

Opinnäytetyö (AMK)
Tietotekniikka
Hyvinvointiteknologia
2015

Riku Viitanen

MUUTOSTENHALLINTA OHJELMISTOPROJEKTISSA



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikka | Hyvinvointiteknologia

Opinnäytetyön valmistumisajankohta 2015 | Sivumäärä 41

Ohjaaja: Teppo Saarenpää

Riku Viitanen

MUUTOSTENHALLINTA OHJELMISTOPROJEKTISSA

Ammattikorkeakoulun ammattiharjoitteluna aloitetun videoneuvottelujen varausjärjestelmän kehitystä jatkettiin yrityksen tuotantokäytön vaatimukset täyttävillä toimintatavoilla. Järjestelmän vaatimuksiin ja alustaratkaisuun tehtiin muutoksia kehitystyön edetessä ja lisäksi projektin toteuttajan rooli vaihtui projektin aikana harjoittelijasta työntekijäksi. Useat muutokset projektin aikana vaativat joustavia hallintametoja, joita ei ollut valmiiksi olemassa. Tarvittavan kokonaisuuden saavuttamiseksi yhdisteltiin ominaisuuksia agile-metodeista. Projektin aikataulu oli lopulta odotettua pidempi mutta järjestelmä saatiin käyttöön viimeisimpien vaatimusten mukaisena. Valmiissa järjestelmässä varataan videoneuvottelu [www-lomakkeella](#) ja järjestelmä luo varaukset tietokantaan sekä Exchangen kalenteriin. Varaajalle ja osallistujille lähetetään kokouskutsu ja ohjeet neuvotteluun osallistumiseen.

ASIASANAT:

sähköposti, videoneuvottelut, sähköiset lomakkeet, tietokannat, ketterät menetelmät, vaatimusmäärittelyt, vaatimusten hallinta

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technology| Health Informatics

Completion year of the thesis 2015 | Total number of pages 41

Instructor: Teppo Saarenpää

Riku Viitanen

MANAGING CHANGES IN A SOFTWARE DEVELOPMENT PROJECT

The development of the videoconference booking application begun as a trainee project and was eventually deployed in a business environment. The requirements and underlying technologies changed during development and the role of the developer changed from trainee to employee. A project with several changes needed flexible methods for project management, which did not exist. To achieve needed functionality, some aspects of different agile methods were combined. The project was finished much later than was originally intended but the deployed application was in accordance to the latest requirements specifications. In the application one books a video conference on a web form and the application creates entries to a database and Exchange calendar. The user and the participants are sent an invitation to the meeting and instructions for the videoconference.

KEYWORDS:

Email, Video conferences, eServices, Databases, Agile methods, Requirements specifications, Managing requirements

SISÄLTÖ

KÄYTETYT LYHENTEET	7
1 JOHDANTO	8
2 TEORIA	9
2.1 Yrityksen muutoksenhallinta	9
2.1.1 Configuration Management Database	10
2.1.2 Minimidokumentaatio	10
2.2 Vaatimusmäärittelyn teoria	11
2.3 Valmiin järjestelmän teoria	12
2.4 Projektin hallinnan metodien teoria	13
2.4.1 Agile-teoria	13
2.4.2 Work Breakdown Structure	14
3 VARAUSJÄRJESTELMÄN KEHITYSPROJEKTI	16
3.1 Projektin vaatimukset	16
3.1.1 Projektin vaatimukset alussa	16
3.1.2 Vaatimusmuutosten hallinta	17
3.1.3 Muutokset projektin vaatimukseen	17
3.2 Projektin toteutus	19
3.2.1 Alkuperäinen suunnitelma toteutukselle	19
3.2.2 Toteutuksen eteneminen käytännössä	19
3.3 Ohjelmiston tila harjoittelun lopussa	20
3.3.1 Valmis toiminnallisuus	21
3.3.2 Puuttuva toiminnallisuus	21
3.3.3 Virheet ohjelmistossa	21
4 PROJEKTIN KEHITTÄMINEN TYÖNTEKIJÄNÄ	23
4.1 Projektin muuttuneet vaatimukset	23
4.1.1 Jatkuva muutos	24
4.1.2 Muuttunut käyttötarkoitus	24
4.1.3 Muutokset järjestelmän käyttöympäristössä	25
4.1.4 Valitun agile-lähestymistavan soveltuvuus projektiin	26
4.2 Kehitystyön painopiste	27

4.3 Ohjelmistokehitys ja muut työtehtävät	27
4.3.1 Kokonaisuuden hallinta	28
4.3.2 Muutosten suunnittelun ryhmä	28
4.4 Ohjelmistokehityksen tukiverkosto	28
4.4.1 Yrityksen resurssien käyttö	28
4.4.2 Työkalut	28
4.5 Projektinhallinta	29
4.5.1 Projektin hallinnan tarpeet	29
4.5.2 Koodin hallinta	30
5 TUOTANTOKÄYTTÖÖN OTTAMINEN	32
5.1 Ympäristö	32
5.2 Käyttäjät	32
5.3 Testaus	32
5.4 Yrityksen ilmeeseen sovittaminen	33
6 VALMIIN OHJELMISTON SIIRTÄMINEN YLLÄPITOON	34
6.1 Ylläpidon tarpeet koodille	34
6.1.1 Kuvaus	34
6.1.2 Mallintaminen	34
6.2 Ohjeet	35
6.3 Koulutus	35
6.3.1 Käyttäjien koulutus	35
6.3.2 Ylläpidon ja tukipalvelun koulutus	36
6.4 Tiedotus	36
7 LOPPUTULOS	37
7.1 Valmiin järjestelmän kuvaus	37
7.2 Projektin onnistuminen	38
7.3 Palaute	39
LÄHTEET	40

LIITTEET

Liite 1. Alkuperäinen vaatimusmäärittely

Liite 2. Kuvaus järjestelmästä harjoittelun lopussa
Liite 3. Järjestelmän pikaohje

KUVAT

Kuva 1. WBS-puu [14]	15
Kuva 2. Laatikon sisältö jäi tarpeettomaksi.	18
Kuva 3. Tilausvahvistusten käsittely.	25
Kuva 4. Varausjärjestelmän käyttöliittymä.	37

KÄYTETYT LYHENTEET

ITIL	Information Technology Infrastructure Library
CAB	Change Advisory Board
RFC	Requests for Change
CMDB	Configuration Management Database
CI	Configuration Item
XP	Extreme Programming
DoD	Definition of Done
WBS	Work Breakdown Structure
IIS	Internet Information Services
POP	Post Office Protocol
TFS	Team Foundation Server

1 JOHDANTO

Työn aiheena on ammattikorkeakoulun ammattiharjoittelussa aloitetun videoneuvottelujen varausjärjestelmän kehitystyön metodien muokkaaminen yrityksen tarpeisiin sopivaksi. Toteutettu järjestelmä on tarkoitettu videoneuvotteluiden varaukseen ja varaustietojen sekä ohjeiden toimittamiseen osallistujille. Harjoittelun lopussa valmiina oleva versio ei ollut täysin toimiva eikä enää sovellettuun tarkoitettuun käyttöön muuttuneiden vaatimusten vuoksi. Vaatimuksia muutettiin projektin aikana, videoneuvottelualustan vaihtumisen seurauksena. Vaatimukseen oli alun perinkin tarkoitus tehdä korjauksia ja tarkennuksia projektin edetessä. Alkuperäisistä vaatimuksista ei lopuksi ollut jäljellä juuri käyttöliittymän runkoa enempää, ja videoneuvottelujen tekniikan uudistuksen myötä koko varausprosessi muuttui täysin erilaiseksi. Harjoittelussa suuren ajan vienyt sähköpostiliikenteen tekstien automaattinen tulkinta ja niihin reagointi poistuivat kokonaan ja tilalle tuli suora yhteys resursseihin. Lopputulos on käyttäjälle huomattavasti käytettävyydeltään parempi ja toimintavarmuuskin parani uuden tekniikan myötä.

Opinnäytetyössä kuvataan järjestelmän toteutuksen eri vaiheet, kehityksen aikana tulleet muutokset vaatimukseen ja toteutukseen sekä näiden muutosten hallintaa. Muutoksia ja vaatimuksia tarkastellaan lähemmin ja esitellään niiden hallinnan taustalla oleva teoria ja prosessit. Vaatimusmäärittelyn prosessien lisäksi esitellään projektinhallinnan ja kehitystyössä käytettyjen työtapojen teoriaa. Pääasiassa projekti on toteutettu agile-malleja hyödyntäen, mitään tiettyä kokonaisuutta tarkkaan noudattamatta.

2 TEORIA

Tuotaessa projektia yritysmaailmaan tulee huomioida projektin moneen osaluueeseen liittyviä asioita. Tässä luvussa on kuvattu kyseisistä asioista suurimpia ja lisäksi projektissa alusta asti käytettyä teoriaa. Järjestelmä toteutettiin Medbit Oy:lle, joka tuottaa ICT-palveluita terveydenhuollon ja sosiaalipalveluiden tarpeisiin. Osa kuvatuista prosesseista on yleisesti Medbitillä käytössä tai tämän käyttöön räätälöityjä.

2.1 Yrityksen muutoksenhallinta

Yrityksen muutoksenhallintaprosessi on peräisin Information Technology Infrastructure Librarysta (ITIL), vaikka muutoksenhallinta itsessään on ollut olemassa jo sitä ennen. ITIL on tietohallinnon parhaiden käytäntöjen kokoelma, joka ei ole yleisesti saatavilla oleva resurssi, vaan se on hinnoiteltu yritysten ostettavaksi ja sitä myydään kirjoina ja koulutuspaketteina. Koska virallista ITIL-materiaalia ei ole kattavasti käytettävissä, tässä työssä käytetään muita lähteitä, joissa käsitellään ITILiä, ja kuvataan yrityksen käyttämä muutoksenhallinnan prosessi.

