
ITERATIIVINEN PROJEKTIMALLI VERKKOPALVE- LUPROJEKTEISSA

Case Ambientia Oy



Ammattikorkeakoulututkinnon opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Visamäki 11.6.2010

Henna Kautto



Tietojenkäsittelyn koulutusohjelma
Hämeenlinna

Työn nimi Iteratiivinen projektimalli verkkopalveluprojekteissa
Case Ambientia Oy

Tekijä Henna Kautto

Ohjaava opettaja Lasse Seppänen

Hyväksytty _____._____.20____

Hyväksyjä

HÄMEENLINNA
Tietojenkäsittelyn koulutusohjelma
Systeemityö

Tekijä	Henna Kautto	Vuosi 2010
Työn nimi	Iteratiivinen projektimalli Case Ambientia Oy	verkkopalveluprojekteissa

TIIVISTELMÄ

Opinnäytetyö tehtiin Ambientia Oy:lle, joka on sähköiseen liiketoimintaan ja viestintään sekä yhteisöllisiin ratkaisuihin erikoistunut yritys. Ambientia suunnittelee ja toteuttaa räätälöityjä verkkosovelluksia ja -palveluita. Ambientia on kehittänyt oman projektimallin, joka pohjautuu ketteriin menetelmiin. Opinnäytetyössä haluttiin selvittää, toimiiko projektimalli projekteissa käytännön tasolla ja mitä mieltä työntekijät ovat projektimallista.

Työn teoriaosuuteen saatiin aineistoa tutustumalla Ambientian dokumentoimaan projektimalliin. Muita prosessimalleja tutkittiin kirjallisuuslähteiden avulla ja prosessimallien ominaisuuksia vertailtiin toisiinsa. Tietoa projektimallin toimivuudesta käytännön työssä saatiin suorittamalla lomakekysely Ambientian työntekijöille ja haastatteleamalla kolmea sovelluskehittäjää.

Oman projektimallin kehittäminen ja sen käyttäminen projekteissa on ollut tärkeä asia Ambientian projektien läpiviennin kannalta. Ilman projektimallia olisi vaikeampaa ohjata projekteja alusta loppuun sovituisissa kustannuksissa ja aikataulussa. Mallia tulisi kuitenkin kehittää ja muokata niin, että sitä voisi hyödyntää erityyppisissä projekteissa. Mallin toteutumista projekteissa tulisi valvoa, jotta malli ei jäisi vain ohjeistukseksi. Projektimallin kehittämiseen tulisi ottaa työntekijöitä eri projektiryhmistä mukaan. Lisäksi projektimallia tulisi käydä työntekijöiden kanssa läpi, jotta malli tulisi jokaiselle tutuksi ja osaksi työntekoa.

Avainsanat prosessimalli, iteratiivinen projektimalli, ketterät menetelmät, verkkopalvelu.

Sivut 30 s. + liitteet 14 s.

HÄMEENLINNA

Degree Programme in Business Information Technology
Software Engineering

Author

Henna Kautto

Year 2010

Subject of Bachelor's thesis

Iterative Project Model in Web Service Projects
Case Ambientia Oy

ABSTRACT

This thesis was commissioned by the Ambientia Oy which specialises in electronic business, communication services and social software solutions. Ambientia designs and implements websites and various web-applications tailored for specific needs. Ambientia has created a project management model of its own. This model is based on agile software development. The aim of the thesis was to explore how Ambientia's iterative project management model works in web service projects in practice and what the employees think about the model.

Background information was gleaned from the documentation of the project model documented by Ambientia. Other software development methodologies were explored from literature. Information about the functionality of Ambientia's project model in practise was gathered by a web-based questionnaire which was sent to all the employees of Ambientia. Three of Ambientia's software developers were interviewed. In addition, properties of the project model were compared to other software development methodologies.

Developing a project model of its own and using it in projects have been an important thing to Ambientia's projects. Without the project model it would be harder to control projects within fixed budgets and schedules. The model should be developed and modified so that it could be used in different kind of projects. The implementation of the model in projects should be monitored so that it wouldn't remain on the instruction level only. Employees from different projects groups should be taken along to developing the model. In addition, the project model should be processed with the employees so that the model would become familiar to everyone and it could be part of everyday working.

Keywords software development methodology, iterative project management model, agile software development, web service.

Pages 30 p. + appendices 14 p.

SISÄLLYS

1	JOHDANTO.....	1
2	PROSESSIMALLIT	2
2.1	Vesiputousmalli.....	2
2.2	Prototyypimalli.....	4
2.3	Spiraalimalli	5
2.4	V-malli	6
2.5	Ketterät menetelmät	7
2.5.1	XP.....	8
2.5.2	Scrum.....	9
2.5.3	RUP	12
3	AMBIENTIA OY:N PROJEKTIMALLI.....	13
3.1	Vaiheet	14
3.2	Iterointiprosessi	16
4	PROSESSIMALLIEN VERTAILU	17
5	VERKKOSOVELLUKSEN SUUNNITTELEMINEN	22
6	LOMAKEKYSELYN TULOKSET.....	23
7	TEEMAHAASTATTELUIJEN TULOKSET	25
8	YHTEENVETO JA JATKOSUOSITUKSET	27

LÄHTEET

LIITE 1	Lomakekyselyn saatekirje
LIITE 2	Lomakekyselyn vastaukset
LIITE 3	Teemahaastattelu Heikki Malkamäki
LIITE 4	Ote Jari Saukkosen teemahaastattelusta
LIITE 5	Teemahaastattelu Tuomas Martinkallio

1 JOHDANTO

Ketterät menetelmät ovat nousseet suosituksi vaihtoehdoksi mietittäessä verkkopalveluprojekteissa käytettävää prosessimallia. Asiakas haluaa nähdä toteutettavan sovelluksen konkreettisena jo projektin varhaisessa vaiheessa, jotta hän saisi oikean kuvan siitä, millaista sovellusta projektissa ollaan toteuttamassa. Näin vältetään siltä, että sovelluksen toimittajan ja asiakkaan näkemykset eroavat toisistaan ja lopputulos ei miellytä asiakasta.

Ketterät menetelmät rohkaisevat ottamaan asiakkaan tiiviisti mukaan projektiin ja kommunikoimaan suoraan asiakkaan ja projektiryhmän välillä. Ne sallivat nopeat muutokset projekteissa, sillä kehitystyötä tehdään lyhyissä aikajaksoissa, iteraatioissa, mikä mahdollistaa muutosten tekemisen hallitusti. Kehitteillä oleva sovellus ja sen toimivuus on tärkeää. Muun muassa nämä ominaisuudet erottavat ketterät menetelmät perinteisistä prosessimalleista.

Ambientia Oy on sähköiseen liiketoimintaan, viestintään ja yhteisöllisiin ratkaisuihin erikoistunut yritys. Yritys työllistää noin 70 henkilöä viidellä eri paikkakunnalla. Ambientia hyödyntää projekteissaan kehittämäänsä iteratiivista projektimallia, joka pohjautuu ketteriin menetelmiin. Opinnäytetyössä on tarkoitus tutustua tähän malliin. Mallin nimi on Ambientian iteratiivinen projektimalli, joten opinnäytetyössä puhutaan tämän mallin osalta projektimallista ja muista malleja käsiteltäessä prosessimalleista.

Opinnäytetyössä keskitytään seuraaviin tutkimuskysymyksiin:

- Miten Ambientian projektimalli eroaa muista prosessimalleista?
- Miten verkkosovellus suunnitellaan Ambientian iteratiivisen projektimallin avulla?
- Miten Ambientia projektimalli toimii projekteissa käytännössä?

Opinnäytetyössä tutustutaan erilaisiin prosessimalleihin ja niiden ominaisuuksiin. Tämän lisäksi vertaillaan, miten Ambientian iteratiivinen projektimalli eroaa muista malleista. Vertailussa on mukana sekä perinteisempiä prosessimalleja että uudempia ketteriä menetelmiä. Tietoa verkkosovelluksen toteutuksesta hankitaan projektimallin vaihetuotosten avulla, jotka dokumentoidaan. Tällä tavoin saadaan tietoa siitä, miten projekti etenee projektimallin avulla. Tietoa siitä, miten Ambientian projektimallia käytetään todellisuudessa ja miten projektimallia voitaisiin kehittää, saadaan lomakekyselyn ja teemahaastattelujen perusteella.

2 PROSESSIMALLIT

Ohjelmistokehityksen avuksi on kehitetty erilaisia prosessimalleja, jotka ovat mahdollisimman yleisesti sovellettavissa olevia ohjeistoja sovellusten ja verkkopalveluiden tuottamiseen. Niiden tarkoitus on hallita ja organisoida projektia niin, että projekti saataisiin päätettyä sovitussa ajassa ja budjetissa. Prosessimallit auttavat toteuttamaan projektit paremmin, nopeammin ja tehokkaammin. (Murch 2002, 57–58.)

Yritys saa lisäarvoa toimivasta prosessimallista, sillä se ehkäisee kehitystyön riskejä, lisää tehokkuutta ja auttaa vähentämään päällekkäisyyksiä. Näiden lisäksi se vähentää kustannuksia, lyhentää iteraatiokierroksia, parantaa ohjelmistojen laatua, auttaa projektipäälliköitä työssään ja tarjoaa puitteet projektitiimin, yhteistyökumppaneiden ja alihankkijoiden kanssa toimimiseen. Prosessimalli on osa yrityksen johtamisjärjestelmää ja päätöksentekomalleja. Sen merkitys tulee sitä tärkeämmäksi mitä enemmän yrityksellä on projekteja. (Murch 2002, 10.) Prosessimallin suunnitteluun käytetty aika näkyy yrityksen kyvyssä hyödyntää omia vahvuuksiaan. (Murch 2002, 57–58.)

Prosessimallit mallintavat ohjelmistojen valmistusprosessia. Näin ohjelmistojen valmistusta voidaan käsitellä systemaattisesti. Perinteisesti prosessi on ollut elinkaarimallin mukainen. Elinkaarimallissa ohjelmiston valmistus pyritään näkemään mahdollisimman laajana, aikaan sidottuna prosessina, jossa ohjelmiston tekninen valmistus on pieni osa kokonaisuutta. Ohjelmiston elinkaari kuvaa ohjelmiston kaikki vaiheet aloituksesta lopetukseen. Erilaisissa projekteissa on erilaisia tehtäviä erilaisessa järjestyksessä ja tämä vaikuttaa käytettävän prosessimallin valintaan. Valitsemalla sopiva malli, resurssit saadaan käyttöön tehokkaasti. Huonolla valinnalla taas menetetään tehokkuutta. (McConnell 2002, 465.)

2.1 Vesiputousmalli

Vesiputousmalli on tavallisin vaihejakomalleista. Vaihejakomalleissa ohjelmiston kehittäminen tai koko elinkaari on jaettu vaiheisiin. Vesiputousmalli on esitelty ensimmäisen kerran 1970 Winston W. Roycen kirjoittamassa artikkelissa. Winston Royce ei kuitenkaan ollut vesiputousmallin kannattaja, vaan esitti mallin olevan esimerkki siitä, miten projektia ei kannata tehdä. Todellisuudessa Royce kannatti iteratiivista kehitystä, mutta hänet leimattiin artikkelinsa takia vesiputousmallin isäksi. (Larman 2007, 102–103.)

Vesiputousmalli on saanut nimensä siitä, että sen suunnittelu- ja toteutusprosessi etenee vaiheittain alaspäin kuin vesiputouksessa. Vesiputousmallissa keskitytään yhteen prosessivaiheeseen kerrallaan. Kun vaihe on valmis, tulokset hyväksytetään ja jäädytetään ennen seuraavaan vaiheeseen siirtymistä. Alkuperäinen vesiputousmalli ei salli paluuta vaiheissa taaksepäin, mutta mallin uusimmissa versioissa tämä on mahdollista. Jokaisen vaiheen lopputuotteena on yksi tai useampi dokumentti. (Haikala & Märijärvi 2004, 36–37.)

Haikalan ja Märijärven mukaan perinteinen vesiputousmalli jakaantuu seitsemään eri vaiheeseen, joista ensimmäinen on esitutkimus (kuva 1). Esitutkimusvaiheessa määritellään ohjelmiston yleiset järjestelmätason vaatimukset. Esitutkimuksessa mietitään vastausta kysymykseen: ”Miksi ohjelmisto tulisi tehdä?”. Näitä vaatimuksia kutsutaan asiakasvaatimuksiksi, sillä niissä ei oteta kantaa siihen, millainen järjestelmä täyttää asiakkaan tarpeet. Ne kuvaavat asiakkaan vaatimuksia toteutettavasta ohjelmistosta.

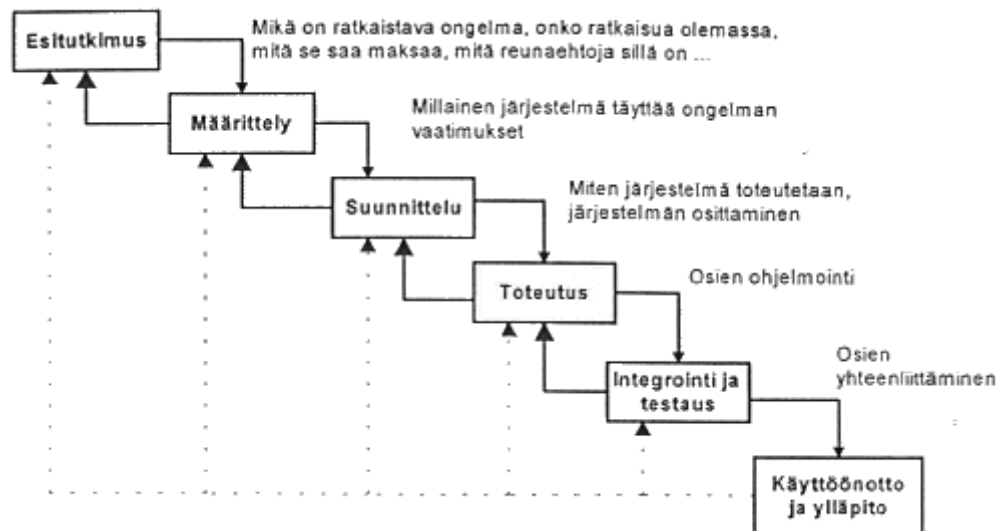
Esitutkimus on vaiheista tärkein, sillä jos asiakasvaatimukset ovat väärät, ei lopputuloksena voi olla hyvä ohjelmisto. On myös tärkeää ymmärtää asiakkaan todelliset tarpeet perusteellisesti. Haastattelu sopii menetelmänä hyvin asiakkaan tarpeiden selvittämiseen. Esitutkimusvaihe saattaa sulautua määrittelyvaiheeseen, sillä asiakasvaatimusten tarkentaminen jatkuu koko määrittelyvaiheen ajan.

Määrittelyvaiheessa analysoidaan asiakkaan vaatimuksia ja niistä johdetaan ohjelmistovaatimukset. Vaihe vastaa kysymykseen: ”Mitä järjestelmä tekee?”. Vaiheen tuloksena syntyy toiminnallinen määrittely. Määrittelydokumentissa kuvataan ohjelmiston toiminnot ja ei-toiminnalliset vaatimukset, kuten käytettävyys, suoritusteho ja vaste-ajat. Ulkoiset rajoitukset kuvataan myös. Niitä voivat olla esimerkiksi toteutusympäristö ja käytössä oleva muistitila.

Suunnitteluvaiheessa suunnitellaan määrittelyvaiheessa kuvatut toiminnot. Vaihe vastaa kysymykseen: ”Miten järjestelmä tekee suunnitellut toiminnot?”. Suunnitteluvaihe jakaantuu kahteen eri vaiheeseen: arkkitehtuuri- ja moduulisuunnitteluun. Arkkitehtuurisuunnittelun lopputuloksena syntyy tekninen määrittely, jossa ohjelmisto jaetaan itsenäisiin moduuleihin. Moduulisuunnittelussa suunnitellaan yksittäisten moduulien rakenne.

Toteutusvaiheessa suoritetaan sovelluksen ohjelmointityö. Samalla kirjoitetaan sovellukseen liittyvät tekniset dokumentit. Vaiheen tavoitteena on valmis ja toimiva sovellus. Kun toteutusvaihe on saatu päätökseen, siirrytään testausvaiheeseen. Tämän vaiheen aikana yritetään löytää järjestelmän virheet. Testauksen osuus projektin kokonaiskustannuksista saattaa olla huomattava, mutta usein käy myös niin, että testaukseen ei panosteta tarpeeksi. Tämä kostautuu myöhemmin sovelluksesta löytyvillä virheillä.

