki. Käyttöönottovaiheen ongelmista huolimatta myös muuhun työskentelyyn oltiin pääpiirteittäin tyytyväisiä, ja projektin arvioitiin sujuneen hyvin. Asiakkaan projektin osa-alueille antamat arvosanat on esitetty taulukossa 3.

Taulukko 3. Asiakkaan antama arviointi projektin osa-alueittain opetussuunnitelman julkaisualustasta. Arvosteluasteikko on yhdestä viiteen.

Osa-alue	Arvosana
Yhteistyö	4
Kommunikointi	5
Joustavuus kehityksessä	4
Tuote vastaa tarpeisiin	4
Tuotteen laatu	3,5
Tuote vastaa odotuksia	4
Käyttöönottovaiheen tuki	5
Yleisarvosana	4

Kehitystiimissä projektin ajateltiin sujuneen hyvin. Haasteita asettivat tuotteen määrittelyn ja asiakkaan tarpeiden muuttuminen projektin aikana. Myös projektin kokoluokka koettiin haastavaksi sekä toteutuksen että projektinhallinnan osalta. Muutostarpeisiin reagoimiseen ja yhteistyöhön, sekä asiakkaan kanssa, että kehitystiimin sisällä, oltiin tyytyväisiä. Kehitystiimi oli erityisen tyytyväinen oman osaamisensa ja työskentelynsä kehittämiseen etenkin suunnittelun ja teknisen toteutuksen osalta. Kehitystiimin projektin osa-alueille antamat arvosanat on esitetty taulukossa 4.

Taulukko 4. Kehitystiimin antama arviointi projektin osa-alueittain opetussuunnitelman julkaisualustasta. Arviointiasteikko on yhdestä viiteen.

Osa-alue	Arvosana
Yhteistyö asiakkaan kanssa	4
Yhteistyö kehitystiimissä	4,5
Kommunikointi asiakkaan kanssa	4
Kommunikointi kehitystiimissä	4,5
Joustavuus kehityksessä	5
Tuote vastaa tarpeisiin	5
Tuotteen laatu	4,5
Tuote vastaa odotuksia	4,5
Yleisarvosana	4,5

Asiakkaan ja kehitystiimin antamista arvoista voidaan päätellä projektin sujuneen tähän mennessä pääpiirteittäin hyvin. Haasteista huolimatta sekä asiakkaan että kehitystiimin työskentelyn arvioitiin onnistuneen ja tuotteen vastaavan valmiilta osiltaan asiakkaan tarpeita. Myös tuoteversion käyttöönoton ajateltiin sujuneen pääosin hyvin, ja siinä ilmenneet ongelmat saatiin ratkaistua.

4.3 Kehitysehdotukset

Asiakastiedon sijoittajasivusto -projektin esiin nostamat kehitysehdotukset

Asiakastiedon pörssiin listautumisesta opittiin paljon projektin aikana. Listautumisen vaiheet ja sijoittajasivuilta vaaditut ominaisuudet tulivat tutuksi projektiin osallistuneille. Laki asettaa tietyt tarkat vaatimukset ja määräajat, joita on noudatettava listautumisprosessin ajan. Kun yritys listautuu pörssiin, projektin kulku on siis samankaltainen. Yrityksen listautuminen pörssin pää- tai jollekin muista listoista asettaa hieman eri vaatimustasot sijoittajasivuille ja muille listautumiseen liittyville tuotteille, kuten esimerkiksi esitteelle.