Muutoksenhallinnan kohteita ovat kaikki yrityksen käytössä olevat sovellukset ja niiden osat, kuten ohjelmistot ja palvelimet. Esimerkiksi käytössä oleva ajanvarausjärjestelmä voi sisältää ohjelman, käyttöliittymän palvelimen ja tietokantainstanssin. Sovelluksen kaikki kohteet on liitetty toisiinsa, jotta voidaan analysoida suunnitellun muutoksen vaikutukset olemassa oleviin kohteisiin. Tarkoituksena on varmistaa uuden kohteen tai käytössä oleviin kohteisiin tehtävän muutoksen vaikutusten yhteensopivuus käytössä olevien kohteiden kanssa. Varmistetaan, että työ on tehty vaadittujen prosessien mukaisesti ja ettei muutoksella aiheuteta häiriötä käytössä oleviin kohteisiin. Muiden kohteiden toiminnan säilyvyyden lisäksi muutoksella on oltava johdon hyväksyntä, ja siitä on oltava yritykselle jotain hyötyä ennen kuin se hyväksytään. Hyödyksi yritykselle katsotaan parantunut palvelu, kasvavat tulot tai vältetyt menot.

Change Advisory Board (CAB) käsittelee viikoittaisissa kokouksissaan kaikki muutosehdotukset eli Requests for Change (RFC). Ensimmäinen CAB-käsittely uudesta kohteesta tehdään ennen kuin toteutusta on aloitettu. Ensimmäisessä käsittelyssä myönnetään hyväksytylle kohteelle valmistelulupa, jonka jälkeen varsinaisen toteutuksen voi aloittaa. Uusi käsittely tarvitaan viimeistään ennen käyttöönottoa, jolloin tarkastellaan tehtyä työtä ja kohde käsitellään päivitettyillä tiedoilla. [1]

2.1.1 Configuration Management Database

CMDB sisältää kaikki yrityksen tuotanto- ja testikäytössä olevat kohteet eli Configuration Items (CI). Lisäksi tietokannassa on tieto kaikista muista kohteista, jotka yritys haluaa pitää muutoksenhallintansa piirissä. Yleisesti kohteina on kaikki yrityksen tietohallinnon alainen toiminta. [2] [3]

Tietokantaan tehtävälle sovelluskortille kirjataan järjestelmän tiedot ja siitä luodaan liitokset muihin järjestelmään liittyviin kohteisiin. Tavallisia liitettuja kohteita ovat alustarakaisut ja palvelimet mutta liitosten verkko kasvaa sovelluksen monimutkaistuessa. [2] [3]

2.1.2 Minimidokumentaatio

Medbitin minimidokumentaatio koskee kaikkia CMDB:n sovelluksia. Dokumentaatioon kuuluvat vähintään arkkitehtuurikuvaus, asennusohje, pikaohje ja dokumentti ylläpidolle. Dokumenttien tarkoitus on tukea muutoksenhallintaa ja mahdollistaa jatkuva palvelu. Kun muutoksenhallinnan kohteeseen tehdään muutoksia, tulee myös kohteen dokumentaatio päivittää vastaamaan kohteen uutta tilaa. Alkuperäinen julkaisuversio ja huomattavilla muutoksilla päivitetty dokumentti on hyväksyttävä kohteesta vastuussa olevan ryhmän kokouksessa. Yrityksen kannalta pienen merkityksen ja muihin kohteisiin nähden vaikutukseltaan vähäisen kohteen dokumentoinnissa voidaan dokumentointi tehdä rajatulla sisällöllä, jolloin suurin osa tuotetusta sisällöstä on pikaohjeessa. Pika-

ohje on tarkoitus tehdä niin että suurin osa odotettavissa olevista jatkuvan palvelun tiedon tarpeista täyttyy. Pikaohjeessa ei kuitenkaan saisi olla tarpeetonta tietoa, ja sen on oltava mahdollisimman tiivis. [1]

2.2 Vaatimusmäärittelyn teoria

Vaatimusmäärittelyn tehtävä on kuvata haluttu lopputulos niin tarkasti ja yksiselitteisesti, että sen perusteella kehitetty ratkaisu on asiakkaan tarpeen mukainen. Lisäksi vaatimusten toteutuminen on oltava todennettavissa testeillä tai mittauksilla. Lopputuotoksen lisäksi vaatimusmäärittelyssä kuvataan projektilta vaadittavat muut asiat ja mahdollinen jatkuva palvelu. [4]

Vaatimusmäärittelyssä tulee jättää tilaa toteuttavalle osapuolelle, tehdä parhaaksi katsomiaan ratkaisuja mutta vaatimusten on kuitenkin varmistettava ratkaisun sopivuus käyttöön. Vaatimuksissa ei siis ole tarkoitus eritellä kehityksessä käytettäviä yksittäisiä tekniikoita tai sitoa kehitystä ennalta määrättyyn ratkaisuun, joka ei välttämättä ole toteutushetkellä paras mahdollinen, ellei sen käyttö ole jostain syystä välttämätöntä. Tilaa jätetään, koska toteuttavalla taholla pitäisi aina lähtökohtaisesti olla parempi tekninen taitotieto tehtävästä ratkaisusta, kuin asiakkaalla. Jos toteuttavalle taholle jätetään liikaa tilaa omille ratkaisuilleen, mahdollistetaan tahallinen vaatimusmäärittelyn värinkäyttö ja tilaajan kannalta huonot ratkaisut toteutuksessa, joten vaatimusten on oltava hyvässä tasapainossa kaikki näkökulmat huomioon ottaen. Vaatimusten on oltava yksiselitteisiä ja kaikkien osapuolten on ymmärrettävä ne samalla tavalla. Määrittelyssä tulee välttää subjektiivisesti tulkinnanvaraisia ilmaisuja ja käyttää todennettavissa olevia määreitä. [4]

Vaatimusmäärittely jäädytetään perinteisessä projektissa jo mahdollisimman aikaisessa vaiheessa, vaikka muutokset ovat yleisiä. Ketterissä projekteissa, joissa käytetään esimerkiksi XP-metodia, vaatimusmäärittelyä muokataan kehityksen edetessä. Jos projektissa aiotaan muuttaa vaatimuksia kehitystyön aikana, tarvitaan muutosten käsittelylle käytäntö. Tässä projektissa käytetty malli pohjautuu Model Stormingiin. Se on Just In Time -mallinnusta, eli asiat käsitel-

lään tarpeen mukaan. Kun tunnistetaan muutostarve, kootaan asian käsittelyyn tarvittava osa tiimistä ja tarvittaessa asiakkaan edustaja, joiden kanssa järjestetään pikainen kokous. Kokoukset kestävät yleensä vain muutaman minuutin. Kokouksessa osallistujat kokoontuvat käytetyn välineen, joka voi olla esimerkiksi valkotaulu, ympärille ja asiaa mallinnetaan yhdessä, kunnes tarvittava ymmärrys asiasta on saavutettu. Tämän jälkeen kaikki osallistuneet voivat palata omiin töihinsä. Näin varmistetaan, että asiakkaan edustaja ymmärtää esittämiensä muutosten seuraukset ja tiimillä on oikea kuva tarvittavista muutoksista. [5]

2.3 Valmiin järjestelmän teoria

Valmiin määritelmä (eli definition of done (DoD)) mahdollistaa kehitystyön ja projektin hallitun päätöksen oikeaan aikaan. Ilman tarkkaa kuvaa siitä, milloin ollaan valmiita, saattaa kehitys jäädä vajaaksi tai jatkua tarpeettoman pitkään. Jos kehitys päätetään liian aikaisin, ylläpidon kustannukset kasvavat, kun joudutaan korjaamaan järjestelmään jääneitä virheitä. Jos kehitystyötä jatketaan pidempään kuin olisi tarpeen, maksetaan turhasta työstä. [6]

Valmiin määritelmä on tärkeä osa monia agilemetodeita, kuten Scrumia. Siinä korostetaan kaikkien kanssa yhdessä sovittujen ja esimerkiksi tarkistuslistana toteutettuna dokumenttina saatavilla olevan kuvausten roolia. DoD:n formaatti ei ole ennalta määrätty mutta sen toteutus on luontevaa tehdä yksinkertaisena tarkistuslistana tai käyttäjätarinoiden avulla. Listan alkioden ei ole tarkoitus olla vaatimusmäärittelyn kohtia, vaan sillä otetaan kantaa työn laatuun. DoD listalla voi olla esimerkiksi tietyn testiajon virheetön tulos tai hyväksyttäväksi katsottu määrä tietynlaisia virheitä. Kun valmiin määritelmä täyttyy, siirrytään seuraavaan tehtävään. Scrumissa DoD määrää voidaanko tietty tehtävä katsoa tehdyksi sprintin aikana. Projektilla kokonaisuutena voi olla erikseen oma tarkistuslistansa tai projekti voidaan katsoa valmiiksi vasta kun kaikki osat ovat valmiita. [6]

Valmiin määritelmä muuttuu kehitystyön aikana. Siihen lisätään tarpeelliseksi havaittuja kohtia tai määritelmää muokataan paremmin tarkoitukseen sopivaksi, jos esimerkiksi vaatimusmäärittely tai ympäristö muuttuu. Se on pidettävä realistisesti toteutettavissa olevana ja riittävän kattavana myös muuttuvassa ympäristössä. DoD pidetään ajan tasalla niin että sitä voisi käyttää vaikka auditoinnin tarkistuslistana. [6]

2.4 Projektin hallinnan metodien teoria

Projektin hallinnan tulee mahdollistaa nopea kehitys, jossa keskitytään olennaiseen ja reagoidaan muutoksiin. Tällä hetkellä on suosittua käyttää ketteriä menetelmiä, joista monissa on tässä projektissa tarvittavia ominaisuuksia. Suosituimmat menetelmät kattavat vain osan kokonaiskuvasta ja ne on kehitetty tiimien työkaluiksi. Vaikka osassa menetelmistä tiimin koko voi olla hyvinkin pieni, lähes kokonaan yksittäisen kehittäjän toteuttamassa projektissa ei voi käyttää niistä mitään sellaisenaan. Useissa metodeissa on kuitenkin hyviä ominaisuuksia ja käytäntöjä, joita yhdistelemällä saadaan yksittäiselle kehittäjälle sopiva kokonaisuus. Tässä luvussa kuvataan näitä osia. [7]