Testauksen jälkeen on käyttöönoton ja ylläpidon vuoro. Käyttöönoton aikana sovellus otetaan asiakkaalla käyttöön. Ylläpidon aikana korjataan järjestelmän virheitä ja ratkotaan asiakkaan ongelmia sovellukseen liittyen. Jos vaatimukset sovelluksen suhteen muuttuvat, sovellusta muokataan vaatimusten mukaisesti ja siihen lisätään mahdollisesti uusia ominaisuuksia. Ylläpito jakaantuu kahteen vaiheeseen: korjaavaan ja täydentävään ylläpitoon. (Haikala & Märijärvi 2004, 37, 39, 40, 41.)



KUVA 1 Vesiputousmalli (Haikala & Märijärvi 2004, 36.)

2.2 Prototyypimalli

Projektin toteutuksessa sovellukseen liittyvät kysymykset ja ongelmat tulevat esiin yleensä vasta silloin, kun asia on toteutettu jo käytännössä. Tämän vuoksi prototyypimallissa tehdään heti projektin alussa yksinkertainen malli sovelluksesta, joka auttaa asiakasta hahmottamaan tulevaa sovellusta. Prototyypimallin käyttö käyttöliittymien kehityksen yhteydessä on tämän takia yleistä. (McConnell 2002, 569.) Prototyypimallissa kehitetään järjestelmää väliversioita hyödyntäen ja asiakkaan tarpeisiin reagoidaan nopeasti. Prototyypimalli on enemmän apuväline kuin prosessimalli, sillä sovelluksen lopullinen kehitystyö voidaan tehdä mitä tahansa prosessimallia käyttäen. (Taina 2009.)

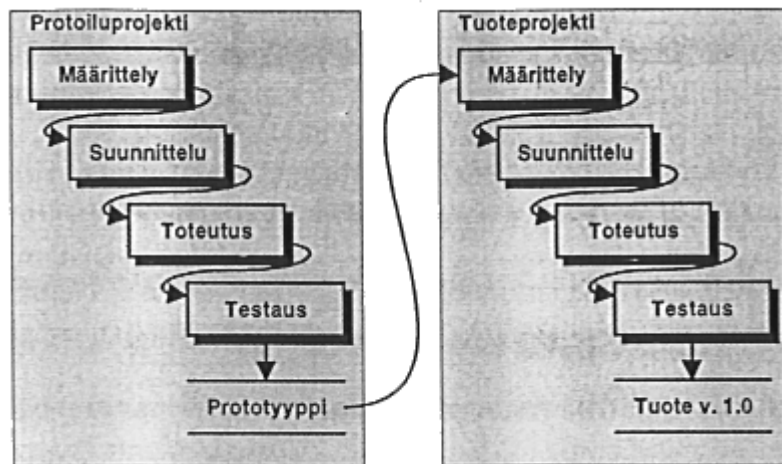
Prototyypimalleja on kaksi erilaista: kehittyvät ja hylättävät prototyypit. Kehittyvästä prototyypistä jalostetaan tuotantoversio, kun taas hylättävä prototyyppi ei siirry koskaan tuotantoon. Kehittyvän prototyypin tapauksessa tehdään alkuperäisen idean perusteella suunnitelma ja kehitetään alustava prototyyppi. Prototyyppi esitellään asiakkaalle ja asiakas antaa siitä palautteensa. Tämän jälkeen prototyyppiä kehitetään niin kauan, kunnes se vastaa asiakkaan tarpeita. Lopuksi prototyyppi viimeistellään ja julkaistaan. (McConnell 2002, 434.)

Sovelluksen kehitys aloitetaan riskialttiimmista osista, koska silloin nähdään, miten kehitys etenee. Jos vastaan tulee ylipääsemättömiä ongelmia, voidaan kehitystyö lopettaa, jolloin kustannuksia ei kerry enempää. On tärkeää saada asiakas mukaan projektiin, testaamaan prototyyppiä ja ymmärtämään näkemänsä. Asiakkailta saatavan palautteen on oltava käyttökelpoista, jotta siitä olisi hyötyä kehitystyössä. (McConnell 2002, 434–437.)

Hylättävän prototyypin teossa käytetään nopeampia tekniikoita kuin lopullisessa versiossa. Yleensä prototyyppiä ei käytetä lopullisen sovelluksen

ytimenä, vaan se heitetään pois. Tämän jälkeen aloitetaan varsinaisen sovelluksen työstäminen. Prototyypin tarkoitus on auttaa esimerkiksi sovelluksen suorituskyvyn testaamisessa, verrata erilaisia toteutusvaihtoehtoja ja selvittää vaatimuksia. (McConnell 2002, 569–570.)

Poisheitettävässä prototyypissä on vaarana, että prototyyppi säilytetään ja sen pohjalta kehitetään lopullinen tuote. Tämä saattaa johtaa sovellukseen, joka on suunniteltu huonosti. Tästä seuraa, että ylläpidettävyys ja suorituskyky ovat huonoja. (McConnell 2002, 571.)



KUVA 2 Prototyypimalli (Haikala & Märijärvi 2004, 42.)

2.3 Spiraalimalli

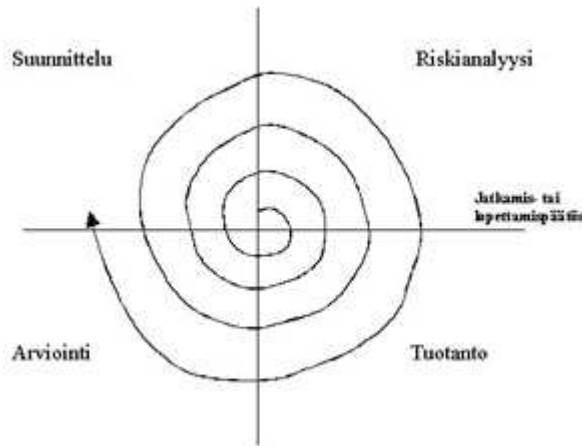
Boehmin vuonna 1988 esittämä spiraalimalli pohjautuu vesiputousmalliin. Toisin kuin vesiputousmallia, spiraalimallia ohjaavat riskit, ei dokumentit. Riskienhallinta on uusi ominaisuus verrattuna vesiputous- ja prototyypimalleihin. Spiraalimalli on iteratiivinen prosessimalli ja se on yhdistelmä vesiputousmallista, prototyypimallista ja riskianalyysistä. Spiraalimallissa on tarkoituksena, että riskit vähenevät kierros kierrokselta, tosin myös uusia riskejä saattaa tulla esiin. (Laine 1998.)

Spiraalimallissa on kuusi kohtaa, joista 3 – 6 suoritetaan jokaisella iteraatiokierroksella:

- Selvitetään tavoitteet, vaihtoehdot ja rajoitukset.
- Tunnistetaan ja selvitetään riskit.
- Puntaroidaan eri vaihtoehdot.
- Kehitetään iteraatioon kuuluvat toimitukset ja varmistetaan, että ne ovat oikein.
- Suunnitellaan seuraava iteraatio.
- Sitoudutaan lähestymistapaan seuraavaa iteraatiota varten, jos on tarpeen.

(McConnell 2002, 141.)

Spiraalimallia luetaan keskustasta ulospäin. Projekti aloitetaan kuvion keskustasta riskien tutkimisella, minkä jälkeen tehdään riskienkäsittelysuunnitelma (kuva 3). Samalla sitoudutaan seuraavan iteraation lähestymistapaan. Jokaisen iteraation jälkeen siirrytään projektissa suurempaan mittakaavaan. (McConnell 2002, 141.)



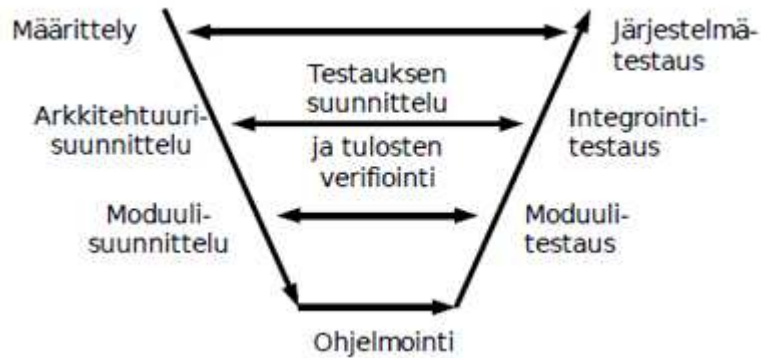
KUVA 3 Spiraalimalli

2.4 V-malli

Vahvasti testaukseen nojautuvaa V-mallia käytetään projektien suunnitteluun ja toteutukseen. V-malli auttaa projektin päämäärien määrittämisessä ja niiden toteuttamisessa. Malli määrittelee jokaiselle projektiin osallistujalle roolin ja tehtävän. V-mallin avulla pyritään minimoimaan projektin riskejä, parantamaan ja takaamaan laatua, pienentämään kustannuksia ja parantamaan kommunikointia. (Fundamentals of the V-Modell 2006.)

V-malli alkaa suunnitteluvaiheesta, jolloin tehdään vaatimusmäärittely. Tämän jälkeen tehdään arkkitehtuurisuunnittelu ja moduulisuunnittelu. Suunnittelu etenee laajemman suunnitelman tekemisestä yksityiskohtaisiin toimintoihin. V-kirjaimen vasen sivu kuvaa suunnitteluvaihetta. Sivua edetään ylhäältä alaspäin (kuva 4). (Haikala & Märijärvi 2004, 289.)

Suunnitteluvaiheen jälkeen siirrytään ohjelmointiin eli toteutukseen. Tämä on V-kirjaimen pohja. Kolmas ja viimeinen vaihe on testaus, joka on V-kirjaimen oikea sivu. Sivua siirrytään alhaalta ylöspäin. Ensimmäinen on moduulitestaus, jonka jälkeen siirrytään integraatiotestaukseen. Viimeisenä on järjestelmätestaus. Testaus etenee tarkemmasta yleisempään. Testauksen suunnittelu tehdään testaustasoa vastaavalla suunnittelutasolla. Esimerkiksi järjestelmätestauksessa testataan, toteutuvatko vaatimusmäärittelyssä kuvatut asiat. (Haikala & Märijärvi 2004, 289.)



KUVA 4 Testauksen V-malli(Haikala & Märijärvi 2004, 289.)

2.5 Ketterät menetelmät

Sovelluskehitysprojektit ovat perinteisesti olleet myöhässä aikataulusta ja ylittäneet budjettinsa. Projektien tulisi olla halvempia, laadultaan parempia ja niiden tulisi ottaa asiakas paremmin huomioon. (Anderson 2004, XXV.) Ketterien menetelmien avulla resurssit voidaan suunnata liiketaloudellisesti tärkeimpiin osa-alueisiin. Tärkeimmät ja riskialttiimmat toiminnot toteutetaan ensin, joten muutoksiin voidaan reagoida nopeasti ja hallitusti. Samalla vältetään tekemästä turhaa työtä. Yhteistyön asiakkaan kanssa tulee olla avointa ja tiivistä, jolloin parhaimmillaan asiakas on osa kehitystiimiä. Haluttuun päämäärään voidaan päästä monella tavalla, joten ketterät menetelmät eivät anna tarkkoja kuvauksia siitä, miten toimitaan projektin edetessä. Ketterien menetelmien taustalla on kokoelma periaatteita, joita noudattamalla uskotaan päästävän hyvään lopputulokseen. (Poimala n.d.)

Ketterät menetelmät yhdistävät iteratiivisen ja inkrementaalisen kehityksen, jolloin sovellusta kehitetään parannellen asiakkaan palautteen perusteella ja kasvattamalla sitä kohti valmista sovellusta.(Poimala n.d.) Ketterät menetelmät perustuvat useisiin peräkkäisiin iteraatioihin. Jokainen iteraatio on oma projektinsa ja sisältää projektisuunnittelun, vaatimusanalyysin, ohjelmistosuunnittelun, koodauksen, testauksen ja dokumentoinnin. Lyhyillä iteraatioilla pyritään minimoimaan riskejä. Jokaisen iteraation lopussa arvioidaan uudelleen projektin tärkeimmät asiat ja päätetään seuraavan iteraation sisällöstä. Ketterissä menetelmissä iteraatiot ovat lyhyitä, joten tästä syntyy ajatus, että kehittäminen on jatkuvaa. (Haikala & Märijärvi 2004, 47.) Yhden iteraation suositeltava pituus on yhdestä kuuteen viikkoa. (Larman 2007, 11.)

Ketterissä menetelmissä on monia yhtäläisyyksiä. Näitä ovat esimerkiksi suora viestintä kirjallisen sijaan, projektiryhmän työskentely samassa työtilassa, asiakkaan mukaan ottaminen projektissa ja nopea reagointi muutoksiin. (Larman 2007, 28.) Nämä toimintatavat ovat lähtöisin kehittäjien ideoista toimintatapojen parantamiseen liittyen. Ketteryydessä on kysymys myös kehittäjien kehittymisestä, ei ainoastaan sovellusten. (Poimala n.d.) Esimerkkejä ketteristä menetelmistä on Scrum, Extreme Programming (XP), Pragmatic Programming ja Agile modeling.

Agile Manifesto on julistus, joka on kirjoitettu vuonna 2001 Utahissa 17 ketterien menetelmien puolestapuhujien toimesta. Julistus toimii yhteisenä pohjana ketterille menetelmille ja levittää ketterien menetelmien ajatusta eteenpäin. Manifestossa määritellään neljä tyypillistä arvoa ketterille menetelmille ja 12 periaatetta, joita menetelmät noudattavat. (Cockburn 2002, 213.)

2.5.1 XP

XP eli Extreme Programming on tunnetuin ketteristä menetelmistä ja sen on luonut Kent Beck yhdessä Ward Cunninghamin ja Ron Jeffriesin kanssa. (Highsmith 2004, xxxiii.) Sen ideana on, että parhaat käytännöt vietään seuraavalle tasolle eli niitä kehitetään jatkuvasti eteenpäin. (Larman 2007, 139.) Mallin pääpaino on toteutuksessa, mutta myös projektinhallintaan on käytäntönsä. (Poimala n.d.)

XP on ohjelmointikeskeinen menetelmä, joka tarjoaa valikoidun joukon käytäntöjä, joita täytyy noudattaa yhdessä. Tämä on tärkeää, sillä käytännöt tasapainottavat toisiaan. XP perustuu neljään arvoon, joita ovat kommunikaatio, yksinkertaisuus, palaute ja rohkeus. XP sisältää neljän arvon lisäksi 12 käytäntöä, joita tulisi noudattaa. (Larman 2007, 36.)

Useat projektin aikaiset ongelmat johtuvat huonosta kommunikaatiosta. Kehittäjien välisiä kommunikaatio-ongelmia pyritään vähentämään pariohjelmoinnin, päivittäisten tapaamisten ja suunnittelupelin avulla. XP:ssä suositetaan 40 tunnin työviikkoa, jotta työteho pysyisi korkeana. (Laine & Paakki n.d.)

Yksinkertaisuudella XP:ssä tarkoitetaan sitä, että asia kannattaa tehdä yksikertaisimmalla toimivalla tavalla. Tämä koskee koko projektin toteuttamista, ei vain ohjelmointia. Sovellus toteutetaan niin, että se vastaa asiakkaan sen hetkisiä tarpeita, eikä mahdollisia tulevaisuuden tarpeita. (Larman 2007, 155.)

XP:ssä asiakas antaa projektin aikana palautetta jatkuvasti. Asiakas kirjoittaa paperikorteille ominaisuuksia, joita haluaa sovellukselle toteutettavan. Näitä kutsutaan käyttäjätarinoiksi. Kehittäjä vastaa palautteeseen välittömästi arvioimalla uuden ominaisuuden. Kehittäjien ajamat yksikkötestit antavat palautetta koodista. Tämä kertoo myös jatkuvasta integraatiosta. Asiakas kirjoittaa hyväksymistestejä, tämä takaa palautteen asiakkaalta omalta osaltaan. (Larman 2007, 155–156.)

Rohkeudella arvona tarkoitetaan rohkeutta kehittää uusia ominaisuuksia nopeasti, tehdä muutoksia ja käyttää uusimpia teknologioita. Ilman yksikkötestausta, hyväksymistestausta ja jatkuvaa integraatiota arkkitehtuuristen muutosten tekeminen olisi vaikeaa. Testaus yhdistettynä yksinkertaiseen suunnitteluun, puhtaaseen koodiin ja sen parantamiseen mahdollistavat rohkeat ratkaisut projektin aikana. (Larman 2007, 156.)

XP:ssä projektin vaiheet koostuvat lyhyistä aikataulutetuista iteraatiokierroksista. Jokainen kierros alkaa suunnittelupelillä, jossa valitaan iteraatiossa toteutettavat käyttäjätarinat ja aikataulutetaan ne. (Laine & Paakki n.d.) Suositeltu iteraatiokierros on pituudeltaan yhdestä kolmeen viikkoa. Ominaisuuksia, joita asiakas ei tarvitse tai ei ole tilannut, ei toteuteta. Asiakkaan ja ohjelmoijien välillä tulee vallita sujuva kommunikaatio. Valmis sovellus pyritään toimittamaan asiakkaalle alle vuoden sisällä. (Larman 2007, 138.)