Koska Agenda Helsingillä on nyt kokemusta yritysten listautumiseen liittyvistä toimenpiteistä ja listautumisprosessista kokonaisuutena, projektin kulku on paremmin ennustettavissa. Tämä mahdollistaa entistä tarkemman tuotteen vaatimusten määrittelyn jo projektin alussa. Myös tuoteversioiden vaatimien ominaisuuksien kehittämissaikataulu voidaan suunnitella vielä aiempaa tarkemmin. Muutoksia kuitenkin tulee kaikissa projekteissa, eikä yrityksen listautuminen ole poikkeus, joten niihin voidaan valmistautua ja reagoida vielä aiempaa tehokkaammin. Etenkin listautumisprojekteissa osa muutoksista voi tulla todella lyhyellä varoitusajalla. Tämä on otettava huomioon sekä suunnittelussa että teknisessä toteutuksessa.

Listautumisprojekteissa työskentelyä voi kehittää vielä tehokkaammaksi ottamalla käyttöön jonkin ketterän kehitysmenetelmän, esimerkiksi scrumin. Scrumin avulla on mahdollista tehostaa entisestään työskentelyä, ja tiukan aikataulun asettama eri ominaisuuksien tarkka priorisointi ja sen muuttaminen tarvittaessa on mahdollista [5, s. 35]. Kun julkaisuajankohdat lähestyvät, tällä voidaan varmistaa, että kaikki prioriteetiltaan tärkeät ominaisuudet on tehty ja myös asiakas on yhtä mieltä siitä, mitä ne ovat.

Helsingin kaupungin opetussuunnitelman julkaisualusta -projektin esiin nostamat kehitysehdotukset

Helsingin kaupungin perusopetuksen opetussuunnitelman julkaisualusta on Agenda Helsingin tähänastisista projekteista suurin. Palveluun kirjautuneita käyttäjiä eli kouluja on yli 100 ja muita käyttäjiä tuhansia. Myös rakenteensa puolesta palvelu on ollut erittäin haastava toteuttaa. Lisäksi koulujen palveluun syöttämän datan määrä on yllättävän

suuri ja palvelua käytettäessä tietokantakyselyjen määrä etenkin kirjautuneelle käyttäjälle kasvoi ominaisuuksien laajentuessa, mikä asetti korkeat vaatimukset tekniselle toteutukselle.

Suurimpana haasteena on kuitenkin ollut asiakkaan tarpeiden jatkuva muuttuminen ja laajentuminen. Koska uusia ominaisuuksia ilmeni projektin edetessä ja edellisiä jouduttiin määrittelemään ja toteuttamaan uudelleen, niiden testaamiselle varattu aika lyhyeksi. Palvelulle ei myöskään ehditty tekemään kuormitustestausta, jonka avulla ongelmia olisi voitu havaita etukäteen.

Jatkossa vastaavissa projekteissa olisi hyvä, jos jo projektin varhaisessa vaiheessa olisi tarkemmin selvillä, mitä lähdetään tekemään. Ominaisuudet pitäisi pystyä priorisoimaan paremmin tärkeisiin ja vähemmän tärkeisiin, jotta tärkeimmät ominaisuudet saadaan varmasti julkaistua ajoissa. Jos kehitystiimi huomaa asiakkaan tarpeiden muuttuvan jatkuvasti, asiakasta pitää yrittää auttaa selvittämään tarkemmin tavoitteitaan ja priorisoimaan tuotteen ominaisuuksia.

Tässä projektissa ketteristä kehitysmenetelmistä etenkin scrum olisi ollut hyödyllinen työkalu. Vaikka mittaustulosten perusteella asiakas ja kehitystiimi ovat olleet tyytyväisiä erityisesti joustavaan työskentelyyn, projektissa työskentely olisi voinut olla hallitumpaa. Scrumin käyttäminen olisi pakottanut kaikki projektin työntekijät määrittelemään tarkemmin tarvittavat ominaisuudet ja jättämään vähemmän tärkeät vähemmälle huomiolle. Scrum olisi myös mahdollistanut kehitystiimin hallitumman työskentelyn.