2.4.1 Agile-teoria

Agile-ohjelmistokehitys on ryhmä metodeita, joita käytetään tuotannon nopeampaan ja joustavampaan ohjaamiseen. Agilella tarkoitetaan käytännön yhteisissä usein yhdistelmää siihen luettavien metodien ominaisuuksista. Agile mainittiin ohjelmistokehityksen yhteydessä ensimmäisenä Ketterän ohjelmistokehityksen julistuksessa (Agile Manifesto) mutta sen juuret ovat 80-luvulla. Julistuksessa listataan ketterän kehityksen periaatteet, eikä se itsessään ole käytönotettava metodi. Ketteriä metodeja on useita ja niistä mainitaan usein esimerkkinä Scrum. Scrumia opetetaan kouluissa esimerkkinä Agilesta mutta sitä käytetään teollisuudessa, puhtaan muotonsa lisäksi, paljon tiettyyn tarpeeseen muokattuna. Muokattua Scrumia kutsutaan nimellä ScrumBut eli Scrum mutta tietyillä muutoksilla. Tässä työssä on hyödynnetty Agilen nimissä Scrumin lisäksi

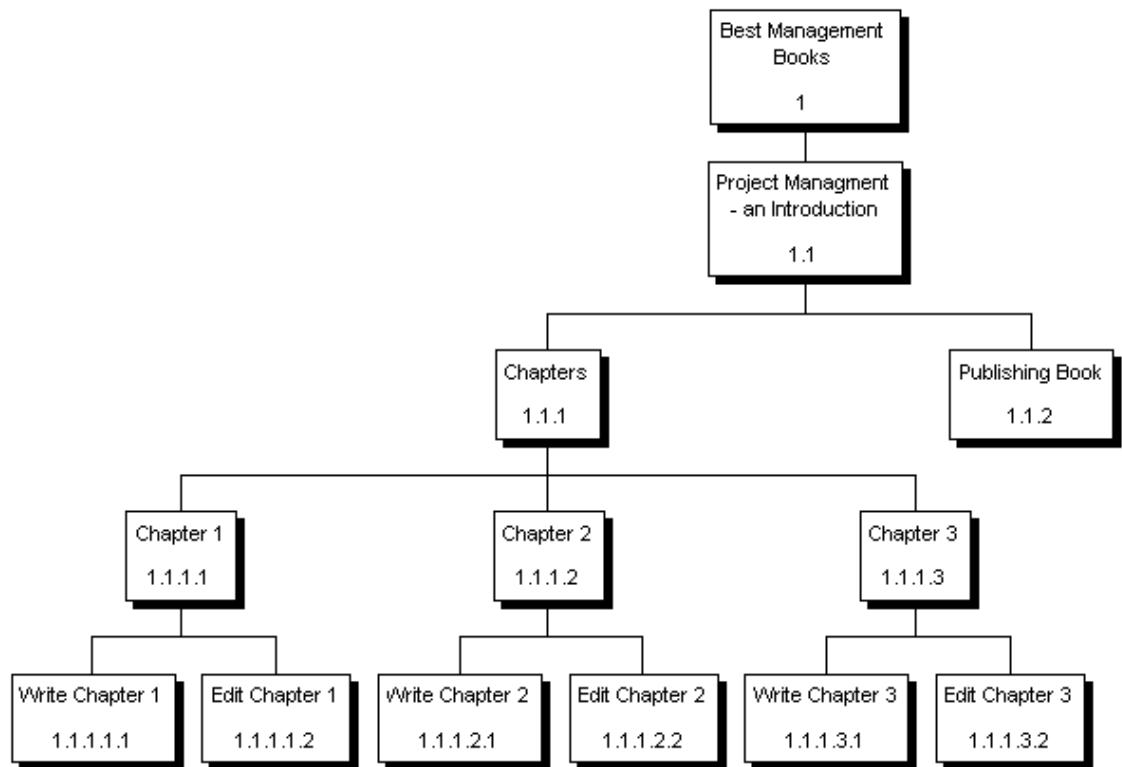
si ominaisuuksia mm. Extreme Programmingista (XP), joka tähtää todella nopeaan kehitykseen ja muuttuviin vaatimuksiin mukautumiseen. [8] [9] [10]

Agile-metodit ovat projektia tekevän tiimin työkaluja, eikä niistä mitään voi sellaisenaan soveltaa projektiin, jota tekee pääasiassa vain yksi henkilö.

Extreme Programming sisältää 12 käytäntöä, joista tässä yhteydessä lainattiin kahta. Olennaisimpana tässä yhteydessä on ohjelmakoodin rakenteen jatkuva tarkastelu ja muuttaminen tarvittaessa. Kun vaatimukset muuttuvat, uudet ominaisuudet ja vanhojen moduulien muutokset eivät enää sovi alkuperäiseen kehykseen. Tarvittaessa koodia uusitaan, ja sitä parannetaan jatkuvasti. Toisena ominaisuutena lainattiin asiakkaan edustajan jatkuvasti käytettävissä oleminen. Asiakkaan edustaja on koko kehityksen ajan tiimin kanssa samoissa tiloissa ja osallistuu tiiviisti kehitystyöhön. [11] [12] [13] [14]

2.4.2 Work Breakdown Structure

WBS on osiin jaettu työ. Ylätason käsite jaetaan ensin osiin, joita jaetaan taas osiin. Osien jakamista jatketaan kunnes on saavutettu hallittavissa olevan kokoisia kohteita. Tulos on kuvan 1 mukainen puurakenne, jossa on listattu kaikki projektissa toteutettavat asiat. [16]



Kuva 1. WBS-puu [15]

Kohteiden ei ole tarkoitus olla tehtäviä, vaan tehtävien tuotoksia, eli tuotteita, dataa, palveluita tai näiden yhdistelmiä. Riittävän pieneksi jaetuista kohteista on kuitenkin helppo tuottaa tarvittava tehtävälista, ja koska WBS sisältää koko projektin kaikki tuotokset, siitä on johdettavissa kattava lista tehtävistä. [15] [16]

3 VARAUSJÄRJESTELMÄN KEHITYSPROJEKTI

Medbitillä aikaisemmin käytössä olleen manuaalisen sähköpostipohjaisen videoneuvottelujen varauskäytännön tilalle haluttiin automaattinen varausjärjestelmä. Järjestelmä annettiin tehtäväksi Turun Ammattikorkeakoulun ammattiharjoittelussa toteutettavaan projektiin. Harjoittelun päätyttyä järjestelmää toteuttanut henkilö jatkoi projektia yrityksen työntekijänä.

3.1 Projektin vaatimukset

Projektin vaatimukset ilmoitettiin pääpiirteittäin jo järjestelmän toteuttajaa haettaessa. Osa vaatimuksista kuitenkin muuttui jo heti alussa ja muutoksia tehtiin jatkuvasti koko projektin ajan. Kun harjoittelu muuttui työsuhteeksi ja käyttöön otettavan version kehitys alkoi, tehtiin vaatimukseen samalla sekä toiminnallisuuden että käytettävien prosessien osalta suuria muutoksia.

3.1.1 Projektin vaatimukset alussa

Tehtävänannossa, harjoittelun alussa, oli jo mukana alustava vaatimusmäärittely. Ensimmäinen määrittelypalaveri tarkensi vaatimuksia jonkin verran. Vaatimuksia mietittiin tussitaulun avulla ja kuva taululle kirjatusta vaatimuksista toimi alkuperäisen vaatimusmäärittelyn lisänä.

Toteuttajan ohjelmointitaidot olivat alkuvaiheessa nettisivujen osalta puutteelliset, eikä vaatimuksissa osattu arvioida tiettyjen vaatimusten toteutuskelpoisuutta olemassa olevilla resursseilla. Vaatimukset painottuivat suurelta osin käyttööntymisen toiminnallisuuteen, ja esimerkiksi dynaamisesti etenevä varauslomake kirjattiin haluttuihin ominaisuuksiin.

3.1.2 Vaatimusmuutosten hallinta

Muutokset vaatimukseen ilmoitettiin useimmiten niin että projektin omistaja kertoi suullisesti tai kirjallisesti projektiryhmälle uusista tai muuttuneista vaatimuksista. Projektin omistajan edustajana tässä yhteydessä oli projektia ohjaava palvelupäällikkö ja toisinaan varausten kohteena olevasta palvelusta vastaava taho. Useimmiten muutoksista keskusteltiin välittömästi tai mahdollisimman pian. Käytännössä uudet tai muuttuneet vaatimukset kirjattiin projektiin usein vapaaehtoisten keskusteluiden pohjalta. Useimmiten muutokset eivät olleet pakollisia vaatimuksia mutta niitä toteutettiin kuitenkin mahdollisuuksien mukaan hyvin kattavasti.