Projektin muutoksiin pyritään vastaamaan joustavasti, myös projektin myöhäisemmissä vaiheissa. Muutoksista huolimatta koodin tulee olla laadukasta. (Larman 2007, 139 – 140.) Ohjelmakoodi on työvälineistä tärkein ja samalla sovelluksen tärkein dokumentti. XP:ssä suositetaan pariohjelmointia, jossa kehittäjä ohjelmoi koodia ja toinen kehittäjä kommentoi tätä välittömästi. XP:ssä testataan kehitettävää sovellusta jatkuvasti. (Haikala & Märijärvi 2004, 47.)

Koodin parantelu tarkoittaa, että rakenteellisesti huonolaatuista koodia korjataan selkeämmäksi ja hyvien tapojen mukaiseksi. Koodin toiminnallisuus ei kuitenkaan saa muuttua. Koodia ei omista kukaan, vaan kuka tahansa voi muuttaa sitä halutessaan. Jos koodia muuttaa, täytyy muuttajan laatia uudet testit. Ohjelmoinnissa noudatetaan yhteisiä tyylisääntöjä, jotta kehittäjät voivat muokata toistensa koodeja. (Laine & Paakki n.d.)

XP:ssä suositetaan yksikkötestausta koko koodille. Testin läpäissyt uusi koodi liitetään heti sovellukseen. Tätä kutsutaan jatkuvaksi integroinniksi. Yksikkö- ja hyväksymistestit suunnitellaan ennen koodausta. Asiakas suunnittelee ja suorittaa hyväksymistestit. Automatisoitua testausta käytetään myös iterointi- ja regressiotestien muodossa. (Laine & Paakki n.d.)



KUVA 5 XP:n prosessi (Poimala n.d..)

2.5.2 Scrum

Scrum on projektinhallinnan menetelmä, jota käytetään erityisesti ohjelmistoprojekteissa. Se soveltuu hyvin myös muihin projektityyppeihin. Scrumissa keskitytään siihen, miten projektia hallinnoidaan, eikä siihen, miten sovellus toteutetaan. (Anderson 2004, 251.) Se esiteltiin ensimmäisen kerran vuonna 1995, minkä jälkeen sitä on käytetty useissa projekteis-

sa. (Holcombe 2008, 15.) Scrumin varsinaisina kehittäjinä pidetään Ken Schwaberiä, Jeff Sutherlandia, John Scumnotalesia ja Mike Beedlea. (Highsmith 2004, xxxii).

Poimalan mukaan Scrum tarjoaa mallin sovelluskehitykseen, jonka avulla projektia ohjataan. Sen tavoitteena on edistää nopeutta ja joustavuutta projekteissa. Scrumissa on viisi arvoa, joita pyritään noudattamaan. Näitä ovat sitoutuminen, keskittyminen, avoimuus, kunnioitus ja rohkeus. Scrum perustuu erimittaisiin sykleihin. Tärkeimmät näistä sykleistä ovat sprintti ja päivä. Sprintti vastaa yhtä kehitysjaksoa, jonka jälkeen sovellus on periaatteessa valmis julkaistavaksi (kuva 6). Yksi sprintti kestää yleisimmin yhden kuukauden, mutta aika voi vaihdella yhdestä viikosta kahteen kuukauteen.

Scrumissa on kolme roolia, jotka ovat Scrum-mestari, tuotteen omistaja ja tiimi. Scrum-mestari on yleensä projektipäällikkö ja vastaa siitä, että tiimillä on parhaat mahdolliset olosuhteet tehdä töitä. Jos tiimi raportoi mestarille ongelmista, jotka hidastavat töiden tekemistä, mestarin tehtävä on ratkaista nämä ongelmat. Hän myös dokumentoi ryhmän työskentelyä. Lisäksi hän on vastuussa päivittäisistä aamupalavereista ja siitä, että Scrumin periaatteita noudatetaan. Scrum-mestarilla ei ole suoraa määräysvaltaa tiimin jäseniin.

Tuotteen omistaja vastaa tuotteen ominaisuuksista viime kädessä. Omistaja on yleensä asiakkaan edustaja. Tuotteen omistaja tekee päätökset liittyen tuotteen ominaisuuksiin ja toiminnallisuuteen. Hän aikatauluttaa projektin ja huolehtii, että projekti on kannattava.

Tiimi koostuu henkilöistä, jotka ovat tekemässä projektia. Ihannekoko tiimille on seitsemän jäsentä. Tiimi koostuu henkilöistä, joilla on tarvittava osaaminen toteuttaa sovellus. Tiimissä ei nimetä erilaisia rooleja, kuten ohjelmoija ja testaaja. Tällä halutaan korostaa sitä, että kaikki ovat vastuussa toteutettavasta sovelluksesta ja kaikki henkilöt ovat projektin onnistumisen kannalta yhtä tärkeitä. Tiimi jakaa työtehtävät itsenäisesti, jolloin vältytään tehtävien siirtämiseltä edestakaisin. Jokainen tekee sitä asiaa, mitä osaa parhaiten.

Ihmiset jaetaan Scrumissa kahteen ryhmään: siat ja kanat. Sikoihin kuuluvat Scrum-mestari, tuotteen omistaja ja tiimi, eli ne, joilla on jokin kolmesta roolista. Kanoja ovat henkilöt, jotka ovat kiinnostuneita projektista. Esimerkkejä kanoista ovat toisen Scrum-tiimin jäsenet ja toimitusjohtaja. Nimet ovat peräisin eräästä vitsistä, josta ne ovat vakiintuneet käyttöön. Siat määräävät projektin toimintatavat, kun taas kanat voivat vain tehdä havaintoja projektissa. (Poimala n.d.)

Scrumissa luodaan ennen projektin aloitusta korkean tason kuva siitä, mitä projektissa ollaan tekemässä ja mikä tulee olemaan projektin lopputuote. Seuraavaksi tehdään työlista, johon kirjataan tuotteen ominaisuudet. Tuotteen omistaja priorisoi ominaisuudet ennen toteutuksen aloittamista. (Larman 2007, 117.)

Ennen kuin ensimmäinen sprintti aloitetaan, tiimi tekee arvion siitä, kuinka monta vaatimusta se pystyy toteuttamaan yhden sprintin aikana. Sprintille asetetaan myös päämäärä, joka mittaa sprintin onnistumista. Sprintin aikana toteutetaan vaatimuksia, jotka valittiin toteutettavaksi sprintin aikana. Näiden vaatimusten muuttaminen on kiellettyä, mutta tiimi saa käyttää kaikkia mahdollisia keinoja, jotta sprintin päämäärä saavutettaisiin. Tiimiläiset päättävät itse, kuka tekee mitäkin työtä. (Poimala n.d.)

Tiimi kokoontuu päivittäin yhteen päiväpalaverissa samaan aikaan samassa paikassa ja jokainen tiimin jäsen vastaa kolmeen kysymykseen:

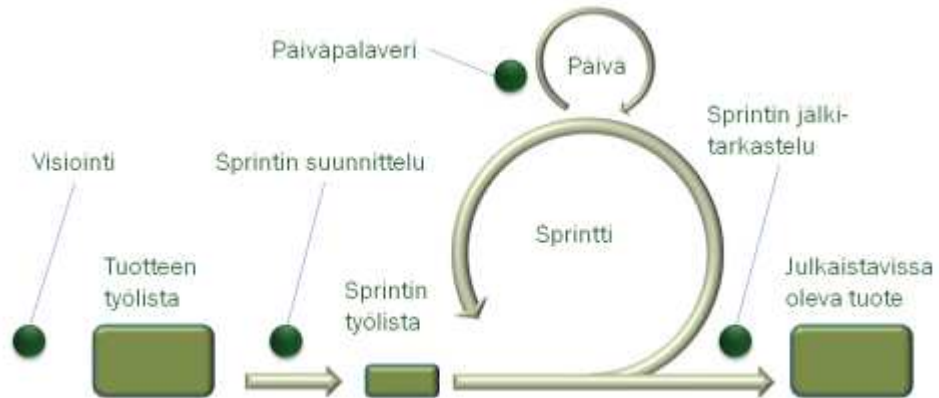
- Mitä olet saanut aikaan viime palaverin jälkeen?
- Mitä aiot tehdä tämän ja seuraavan palaverin välisenä aikana?
- Mitkä tekijät estävät tai hidastavat sinua saavuttamasta sprintin tavoitteita?

(Larman 2007, 120–121.)

Palaveriin voivat osallistua Scrum-mestari ja tiimi sekä henkilöt, jotka ovat kiinnostuneita projektista. (Poimala n.d.) Palaveri pidetään yleensä seisten ympyrässä, jotta palaveriin käytetty aika ei venyisi. Muut kuin tiimin jäsenet eivät kuitenkaan saa puhua ja he seisovat ringin ulkopuolella. Muista aiheista kuin yllä olevista, ei saa keskustella. Suositeltu palaverin kesto on 15 minuuttia. Palaverin tarkoitus on saada selville projektin eteneminen ja mahdolliset ongelmat. (Larman 2007, 12 –122.)

Jokaisen sprintin jälkeen pidetään katselmointitilaisuus, johon saavat osallistua kaikki halukkaat. Tällöin tiimi esittelee valmista tuotetta tuotteen omistajalle. Tuotteen pitäisi olla jokaisen sprintin loppuun mennessä käytönotettavissa, sillä tuote on dokumentoitu ja toteutettu. Tuotteen omistaja päättää joko seuraavan sprintin toteuttamisesta tai tuotteen käyttöönotosta. (Poimala n.d.)

Scrum-mestari, tuotteen omistaja ja tiimi kokoontuvat jokaisen sprintin lopussa keskustelemaan päättyneestä sprintistä. Tiimin jäsenet kertovat omat mielipiteensä sprintin onnistumisesta ja parannusehdotuksista. Sitoumuksia ei kuitenkaan vielä tehdä. Seuraavassa tapaamisessa tiimi päättää kehityskohteet, jotka pyritään toteuttamaan seuraavan sprintin aikana. Jälkitarkastelun jälkeen alkaa uuden sprintin suunnittelu. Tuotteen omistaja voi tässä vaiheessa muokata työlisteriä ja tiimi arvioi, mitkä ominaisuudet tullaan toteuttamaan seuraavassa sprintissä. (Larman 2007, 120.)



KUVA 6 Scrum (Poimala n.d.)

2.5.3 RUP

Ivar Jacobson on kehittänyt RUP-prosessin, joka on lyhenne sanoista Rational Unified Process. (Larman 2007, 207.) RUP juontaa juurensa UPista, joka on iteratiivisen ohjelmistokehityksen prosessikehys. RUP on UPin tunnetuin ja kattavimmin dokumentoitu versio. RUPin mukaan ohjelmistokehitys koostuu neljästä päävaiheesta, joita ovat aloittaminen, tarkentuminen, rakentuminen ja siirtyminen (kuva 7). Kukaan päävaihe sisältää yhden tai useamman iteraation. (Haikala & Märijärvi 2004, 47.) Prosessi koostuu peräkkäisistä aikataulutetuista iteraatioista, joista jokainen on oma vesiputouksensa. (Haikala & Märijärvi 2004, 45.)

Aloittamisvaihe kestää vain muutaman päivän. Iteraatiot tässä vaiheessa ovat mahdollisia, mutta harvinaisia. (Larman 2007, 180.) Vaiheen tarkoituksena on selvittää sovelluksen ominaisuudet, alustavat mallit, tehdä alustava tuotearkkitehtuuri, prototyyppi, selvittää mahdolliset riskit, tehdä alustava projektisuunnitelma ja määrittellä alustavat onnistumiskriteerit. Aloittamisvaiheessa tutkitaan myös erilaisia tuotekonseptin vaihtoehtoja. (Haikala & Märijärvi 2004, 46.)

Tarkentumisvaiheessa ohjelmoidaan arkkitehtuurisesti tärkeät elementit ja ne myös testataan. Nämä tehdään lyhyissä aikarajatuissa iteraatioissa. Tämä vaihe sisältää jo ohjelmointityötä. Workshoppeja saatetaan järjestää, jotta saadaan selville muuttuneita tai uusia tarpeita sovellukseen liittyen. Tämän vaiheen jälkeen sovelluksen ydin ja useimmat vaatimukset ovat päätetty. (Larman 2007, 181.) Tuotteen perusarkkitehtuuri lyödään lukkoon ja toteutetaan. Tarkentumisvaiheen tuloksena ovat seuraavat asiat: täydennetyt mallit, toteutettu toimiva perusarkkitehtuuri, arkkitehtuurikuvaus, riskit, seuraavan vaiheen projektisuunnitelma, onnistumiskriteerit ja alustava käyttöohje. (Haikala & Märijärvi 2004, 46.)

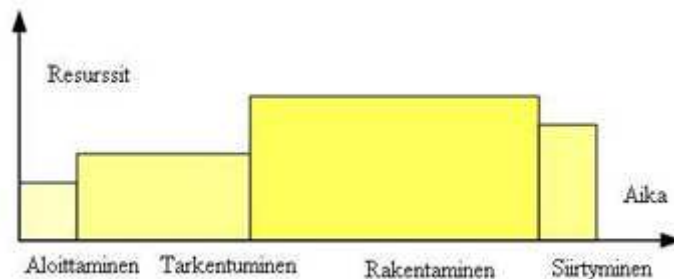
Sovellus tehdään valmiiksi lyhyissä iteraatioissa rakennusvaiheen aikana. Joka iteraatiokierröksellä syntyy uusi beta-versio, jota asiakas testaa. Vaatimukset saattavat muuttua, mutta parhaassa tapauksessa suurimmat yllätykset on huomattu jo aiemmissa vaiheissa. (Larman 2007, 173.) Rakennusvaiheen tuotoksina saadaan seuraavat asiat: lähes täydelliset mallit, be-

ta-versio, arkkitehtuurikuvaus, seuraavan vaiheen projektisuunnitelma, onnistumiskriteerit ja käyttöohje. (Haikala & Märijärvi 2004, 46.)

Siirtymisvaiheessa sovellus otetaan käyttöön. Ennen tätä asiakas saa nähtäväkseen beta-version, jota asiakas tutkii ja antaa siitä palautetta. Tähän voi sisältyä useita iteraatiokierroksia. Lopulta valmis sovellus otetaan käyttöön. (Larman 2007, 181.) Siirtymisvaiheessa valmistuvat seuraavat asiat: installointivalmis ohjelmisto, byrokraatiadokumentit, täydelliset mallit, arkkitehtuurikuvaus, käsikirjat ja www-palvelut. (Haikala & Märijärvi 2004, 46.)

Larmanin mukaan RUPissa on määritelty kuusi parasta tapaa, joilla minimoidaan ohjelmiston viat ja maksimoidaan tehokkuus. Iteratiivisesti kehittäminen on pääasia. Iteraatiot ovat pituudeltaan kahdesta kuuteen viikkoa. Sovelluksen ohjelmointi aloitetaan varhaisessa vaiheessa ja sitä kehitetään palautteen perusteella. Raskaita arkkitehtuurisuunnitelmia ja vaatimusmäärittelyjä ei suosita.

Jokaisessa iteraatiossa ennen ohjelmoinnin aloittamista, tehdään pieni visuaalinen malli sovellusta kuvaamaan. Ohjelmoidessa ensimmäisenä toteutetaan riskialttiimmat toiminnot ja ominaisuudet. Olemassa olevia komponentteja hyödynnetään, jos se on mahdollista. Sovelluksen laatua valvotaan jatkuvasti tehtävien testien avulla. Vaatimusten ja muutosten hallinnointia tehdään jatkuvasti. (Larman 2007, 188–189.)



KUVA 7 RUPin neljä päävaihetta (Partanen, 2009.)

3 AMBIENTIA OY:N PROJEKTIMALLI

Ambientia Oy on kehittänyt projektimallin helpottamaan projektien läpi viemistä sovitussa budjetissa ja ajassa. Malli on iteratiivinen ja inkrementaalinen eli sovellusta kehitetään pienissä jaksoissa ja se kasvaa kohti lopullista muotoaan projektin loppua kohti (kuva 8). Asiakkaan muutospyyntö on mahdollista toteuttaa, sillä muutostyöt suunnitellaan pienissä kokonaisuuksissa. Tällöin muutokset pysyvät hallinnassa. Projektimallin tarkoituksena on toimittaa asiakkaalle näkyviä tuloksia mahdollisimman aikaisessa vaiheessa. Tämä auttaa asiakasta hahmottamaan tulevaa verkkopalvelua ja ehkäisee mahdollisia väärinymmärryksiä toimittajan ja asiakkaan välillä.