Vaikka kehitettävää tämän projektin työskentelyssä on jonkin verran, suurin osa projektista sujui hyvin ja muutoksiin pystyttiin reagoimaan riittävän tehokkaasti. Tähän mennessä palveluun kirjautuneet koulut ovat saaneet sisältöjään syötettyä hyvin. Palvelua on osattu myös käyttää pääosin hyvin huolimatta käyttäjien hyvin erilaisesta tietotekniikan tunteuksesta ja taitotasosta.

Yleisellä tasolla

Koska jokainen sovelluskehitysprojekti on aina erilainen kuin muut, projektin kulkua ja sen eri vaiheita on mahdotonta ennustaa tarkasti etukäteen [5, s. 23–24]. Tämä johtaa väistämättä asiakkaan tarpeen ja sitä kautta myös tuotteen muutoksiin, joihin on kyettävä reagoimaan niin hyvin kuin mahdollista vallitsevissa olosuhteissa. Etenkin pitkissä ja laajoissa projekteissa, joissa on mukana paljon ihmisiä, on niin paljon muuttujia, että niiden hallitseminen on erittäin haastavaa. Lisäksi vaatimusten lista kasvaa jatkuvasti ja eri henkilöillä on erilaiset mielipiteet siitä, minkälainen tuote pitäisi olla.

Kaiken edellä mainitun hallitsemiseksi on enemmän kuin suotavaa omaksua ketterien kehitysmenetelmien perusteet ja kokeilla niitä käytännössä. Vaikka Agenda Helsinki on tällä hetkellä mikroyritys ja projektityöskentely täten hyvin hallittavissa, esimerkiksi scrumin käyttöönotto projekteissa toisi lisää tehokkuutta työskentelyyn. Sen käyttöönotto tehostaisi koko yrityksen resurssien käyttöä ja auttaisi myös asiakkaita hahmottamaan paremmin projektin kulkua. Scrumin käyttö projekteissa myös kehittäisi yksittäisen työntekijän työskentelyä ja oman työn arvioimista. Kehitystiimin taidot kasvaisivat projekti projektilta, ja tätä prosessia olisi helpompi tarkastella ja arvioida. Menneiden projektien prosessien kautta on helpompi arvioida tulevien projektien etenemistä, ennaltaehkäistä ongelmia ja reagoida arvaamattomiin muutoksiin.

Vaikka työskentely sujuu nykyisellään erinomaisesti, yrityksen kasvaessa projektinhallinnan asettamat haasteet kasvavat samassa suhteessa. Etenkin isoissa projekteissa on Agenda Helsingin lisäksi mukana paljon eri henkilöitä ja muita muuttujia, jotka asettavat sovelluskehityksen projektinhallinnan uudelle tasolle. Helpoin tapa hallita isoja kokonaisuuksia on jakaa ne pienempiin osiin. Ketterät kehitysmenetelmät on kehitetty juuri tätä varten, ja niiden käyttöönotto on helppoa. Esimerkiksi scrum sopii erinomaisesti kaikenkokoisiin projekteihin ja saattaisi osittain juuri skaalautuvuutensa perusteella olla yksi parhaista vaihtoehdoista projektityöskentelyn kehittämiseksi Agenda Helsingissä.

5 Yhteenveto

Projektinhallinta on haastavaa kaikissa projekteissa, ja mitä isommasta projektista on kyse, sitä vaikeampaa se on. Koko projekti koostuu lukemattomista muuttujista, joista osaan voi vaikuttaa ja osaan ei. Tämä johtaa väistämättä ennalta arvaamattomien tilanteiden ja ongelmien syntymiseen, joihin on pystyttävä reagoimaan niin tehokkaasti, etteivät ne vaikuta kriittisesti tuotteeseen tai mihinkään sen ominaisuuksista. Omat haasteensa projekteihin tuovat myös henkilöiden välinen kanssakäyminen ja kommunikointi kaikkien osapuolien välillä.