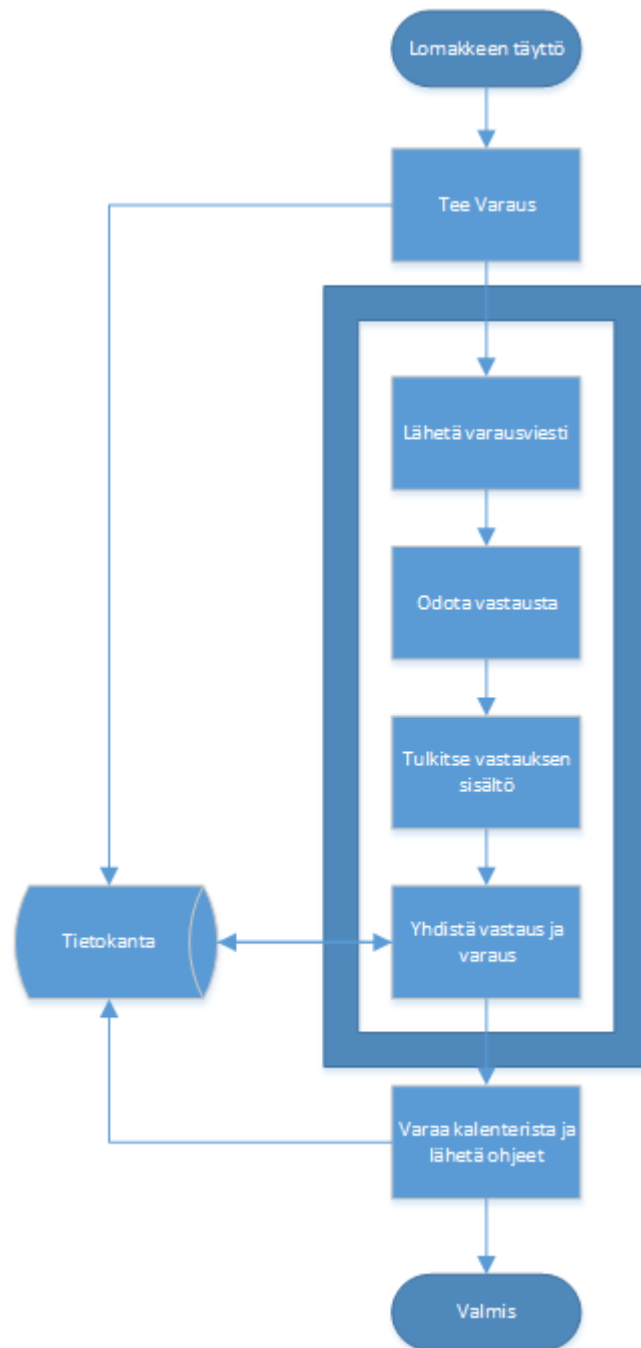
Kun muutosten työmäärä arvioitiin pieneksi, ne toteutettiin välittömästi tai kirjattiin paperille tilapäiseen tehtävälistaan, jonka tehtävät toteutettiin mahdollisimman pian. Suuremmat muutokset kirjattiin varsinaiseen tehtävälistaan, josta tehtävät poimitaan kun niiden toteuttaminen on projektin etenemisen kannalta ajankohtaista.

3.1.3 Muutokset projektin vaatimukseen

Osa muutoksista, kuten esimerkiksi muutoksenhallintaprosessi, oli suoraa seurausta yrityksen kehitysympäristöön siirtymisestä. Muutoksia syntyi myös ajallisesti pitkäksi muuttuneen kehitystyön seurauksena, kun käyttötarpeeseen ja itse varausten kohteena oleviin videoneuvotteluihin tehtiin päivityksiä ja muutoksia.

Suurimmat muutokset vaatimukseen kehitystyön näkökulmasta tulivat, kun videoneuvottelut siirtyivät yrityksen omiin resursseihin. Neuvottelut varattiin aluksi kolmannelta osapuolelta sähköpostilla ja viestien tulkinta oli suuressa osassa alkuperäisissä vaatimuksissa. Uusien resurssien myötä sisään tulevan sähköpostin sisältöä ei tarvinnut enää tulkita mitenkään ja siihen toteutettu moduuli jäi kokonaan tarpeettomaksi. Toinen suuri haaste toteutuksessa oli tilausten ja varausten yhdistäminen. Palveluntarjoajalla ei ollut viestiliikenteessään käytössä

mitään varauksen yksilöivää tietoa, joten ongelma oli ratkaistava ohjelmallisesti. Uusien resurssien myötä ei tarvittu enää erillistä tilausvaihetta ja tämäkin moduuli poistettiin tarpeettomana. Kuvassa 2 on nähtävissä muutosten seurauksena pois jääneiden tehtävien ja käyttöön jäävien tehtävien määrän suhde.



Kuva 2. Laatikon sisältö jäi tarpeettomaksi.

3.2 Projektin toteutus

Ketterän ohjelmistokehityksen julistuksessa ohjataan arvostamaan ”Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa” ja ”Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota”. Jälkimmäisiä asioitakin pitäisi kuitenkin toteuttaa mahdollisuuksien mukaan. Alkuperäisestä suunnitelmasta ei ollut lopullisessa toteutuksessa kuitenkaan paljoa jäljellä ja dokumentaation kattavuus ei ollut riittävä edes yrityksen minimidokumentaation mukaisesti. [11]

3.2.1 Alkuperäinen suunnitelma toteutukselle

Toteutuksessa päätettiin käyttää Agile-lähestymistapaa, jolloin projektisuunnitelmaan tehdään jatkuvasti muutoksia kehityksen edetessä. Koska vaatimuksiin oli tulossa muutoksia eikä projektin työkuormaa osattu alussa arvioida, ei projektisuunnitelmaan tehty tarkkaa aikataulua. Työ suunniteltiin aloitettavaksi toiminnan kannalta välttämättömistä osista ja mahdollisimman aikaisessa vaiheessa toimivaan prototyyppiin pyrkien.

3.2.2 Toteutuksen eteneminen käytännössä

Projekti aloitettiin tutustumalla mahdollisiin toteutustapoihin ja tekniikoihin. Vaatimusmäärittely käsiteltiin ensimmäisessä kokouksessa ja sovittiin vapaamuotoisesti projektisuunnitelmasta. Ennen ohjelmakoodin kirjoittamista mallinnettiin aiottu ratkaisu olennaisimmilla kaavioilla. Kehityksen edetessä sovittiin pidettäväksi säännölliset tilannekatsauskokoukset, joissa varmistetaan projektin etenevän oikeaan suuntaan. Lisäksi asiakkaan edustaja oli jatkuvasti tavoitettavissa.

Ensimmäinen toteutuskohte oli käyttöliittymän luonnosversio. Toisena kohteena oli tietokannan pohjatoteutus, josta Visual Studio generoi käyttöliittymän aikaisemmin tehdyn luonnosversion tilalle. Seuraavaksi siirryttiin sähköpostiliiken-

teen toteutukseen. Sähköpostin toteutuksen yhteydessä luotiin automatisoitu testi sähköpostimoduulille. Pohja kaikelle toiminnallisuudelle oli tässä vaiheessa jo olemassa ja toiminnassa mutta ratkaisut olivat suurelta osalta tilapäisiksi tarkoitettuja. Vaikka alkuperäiset vaatimukset olisivat täyttyneet, kehitystyötä jatkettiin käyttökelpoisemman järjestelmän saavuttamiseksi.

Sähköpostiliikenne piti siirtää tilapäiseksi tarkoitettusta ympäristöstä Exchangeen ja tietokannat palvelimille. Tässä vaiheessa projektia odotettiin alustojen käyttöönottoja ja asennettiin yrityksen kehitysympäristö. Ohjelmointityö ei näistä syistä edennyt useaan päivään. Ympäristö saatiin käyttöön juuri ennen projektin siirtymistä harjoitteluprojektista yrityksen projektiksi.

Palvelinten tietoliikenneyhteyksien avaukset ja niihin liittyvät erikoistapaukset vaativat useita tukipyyntöjä ja projektin valmistuminen lykkääntyi. Samalla projektin rinnalle tuli muita työtehtäviä ja projektiin käytettävissä oleva aika supistui. Palvelinten käyttöönoton jälkeen projektissa keskityttiin uusien ja muuttuneiden vaatimusten edellyttämien muutosten toteuttamiseen ja kokonaisuuden julkaisukelpoiseksi saattamiseen.

3.3 Ohjelmiston tila harjoittelun lopussa

Projekti oli harjoittelujakson aikana täyttänyt alkuperäisen vaatimusmäärittelyn vaatimukset, mutta tuotantokäyttöön ei riittänyt toteutettu ratkaisu. Sähköpostiliikenteen vaatimukset muuttuivat ennen harjoittelun loppua ja tästä seurannut muutostyö oli kesken. Koska sähköpostin käsittely ei toiminut, kokonaisuus ei ollut senhetkisten vaatimusten mukainen.

Aikaisemmin kehitetty versio, jossa sähköpostitoiminnallisuus täytti alkuperäiset vaatimukset, olisi ollut ensimmäisten vaatimusten mukainen. Tässäkin versiossa ilmeni testauksessa virheitä mutta ilman vaatimusten muutoksia, järjestelmä olisi ollut valmis ja korjaukset olisi voitu tehdä julkaisun jälkeen.

3.3.1 Valmis toiminnallisuus

Harjoittelun aikana, alkuperäisten vaatimusten mukaiseksi toteutetussa versiossa, oli tilapäiseksi tarkoitettuja ratkaisuja. Sähköpostiliikenne oli toteutettu Gmail-pohjaisena ja tietokanta oli kehityskoneen paikallinen kanta, jonka kehitysalusta oli generoinut. Käyttöliittymäkin oli kehitysalustan generoima ja vaikka se täytti vaatimukset, se oli ulkonäöltään ja käytettävyydeltään selkeästi kehitysversion oloinen. Kaiken vaatimusmäärittelyssä pakollisena vaaditun toiminnallisuuden toteutus oli kuitenkin olemassa, vaikka kokonaisuutta ei voinut katsoa julkaisukelpoiseksi.

3.3.2 Puuttuva toiminnallisuus

Vaatimusmäärittelyssä oli joitakin ominaisuuksia, jotka eivät olleet pakollisia. Kehitystyössä priorisoitiin järjestelmän toiminnan kannalta välttämättömät asiat. Tarkoitus oli saada toimiva prototyyppi mahdollisimman aikaisessa vaiheessa ja optionaaliset vaatimukset eivät olleet toiminnan kannalta välttämättömiä. Pakollisten vaatimusten toteutuksen yhteydessä huomioitiin mahdollisuuksien mukaan myös optionaaliset vaatimukset mutta niitä ei yritetty toteuttaa erikseen ennen ensimmäisen prototyypin valmistumista.