Asiakas priorisoi, mitkä toiminnallisuudet tulevassa sovelluksessa ovat tärkeimpiä. Projektia edistetään tämän perusteella. Asiakas ja projektiryhmä työskentelevät yhdessä workshoppeissa, joiden aihe on sovittu etukäteen. Projektia viedään konkreettisesti eteenpäin workshopien aikana. Jokaisen iteraatiokierroksen jälkeen asiakas antaa palautetta, jonka perusteella projektia ohjataan oikeaan suuntaan.

	Konseptointi	Perustaminen	Rakentaminen
Ulkoasu	Maailma	Leiska	Yhdistetään
XHTML	Malli	MOCK	
Sovelluskoodi			
Tietokanta			

KUVA 8 Ambientia Oy:n projektimalli yksinkertaistettuna

3.1 Vaiheet

Ambientian projektimalli koostuu kuudesta eri vaiheesta (kuva 9). Ensimmäinen vaihe on tarvekartoitus, johon osallistuu asiakas, konsultti ja asiantuntija. Tapaamista kutsutaan workshopiksi. Osallistujat miettivät yhdessä, miksi nimenomainen projekti toteutetaan ja mitä projektissa on tarkoitus tehdä. Vaiheen lopputuloksena syntyy tarvekartoitus, joka vastaa tähän kysymykseen. Tarvekartoitus tiivistyy projektin visioon ja tarvekartoitusdokumenttiin. Dokumentissa on korkean tason vaatimuksia projektiin liittyen. Workshopin jälkeen Ambientia järjestää oman kokouksensa, jossa käydään läpi projektin seuraavat vaiheet ja jaetaan tehtävät. Kokouksen jälkeen kirjoitetaan kokousmuistio ja lähetetään se asiakkaalle.

Kun tarvekartoitus on valmis, siirrytään konseptointivaiheeseen. Konseptoinnissa ovat mukana asiakas, konsultti ja projektiryhmä. Ennen konseptointia projekti luokitellaan ja konsultti kertoo projektiryhmälle tulevasta projektista. Verkkopalvelusta toteutetaan klikkailtava xhtml-malli, jonka avulla visualisoidaan ja tehdään asiakkaalle konkreettiseksi verkkopalvelun käyttöliittymä ja sen toiminnot. Xhtml-malliin toteutetaan verkkopalvelun kokonaisuuden kannalta tärkeimmät näkymät. Asiakas päättää, mitkä näkymät ovat tärkeimpiä, sillä ne toteutetaan ensin. Lopputuotteena

vaiheesta syntyy joko xhtml- tai Axure-malli ja arkkitehtuurisuunnitelma. Kaikista tärkeistä käyttötapauksista kirjoitetaan käyttötapaukset.

Perustamisvaiheessa sovellukseen liitetään näkymät ja niihin yhdistetään mock-toteutus arkkitehtuurista. Tämän lisäksi tehdään spike-testit kriittisistä toiminnoista. Vaihe tehdään iteratiivisesti asiakkaan priorisoinnin perusteella. Mahdolliset valmiit moduulit asennetaan tässä vaiheessa. Ennen perustamista ensimmäinen auditointi asiakkaan kanssa täytyy olla järjestetty ja suunnitteluvaiheesta tulee olla tehtynä tilaus. Kehitysympäristön täytyy olla asennettu ja iteraatiosuunnitelma tehty. Perustamisvaiheessa on mukana asiakas, konsultti ja projektiryhmä. Lopputuotteena vaiheesta on mock-mallitoteutus, hyväksyty visuaalinen ilme, työmääräarvio toteutuksesta ja tekniset kokeilut.

Rakentamisvaiheen aikana rakennetaan iteroiden asiakkaan kanssa priorisoidut ominaisuudet valmiiksi asti. Lopputuloksena tuotetut versiot testataan muuttuneilta osiltaan. Rakentamisessa on mukana asiakas, projektiryhmä ja testaus. Ennen rakentamisvaihetta tulee olla tilaus toteutusvaiheesta valmis. Lopputuotteena syntyy valmis toiminnallisuus ja sovitun ilmeen mukainen sovellus. Graafinen ohjeistus tehdään, jotta verkkopalvelun ulkoasu olisi selvä. Tämä pitää sisällään esimerkiksi väreit, fontit, muotoikielen ja logon käytön.

Julkaisutestausvaiheen aikana yhdistetään asiakkaan edellisten vaiheiden aikana tuottama sisältö palveluun ja integraatiotestataan kokonaisuus. Julkaisutestausvaiheessa on mukana asiakas, projektiryhmä, testaus ja ylläpito. Ennen vaiheen toteutusta tulee olla julkaistava toiminnallisuus valmiina. Lopputuotteena vaiheesta syntyy ”Known Issues”- ja kehitysidealista.

Käyttöönottovaiheessa palvelu otetaan käyttöön ja julkaistaan. Käyttöönotossa on mukana asiakas, projektiryhmä, testaus ja ylläpito. Ennen vaihetta palvelun tulee olla julkaisukunnossa. Toinen auditointi on pidetty. Kun palvelu luovutetaan ylläpitoon, suoritetaan vielä kolmas auditointi. Lopputuotteita ovat julkaistu palvelu, päätetty projekti ja asiakaspalaute. Projektin päätyttyä järjestetään tapaaminen asiakkaan kanssa, jossa projekti ja sen kulku käydään läpi.



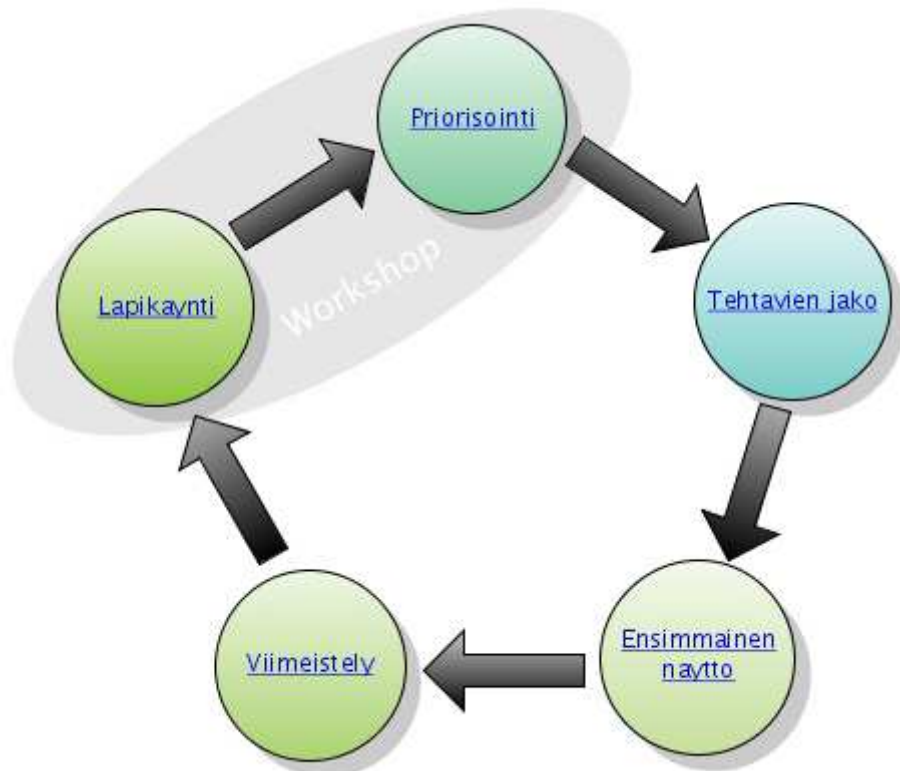
KUVA 9 Ambientian iteratiivinen projektimalli

3.2 Iterointiprosessi

Iterointia käytetään kaikissa muissa projektimallin vaiheissa, paitsi tarvekartoituksessa ja käyttöönotossa. Iterointiprosessi koostuu viidestä kohdasta, joista ensimmäinen on priorisointi (kuva 10). Priorisoinnissa toteutettavat asiat esitellään tehtävähallintajärjestelmässä, jonka jälkeen valitaan, kuinka kriittisiä nämä ominaisuudet ovat asiakkaalle ja mitkä ominaisuudet tuovat asiakkaalle lisäarvoa. Tämän jälkeen tehdään riskinarviointi, kirjataan ominaisuudet tehtävähallintajärjestelmään ja sovitaan aikataulusta asiakkaan kanssa.

Seuraavaksi jaetaan tehtävät ja tarkennetaan ne eri ominaisuuksien alle. Projektiryhmän jäsenet valitsevat itse tehtävänsä. Tämän jälkeen on ensimmäinen näyttö, jolloin asiakas näkee ensimmäisen version sovelluksen toiminnallisuudesta ja visuaalisuudesta. Asiakkaan kanssa käydään läpi ilmenneet ongelmat ja kysymykset, jonka jälkeen viimeistellään toteutettavat ominaisuudet.

Iterointiprosessin viimeinen vaihe on läpikäynti, joka järjestetään asiakkaan kanssa. Asiakas ja projektiryhmä kommentoivat versiota ja projektiryhmä tekee mahdolliset korjaukset. Jos korjauksia ei enää tehdä, versio jäädytetään ja se on valmis testattavaksi. Lopputuloksena on minor-versio sovelluksesta. Iteraation kulusta pidetään palautekeskustelu yhdessä asiakkaan kanssa.



KUVA 10 Iterointiprosessi

4 PROSESSIMALLIEN VERTAILU

Tutkimuskysymykseen ”Miten Ambientian projektimalli eroaa muista prosessimalleista?” etsitään vastausta tutkimalla tarkemmin prosessimallien ominaisuuksia ja vertailemalla malleja toisiinsa.

Riskien hallinnan kannalta Ambientian projektimalli ja ketterät menetelmät eli XP, Scrum ja RUP, ovat hyvä valinta prosessimalleiksi, sillä niissä otetaan huomioon muutokset, joita voi tulla projekteissa sekä asiakkaan suunnalta että yleisesti. Myös spiraalimalli ottaa riskit hyvin huomioon. Huonoiten riskien hallinnan huomioi vesiputous- ja V-malli. Prototyypimalli huomioi riskit keskinkertaisesti.

Spiraalimallin vahvuus on riskinhallinta. Toisaalta riskintuntemus saattaa olla vajavaista ja siihen on silti luotettava. Hoitamattomat riskit saattavat kertaantua myöhemmillä kierroksilla. Spiraalimalli on prosessimallina iteratiivinen ja se kannustaa kokeilemaan projektissa erilaisia vaihtoehtoja. Virheelliset ratkaisut voidaan korjata ja huonot ratkaisut hylätä. Projektia ohjataan riskianalyysin tulosten perusteella oikeaan suuntaan. Spiraalimallin huonoja puolia on, että asiakasta saattaa olla vaikea aktivoida yhteistyöhön. Lisäksi mallin soveltaminen on aikaa vievää, joten aikataulut saattavat venyä. Malli soveltuu parhaiten projekteihin, joissa vaatimukset eivät ole selvillä. Spiraalimallia voidaan hyödyntää myös vasta siinä vaiheessa, kun sovellus on otettu asiakkaalla käyttöön eli ylläpitovaiheessa. (Laine 1998.)

Muutosten hallinnan kannalta parhaat prosessimallit ovat Ambientian projektimallin ja ketterien menetelmien lisäksi prototyypimalli ja spiraalimalli. Vesiputous- ja V-malleissa muutokset ovat hankalia toteuttaa. Tämä johtuu mallien kankeasta rakenteesta. Havainnollisuuden asiakkaille ottaa parhaiten huomioon prototyypimalli, spiraalimalli, ketterät menetelmät ja Ambientian projektimalli. Vesiputousmallissa ja V-mallissa sovelluksen havainnollisuutta asiakkaalle ei ole otettu tarpeeksi huomioon. Näissä malleissa sovelluksesta ei esimerkiksi toteuteta käyttöliittymämallia, joka havainnollistaisi asiakkaalle tulevaa sovellusta.

Prototyypimallissa asiakas saa jo varhaisessa vaiheessa nähtäväkseen prototyypin, joka kuvaa sovelluksen toimintaa. Tällöin asiakkaan visio tulevasta sovelluksesta on selvempi kuin ilman visuaalista mallia. Prototyypin olemassaolo voi aiheuttaa myös hämmennystä asiakkaalle, sillä asiakas ei aina ymmärrä, että prototyyppi ei ole valmis sovellus. Asiakas saattaa ihmetellä, miksi sovellusta ei voi heti käyttää tai miksi sen kehittämiseen kuluu vielä paljon aikaa. (Taina 2009.)

Pois heitetty prototyyppi saatetaan nähdä resurssien ja rahan hukkaamisena ja saman työn tekemisenä kahteen kertaan. Tällöin täytyy ymmärtää, että prototyyppi on toiminut vain apuna, eikä sen ole tarkoituskaan olla lopullinen tuote. Vaikka ulkoasu on kunnossa, sovelluksen logiikka on vajaa. Prototyypimalli ei sovi suuriin projekteihin ja se saattaa muuttua ”koodaa ja korjaa” -tyyliseksi kehittämiseksi. Prototyypimallilla on etsitty vastausta muuttuvien vaatimusten hallintaan projekteissa. (McConnell 2002, 572.)

Projektien aikataulutuksessa mikään malleista ei saa huonoa arviota. Keskinertaisesti projektien aikataulutuksessa pärjäävät kaikki muut paitsi ketterät menetelmät ja Ambientian projektimalli, jotka saavat hyvän arvioidon aikataulutuksesta. Esimerkiksi XP:ssä sovellus pyritään toimittamaan asiakkaalle alle vuoden sisällä. Scrumissa ja RUPissa tämä aika on pidempi. Ketterissä menetelmissä on kuitenkin oltava tarkka aikataulutuksen kanssa, sillä ilman hyvää suunnittelua aikataulu saattaa venyä.

Selvittämättömien vaatimusten kartoittaminen sujuu parhaiten ketterillä menetelmillä, Ambientian projektimallilla sekä prototyypin- ja spiraalimallilla. Vesiputous- ja V-malli eivät sovellu hyvin tällaiseen projektiin, jossa vaatimukset eivät olisi selvillä jo projektin alkuvaiheessa. Vaatimusten muuttuminen aiheuttaa näissä malleissa siirtymisen taaksepäin projektissa. Vaatimusten selvittämiseen auttaa klikkailtava malli, jonka avulla asiakas saa kuvan kehitettävästä sovelluksesta. Kun asiakkaalla on tällainen malli tutkittavana, hänelle tulee paremmin mieleen, mitä ominaisuuksia tulevan sovelluksen tulee sisältää.

Ketterät menetelmät ja Ambientian projektimalli pärjäävät vertailussa parhain arvostanoin muuten paitsi dokumentoinnin osalta. Siinä ne saavat arvioiksi ”keskinertainen”. Tämä on seurausta siitä, että ketteriä menetelmiä käytettäessä dokumentaatio ei ole itseisarvo ja tärkeämpää on sovelluksen kehittäminen. Ohjelmakoodi toimii näissä malleissa dokumentaati-

tiona. Vaarana kuitenkin on, että dokumentointi jää projekteissa liian vähälle huomiolle.

Huonointen vertailussa pärjäsivät vesiputous- ja V-malli. Vesiputousmalli on tunnetuin prosessimalleista ja sen pohjalta on rakennettu muita malleja. Vesiputousmallin rakenne on selkeä ja helppo ymmärtää, sillä siinä edetään tietyssä järjestyksessä projektin aloituksesta loppuun ja jokaisessa vaiheessa on omat tehtävänsä. Malli kuvaa ohjelmistotuotantoa hyvin, mutta vain teoriatasolla. Dokumentointi on mallin hyviä puolia, sillä erilaisia dokumentteja syntyy projektin aikana paljon. Ulkopuolisen henkilön on helppo tulla projektiin mukaan, kun vaiheet ovat hyvin dokumentoituja. Vesiputousmalli ei ota kantaa käytettäviin tekniikoihin, joten se ei rajoita projektin toteutusta teknisellä tasolla. (Laine 1998.)

Vesiputousmallissa projektille luodaan kustannusarvio jo projektin alussa, joten kustannustietous mallissa on parhaassa tapauksessa hyvä. Kustannuksissa voidaan säästää, jos suunnittelu on tehty oikein. Usein kustannukset eivät kuitenkaan pysy sovitussa rajoissa, koska sovellusta ei ole osattu suunnitella oikein. (Laine 2009.)