Ketterät kehitysmenetelmät tarjoavat isojen ja monimutkaisten kokonaisuuksien hallitsemiseen työkaluja, joilla koko projektin saa jaettua helpommin hallittaviin kokonaisuuksiin. Lähes mihin tahansa muutokseen pystytään reagoimaan kaikissa projektin vaiheissa ilman muun kehitystyön merkittävää kärsimistä, eikä projektia tarvitse keskeyttää osittaisien muutosten takia. Resurssien käyttö on hallitumpaa ja projekteissa työskentelystä opitaan tehokkaasti.

Vaikka kummassakaan tässä työssä esitellyistä projekteista ei käytetty määriteltyjä ketteriä kehitysmenetelmiä, monet Agenda Helsingin projektinhallinnan osat ovat ketteriä sanan varsinaisessa merkityksessä ja muutoksiin kyettiin reagoimaan tehokkaasti. Asiakastiedon sijoittajasivuston vaiheet ennen jatkokehitystä asettivat projektinhallinnalle todellisia haasteita, joista kaikista selvittiin mallikkaasti. Helsingin kaupungin opetussuunnitelman julkaisualustan kehittämisesä jokin ketteristä kehitysmenetelmistä olisi todennäköisesti vähentänyt jonkin verran ylimääräistä työtä ja tuotteen ominaisuudet olisi saatu asiakkaan kanssa määriteltyä ja etenkin priorisoitua tarkemmin ennen niiden kehittämistä.

Ketterien kehitysmenetelmien käyttäminen sovelluskehitysprojekteissa on siis enemmän kuin suotavaa, sillä se tehostaa työskentelyä ryhmässä, mikä johtaa parempien tuotteiden julkaisuun ja kehittää samalla myös itse työprosessia. Lisäksi asiakkuuksien hallitseminen on helpompaa ja riskejä pystytään minimoimaan tehokkaasti. Voikin todeta, että on hyvin vaikeaa löytää yhtäkään todellista syytä, miksi missä tahansa yrityksessä ei kannattaisi ottaa käyttöön ja soveltaa tarvittaessa jotakin ketterää kehitysmenetelmää.

Lähteet

- 1 Maylor, Harvey. 2010. Project Management. Edinburgh Gate: Pearson Education.
- 2 Why Project Management Is Different to Operations Management. 2016. Verkkodokumentti. Leadership Thoughts. <<http://www.leadershipthoughts.com/difference-between-project-management-and-operations-management/>>. Päivitetty 3.4.2016. Luettu 28.3.2016.
- 3 Clark, Wallace. 1923. The Gantt Chart. New York: The Ronald Press Company.
- 4 Bowen, Ronda. 2011. Common Project Management Charts. Verkkodokumentti. <<http://www.brighthubpm.com/project-planning/53691-common-project-management-charts/>>. Päivitetty 19.5.2011. Luettu 28.3.2016.
- 5 Schwaber, K., Beedle, M. 2002. Agile Software Development with Scrum. Upper Saddle River: Prentice Hall.
- 6 Storbacka, Kaj. 2003. Asiakkuuden arvon lähteillä. Helsinki: WSOY.
- 7 To Agility and Beyond: The History and Legacy of Agile Development. 2015. TechBeacon. <<http://techbeacon.com/agility-beyond-history%E2%80%94legacy%E2%80%94agile-development>>. Päivitetty 26.8.2015. Luettu 30.3.2016
- 8 Agile Development - A Brief History. 2013. Verkkodokumentti. VersionOne. <http://www.agilesherpa.org/intro_to_agile/a_brief_history_of_agile/>. Päivitetty 18.10.2013. Luettu 30.3.2016.
- 9 Andrei, Kyle. 2014. Tools for Gantt Charts. Verkkodokumentti. Idealware. <<http://www.idealware.org/blog/tools-gantt-charts>>. Päivitetty 17.7.2014. Luettu 28.3.2016.
- 10 Johdatus tietojärjestelmiin. 2006. Verkkodokumentti. Oulun seudun ammattiotopisto. <http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elinkaarimallit_asia.htm>. Päivitetty 23.2.2006. Luettu 30.3.2016.
- 11 Manifesto for Agile Software Development. 2001. Verkkodokumentti. Agile Alliance. <<http://agilemanifesto.org/> Luettu 30.3.2016>. Päivitetty 14.4.2016. Luettu 30.3.2016.
- 12 Cockburn, Alistair. 2007. Agile Software Development. Second Edition. The Cooperative Game. Boston: Pearson Education, Inc.