3.3.3 Virheet ohjelmistossa

Ensimmäisessä prototyypissä oli paljon tunnettuja vikoja. Järjestelmän käyttö edellytti ohjeiden tarkkaa seuraamista, eikä käyttäjän virheitä käsitelty oikein. Väärin täytetty lomake saattoi viedä varausprosessin osittain valmiiksi ja generoida tietokantaan merkinnän tai sähköpostiviestin, jotka mahdollisesti haittaavat kyseisen tilauksen oikein tekemistä seuraavalla kerralla. Ongelmia oli varsinkin loppukäyttäjien käyttämän vanhan selaimen kanssa, joka ei tue html5 toiminnallisuutta.

Loppukäyttäjälle ei annettu selkeää virheilmoitusta epäonnistuneesta varauksesta, vaan näytettiin tarkka selvitys kooditasolla, koska sitä tarvittiin kehitystyössä. Tässä vaiheessa ei ollut vielä jakoa kehittäjän ja muiden käyttäjien virheilmoituksissa. Julkaisuversiossa näytetään loppukäyttäjälle informatiivisempi virheilmoitus, vaikka siihen ei pitäisi enää päästä edes vanhoilla selaimilla.

4 PROJEKTIN KEHITTÄMINEN TYÖNTEKIJÄNÄ

Projektin kulku jakautuu monessa suhteessa kahteen osaan. Alku toteutettiin työharjoittelijana ja loppu palkallisena työntekijänä. Työntekijäksi siirtyminen mahdollisti virallisemman roolin. Yrityksen resurssien käyttö oli aikaisemmin ollut yksittäisten työntekijöiden aikataulujen ja auttamishalukkuuden varassa mutta työntekijänä tarvittavan resurssin voi tilata asianomaiselta tiimiltä. Toisaalta työntekijänä on vastuussa yrityksen käytäntöjen ja prosessien noudattamisesta, joka tuo huomattavan työmäärän koulutyönä toteutettavaan harjoitteluprojektiin nähden.

Harjoittelija-statusella pyydyt toimenpiteet yrityksestä irralliseen projektiin ratkaistaan usein tilapäisillä ratkaisulla, kun taas työntekijän tilaukset yrityksen sisäiseen projektiin toteutetaan yrityksen käyttöön sopivalla pysyvällä ratkaisulla. Tämän seurauksena työntekijäksi siirtyminen avasi mahdollisuuden lopullisten ratkaisuiden toteuttamiseen ja samalla loi tarpeen tilapäisratkaisuiden korjaamiseen suunnittelusta lähtien. Kuvaus järjestelmän toiminnasta, pian työntekijäksi siirtymisen jälkeen valmistuneessa versiossa, on liitteessä 2.

4.1 Projektin muuttuneet vaatimukset

Suurin muutos prosessinäkökulmasta tuli käyttöönoton tullessa ajankohtaisemmaksi. Yrityksessä oli otettu käyttöön ITIL ja siihen pohjautuva muutoksenhallintaprosessi. Tätä projektia ei viety alussa muutoksenhallintaan prosessin mukaisesti yrityksen toiminnasta irrallisen toteutuksen luonteesta johtuen, ja kaikki tarvittavat muutoksenhallinnan toimenpiteet oli tehtävä jälkikäteen, ennen käyttöönottoa.

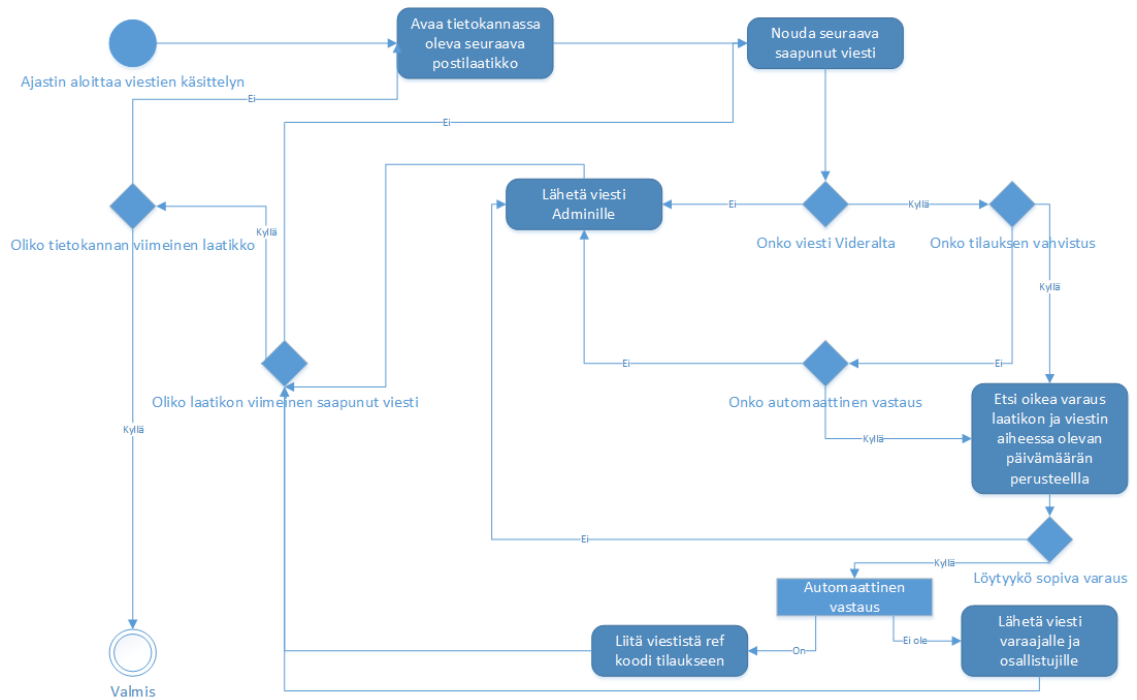
Samoihin aikoihin prosessimuutosten kanssa ilmeni tarve suurille muutoksille toiminnallisiin vaatimuksiin. Tämä tarve tuli varausten kohteena olevien resurssien muutoksista mutta ajoitus osui samaan aikaan prosessimuutosten kanssa.

4.1.1 Jatkuva muutos

Projektissa oli alusta alkaen tarkoitus kehittää myös suunnitelmaa toteutuksen edetessä, eikä jatkaa loppuun asti vesiputousmallimaisella alkuperäisen suunnitelman toteutuksella. Muutoksia alkuperäiseen suunnitelmaan tehtiinkin koko toteutuksen ajan. Suurimmat muutokset tulivat projektin vaihtuessa harjoitteluprojektista yrityksen projektiksi ja palvelun alustana olevan resurssin vaihtuessa. Näiden muutosten lisäksi tulleista muutoksista suurin osa oli suunnitelmassa havaitun puutteen tai virheen korjauksia. Loput muutokset olivat optionaalisia vaatimuksia.

4.1.2 Muuttunut käyttötarkoitus

Alkuperäinen käyttötarkoitus oli aikaisemmin käsityönä tehdyn sähköpostiliikenteen automatisointi. Loppukäyttäjä varasi videoneuvottelun sähköpostitse Medbitiltä. Medbit varasi videoneuvottelun palveluntarjoajalta sähköpostitse ja palveluntarjoajan vahvistusviestistä pääteltiin kyseessä oleva varaus. Oikean tilauksen löydyttyä, Medbit lähetti ohjeet ja kuittauksen tilaajalle. Järjestelmän oli tarkoitus tehdä samat asiat ja korvata tilaajan sähköposti nettisivulle tulevalla käyttöliittymällä. Muutosten jälkeen järjestelmä ei ollut enää yhteydessä erillisen palveluntarjoajan kanssa, vaan järjestelmä tekee itse kaikki tarvittavat toimenpiteet.



Kuva 3. Tilausvahvistusten käsittely.

4.1.3 Muutokset järjestelmän käyttöympäristössä

Alkuperäisten vaatimusten mukaan järjestelmän oli tarkoitus tilata neuvottelu palveluntarjoajalta sähköpostitse ja reagoida tilausvahvistukseen. Sähköpostiliikenteen tulkinnan toteuttaminen vei huomattavan osan ohjelmointityöhön käytetyistä aikaresursseista. Sähköpostiliikenteen ja viestien tulkinnan toteutuksen ollessa jo valmiina, tilauksen prosessi muuttui täysin. Aikaisemmin palveluntarjoajalta erikseen tilattuja videohuoneita ostettiin yrityksen käyttöön jatkuvana resurssina.

Uusista videohuoneista ei tarvinnut enää varata erikseen palveluntarjoajalta aikaa, vaan riittää, ettei samaan videohuoneeseen varata samanaikaisesti neuvottelu kuin yhdelle taholle. Tämän seurauksena kaikki sisäänpäin tulevan postin ja tekstin tulkinta jäi tarpeettomaksi. Varaus tehtiin ennen sähköpostilla ja nyt palveluntarjoajaan ei tarvitse ottaa ollenkaan yhteyttä. Varaukset tehdään käyt-

töliittymällä ja järjestelmä varaa ajan videuhuoneen Exchange-kalenterista. Kaikki sähköpostiliikenne on ulospäin lähtevää, eikä käyttöliittymässä ole vapaita tekstikenttiä, joiden sisältöä olisi tulkittava. Uusien resurssien myötä jäi tarpeettomaksi tekstin tulkinta ja käsittely.