Vesiputousmallin huonoja puolia on sen joustamattomuus. Perinteisessä vesiputouksessa ei sallita paluuta taaksepäin. Lisäksi mallista puuttuu iteratiivisuus. Jos sovelluksen ohjelmointivaiheessa huomataan, että suunnittelussa on tehty virhe, on aikaa vievää ja kallista korjata se. Harvoin myöskään suunnitteluvaiheessa osataan ottaa huomioon kaikki tarvittavat asiat, sillä usein vasta projektin edetessä tulee vastaan uusia asioita, jotka täytyy ottaa huomioon. (Laine & Paakki n.d.)

Vesiputousmalli ei korosta yhteyttä asiakkaaseen. Jos asiakas on kiireinen tai ei halua osallistua tiiviisti projektitoimintaan, vesiputousmalli sopii tähän. Toisaalta asiakkaan näkökulmasta vesiputousmalli ei ole hyvä, sillä mallissa syntyy konkreettista vasta projektin loppupuolella, joten asiakkaan on vaikea tietää, mitä hän todellisuudessa tulee saamaan. Jos asiakas huomaakin, että sovellus ei vastaa hänen toiveitaan, muutosten toteuttaminen on taas kallista ja aikaa vievää.

Vesiputousmalli sopii jäykän luonteensa takia suuriin ohjelmistoprojekteihin. Samalla jäykkyys kuitenkin estää sen kirjaimellisen käytön ohjelmistokehityksessä. Projektin edetessä vaatimukset muuttuvat ja osa vaatimuksista selviää vasta projektin aikana. Suuri osa työajasta menee dokumentointiin. Suurin ongelma on aikataulussa pysyminen. Jos aikataulussa ei pysytä, testaustalle ei jää tarpeeksi aikaa.

V-malli on sukua vesiputousmallille, joten vesiputousmallin hyödyt ja haitat ovat suurelta osin samat kuin V-mallille. V-mallin hyötyjä on se, että mallia voidaan muokata jokaiseen projektiin ja organisaatioon sopivaksi. V-malli auttaa hahmottamaan projektin roolit ja tehtävien suorittamisajankohdan. Se saattaa myös säästää aikaa projektissa, koska testaus otetaan mukaan suunnittelussa jo varhaisessa vaiheessa. Vaikka aikataulu tai budjetti ylittyisi, testausta ei unohdeta. V-malli saattaa olla liian muokattavissa

sa oleva, jolloin mallia käyttävä ei osaa päättää, mitä asioita projektiin otetaan mukaan ja mitä ei. (Taina 2009.)

Prototyypin ja spiraalimallin asettuvat arvostelussa keskilinjalle. Nämä mallit ovat sellaisia, joita on mahdollista käyttää apuna projekteissa esimerkiksi toisen prosessimallin tukena. Erilaisia prosessimalleja on kehitetty, sillä eri mallit sopivat erilaisiin projekteihin. Ketterät menetelmät eivät luettele tarkkoja sääntöjä, miten projektissa täytyy edetä. Tämä saattaa olla yksi syy niiden kasvaneeseen suosioon.

XP:n etuja ovat käytännönläheinen suhtautuminen projektiin, monet mallin tekniikoista on peräisin kehittäjiltä. Asiakas on mukana projektissa tiiviisti. Tämä on otettu huomioon myös Ambientian projektimallissa. Mahdollisia ongelmakohtia XP:ssä ovat käyttäjäkonfliktit, joita ei ole dokumentoitu sekä suunnitelmien ja dokumentoinnin puute. (Laine & Paakki n.d.) Scrum on monikäyttöinen prosessimalli, sillä sitä voidaan käyttää myös markkinoinnissa ja ylläpidossa. Normaalisti malleja käytetään sovelluksen toteutusprojektissa. Ongelmana tässä mallissa saattaa olla se, että projektiryhmän aika saattaa kulua asiakkaan saaman version korjaamiseen ja ongelmien selvittelyyn. (Haikala & Märijärvi 2004, 47.)

Vertailu osoitti, että Ambientian projektimalli ja ketterät menetelmät ovat ominaisuuksiltaan parhaita prosessimalleja. Vesiputousmalli on kankea ja ei siksi sovellu sellaisenaan verkkopalveluprojekteihin. Ambientian projektimalli eroaa muista malleista siten, että se on räätälöity Ambientian omaan käyttöön. Siinä on mietitty asioita yrityksen näkökulmasta, joten monilta sudenkuopilta on vältytty. Malli pohjautuu ketteriin menetelmiin, joten mahdollisten muutospyyntöjen jälkeen projektin suuntaa voidaan muuttaa. Sovellusta kehitetään pienissä iteraatioissa, mikä auttaa muutosten toteuttamisessa. Asiakas saa jo projektin alkuvaiheessa nähtäväkseen mallin tulevasta sovelluksesta. Asiakas saa myös päättää, mitkä ominaisuudet ovat tärkeimpiä ja toteutetaan ensimmäisenä.

Prosessimalleja vertaillessa ketterät menetelmät ohittivat perinteiset prosessimallit lähes kaikissa ominaisuuksissa. Hyödyt yritykselle ja asiakkaalle ovat selvät vertailun perusteella. Ajankäyttö ja muutosmahdollisuudet projekteissa ovat kattavammat ketterissä malleissa kuin perinteisemmissä prosessimalleissa. On kuitenkin tilanteita, jossa perinteisen vesiputousmallin käyttö voi olla perusteltua, joten vanhojakaan malleja ei kannata unohtaa. Koska kaikki prosessimallit polveutuvat vesiputousmallista, siinä on oltava hyviäkin puolia.

Alla olevissa taulukoissa (taulukko 1, taulukko 2) kuvataan eri prosessimallien ominaisuuksia seuraavilla sanallisilla arvioilla: huono, keskinkertainen, hyvä.

TAULUKKO 1 *Vertailutaulukko eri prosessimallien ominaisuuksista*

Ominaisuus	Vesiputousmalli	Prototyypimalli	Spiraalimalli	V-malli
Riskien hallinta	Huono	Keskinkertainen	Hyvä	Huono
Muutosten hallinta	Huono	Hyvä	Hyvä	Huono
Havainnollisuus asiakkaalle	Huono	Hyvä	Hyvä	Huono
Aikataulutus	Keskinkertainen	Keskinkertainen	Keskinkertainen	Keskinkertainen
Vaatimukset, jotka eivät ole selvillä	Huono	Hyvä	Hyvä	Huono
Dokumenttien määrä	Hyvä	Huono	Keskinkertainen	Hyvä
Yhteydenpito asiakkaaseen	Huono	Keskinkertainen	Keskinkertainen	Huono
Sopii parhaiten	Projekteihin, joissa vaatimukset ovat alusta asti tiedossa	Käytettäväksi apuna jonkun muun prosessimallin tukena	Projekteihin, jossa tavoitteet eivät ole selkeitä	Projekteihin, joissa vaatimukset ovat alusta asti tiedossa

TAULUKKO 2 *Vertailutaulukko eri prosessimallien ominaisuuksista*

Ominaisuus	XP	Scrum	RUP	Ambientian projektimalli
Riskien hallinta	Keskinkertainen	Hyvä	Hyvä	Hyvä
Muutosten hallinta	Hyvä	Hyvä	Hyvä	Hyvä
Havainnollisuus asiakkaalle	Hyvä	Hyvä	Hyvä	Hyvä
Aikataulutus	Hyvä	Hyvä	Hyvä	Hyvä
Vaatimukset, jotka eivät ole selvillä	Hyvä	Hyvä	Hyvä	Hyvä
Dokumenttien määrä	Keskinkertainen	Keskinkertainen	Keskinkertainen	Keskinkertainen
Yhteydenpito asiakkaaseen	Hyvä	Hyvä	Hyvä	Hyvä
Sopii parhaiten	Moniin erityyppisiin projekteihin	Keskittyy projektin läpivientiin	Moniin erityyppisiin projekteihin	Moniin erityyppisiin projekteihin

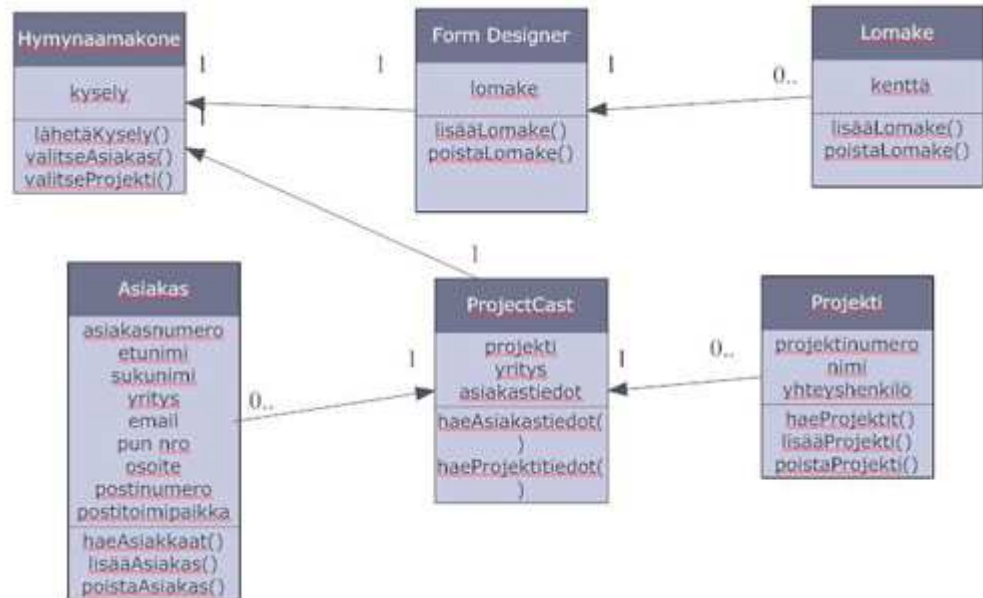
5 VERKKOSOVELLUKSEN SUUNNITTELEMINEN

Tutkimuskysymykseen ”Miten verkkosovellus suunnitellaan Ambientian iteratiivisen projektimallin avulla?” etsitään tuloksia kuvaamalla verkkosovelluksen suunnittelu projektimallin vaiheiden mukaisesti. Toteutusvaihe kuvataan vain pintapuolisesti. Alla on kerrottu tietoja suunnitellusta verkkosovelluksesta.

Suunniteltava sovellus on asiakastytyvyyden mittaamiseen tarkoitettu sovellus, jonka avulla kerätään asiakkailta palautetta tehdyistä projekteista ja sovelluksista. Jatkuva palaute auttaa projektiryhmää reagoimaan nopeammin asioihin ja antaa ryhmälle tietoa siitä, miten asiakas reagoi erilaisiin asioihin. Työntekijät saavat sovelluksen avulla jatkuvaa palautetta, mikä motivoi jokapäiväisessä työskentelyssä. Palaute voi olla negatiivista tai positiivista, joten palaute auttaa työntekijää muuttamaan työskentelyään oikeaan suuntaan. Palautteen saaminen jatkuvasti kerryttää tilastotietoa asiaan liittyen, joten Ambientian on helppo tutkia, millaista palautetta saadaan. Kyselyiden kautta saadaan selville myös oma projektiyrityskuva. Kysely on tarkoitus lähettää asiakkaille kerran kuukaudessa ja asiakkaiden on mahdollista kieltäytyä osallistumasta kyselyyn. Kyselystä kerrotaan asiakkaalle heti projektin alussa ja asiakkaalla on mahdollisuus nähdä tulokset. Projekti toteutetaan sisäisenä projektina.

Projekti aloitetaan tarvekartoitusvaiheella ja heti projektin alussa sovitaan tapaaminen asiakastytyvyyssovelluksen tilaajan kanssa. Tapaamisessa on mukana myös järjestelmärkkitehti asiantuntijan roolissa. Tapaamisen aikana mietitään, miksi projekti tulotisiin toteuttamaan ja mitä asioita projektin aikana tehtäisiin. Vaiheen lopputuotteena saadaan projektin visio ja tarvekartoitusraportti.

Tarvekartoituksen jälkeen siirrytään konseptointivaiheeseen, jossa asiakastytyvyyssovelluksesta toteutetaan klikkailtava XHTML-malli. Malli auttaa hahmottamaan toteutettavan sovelluksen ulkoasua ja toimintaa. Sovelluksessa ei tule olemaan kovin montaa näkymää, joten kaikki näkymät toteutetaan jo tässä vaiheessa. XHTML-mallin lisäksi tehdään arkkitehtuurisuunnitelma, domainmalli ja käyttötapauskuvaukset (kuva 11). Vaihe etenee iteratiivisesti eli pienenissä vaiheissa eteenpäin. Henri Sora antaa palautetta sovelluksesta, jonka perusteella sitä edistetään. Hän myös hyväksyy sovelluksen visuaalisen ilmeen.



KUVA 11 Domainmalli sovelluksesta

Perustamisvaiheessa asiakastytyväisyyssovellukseen liitetään näkymät ja mallitoteutus arkkitehtuurista yhdistetään siihen. Kriittiset toiminnot testataan. Tätä vaihetta toteutetaan iteroiden pienissä sykleissä, sovelluksen tilaajan vaatimusten mukaan. Tämän jälkeen siirrytään rakentamisvaiheeseen, jonka aikana sovellus toteutetaan iteroiden valmiiksi asti. Sovellus testataan muuttuneilta osin, jos on tarvetta. Julkaisutestauksen aikana palvelu testataan kokonaisuudessaan. Testaus on tärkeä vaihe, jonka huono toteutus näkyy myöhemmin virheinä ja käyttöongelmina sovelluksessa. Vaikka sovelluskehittäjät olisivat testanneet sovellusta toteutuksen aikana, jää virheitä silti sovellukseen. Käyttöänoton aikana asiakastytyväisyyssovellus otetaan Ambientialla käyttöön. Käytön aikana esille tulleet ongelmat korjataan.

6 LOMAKEKYSELYN TULOKSET

Tutkimusongelmaan ”Miten Ambientian projektimalli toimii projekteissa käytännössä?” käytettiin yhtenä tutkimusmenetelmänä lomakekyselyä, josta vastaukset saataisiin mahdollisimman laajalta vastaajajoukolta. Menetelmänä kysely on tehokas ja nopea suorittaa, mutta samalla siihen liittyy heikkouksia. Koska vastaajajoukko saattaa jäädä pieneksi tai vastaajat eivät vastaa toden mukaisesti, vastaajajoukon valinta on tärkeää. (Hirsjärvi, Remes & Sajavaara, 2007, 190.)

Kyselyn saatekirje lähetettiin sähköpostitse Ambientian työntekijöille, yhteensä 65 henkilölle. Kysely oli näkyvillä Ambientian omassa kyselyiden luomiseen tarkoitettussa palvelussa. Kyselyjoukkoon kuului henkilöitä eri toimenkuvista ja projektiryhmistä. Kyselyyn vastasi 18 henkilöä 65:stä, vastausprosentin ollessa 27,7 %. Kyselyyn vastanneista naisia oli 22 % ja miehiä 77 %. Tämä selittyy sillä, että valtaosa Ambientian työntekijöistä on miehiä. Vastaajista 22 % on työskennellyt Ambientialla vähintään seit-

semän vuotta. 4-6 vuotta työskennelleitä on 27 % ja 38 % vastaajista on työskennellyt Ambientialla yhdestä kolmeen vuotta. Vastaajista 11 % on työskennellyt Ambientialla alle vuoden.

Ambientian iteratiivinen projektimalli tunnetaan työntekijöiden keskuudessa yleisellä tasolla melko hyvin, sillä yli puolet vastaajista vastasi tietävänsä projektimallin kuusi eri vaihetta. 33 % vastaajista vastasi, että ei tunne vaiheita ja 11 % vastaajista ei osannut sanoa, että tuntee vaiheet. Koska kolmasosa vastaajista ei tunne vaiheita, näyttää siltä, että projektimallin läpikäynnille olisi tarvetta. Toisaalta osa vastaajista saattaa olla työtehtävissä, jossa projektimallia ei suoranaisesti käytetä.

Kysymykseen ”Kuinka hyvin tunnet projektimallin eri vaiheet” 38 % vastaajista vastasi tuntevansa ne tyydyttävästi. Vastaajista 27 % vastasi tuntevansa vaiheet melko hyvin. Sama prosenttimäärä vastaajista vastasi tuntevansa vaiheet huonosti. 5 % vastaajista eli yksi henkilö vastasi, että ei osaa sanoa kuinka hyvin vaiheet tuntee. Vastausten perusteella projektimallin eri vaiheet eivät ole kovin tuttuja. Yksikään vastaajista ei vastannut, että tuntisi vaiheet erittäin hyvin. Nämä vastaukset korostavat projektimallin läpikäynnin tärkeyttä.

55 % vastaajista vastasi tunnistavansa projektimallin eri vaiheet omassa työskentelyssään, kun taas 27 % vastasi, että ei tunnista vaiheita omassa työssään. 16 % vastaajista vastasi, että ei osaa sanoa. Vastausten perusteella yli puolet vastaajista tunnistaa projektimallin vaiheet omassa työskentelyssään, mikä on hyvä asia.