- 13 Comparing Traditional System Analysis and Design with Agile Methodologies. 2009. Verkkodokumentti. University of Missouri. <<http://www.umsl.edu/~hugh-eyd/is6840/waterfall.html>>. Päivitetty 9.11.2009. Luettu 28.3.2016.
- 14 Dybå, T., Dingsøy, T. 2008. Empirical studies of agile software development: A systematic review. Verkkodokumentti. ScienceDirect. <https://www.researchgate.net/profile/Torgeir_Dingsoyr/publication/222827396_Empirical_studies_of_agile_software_development_A_systematic_review/links/0fcfd506ec26a17395000000.pdf>. Päivitetty 24.1.2008. Luettu 31.3.2016.
- 15 Voigt, Benjamin J. J. 2004. Dynamic System Development Method. Verkkodokumentti. University of Zurich. <https://files.ifi.uzh.ch/roerg/arvo/courses/seminar_ws03/14_Voigt_DSMD_Ausarbeitung.pdf>. Päivitetty 20.1.2004. Luettu 31.3.2016.
- 16 Feature Driven Development (FDD) and Agile Modeling. 2014. Verkkodokumentti. Agile Modeling. <<http://www.agilemodeling.com/essays/fdd.htm>>. Päivitetty 9.10.2014. Luettu 31.3.2016.
- 17 What is Lean? 2016. Verkkodokumentti. Lean Enterprise Institute. <<http://www.lean.org/WhatsLean/>>. Päivitetty 11.4.2016. Luettu 31.3.2016.
- 18 Wells, Don. 2013. Extreme Programming: A gentle introduction. Verkkodokumentti. Don Wells. <<http://www.extremeprogramming.org/>>. Päivitetty 8.10.2013. Luettu 31.3.2016.
- 19 Schwaber, K., Sutherland, J. 2014. The Scrum Guide. Verkkodokumentti. Scrum.org. <<https://scrumwell.files.wordpress.com/2014/03/scrum-guide-2013-fi-v1-1.pdf>>. Päivitetty 14.3.2014. Luettu 28.3.2016.
- 20 The History of Scrum. 2014. Verkkodokumentti. Scrum Guides. <<http://www.scrumguides.org/history.html>>. Päivitetty 26.9.2014. Luettu 28.3.2016.
- 21 Scrum Team. 2016. Verkkodokumentti. Mountain Goat Software. <<https://www.mountaingoatsoftware.com/agile/scrum/team>>. Päivitetty 11.4.2016. Luettu 28.3.2016.
- 22 Leskinen, Tommi. 2015. Agenda Helsinki. Verkkodokumentti. <<http://investors.asiakastieto.fi/>>. Päivitetty 11.4.2016. Luettu 10.4.2016.
- 23 Tietoa pörssistä. 2016. Verkkodokumentti. Nasdaq Helsinki Oy. <<http://www.nasdaqomxnordic.com/tietoaporssista>>. Päivitetty 11.4.2016. Luettu 8.4.2016.
- 24 Leskinen, Tommi. 2016. Suunnittelija. Agenda Helsinki. 10.4.2016.

Asiakastiedon sijoittajasivuston rakenne listautumista varten

Asiakastiedon sijoittajasivuston rakenne suunniteltuna ennen listautumista. Rautalankamallissa on kuvattuna sekä sivuston rakenne että alkuvaiheessa suunnitellut yksittäisten sivujen sisällöt.