Käyttöympäristön muutokset mahdollistivat nopeamman ja yksinkertaisemman prosessin. Kun sähköpostia ei tullut enää sisään, poistui tarve monelle sähköpostilaatikolle, joita oli käytetty tilauksien yksilöinnin tukena. Sähköpostilaatikoita oli aiemmin tarkkailtava saapuneiden viestien käsittelyyn ottamiseksi ja tähän oli toteutettu useita sähköpostilaatikoita. Laatikoita tarvittiin suuri määrä, koska tilausta yksilöivää tietoa ei päivämäärän lisäksi ollut ja jokaiselle päivälle varataan useita neuvotteluja. Tilaukset yhdistettiin varauksiin yksilöllisillä lähettäjän tiedoilla ja päivämäärällä. Uusi nopeampi prosessi vaatii siis vain yhden sähköpostiosoitteen. Tilauksen resurssit varataan välittömästi tilauksen teon yhteydessä ja kaikki sähköpostiliikenne tapahtuu heti. Varauksen käsittelyaika lyheni useasta päivästä muutamaan sekuntiin ja palvelimen kuormitus keveni huomattavasti. Ylimääräisten sähköpostiosoitteiden lisäksi voitiin jättää pois laatikoita tarkkaileva ominaisuus, joka toistui säännöllisin väliajoin palvelimella ja käytti resursseja jatkuvasti. Ajastettuna tehtävänä jätettiin kuitenkin palvelimella lähetyshäiriöitä tarkkaileva toiminnallisuus mutta tätä tarkistusta ei tarvitse tehdä usein, eikä se kuormita palvelinta ja verkkoa sähköpostitarkistusten tapaan jatkuvasti.

4.1.4 Valitun agile-lähestymistavan soveltuvuus projektiin

Agilen mukanaan tuoma kehityksen nopea mukautuvuus oli avainasemassa mahdollistamassa harjoitteluprojektin muutosta tuotantoon. Vaatimukseen tehtyjä muutoksia oli helpompi soveltaa toteutustyöhön kun projektisuunnitelmassa oli valmiiksi liikkumavaraa ja työn pääpaino oli toteutuksessa, eikä suunnitelmissa tai dokumentoinnissa.

Haittapuolet ilmenivät kun muutoksia tuli jo valmiiksi prototyypiksi kehitettyihin moduuleihin. Toimivan prototyypin tärkeimpiä osia oli sähköpostin käsittely, jota

ei lopulta tarvittukaan. Perinteisellä projektin toteutuksella muutoksen kohteena olevat moduulit eivät olisi olleet vielä valmiita ja hukkaan mennyt työmäärä olisi ollut todennäköisesti pienempi. Muutosten toteuttamisen työmäärä olisi kuitenkin ollut suurempi, joten agile oli tässä lopuksi parempi vaihtoehto.

4.2 Kehitystyön painopiste

Kehitys aloitettiin niistä osista, jotka katsottiin välttämättömimmiksi järjestelmän perustoiminnallisuuden kannalta. Tärkeimmäksi katsottiin sähköpostin käsittely, koska se oli ollut eniten henkilöresursseja vievä osa ennen järjestelmää. Ilman sähköpostin käsittelyominaisuutta olisi jäänyt sama määrä manuaalista työtä ja uudesta järjestelmästä ei olisi mitään hyötyä.

Sähköpostin tiedot ja varaukset oli saatava johonkin, joten seuraava luonteva painopiste oli tietokannassa. Tietokantojen kanssa samaan aikaan oli työn alla tilausten käsittely ja automatisointi.

Painopiste jakautui karkean perustoiminnallisuuden valmistuttua monelle alialle, teknisestä näkökulmasta. Käsittelyssä oli yksittäisiä ominaisuuksia, joiden tarvitsema toiminnallisuus rakennettiin kerralla kaikkiin osiin. Näin välttyttiin tarpeettoman raskaalta rajapintasuunnittelulta, jota ei olisi ollut järkevää tehdä muuttuvien vaatimusten ja rajallisen osaamisen takia. Kun tietty ominaisuus toteutetaan kerralla koko sovellukselle, saadaan ominaisuus käyttöön seuraavassa prototyypissä.

4.3 Ohjelmistokehitys ja muut työtehtävät

Ohjelmistokehityksen rinnalle kerääntyi nopeasti paljon muita työtehtäviä, jotka eivät liittyneet tähän projektiin mitenkään. Useat muista työtehtävistä olivat kiireellisiä ja tämä projekti jäikin pian muiden tehtävien lomassa tehtäväksi työksi, koska tämän projektin tuotoksella ei ollut kiire.

4.3.1 Kokonaisuuden hallinta

Kun projektin ulkopuoliset tehtävät alkoivat täyttää järjestelmän toteuttajan kalenteria, tarvittiin projektin ja muiden tehtävien yhteensovittamiseen työkaluja ja -tapoja. Yrityksellä ei ollut valmiina ratkaisua tällaiseen käyttöön. Projektinhallintatyökalut eivät sovellu hyvin yhden työntekijän monipuolisen työkuorman hallintaan, eikä niitä ollut saatavilla kohderyhmän ulkopuolisille työntekijöille. Tarpeeseen kokeiltiin useita ratkaisuja, jotka eivät olleet liian kalliita mahdolliseen käyttöönottoon. Sama tarve oli muillakin tiimin jäsenillä, joten samalla esitettiin mahdollisesti yrityksen yleiseen käyttöön tulevaa ratkaisua.

4.3.2 Muutosten suunnittelun ryhmä

Muutosehdotuksia esitettiin suoraan järjestelmän toteuttajalle, joka muutti pienet ja toteutuskelpoiset ehdotukset suoraan tehtäviksi, jotka lisättiin projektin tehtävälistaan. Suurempia muutoksia käsiteltiin ohjausryhmän kesken. Muutosehdotukset tulivat ohjausryhmän jäseniltä tai toteuttajalta itseltään.

4.4 Ohjelmistokehityksen tukiverkosto

4.4.1 Yrityksen resurssien käyttö

Työntekijöille on tekniikan lisäksi saatavilla muiden työntekijöiden työpanos. Harjoittelun aikanakin oli mahdollisuus pyytää apua mutta työntekijänä tarvittavan asian voi tilata käytetyn prosessin mukaisesti ja se tehdään tietyn aikataulun mukaan.

4.4.2 Työkalut

Kehitystyön muututtua yrityksen työntekijän projektiksi, saatiin käyttöön useita uusia resursseja. Järjestelmälle tilattiin virtuaalipalvelin ja tietokanta yrityksen

klusterista, joiden tilalla oli harjoittelun aikana ollut kehityskoneen Internet Information Services (IIS) -alusta ja paikallinen localdb-tietokanta. Harjoittelun aikainen Post Office Protocol (POP) ratkaisu korvattiin lopulliseen käyttöön tarkoitettulla Exchange ratkaisulla. Exchangen myötä tuli myös mahdollisuus käyttää kalenteria, joka paransi käytettävyyttä huomattavasti.

Yrityksessä ei ollut aikaisemmin juuri tehty suuren mittakaavan ohjelmointia itse, vaan koodi ostettiin ja oma ohjelmointityö oli useimmiten skriptejä tai makroja. Tämän järjestelmän kehitystyön aikana aloitettiin yrityksessä muutakin ohjelmointia ja rakennettiin versiohallintaa ohjelmakoodille. Uutta Team Foundation Server (TFS) versiohallintaa testattiin myös tässä projektissa mutta sen tietoliikenneyhteydet olivat vielä kesken, kun sitä olisi tarvittu, eikä ratkaisua otettu käyttöön tämän projektin kehitystyön yhteydessä. Ohjelmakoodi siirretään julkaisun jälkeen versiohallintaan talteen ja sitä käytetään jatkokehityksessä.

4.5 Projektinhallinta

4.5.1 Projektin hallinnan tarpeet

Kun projektia oli toteuttamassa pääosin vain yksi henkilö, voitiin henkilöresurssien hallinta jättää tässä tapauksessa lähes kokonaan huomiotta. Henkilöiden aikaresursseja tai saatavuutta ei arvioitu, vaan projektin odottaessa projektiyöryhmän ulkopuolisen resurssin tuotosta, projektin työryhmä keskittyi muihin tehtäviin. Muita resursseja tai laatua ei muutettu viivytyksen korjaamiseksi, vaan lopputuotteen julkaisu lykkääntyi.

Pääosassa projektin hallinnassa oli työn jakaminen sopiviin osiin ja tehtäviin, jotka pystyttiin toteuttamaan suunnitelmallisesti. Työ jaettiin suuriin kokonaisuuksiin, joista eroteltiin vielä pienempiä kokonaisuuksia. Suuria kokonaisuuksia ohjelmoinnin osalta olivat esimerkiksi käyttöliittymä, tiedon tallentaminen ja tiedon jakaminen. Nämä jaettiin pienempiin osiin toiminnallisuuden näkökulmasta. Kokonaisuudet jaettiin tarpeen mukaan tehtäviin viimeistään kun niiden toteuttaminen oli lähitulevaisuudessa ajankohtaista. Tehtävien suoritusjärjestyk-

seen vaikutti niiden tarpeellisuus toimivan prototyypin kannalta ja muiden tehtävien riippuvuus kyseisen tehtävän tuotoksesta.

Tehtävien hallinnassa kokeiltiin joitakin siihen tarkoitettuja ratkaisuja mutta käytännössä tarkoitukseen sopi parhaiten tekstipohjainen lista. Lista oli helppo päivittää muuttuneita vaatimuksia, eikä tässä projektissa ollut juuri tarvetta henkilöresurssien hallintaa helpottaville työkaluille, joita useissa muissa projektin hallintatyökaluissa on.

4.5.2 Koodin hallinta

Vaikka versiohallinta on tärkeää yhdenkin tekijän projektissa, sen merkitys korostuu kun mukaan tulee enemmän kehittäjiä. Lähdekoodin versiohallinta oli projektin opiskeluaikana käytännössä varmuuskopiointia. Kun saataville tuli mahdollisuus käyttää toisenkin kehittäjän resursseja, tarvittiin oikeaa versiohallintaa. Yrityksen oma versiohallinta oli tässä vaiheessa vasta ajatustasolla, joten projektissa päädyttiin käyttämään kolmannen osapuolen ilmaista ratkaisua. Saatavilla olevista vaihtoehdoista päädyttiin BitBucketiin sen tarkoitukseen hyvin sopivan hinnoittelumallin vuoksi. Vaihtoehdoissa oli hinnoittelu tehty niin, että yksi projekti kahdella kehittäjällä edellyttää jo maksullista tilausta.