Kysymys ”Oletko käynyt projektimallia läpi yhdessä projektiryhmän/päällikön kanssa?” jakoi vastaukset lähes tasan, sillä tasan puolet vastaajista vastasi ”kyllä” ja 44 % vastaajista ”ei”. 5 % vastaajista ei osannut sanoa, onko mallia käyty yhdessä läpi. Tarkentava kysymys ”Kuinka tarkasti projektimallia on mielestäsi käyty läpi?” paljasti, että tarvetta tarkemmalle läpikäynnille olisi, sillä puolet vastaajista oli sitä mieltä, että malli on käyty läpi kohtalaisen tarkasti. 38 % mielestä projektimallia ei ole käyty läpi lainkaan tarkasti. Ainoastaan 5 % vastaajista tunsikin, että mallia on käyty hyvin tarkasti läpi. 5 % vastaajista oli sitä mieltä, että mallia on käyty läpi melko tarkasti.

Projektimalli vaikuttaa vastaajajoukosta puolen työskentelyyn. 33 % vastaajista ei osaa sanoa, että vaikuttaako projektimalli heidän työskentelyynsä. 16 % vastasi, että malli ei vaikuta heidän työskentelyynsä. Jatkokysymyksenä oli, että miten projektimalli vaikuttaa vastaajien työskentelyyn. Kysymykseen sai kirjoittaa vapaan vastauksen. Vain seitsemän vastaajaa vastasi tähän kysymykseen. Vastauksissa oli esillä eri vaiheiden liittyminen omaan työskentelyyn eli kerrottiin, mitä vaiheita omaan työskentelyyn kuuluu. Projektimallin vaiheittainen eteneminen nousi myös esiin.

Kysymykseen ”Miten projektimalli toimii mielestäsi käytännössä” 38 % vastaajista vastasi, että tyydyttävästi. 27 % vastaajista vastasi, että ei osaa sanoa. Vain 16 % vastaajista vastasi, että projektimalli toimii käytännössä

melko hyvin. 11 % vastasi, että malli ei toimi kovin hyvin ja 5 %, että ei lainkaan hyvin. Vastajaat eivät ehkä halunneet ottaa kantaa tähän asiaan, koska lähes kolmas osa vastaajista ei osannut sanoa projektimallin toimivuudesta käytännössä. Annettujen vastausten perusteella projektimallin toimivuuteen käytännössä on kiinnitettävä huomiota.

17 vastaajaa 18:sta vastasi vapaamuotoiseen kysymykseen ”Mitä hyvää projektimallissa on?”. Vastauksista nousi selvimmin esiin mallin iteratiivisuus ja ketteryys, joiden avulla kehitys tapahtuu hallitusti. Projektimallin vaihteellisuus ja mallin selkeys mainittiin myös. Projektimallista tehty dokumentointi on myös hyvä asia. Mallia pidettiin johdonmukaisena ja projektimallin vaiheiden mukaisesti edetessä asiakas saa projektin varhaisessa vaiheessa nähtäväkseen klikkailtavan sovelluksen.

Vastauksissa kysymykseen ”Mitä huonoa projektimallissa on?” tuli esille se, että mallin eri vaiheita ei noudateta aina käytännössä. Tämä voi olla seurausta esimerkiksi kiireestä tai mallin monimutkaisuudesta. Joitakin vaiheita ei vastaajien mielestä ole ohjeistettu tarpeeksi hyvin ja siksi toimintamallien omaksuminen voi olla hankalaa. Vastaajien mielestä projektimalli ei sovi kaikkiin projekteihin. Tähän kysymykseen vastasi 16 vastaajaa 18:sta.

Kysymykseen ”Miten projektimallia pitäisi mielestäsi kehittää?” vastasi 12 vastaajaa 18:sta. Projektimallin noudattaminen ja iteraatioiden käyttäminen nousivat esiin vastauksista. Tämä liittyy ehkä enemmän mallin noudattamiseen kuin sen kehittämiseen. Vastajaat toivoivat myös projektimallin eteenpäin kehittämistä ja sen havainnollistamista. Projektimalli tuntuu vastaajista monimutkaiselta, joten sitä voisi yksinkertaistaa tai käydä läpi työntekijöiden kanssa. Projektipäälliköille toivottiin enemmän projektipäällikön tehtäviä. Malli pitäisi saada paremmin vietyä käytännön tasolle, sillä tällä hetkellä se on epäkonkreettinen, eikä tarjoa toimintatapojen kuvauksia.

7 TEEMAHAASTATTELUIJEN TULOKSET

Lomakekyselyn lisäksi tutkimustapana käytettiin teemahaastattelua, sillä niiden avulla oli tarkoitus saada syvällisempää palautetta projektimallista. Teemahaastatteluissa pyrittiin saamaan selville haastateltavien henkilökohtaiset kokemukset aiheesta. Haastattelujen pohjana oli kysymyksiä projektimalliin liittyen, mutta jokainen haastattelu eteni omalla tahdillaan ja haastattelutilanteen ehdoilla. (Hirsjärvi & Hurme, 2009, 34.)

Teemahaastatteluissa haastateltiin kolmea Ambientilla työskentelevää sovelluskehittäjää. Kaksi haastatteluista toteutettiin kasvokkain ja keskustelut nauhoitettiin. Yksi haastatteluista tehtiin Skype-ohjelman välityksellä. Jokainen haastateltavista kertoi tuntevansa projektimallin vähintään pääpiirteittäin. Haastateltavat henkilöt ovat olleet joko suunnittelemassa projektimallia tai omatoimisesti tutustuneet malliin. Jokainen haastateltava oli kuitenkin sitä mieltä, että projektimallin ominaisuuksiin ei ole perehtytty tarpeeksi syvällisesti yhdessä projektipäällikön tai -ryhmän kanssa.

Projektimallin olemassaolo koettiin hyvänä asiana, sillä malli toimii kehyksenä projekteissa. Projektimallia tulisi kuitenkin kehittää, jotta siitä saadaan entistä hyödyllisempi. Joihinkin projektityyppisiin malli ei sovi lainkaan, joten mallia pitäisi pystyä muokkaamaan niin, että sitä voi hyödyntää erityyppisissä projekteissa. Haastatteluissa selvisi, että erityisesti niissä tapauksissa, joissa projekti on projektimallimuottiin sopimaton, projektiryhmän olisi hyvä käydä yhdessä projektimallia läpi ja miettiä, että miten malli saadaan sovitettua projektiin. Myös kehitysprojekteihin olisi hyvä saada oma mallinsa, sillä kehitysprojektia ei tehdä samalla laajuudella kuin uutta järjestelmää.

Haastateltavat olivat sitä mieltä, että malli ei näy heidän jokapäiväisessä työskentelyssään, tai sitten projektimallin toteutumista omassa työssä ei tule ajatelleeksi. Yleisesti projektimallin käyttö näkyy muun muassa niin, että konseptointi tehdään projektissa ennen toteutusvaihetta, jolloin saadaan järkky kokonaisuus toteutusvaiheeseen. Malli toimii korkealla tasolla, mutta projektimallin kaikkia osia ei ole saatu tarpeeksi hyvin käyttöön, esimerkiksi käytäntöjä. Mallin toteutumista projekteissa olisi hyvä valvoa, jotta edettäisiin projektimallin mukaan, eikä siirryttäisi seuraaviin vaiheisiin ennen kuin aika on oikea. Projektipäällikkö voisi valvoa mallin toteutumista projekteissa ja raportoida mallin toimivuudesta.

Kellään haastateltavista ei ole projektimallia tulostettuna huoneensa seinään, eikä mallin vaiheita käydä katsomassa joka päivä dokumentaatiosta. Haastateltavien mukaan projektimalli on kuitenkin projekteissa käytössä ja se jaksottaa projekteja. On hienoa, että projektimalli on dokumentoitu. On projektista kiinni, miten hyvin projektimallia noudatetaan projektin aikana. Pienemmissä projekteissa saatetaan oikoa eri vaiheissa ja vaiheita tehdä rinnakkain tai jättää niitä kokonaan pois. Vaiheiden välillä ei välttämättä ole selkeitä rajoja. Sovelluskehittäjillä saattaa olla työn alla useita eri projekteja samaan aikaan ja eri vaiheissa, mikä vaikuttaa rajojen hahmottamiseen.

Haastateltavien mukaan projektimallin yksi hyvistä puolista on se, että saadaan vähennettyä yritykseen kohdistuvaa taloudellista riskiä. Projektin alussa asiakkaalle myydään tarvekartoitus- ja konseptointivaiheet, jotta myöhemmin voidaan paremmin arvioida, millaista verkkopalvelua ollaan kehittämässä. Jos konseptointivaihe on tehty perusteellisesti, on todennäköistä, että toteutetaan sellaista järjestelmää, jonka asiakas on tilannut. Positiivista on se, että projektin aikana asioita tehdään yhdessä asiakkaan kanssa. Tämä on hyvä, sillä projektin alussa asiakas harvoin pystyy tarkasti kuvailemaan, mitä tarvitsee. Jos projektia ei alettaisi purkaa tekemällä mallia toteutettavasta sovelluksesta, sorruttaisiin ikuisen määrittelyvaiheeseen.

Iteratiivisuus koettiin haastatteluissa projektimallin vahvuudeksi. Ambientilla ei ole kuitenkaan selkeää iteraatiosykliä, jolle olisi sovittu tietty pituus. Dokumentointi on asia, jonka yksi haastateltavista nosti esiin. Ajan puutteen vuoksi se jää usein keskeneräiseksi tai jopa kokonaan tekemättä. Joskus saattaa tulla tilanteita, jossa asiakas toivoo saavansa dokumentteja

toteutettavaan verkkosovellukseen liittyen. Jos sovellusta ei ole dokumentoitu tarpeeksi tarkasti, ei dokumentteja voida toimittaa asiakkaalle. Huolellisella dokumentoinnilla tällaiset tilanteet saataisiin estettyä.

Projektimallia pitäisi kehittää eteenpäin erityisesti käytännön osalta, eikä vain projektia ohjaavan henkilön osalta. Projektimallin heikkouksia on, että se yrittää kattaa kaiken projektikokonaisuuksien ohjaamisesta käytännön työhön. Järkevämpää olisi, että projektisalkun hallinta ja käytännön tekeminen olisivat erikseen, erotettuina toisistaan. Projektimalli on haastattavien mielestä tällä hetkellä melko pienessä osassa projektin toteutuksessa. Malli ohjaa työn vaiheita ja toimii suuntaviittana, jonka avulla projekti tehdään alusta loppuun asti. Projektimalli toimii siis selkärankana projektille.

8 YHTEENVETO JA JATKOSUOSITUKSET

Prosessimallien vertailun perusteella Ambientian projektimalli on ominaisuuksiltaan hyvä. Malli ottaa projekteissa huomioon riskit ja mahdolliset muutostarpeet. Projektit pystytään aikatauluttamaan tarkemmin kuin ilman mallia, koska projekteissa edetään vaiheittain. Asiakas saa projektin varhaisessa vaiheessa nähtäväkseen konkreettisen sovelluksen. Tämän vuoksi Ambientia pystyy vastaamaan myös vaatimuksiin, jotka eivät välttämättä ole alusta asti selvillä. Asiakkaaseen pidetään projektin aikana tiiviisti yhteyttä, joten tietokatkoja asiakkaan ja toimittajan välille ei synny. Riski siihen, että Ambientia toteuttaisi verkkosovellukseen sellaisia ominaisuuksia, joita asiakas ei olisi tilannut, pienenee.

Projektien aikana verkkosovellusten dokumentointiin tulisi kiinnittää huomiota. Dokumentointi on vesiputousmallissa otettu paremmin huomioon kuin ketterissä menetelmissä. Tällä on hyvät ja huonot puolensa. Jos sovellusta ei ole dokumentoitu, tieto jää vain muutamien tai pahimmassa tapauksessa yhden ihmisen tietoon. Ajan kuluessa asiat unohtuvat tai yksittäinen henkilö saattaa esimerkiksi vaihtaa työpaikkaa. Tällaisissa tilanteissa tietous sovelluksesta on heikko. Tällaisen sovelluksen kehitystyö on haastavaa, sillä kehittäjä joutuu tutustumaan omin avuin sovelluksen ominaisuuksiin. Jos tietoa olisi saatavilla dokumenteista, kehitystyön jatkaminen olisi helpompaa.

Verkkosovelluksen suunnitteleminen Ambientian iteratiivisen projektimallin avulla osoittautui toimivaksi ratkaisuksi. Eteneminen pienissä sykleissä ja sovelluksen kasvattaminen kohti lopullista muotoaan, antaa mahdollisuuksia tarvittaessa suunnan muutoksiin projektissa. Jos toteutettavaan sovellukseen tulee muutostarpeita, niitä ei ole liian myöhäistä toteuttaa, koska sovellusta on kehitetty pienissä iteraatioissa. Koska verkkosovelluksen tilaaja on mukana jo projektin alusta asti, on pienempi riski epäonnistua sovelluksen suunnittelussa. Asiakas pystyy antamaan oman panoksensa tulevaan sovellukseen, koska on mukana suunnittelemassa sitä. Sovelluksen suunnittelu ja toteutus etenee mallin ja asiakkaan pyyntöjen mukaisesti, joten verkkosovelluksesta tulee juuri sellainen, millaisen asiakas on tilannut.

Ambientian projektimalliin liittyvät lomakekyselyn vastaukset ja haastatteluissa esiin tulleet mielipiteet olivat hyvin samankaltaisia toistensa kanssa. Kyselyn vastaukset antoivat viitteitä Ambientian projektimallin hyvistä ja huonoista puolista, mutta haastatteluissa päästiin aiheissa syvemmälle tasolle. Tämän vuoksi oli hyvä asia, että haastattelut toteutettiin. Ilman niitä tutkimus olisi jäänyt vain pintaraapaisuksi. Teemahaastatteluihin valitut henkilöt oli valittu hyvin, sillä he toivat erilaisia näkökantoja esiin. Jokainen haastateltava tunsi mallin entuudestaan ja jokaisella oli mielipiteitä projektimalliin liittyen.

Kyselyn vastausten ja haastatteluiden pohjalta voidaan sanoa, että projektimallin olemassaolo ja sen dokumentointi on auttanut projektien läpivienneissä. Malli toimii kehyksenä projekteille ja ohjaa projekteja oikeaan suuntaan. Mallin vaihteellisuus auttaa hahmottamaan projekteja, vaikka kyseessä olisi laajakin projektikonaisuus. Jatkossa Ambientian tulisi kuitenkin kehittää projektimalliaan eteenpäin. Olisi hyvä, jos tätä tarkoitusta varten perustettaisiin työryhmä, jossa olisi osanottajia eri projektiryhmistä ja työtehtävistä. Näin projektimalliin saataisiin uusia näkökulmia ja kehitystyötä saataisiin jatkettua.

Projektipäälliköiden tulisi huolehtia, että mallia noudatetaan projekteissa, jotta se ei jäisi vain käyttämättömäksi ohjeistukseksi. Projektien laatu tulisi varmasti paranemaan, jos mallia saadaan kehitettyä entistä paremmaksi. Projektimallia tulisi käydä kaikkien työntekijöiden kanssa yhdessä läpi, jotta malli tulisi tutuksi kaikille. Vaikka omassa työssään ei mallia käyttäkään, on hyvä tietää, miten yritys toimii projekteissa.

Ambientian projektimallissa on paljon positiivisia ominaisuuksia, mutta samalla myös kehitettäviä asioita. Erityisesti erilaisissa projekteissa eteneminen olisi hyvä ottaa huomioon. Vaikka malli ei ole tarkka ohjeistus siitä, miten projektissa pitää edetä, vaan se on enemmänkin kehys, olisi hyvä, että erilaisille projekteille olisi tarkemmat ohjeistukset. Esimerkiksi projektit, joissa liitetään kaksi erilaista järjestelmää yhteen tai versiopäivitysprojekti, eivät etene samalla tavalla kuin sellainen, jossa toteutetaan verkkosovellus alusta loppuun asti. Malliin olisi hyvä saada myös käytännön ohjeita ja menetelmiä, miten projekteissa toimitaan. Projektinhallinta olisi oma osuutensa ja tämän lisäksi olisi työohjeita projekteissa työskenteleville. Näin malli ei jäisi pelkäksi kehykseksi projekteille, vaan siitä olisi hyötyä jokapäiväisessä työskentelyssä.

Opinnäytetyöprosessin aikana tutustuin Ambientian projektimalliin, jota en entuudestaan tuntenut hyvin. Prosessimalli käsitteenä ei myöskään ollut kovin tuttu. Opinnäytetyötä tehdessä ymmärsin, kuinka paljon yritykselle voi olla hyötyä toimivasta prosessimallista. Prosessimallin kehittäminen on työ, joka ei lopu koskaan, sillä toimintatapoja voi aina kehittää paremmiksi. Erilaisiin prosessimalleihin tutustuminen ja niiden vertailu oli opettavaista. Kyselyn ja haastattelujen toteuttaminen oli mielenkiintoista ja niiden avulla opin vastausten analysointia.

LÄHTEET

Anderson, D.J. 2004. Agile management for software engineering. Pearson Education Inc.

Cockburn, A. 2002. Agile software development. Person Education, Inc.

Fundamentals of the V-Modell. 2006. Viitattu 7.3.2010.
<http://v-modell.iabg.de/v-modell-xt-html-english/index.html>

Haikala, I & Märijärvi, J. 2004. Ohjelmistotuotanto. Karisto Oy.

Helsingin yliopisto. Tietojenkäsittelytieteen laitos. Ohjelmistotuotanto. Harri Laine. Päivitetty 16.9.1998. Viitattu 3.3.2010.
<http://www.cs.helsinki.fi/u/laine/ot/s98/prosessi.pdf>

Helsingin yliopisto. Tietojenkäsittelytieteen laitos. Ohjelmistotuotanto. Harri Laine & Jukka Paakki. Viitattu 3.3.2010.
<http://www.cs.helsinki.fi/u/paakki/ohtuk03-luento2-bw.pdf>

Helsingin yliopisto. Tietojenkäsittelytieteen laitos. Ohjelmistotuotanto. Juha Taina. Viitattu 3.3.2010.
http://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luku-6_6.pdf

Hirsjärvi, S, Remes, P & Sajavaara, P. 2007. Tutki ja kirjoita. Otavan kirjapaino Keuruu.

Hoghsmit, J. 2004. Agile Software development ecosystems. Pearson Education, Inc.

Holcombe, M. 2008. Running an agile software development project. John Wiley & Sons, Inc.

Larman, C. 2007. Agile & iterative development: a manager's guide. Pearson Education Inc.

McConnell, S. 2002. Ohjelmistotuotannon hallinta. Edita Prima Oy.

Murch, R. 2002. IT-projektinhallinta. Edita Prima Oy.

Partanen, J. RUP. 2009. Viitattu 7.3.2010.
<http://upload.wikimedia.org/wikipedia/fi/0/0d/RUPVaiheet.JPG>

Poimala. Periaatteet. Viitattu 7.3.2010.
<http://www.ketteratkaytannot.fi/Ketteryys/Periaatteet/>

Poimala. Scrum. Viitattu 7.3.2010.
<http://www.ketteratkaytannot.fi/fi-FI/Menetelmat/Scrum/>

Poimala. XP. Viitattu 7.3.2010.

<http://www.ketterakaytannot.fi/fi-FI/Menetelmat/XP/>

LOMAKEKYSelyn SAATEKIRJE














Olen tekemässä opinnäytetyötä, jossa tutkin Ambientian iteratiivista projektimallia. Kyselyn tuloksien avulla saadaan selville, miten projektimallia oikeasti hyödynnetään käytännössä.

Olisin kiitollinen, jos vastaisit kyselyyn mahdollisimman pian, kuitenkin tämän viikon aikana.

Terveisin,

Henna Kautto

KYSELYN TULOKSET

Sukupuoli	vaihtoehto	vastaukset	
	Mies	14	77 % 
	Nainen	4	22 % 
Yhteensä 18 vastausta tähän kenttään			
Olen työskennellyt Ambientialla	vaihtoehto	vastaukset	
	1-3 vuotta	7	38 % 
	4-6 vuotta	5	27 % 
	7 vuotta tai yli	4	22 % 
	Alle vuoden	2	11 % 
Yhteensä 18 vastausta tähän kenttään			
Tiedätkö mitkä ovat Ambientian iteratiivisen projektimallin kuusi eri vaihetta?	vaihtoehto	vastaukset	
	Kyllä	10	55 % 
	Ei	6	33 % 
	En osaa sanoa	2	11 % 
Yhteensä 18 vastausta tähän kenttään			
Kuinka hyvin tunnet projektimallin eri vaiheet?	vaihtoehto	vastaukset	
	Tyydyttävästi	7	38 % 
	Melko hyvin	5	27 % 
	Huonosti	5	27 % 
	En osaa sanoa	1	5 % 
Yhteensä 18 vastausta tähän kenttään			

Tunnistatko projektimallin eri vaiheet omassa työskentelyssäsi?	vaihtoehto	vastaukset	
	Kyllä	10	55 % 
	Ei	5	27 % 
	En osaa sanoa	3	16 % 
Yhteensä 18 vastausta tähän kenttään			
Oletko käynyt projektimallia läpi yhdessä projektiryhmän/päällikön kanssa?	vaihtoehto	vastaukset	
	Kyllä	9	50 % 
	Ei	8	44 % 
	En osaa sanoa	1	5 % 
Yhteensä 18 vastausta tähän kenttään			
Kuinka tarkasti projektimallia on mielestäsi käyty läpi?	vaihtoehto	vastaukset	
	Kohtalaisen tarkasti	9	50 % 
	Ei lainkaan tarkasti	7	38 % 
	Melko tarkasti	1	5 % 
	Hyvin tarkasti	1	5 % 
Yhteensä 18 vastausta tähän kenttään			
Vaikuttaako projektimalli työskentelyysi?	vaihtoehto	vastaukset	
	Kyllä	9	50 % 
	En osaa sanoa	6	33 % 
	Ei	3	16 % 
Yhteensä 18 vastausta tähän kenttään			

Jos vaikuttaa niin miten?



Mitä hyvää projektimallissa on?



Mitä huonoa projektimallissa on?

Yhteensä 10 erilaista vastausta

Sivut: 1 2 kaikki

1 -	vastaukset: 1	5 %
2 Ei sovi moniprojektitympäristöön	vastaukset: 1	5 %
3 Vaikea sanoa näin lyhyen ambientia-uran jälkeen...	vastaukset: 1	5 %
4 Vaiheita ei toteuteta käytännössä.	vastaukset: 1	5 %
5 Tällä hetkellä se eteneminen on epäselkeä ja vaiheet eivät etene selkeästi järjestyksessä, vaan edelleen osasta projekteista on vesiputous-projekteja	vastaukset: 1	5 %
6 Siitä ei pidetä kiinni ja vaiheistusta ei ole ohjeistettu tarpeeksi hyvin	vastaukset: 1	5 %
7 Iteratiivisuus jää helposti kiireen jalkoihin.	vastaukset: 1	5 %
8 Liian monimutkainen, ei käytännönläheinen, sitä ei noudateta riittäväällä tasolla johtuen juurikin monimutkaisuudesta.	vastaukset: 1	5 %
9 Kts allat	vastaukset: 1	5 %
10 tyhjä	vastaukset: 1	5 %
11 Ei ole käytössä.	vastaukset: 1	5 %
Projektimalli voi olla ongelmallinen uusasiakashankinnassa, jos asiakas ei uskalla sitoutua konseptoinnin jälkeiseen "avoimeen shekkiin", kun kokonaisuutta ei voi määritellä. Jos lopputulosta ei määritellä kunnolla asiakkaan kanssa, saattaa johtaa epäselvyyksiin siitä, mitä projekti pitkään sisältää ja mitä ei.		
12 Saattaa johtaa (tai on joskus johtanut - vika ei välttämättä projektimallissa) sovelluksen arkkitehtuuriltaan vääränlaiseen toteutukseen, jos laajuus arvioidaan aivan väärin ennen toteutusta (tehdään pientä sovellusta, kun pitäisi tehdä valtavaa).	vastaukset: 1	5 %
13 Budjetissa pysyminen	vastaukset: 1	5 %
14 Toteuttaminen käytännössä. Jossain projektissa varmasti sopiva, mutta harvemmin löytyy tunnistettavia iteraatiokierroksia.	vastaukset: 1	5 %
15 Raskas pieneen projektiin.	vastaukset: 1	5 %
16 En tiedä	vastaukset: 1	5 %
17 Toimintamallien omaksuminen on hankalaa.	vastaukset: 1	5 %
18 Liian isot kokonaisuudet. Mallissa on osuuksia, jotka pitäisi vielä jakaa pieniin iteraatioihin.	vastaukset: 1	5 %

Miten projektimallia pitäisi mielestäsi kehittää?

Yhteensä 17 erilaista vastausta

Sivut: 1 2 kaikki

1 En osaa sanoa.	vastaukset: 2	11 %
2 Projektipäälliköille selkeästi enemmän projektipäällikön tehtäviä.	vastaukset: 1	5 %
3 -	vastaukset: 1	5 %
4 Eteenpäin	vastaukset: 1	5 %
5 Viedä paremmin käytäntöön.	vastaukset: 1	5 %
6 Koen, että ongelmana on enemmänkin projektin seuraaminen, jotta asiat menevät mallin mukaisesti, eikä muutu vesiputous-projektiksi.	vastaukset: 1	5 %
7 Tuntikertymien seurannan kehittäminen ja automaattisia ennakkovaroituksia työkalujen kehittämisen avulla PP:n avuksi. Ryhmän oma tuntikertymäseuranta. Priorisoinnin korostaminen.	vastaukset: 1	5 %
8 Tarvitsisi selkeyttä ja muodostaa sellaiseksi, että pääsee nopeasti tarkistamaan "työohjeet". Pitäisi myös käytännönläheistä, nyt turhan korkealentoinen, epäkonkreettinen. Penään siis toimintatapojen kuvauksia ja projektityöprosessia, miten projekti etenee alusta loppuun mallin mukaisesti.	vastaukset: 1	5 %
9 Kuulopuheiden mukaan sitä ei noudateta tai noudattamisessa ei käytäntöjä riippuen projektipäälliköstä. Mutta korostan, että tämä on siis kuulopuhetta.	vastaukset: 1	5 %
10 tyhjä	vastaukset: 1	5 %

Yhteensä 17 erilaista vastausta

Sivut: [1](#) [2](#) [kaikki](#)

11	Ottaa käyttöön ja oikeesti toimia iteraatioiden mukaan!! Vaiheiden välistä siirtymistä (erityisesti konsultointi ja konseptointi) pitäisi tarkentaa yhdessä eri timien välillä.	vastaukset: 1	5 %
12	Perustamisvaiheen rooli jää joskus hieman epäselväksi, kun ei tehdä jotain laajaa räätälöintiä ja integraatioita sisältävää projektia. Projektin päätös on meillä vielä epäselvä - sen voisi lisätä omaksi vaiheeksi, joka sisältää aina sovitut asiat, nyt projektien jälkeen asiakkuuden kehittäminen jää helposti hieman epäselvään tilaan. Projektin päätöskokous pitäisi tulla joka projektin jälkeen, mutta nyt niitä ei aina pidetä.	vastaukset: 1	5 %
13	Muutoshallinnan seurannalla budjetin osalta	vastaukset: 1	5 %
14	Jotta tätä voitaisiin menestyksekkäästi käyttää pitäisi toimia sen mukaisesti ja saada myös asiakas mukaan.	vastaukset: 1	5 %
15	En tiedä	vastaukset: 1	5 %
16	jaa-a	vastaukset: 1	5 %
17	Projektimallin eri osiot voisi jakaa pieniin osiin.	vastaukset: 1	5 %

TEEMAHAASTATTELU HEIKKI MALKAMÄKI

[14.4.2010 14:24:50] Henna Kautto: kuinka monta vuotta olet työskennellyt Ambientialla?

[14.4.2010 14:24:53] Heikki Malkamäki: hyvä, ettei tarvinnutkaan tohon vastata..

[14.4.2010 14:25:12] Heikki Malkamäki: hiukan yli 3 vuotta

[14.4.2010 14:25:34] Heikki Malkamäki: tai no.. jos lasketaan vain työaika eikä harjoittelu-aikaa, niin sitten tasan 3

[14.4.2010 14:26:02] Henna Kautto: Osaatko nimetä ulkoa Ambientian iteratiivisen projektimallin kuusi eri vaihetta?

[14.4.2010 14:27:19] Heikki Malkamäki: kokeillaanpa..

[14.4.2010 14:27:37] Heikki Malkamäki: kartoitus, konseptointi, toteutus, testaus, julkaisu

[14.4.2010 14:29:23] Henna Kautto: lähes täydellistä, eli aika hyvin osasitte ne nimetä

[14.4.2010 14:29:43] Heikki Malkamäki: nooh.. toi nyt on oikeastaan melko johdonmukainen

[14.4.2010 14:30:09] Henna Kautto: kuinka hyvin tunnet kunkin vaiheen sisällön eli mitä vaiheen aikana tehdään ja keitä henkilöitä vaiheessa pitäisi olla mukana?

[14.4.2010 14:32:35] Heikki Malkamäki: kartoituksessa on mukana lähinnä myyjä ja asiakas.. (ja sitten saukkonen :D)

konseptoinnissa visuaalinen suunnittelija ja projektipäällikkö. ehkä myös joku koodari. myyjä voi myös olla mukana "häiritsemässä", mutta ei aina kyllä ole.

[14.4.2010 14:32:40] Heikki Malkamäki: jatkuu vielä

[14.4.2010 14:33:54] Heikki Malkamäki: toteutuksessa on asiakas ja projektiryhmä tällä kertaa kokonaisuudessaan.

testauksessa testaus ja projektiryhmä, toisinaan myös asiakas.

julkaisussa on mukana projektiryhmä, testaus, NOC ja asiakas.

[14.4.2010 14:34:02] Heikki Malkamäki: ainiin.. sit vielä noi, että mitä tehdään

[14.4.2010 14:41:22] Heikki Malkamäki: tarvekartoituksessa nimensä mukaan selvitetään asiakkaan liiketoimintaympäristö ja tarpeet rakennettavalle järjestelmälle. kartoituksessa tuotetaan jonkinlainen dokumentaatio asiakasyrityksestä ja hankinnan kohteena olevasta järjestelmästä

konseptoinnissa rakennetaan järjestelmä ulkoisesti ja käydään läpi asiakkaan kanssa välillä useaan kertaan esim. värejä hinkaten. (jäi pois aikaisemmin, että asiakas on tässä siis myös mukana). välillä on ollut puhetta, että tässä vaiheessa tarvitsisi olla testaajankin, mutta syystä tai toisesta tähän ideaan ei oikein ole tartuttu. mielestäni idea on hyvä.

toteutuksessa järjestelmään toteutetaan varsinaiset toiminnot, eli siirrytään mock-mallista kohti toimivaa järjestelmää.

testausvaiheessa palvelu on noin periaatteessa valmis. joskin tämä ei toteudu ambientilla, koska ihmiset tekevät työnsä pääosin testaamatta niitä itse (tai tämä on minun käsitykseni). tällöin tapahtuu väistämättä niin, että testaus ja toteutus limittyvät enemmän tai vähemmän. eniveis, testausvaiheessa palvelu testataan ja virheet raportoidaan. tämän jälkeen ne korjataan = iteroidaan toteutuksen ja testauksen välillä. terve iterointi on eri asia, kuin mainitsemani limittyminen, jossa koodaaja ei vastuullisesti toteuta / korjaa asioita ja testaus joutuu sitten ikään kuin tekemään koodarin työtä vain siksi, että koodari ei ole jaksanut kertaakaan klikata tuotostaan läpi.

julkaisuvaiheessa palvelu hyväksytetään asiakkaalla ja julkaistaan näkyväksi maailmalle (verkkopalvelu) tai asiakkaalle käyttöön (esm. KC:lla paljon sovelluksia, jotka vain sisäverkossa).

[14.4.2010 14:46:14] Henna Kautto: tunnistatko projektimallin eri vaiheet omassa työskentelyssäsi?

[14.4.2010 14:56:19] Heikki Malkamäki: meni hetki, kun pankista soittivat

[14.4.2010 14:56:48] Heikki Malkamäki: tunnen kyllä vaiheet. niihin vain on sen verran tottunut, että ei niitä varsinaisesti erota toisistaan.

[14.4.2010 14:57:22] Heikki Malkamäki: lisäksi kun ollaan niin kovin ketteriä, niin pompitetaan vaiheesta toiseen ja tehdään vaiheita toisinaan myös päällekkäin, niin mitään selkeitä rajoja ei ole mitenkään havaittavissa.

[14.4.2010 14:57:44] Heikki Malkamäki: tuohon kun lisää vielä sen, että on useampi projekti työnalla ja eri vaiheessa, niin eihän siinä sillä tavalla selkeyttä ole.

[14.4.2010 14:58:09] Heikki Malkamäki: mutta jos kysytään, että missä vaiheessa jokin projekti on, niin kyllä pystyn 10 minuutin sisään vastaamaan, että mikä on tilanne :)

[14.4.2010 14:59:51] Henna Kautto: kuulostaa hyvältä, että osaisit sanoa projektin tilanteen

[14.4.2010 14:59:52] Henna Kautto: Oletko käynyt projektimallia läpi yhdessä projektiryhmän/päällikön kanssa?