Koska järjestelmää tehtiin Varsinais-Suomen Sairaanhoidopiiriin käyttöön eikä voitu käyttää normaalisti omissa konesaleissa tuotettavia palveluja, piti kaikki salasanat ja väärinkäytöksille altistavat osat koodista poistaa ennen palveluun lataamista. Vaikka palvelua pidettiin tietoturvallisena, katsottiin salasanojen lataaminen ulkoiseen palveluun tarpeettomaksi riskiksi. Käytännössä ainoa riskinä pidettävä salasana koodissa oli Exchangen salasana, jonka salaus olisi ollut ehkä mahdollista purkaa muun koodin avulla. Kyseinen salasana poistettiin ennen jokaista synkronointia ja lisättiin takaisin paikalliseen ohjelmakoodiin heti sen jälkeen.

Versiohallinta jäi kuitenkin enimmäkseen varmuuskopioksi tässäkin yhteydessä, koska ohjelmakoodiin pyydettiin apua vain muutaman kerran. Näissäkään tapauksissa ei päädytty käyttämään versiohallinnan työkaluja, vaan avustava kehittä-

täjä tarkasteli ongelmakohtaa versiohallinnan kanssa ja ratkaisuehdotus käsiteltiin yhdessä ja lisättiin suoraan pääkehittäjän paikalliseen ohjelmakoodiin.

5 TUOTANTOKÄYTTÖÖN OTTAMINEN

5.1 Ympäristö

Järjestelmä otettiin käyttöön virtuaalisella Windows-palvelimella. Järjestelmän käyttämä SQL-tietokanta on jo aikaisemmin käytössä olleesta klusterista. Käyttöliittymän palvelimelle pääsee vain asiakkaan sisäverkosta, joten erillinen kirjautuminen ja tunnistautuminen jätettiin toteuttamatta. Käyttäjätunnus saadaan Windows-kirjautumisesta, eikä käyttäjiä tarvinnut rajata työntekijöiden keskuudessa. Aikaisemmin varatut neuvottelut näkyvät kuitenkin vain niitä varanneen käyttäjän tunnuksilla ja järjestelmässä määritetyillä admin-käyttäjillä. Sähköpostiliikenne ja kalenteritoiminnot on toteutettu Exchangella.

5.2 Käyttäjät

Järjestelmän mahdollisia loppukäyttäjiä ovat kaikki sairaanhoitopiirin työntekijät, joilla on pääsy sisäverkkoon. Käytännössä järjestelmää käyttää eniten pieni joukko lääkäreitä, hoitajia ja hallinnon henkilökuntaa. Valtaosa käyttäjistä on terveydenhuollon ammattilaisia, joiden tietotekninen osaaminen vaihtelee suuresti.

5.3 Testaus

Järjestelmän testausta tehtiin jatkuvasti kehitystyön edetessä. Moduuli- ja integraatiotestaus oli ohjelmoijan tehtävä mutta validointi ja lopputestaus tehtiin yhdessä palvelusta vastanneen henkilökunnan kanssa. Tämän lisäksi käyttöönotto aloitettiin pilottikäytöllä, jossa varmistettiin järjestelmän soveltuminen oikeaan käyttöön. Moduuli- ja integraatiotestauksessa löytyneet virheet korjattiin mahdollisuuksien mukaan heti ja validoinnissa tai lopputestauksessa havaitut puutteet korjattiin ennen pilotin aloitusta.

5.4 Yrityksen ilmeeseen sovittaminen

Alkuperäisessä suunnitelmassa oli tarkoitus muokata käyttöliittymän ulkoasu yrityksen brändin mukaiseksi. Visuaalinen ilme suunniteltiin toteutettavan toiminnallisuuden jälkeen. Kehitystyön aikana selvisi kuitenkin, ettei yrityksen vastaavan kokoisia järjestelmiä oltu juuri kehitetty brändin mukaiseksi. Tämäkin käyttöliittymä jätettiin alustasta tulleen tyylin mukaiseksi ja mahdollinen visuaalinen tyylittely kirjattiin mahdollisiin jatkokehityskohteisiin.

6 VALMIIN OHJELMISTON SIIRTÄMINEN YLLÄPITOON

6.1 Ylläpidon tarpeet koodille

Lähdekoodi on saatavilla tarvittaessa tehtäviä korjauksia varten mutta tämä jäänee alkuperäisen toteuttajan tehtäväksi. Ylläpidon ohjeet ja materiaali sisältävät kuvauksen järjestelmän toiminnasta ja sen liittymäkohdista muihin järjestelmiin. Dokumentaatioissa on mainittu todennäköisiä ongelmakohtia ja mahdolliset ratkaisut niihin.

6.1.1 Kuvaus

Ylläpito edellyttää järjestelmältä tiettyjen asioiden kuvausta. Käytännössä näin pienessä järjestelmässä on suuri osa tiedosta päällekkäistä, eikä kaikkien dokumenttien täydellinen sisältö ole tarpeen. Luvussa 2.1.2 on kerrottu yrityksen minimidokumentaatiosta. Sairaanhoidopiirin normaalisti käyttämiin järjestelmiin verrattuna on kuitenkin kyse niin pienestä järjestelmästä, että tässä tärkeimpiä dokumentteja olivat pikaohje, asennusohje ja arkkitehtuurikuvaus. Näistä todennäköisesti tullaan käyttämään vain pikaohjetta, jossa on tiivistetysti kuvattu muiden dokumenttien olennaisin sisältö. Pikaohjeessa on tarkoitus olla selkeästi kuvattu kaikki todennäköisimmin jatkossa tarvittava sisältö mahdollisimman lyhyesti ja se tehtiin tässäkin tapauksessa ensimmäisenä. Muita dokumentteja täydennettiin osittain mutta ei tarpeettomasti käytännön tarpeiden yli. Käytännön tarpeet arvioitiin muutoksenhallinnan koordinaattorin opastuksella melko pieniksi.

6.1.2 Mallintaminen

Minimidokumentaatioon kuuluu arkkitehtuurikuvaus, jossa tulee olla mallinnettu järjestelmän arkkitehtuuri. Suuren osan mallinnuksesta saa tuotettua kokonaisuuden hallintaan käytetyillä työkaluilla, eikä käsin tarvitse piirtää mitään. Mic-

rosoft Visual Studiolla saa tuotettua ohjelmakoodista tarvittavat kuvaukset ja yrityksen käyttämällä Efecte-sovelluksella saa tuotettua arkkitehtuurikuvauksen. Efecteen on kirjattu järjestelmän yhteys muihin resursseihin jo kehitystyön aikana. Yrityksen muutoksenhallinta käyttää Efecteä työkaluna, jolla hallitaan tiettyyn resurssiin tehdyn muutoksen vaikutus muihin resursseihin ja sinne on siksi jo luotu tieto kaikista järjestelmän kytköksistä muihin järjestelmiin ja resursseihin.

6.2 Ohjeet

Loppukäyttäjän ohjeet liitettiin järjestelmän varausten kohteena olevien videoneuvottelujen ohjeiden yhteyteen. Loppukäyttäjälle näkyvä osa järjestelmästä ei tarvitse paljoa opastusta, joten käyttöohjeet ovat suppeat. Käyttöliittymällä on pyritty tekemään helppokäyttöisiä ratkaisuja ja kaikki toiminnallisuus on vielä sanallisesti ohjeistettu.

Ylläpidolle tehtiin erilliset ohjeet, joiden avulla voidaan ratkaista todennäköisimpiä loppukäyttäjän ongelmia ja vikoja järjestelmän toiminnassa. Näiden avulla tulisi olla mahdollista ratkaista järjestelmän käytössä ilmenneet viat, itse järjestelmää tuntematta.

6.3 Koulutus

6.3.1 Käyttäjien koulutus

Käyttäjille annetaan koulutus tarvittaessa. Järjestelmä on niin yksinkertainen, että koulutuksen järjestäminen ilman pyyntöä ei kannata. Varausjärjestelmän lyhyt esittely sisällytetään kuitenkin tuleviin videoneuvottelukoulutuksiin.

6.3.2 Ylläpidon ja tukipalvelun koulutus

Ylläpito ei tarvitse erikseen koulutusta näin pienen järjestelmän tukitoimintoihin, vaan mahdolliset tukipyynnöt käsitellään ratkaisutietokannan ja käyttöohjeiden avulla. Haastavammat tukipyynnöt ohjataan suoraan kehittäjälle.

6.4 Tiedotus

Tiedotus on kohdistettu varauksia tekeville henkilöille. Vanhaa varaustapaa käyttävät varaajat saavat ohjeet uuteen toimintatapaan varauspyynnön vastausviestinä. Uusille varaajille ei tarvita erikseen tiedotusta, koska videoneuvottelujen ohjeissa ohjataan tekemään varaus tällä järjestelmällä.

7 LOPPUTULOS

7.1 Valmiin järjestelmän kuvaus

Järjestelmän julkaisuversio koostui tietokannasta, internetselaimella käytettävästä käyttöliittymästä ja Exchangen palveluista. Käyttöliittymä on yrityksen omassa konesalissa virtuaalisella Windows-palvelimella, josta on yhteys omaan tietokantaan. Tietokanta on osa yrityksen käytössä ollutta tuotantokäyttöön tarkoitettua klusteria. Testikäyttöä varten on oma kantansa yrityksen testiklusterista. Exchangea käytetään sähköpostiliikenteeseen ja kalenterien hallintaan. Tähän tarkoitukseen on luotu käyttäjätili, jonka kalentereissa on tieto kaikista varauksista ja sähköpostissa kaikki järjestelmän sähköpostiliikenne.