[14.4.2010 15:00:17] Heikki Malkamäki: hyvin vähän

[14.4.2010 15:02:10] Heikki Malkamäki: en muista, että siitä olisi ollut yhtä ainutta erittäin selkeää läpikäynti vain sitä tarkoitusta varten. sitten on vielä niin, että aivan joka projektissa ei välttämättä vaiheet toteudu sellaisinaan, jolloin olisi todellakin tärkeää käydä malli läpi ja esimiehen toimesta kertoa, miten se sovitetaan tähän projektiin. lisäksi mielestäni tuosta mallista pitäisi jalostaa kehitysprojektia varten oma poikkeava mallinsa, koska eihän valmiiseen projektiin tehtäviä muutoksia samalla kaavalla vedetä

[14.4.2010 15:03:06 | Muokattu 15:03:10] Henna Kautto: hyviä näkökoh-
tia

- [14.4.2010 15:03:51] Henna Kautto: eli mielestäsi mallia pitäisi käydä tarkemmin läpi ja soveltaa eri projekteihin?
- [14.4.2010 15:09:09] Heikki Malkamäki: joo, mun mielestä sitä olis hyvä painottaa ylipäättään ryhmille ja valvoa sen toteutumista
- [14.4.2010 15:09:44] Heikki Malkamäki: tyyliin: ollaan tässä vaiheessa => pysytään ruodussa eikä aleta karkaileen muihin vaiheisiin, ennen kuin homma on tehty loppuun
- [14.4.2010 15:10:11] Heikki Malkamäki: tohon liittyy mun mielestä läheisesti se oman työn laadun tarkkailu, mikä käytännössä on testaamista paikallisesti itse
- [14.4.2010 15:11:11] Henna Kautto: Vaikuttaako projektimalli työskentelyysi eli ohjaako se sinua tekemään asiat tietyllä tavalla? voisit antaa esimerkin, jos keksit
- [14.4.2010 15:12:47] Heikki Malkamäki: kysymys onkin hiukan hankalampi.. käytännössä malli on aika pienessä osassa projekteissa ja pyrin vain itse tekemään työni mahdollisimman hyvin ja loppuun asti oikein, jotta se ei tule takaisin tehtäväksi.
- [14.4.2010 15:15:40] Heikki Malkamäki: mielestäni malli ohjaa enemmänkin vain työn vaiheita, jotka ovat tietysti jo oletuksena melko intuitiiviset (esim. kukaan ei tekisi testausta ennen toteutusta). malli ei siis varsinaisesti ohjaa työtäni itsessään vaan on ikään kuin suuntaviittoja, joilla löydetään alusta loppuun.
- [14.4.2010 15:17:05] Henna Kautto: Onko sinulla lisäajatuksia kysymykseen: "miten projektimalli toimii mielestäsi käytännössä?" liittyen
- [14.4.2010 15:17:32] Henna Kautto: olet jo kertonut tuohon liittyen asioita, mutta tuleeko jotain uutta mieleen?
- [14.4.2010 15:18:33] Heikki Malkamäki: kyllä projektimalli toimii, mutta sen toteutumista tulisi ehkäpä valvoa tiukemmin
- [14.4.2010 15:20:30] Henna Kautto: Mitä hyviä puolia projektimallissa on?
- [14.4.2010 15:25:51] Heikki Malkamäki: en osaa kyllä tuohon suoraan vastata.
- [14.4.2010 15:26:17] Henna Kautto: iteratiivisuus , ketteryys vs. vesiputous?
- [14.4.2010 15:26:55] Heikki Malkamäki: no juu, on se tietysti iteratiivinen ja varmasti elastisempi, kuin vesiputous. onhan se hyvin dokumentoitukin.
- [14.4.2010 15:27:58] Henna Kautto: tuleeko huonoja puolia vielä mieleen?
- [14.4.2010 15:28:17] Heikki Malkamäki: siihenkin on vastaus aika hiljainen
- [14.4.2010 15:29:51] Henna Kautto: nyt vielä mahdollisuus antaa kehitysajatuksia: Miten projektimallia pitäisi mielestäsi kehittää?
- [14.4.2010 15:30:17] Henna Kautto: ja voit kertoa vapaasti jos on vielä jotain sydämellä
- [14.4.2010 15:32:04] Heikki Malkamäki: tuohonkaan en aivan suoraan osaa mitään vastata. sen toteutumista pitäisi ehkäpä projektipäällikön seurata tarkasti ja tehdä arviota sen toimivuudesta. itse teen työt ja tarvittaessa

joustan esimerkiksi asiakkaan toiveiden mukaan. dokumentit tuotetaan kun niitä tarvitaan. esim. arkkitehtuurisuunnitelmaa tuskin on olemassa kaikista projekteista, mutta siinä on lähinnä kysymys asioiden toteutumisen valvonnasta.

[14.4.2010 15:33:41] Henna Kautto: jos tässä oli kaikki, mitä sinulla oli kerrottavanasi, niin kiitän haastattelusta (bow)

[14.4.2010 15:33:54] Heikki Malkamäki: ahaa, selvä. kiitokset :)

[14.4.2010 15:34:01] Heikki Malkamäki: ja ole hyvä

[14.4.2010 15:35:52] Henna Kautto: hyvä hyvä

OTE JARI SAUKKOSEN TEEMAHAASTATTELUSTA

Haastateltavana Jari Saukkonen ja aiheena on Ambientian iteratiivinen projektimalli.

Kuinka monta vuotta olet työskennellyt Ambientialla?

Pitää laskea, syksyllä tulee 12 vuotta täyteen.

Tiedätkö mitkä ovat Ambientian iteratiivisen projektimallin kuusi eri vaihetta?

Kyllä.

Kuinka hyvin tunnet eri vaiheet? Osaisitko kertoa, mitä asioita eri vaiheisiin liittyy?

Todennäköisesti kyllä.

Käytätkö projektimallia työssäsi?

Niin, se on eri asia, millä tasolla projektimallia noudatetaan. Projektimalli on käytössä ja suurin piirtein mennään vaiheistuksen mukaan. Tällä hetkellä se ajaa sitä, miten projekteja jaksotetaan, se ei sen syvemmälle käytännössä mene. Näinhän se menee juu.

Kuinka paljon olette käyneet mallia läpi projektiryhmän tai päällikön kanssa?

Olet itse ollut projektimallia tekemässä, joten olet varmaan paljon käynyt tätä läpi.

Sitten vastaan varmaan, että en ollenkaan.

Kuinka tarkasti projektimallia on mielestäsi käyty läpi?

Huono kysymys sinulle, koska olet itse ollut tätä toteuttamassa.

Vaikuttaako projektimalli työskentelyysi ja miten se vaikuttaa? Vaikka jokapäiväiseen työskentelyyn.

Kyllähän se näkyy projektin luonteessa, että minkälaisia asioita projektiin on palasteltu. Sillä tavalla aika hyvin, että pyritään tekemään konseptointia alkuvaiheessa, jotta saadaan järkevämpää kokonaisuutta sinne tekemispuolelle. Mutta yhden päivän aikana siitä on hankala sanoa.

Työskenteletkö samalla tavalla kuin projektiryhmät vai eroaako työskentelysi siitä?

Teen samanlaista työtä kuin projektiryhmät, mutta prosenttiosuus mitä teen on hyvin vaihtelevaa. Se on nollan ja viidenkymmenen välillä tilanteesta riippuen.

Sitten päästään näihin mielenkiintoisimpiin kysymyksiin, eli miten malli toimii mielestäsi käytännössä?

Korkealla tasolla ihan hyvin, mutta sitten... Alkuperäisen projektimallin visio, siellähän on näitä alakohtia, eli käytäntöjä ja muuta tällaista. Siinä ei ihan hirveästi niitä asioita ole saatu käytäntöön. Ihan korkean tason asioita.

Mitä hyvää mallissa mielestäsi on?

Jos lähdetään tarvekartoitusvaiheesta. Käydään ihan vaiheittain.

Ihan yleisesti mallissa on hyvää se, että saadaan vähennettyä riskiä, mikä meille kohdistuu. Eli niin tota, käytännössä jos on hyvin tehty konseptointi ja muu, niin todennäköisyys sille, että tehdään oikeasti sitä, mitä asiakas on halunnut, on aika paljon isompi. Ja sama juttu, meidän ei (kun mennään ton mukaan) yleensä tarvi alkuvaiheessa perustelemaan mitään hirveitä könttäsummia, että arvioi tästä 80 000 euron projekti nak naks vaan yleensä saadaan myytyä se tarvekartoitus ja konseptointi sinne alkuun. Ja sitten sen jälkeen aletaan tarkentamaan vasta oikeita arvioita, että mitä ollaan tekemässä ja mitä se oikeasti tarkoittaisi.

Mitä olet mieltä rakentamisvaiheesta, että päästäänkö rakentamisvaiheessa etenemään iteroiden vai tuleeko sellaisia, että ei mennäkään ihan niin?

No, sanotaan, että mennään iteratiivisuuden määritelmään, ei emme mene vesiputousmallin mukaan. Emme ensin tee ja tässä välissä ei ole mitään palautetta. Kaikki toiminta on tietyllä tapaa iteratiivista. Määritellään ensin ja kysymys on sitten enemmän siitä, että formaaleja määreitä iteratiivisuus.. esimerkiksi siitä että kun tehdään projektia, niin siinä meillä ei ole määritelty iteraation pituutta, että tehdäänkö tietyllä aikavälillä vaan niitä tota, se on yleensä projekti menee eteenpäin ja sitten iteratiiviset asiat muuttuu matkan varrella. Mutta sellaista varsinaista selkeää iteraatiosykliä meillä ei ole. Ja mä en laske iteraatioksi kahden viikon päästä olevaa palaveria.

Keksitkö projektimallista parannettavaa?

Mallia pitäisi saada vietyä pidemmälle, nimenomaan käytännön työskentelyn osalta eikä vain projektia ohjaavan osalta. On ollut hyviä keskusteluita liittyen projektin ohjausmalliin, että iteratiivisen projektimallin, mitä pari vuotta sitten säädettiin, ja pohdittiin että sen yksi ongelma on se, että se yrittää haukata sen koko palan. Ylhäältä projektikokonaisuuksien ohjaamisesta käytännön työhön. Olisi järkevämpää, että homma jaettaisiin niin, että projektisalkun hallinta olisi erikseen ja käytännön tekeminen, niillä on erilaiset vaatimukset. Ei yritetä haukata koko palaa.

TEEMAHAASTATTELU TUOMAS MARTINKALLIO

Haastateltavana Tuomas Martinkallio ja aiheena Ambientian iteratiivinen projektimalli.

Kuinka monta vuotta olet ollut Ambientialla töissä?

1,5.

Hyvin tarkasti.

Tiedätkö mitkä ovat Ambientian iteratiivisen projektimallin kuusi eri vaihetta?

Kyllä, uskoisin tuntevani mallin noin yleisesti.

Mutta periaatteessa tunnet ja tiedät vaiheet ja miten ne menee?

Jep.

Tunnistatko projektimallin eri vaiheet omassa työskentelyssäsi? Jokapäiväisessä työskentelyssä, kun teette projekteja.

Aika huonosti johtuen siitä, että projektit, ainakin tässä ryhmässä missä itse olen, projektit ovat aika pieniä ja niissä enemmän tai vähemmän oiotaan. Se ei aina periaatteessa mee sillä mallilla enemmän tai vähemmän sellaista rinnakkaista tekemistä tai joidenkin vaiheiden yli hypätään.

Osaatko sanoa sellaista vaihetta, jonka yli usein hypätään?

Eli vaiheet ovat tarvekartoitus, konseptointi, perustaminen, rakentaminen, julkaisutestaus ja käyttöönotto.

Itse mä en ole itse useinkaan mukana tarvekartoituksessa ja konseptoinnissa edes mukana, mutta esimerkiksi perustaminen ei nyt jää pois, mutta sanotaan näin, että mun kokemuksen mukaan menee aika kursiiivisesti läpi että .. se että olisi täydellinen maailma, jossa ryhmä briiffattaisiin hyvin ja ois hyvin dokumentointi ennen kuin lähdetään tekemään, on vielä toteutumatta.

Tunnetko, että projektissa mentäisiin kuitenkin iteratiivisesti eteenpäin?

Joo, kyllä.

Tuntuuko, että dokumentointi jää jalkoihin eli olisi hyvä, että projekti dokumentoitaisiin paremmin?

En tiedä johtuuko se iteratiivisesta projektimallista suoranaisesti. Se on ehkä se kulttuuri vain sellainen, että meillä ei useinkaan ole velvollisuutta tehdä dokumentointia kovin hyvin ja se mitä ei ole pakko tehdä, siitä ei varmasti saa rahaa... vaikea juttu. Jos

sain tapauksissa asiakkaat olettaa saavansa dokumentaation ja tulee hankala tilanne, kun oletuksena sitä ei tehdä kovin hyvin.

Oletko käynyt projektimallia läpi yhdessä projektipäällikön tai ryhmän kanssa?

No ei. Mulle on tää kerran käyty läpi, se oli perehdytyksessä ja se ei ollut projektipäällikkö, joka sen kävi läpi.

Jatkokysymys, että kuinka tarkasti mallia on mielestäsi käyty läpi?

Ei lainkaan tarkasti. Malli ei suoranaisesti ohjaa tekemistä. Se ei ole ohje, minkä mukaan projektia sinällään tehtäisiin. Se on prosessi, minkä mukaan projekti menee meillä läpi. Siinä on pieni ero. Siinä ei suoraan oteta kantaa, että tehkää näin, vaan siinä kuvataan, että näin tehdään.

Toivoisitko, että projektit menisivät tarkemmin mallin mukaan vai pitäisikö mallia ensin kehittää?

Se ei joissain tapauksissa kerta kaikkiaan sovi.. jossain tapauksissa se on vieras ja joissain sopimaton. Meillä tehdään hyvin laidasta laitaa projekteja..

olisi enemmänkin järkeä dokumentoida, mitä mihin pureutua teasereihin, mutta joissain tapauksissa pitäisi ilman muuta tarkemmin noudattaa, mutta toisaalta kolikon kääntöpuoli on se, että pitäiskö projektimallia sovittaa erityylyisiin projekteihin eli tavallaan skaalattas..

Osaatko sanoa, miten projektimalli vaikuttaa jokapäiväiseen työskentelyysi? Sitä käytinkin jo tuossa läpi.

Se on kehys, jonka mukaan toimitaan, että kaikilla olisi se suuri kuva, että menisi sen mukaan. Asiakkaalle kerrotaan, että näin se sitten menee. Se on sellaista selkärankaa, että voidaan asiakkaalle kertoa, että tämä on meidän toimintatapa. Sitä ei ole printattu mun huoneeni seinään, että sitä käytäisiin koko ajan läpi ja mulle kerrottaisiin, että olet nyt tässä kohtaa projektimallissa. Se on kuitenkin niin ylhäällä tasolla.

Mitä hyviä puolia mallissa on?

Varmaan, kun nimikin on iteratiivinen, niin se hyvä puoli. Meillähän on semmonen niinku toimintamalli, mikä on varmaan jonnekin kirjoitettu, että me tehdään näitä asioita asiakkaan kanssa. Asiakas ei aina pysty kauheen tarkasti kertomaan tarvettaan, vaan meidän täytyy jotenkin lähteä purkamaan sitä tekemällä, muuten sorruttaisiin ikuisen määrittelyyn. Iteratiivisuus on itsessään tärkeä juttu.

Malli itsessään on hyvä asia, että se on olemassa, vaikka siinä varmastikin on paljon kehitettävää. On kuitenkin joku pohja ja käsitys siitä, miten minun on tarkoitus tämä asia tehdä.

Joku on käynyt asian läpi, että se ei ole täynnä sudenkuoppia ja se on varmin tie tuhoon, vaan tällä voi oikeasti onnistuakin.

Mitä huonoja puolia löytyy?

Sanoinkin varmaan jo, että se on projektimalli, joka ei ota kantaa siihen, miten projekti pitäisi tehdä ja mitä asioita siinä pitäisi ottaa huomioon. Yksinkertaisia asioita, vaikka että olisi määritelty että tämän vaiheen jälkeen sulla on kädessä dokumentaatio tai tällä tasolla (ennen seuraavaa askelta). Se on hyvä siihen, mikä se on, mutta se ei ole.

Siitä puuttuu tärkeitä juttuja ilman muuta, mitkä eivät ole itsestään selviä. Ei niinkään huonoja asioita, vaan kehittämistä.

Kiitoksia haastattelusta.

Kiitos.