Julkaisuversion käyttöliittymästä poistettiin kaikki järjestelmän halutun toiminnallisuuden kannalta tarpeeton. Jäljelle jäi vain etusivu, tilauslomake, lista omista varauksista ja yhteystietoihin sekä tukeen liittyvät sivut. Loppukäyttäjä voi tehdä uuden varauksen, tarkastella aikaisemmin tekemiään varauksia ja etsiä tuen yhteystietoja.



Kuva 4. Varausjärjestelmän käyttöliittymä.

Loppukäyttäjä aloittaa järjestelmän käytön siirtymällä etusivulle tai suoraan tilauslomakkeelle. Tilauslomakkeella täytetään omat ja osallistujien yhteystiedot sekä neuvottelun haluttu ajankohta ja varattu neuvotteluhuone. Järjestelmä tarkastaa kalenterista ensimmäisen annettuun aikaan vapaana olevan neuvotteluhuoneen, ja varaa sen tekemällä kalenteriin kokouksen. Tämän jälkeen järjestelmä lähettää kokouskutsun ja ohjeviestin kaikille osallistujille.

Kaikki varaukset tallentuvat myös tietokantaan mutta videoneuvotteluhuoneiden saatavuustiedot tarkistetaan aina Exchange-kalenterista. Näin mahdollistetaan kalenterin manuaalikäyttö esimerkiksi toistuvien- tai erikoisvarausten tapauksissa. Ylläpito voi tehdä järjestelmän kalenteriin merkinnän, jolla käytännössä saa varattua tietyn videoneuvotteluhuoneen. Käyttäjä voi itse käydä jälkikäteen katsomassa listaa aikaisemmin järjestelmästä varaamista neuvotteluista mutta niihin ei voi enää tehdä muutoksia. Käyttäjälle esitettävä lista aikaisemmin varatuista neuvotteluista tuotetaan tietokannasta mutta niistä esitetään samat tiedot kuin kalenterikutsuissa. Järjestelmässä ei kuitenkaan näy ylläpidon manuaalisesti tekemät muutokset tai varaukset. Muutokset pyydetään ylläpidolta, joka voi siirtää järjestelmän kalenterissa olevaa kokousta tai luoda toistuvan varauksen. Järjestelmä lähettää kaikille osallistujille yhteisen sähköpostin, johon kaikille vastaamalla saa tarvittaessa kuka tahansa osallistujista tiedotettua kaikkia muita osallistujia. Loppukäyttäjän kaikista omista varauksista ja kaikista varauksista, joihin tämä on osallistunut, on tieto myös kyseisen käyttäjän omassa kalenterissa.

7.2 Projektin onnistuminen

Projektin aikana tulleet muutokset vaatimuksiin siirsivät lopputuotoksen valmistumista ja lopullinen julkaisupäivämäärä olikin useita kuukausia alussa sovitusta aikataulusta jäljessä. Aikataulua venyttivät myös palvelintilauksen jälkityöt, kuten palomuurien avaukset, nimipalvelinten konfigurointi ja käyttöliittymäpalvelimen haastava yhteys tietokantapalvelimeen. Useat näistä työvaiheista vaativat erillisen tilauksen, eikä näistä ollut etukäteen tietoa. Seuraava tilauksen tarve

selvisi aina kun huomattiin, ettei tehty työ tuonutkaan toivottua toiminnallisuutta. Syyinä monivaiheiseen palvelintilaukseen oli projektiryhmän kokemattomuus tällaisissa palvelintilauksissa ja tarvittavien työvaiheiden vastuun jakaminen monelle taholle.

Julkaisuversio täytti vaatimusmäärittelyssä kuvatut pakolliset vaatimukset mutta useita optionaalisia vaatimuksia ei toteutettu. Optionaalisten vaatimusten toteuttaminen jätettiin jatkokehityksen tehtäväksi.

7.3 Palaute

Projektin aikana saatu palaute on ollut pääasiassa positiivista. Aikataulusta saatu palaute oli neutraalia, koska alkuperäinen aikataulu ei toteutunut mutta tälle oli hyväksyttävä syy. Toinen neutraali palaute tuli ohjelmakoodista. Mahdollisimman aikaisessa vaiheessa toimivaan prototyyppiin tähänneen kehitystyön sivutuotteena oli koodiin jäänyt paljon tilapäisratkaisuuksi tarkoitettuja osia. Lisäksi suurten toiminnallisten vaatimusten muututtua oli ohjelmakoodissa tehty muutoksia, jotka olisi ollut hyvä toteuttaa eri tavalla mutta käytetty tapa oli työmäärältään pienempi ja johti nopeampaan aikatauluun. Käytettävyydestä tai käyttötarkoitukseen soveltuvuudesta ei ole vielä saatu palautetta loppukäyttäjiltä hiljaisen käyttöönottoajankohdan seurauksena.

LÄHTEET

- [1] Medbit Oy, *Sisäiset ohjeet*, Turku: Medbit Oy, 2015.
- [2] Wikimedia Foundation, Inc., "Wikipedia," Wikimedia Foundation, Inc., 10 7 2014. [Online]. Osoite: [https://en.wikipedia.org/wiki/Change_management_\(ITSM\)](https://en.wikipedia.org/wiki/Change_management_(ITSM)). [Viitattu 21 07 2015].
- [3] International Organization for Standardization, *ISO/IEC 20000-1:2011*, 2011.
- [4] Mikkonen, T. ja Haikala, I., *Ohjelmistotuotannon käytännöt*, Talentum Media Oy, 2011.
- [5] Ambler, S. W., "agilemodeling.com," Ambyssoft Inc., 2014. [Online]. Osoite: <http://agilemodeling.com/essays/modelStorming.htm>. [Viitattu 21 07 2015].
- [6] Panchal, D., "scrumalliance.org," SCRUM ALLIANCE®, Inc., 2015. [Online]. Osoite: [https://www.scrumalliance.org/community/articles/2008/september/what-is-definition-of-done-\(dod\)](https://www.scrumalliance.org/community/articles/2008/september/what-is-definition-of-done-(dod)). [Viitattu 21 07 2015].
- [7] Ambler, S. W. ja Lines, M., *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*, IBM Press, 2012.
- [8] Wikimedia Foundation, Inc., "Wikipedia," Wikimedia Foundation, Inc., 08 05 2015. [Online]. Osoite: https://en.wikipedia.org/wiki/Extreme_programming. [Viitattu 21 07 2015].
- [9] Copeland, L., "Computerworld.com," Computerworld, Inc., 2015. [Online]. Osoite: <http://www.computerworld.com/article/2585634/app->

development/extreme-programming.html. [Viitattu 21 07 2015].

- [10] scrum.org, "scrum.org," 2015. [Online]. Osoite:
<https://www.scrum.org/ScrumBut>. [Viitattu 15 11 2015].
- [11] Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R. C.; Mellor, S.; Schwaber, K.; Sutherland, J. ja Thomas, D., "Ketterän ohjelmistokehityksen julistus," 2001. [Online]. Osoite: <http://agilemanifesto.org/iso/fi/>. [Viitattu 7 6 2015].
- [12] Wikimedia Foundation, Inc., "Wikipedia," Wikimedia Foundation, Inc., 19 07 2015. [Online]. Osoite:
https://en.wikipedia.org/wiki/Agile_software_development. [Viitattu 21 07 2015].
- [13] Wikimedia Foundation, Inc., "Wikipedia," Wikimedia Foundation, Inc., 04 06 2015. [Online]. Osoite:
https://en.wikipedia.org/wiki/Lean_software_development. [Viitattu 21 07 2015].
- [14] Ambler, S. W., "Agilemodeling.com," Ambysoft Inc., 2012. [Online]. Osoite:
<http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm>. [Viitattu 21 07 2015].
- [15] Booz, Allen ja Hamilton, "Earned Value Management Tutorial Module 2: Work Breakdown Structure," U.S. Department of Energy. Office of Science, Tools & Resources for Project Management, 2011. [Online]. Osoite:
http://science.energy.gov/~media/opa/powerpoint/Final_Module_2.ppt. [Viitattu 27 12 2011].
- [16] Wikimedia Foundation, Inc., "Wikipedia," Wikimedia Foundation, Inc., 20

06 2015. [Online]. Osoite:

https://en.wikipedia.org/wiki/Work_breakdown_structure. [Viitattu 21 07 2015].

Alkuperäinen vaatimusmäärittely

Käyttötarkoitus

Järjestelmä varaa pyydetyn tilan Outlook kalenterista ja tilaa Videralta videoneuvottelun käyttäjän määrittämälle ajalle. Järjestelmä lähettää tiedot tapahtumasta tilaajalle ja osallistujille. Järjestelmä pitää tietokantaa tapahtumista, jossa ne ovat muokattavissa ennen tapahtuman alkua.

Käyttöliittymä

Käyttöliittymässä voi valita neuvottelun ajan ja osallistujamäärään soveltuvan tilan. Järjestelmään syötetään osallistujien sähköpostiosoitteet ja tilaajan yhteystiedot. Käyttöliittymän tulee edetä vaihe kerrallaan, tarjoten vain tilaukseen soveltuvia vaihtoehtoja valittavaksi. Yhteystietoihin täytyy syöttää sähköpostiosoite, joka sisältää @ -merkin. Tilaus vahvistetaan tietojen syöttämisen jälkeen erikseen. Vahvistusviestissä on linkki tilauksen peruuttamiseen.

Toiminta

Järjestelmään kirjaudutaan aluksi sisään ja lopuksi ulos. Tila lukitaan heti, kun se on valittu käyttöliittymästä. Tilauksen voi jättää kesken milloin tahansa ja mahdolliset tehdyt tilavaraukset perutaan. Tila jää varatuksi ja sähköpostien lähetys alkaa kun tilaus vahvistetaan käyttöliittymästä. Järjestelmään syötetään nettisivulla olevan käyttöliittymän kautta tilauksen tiedot. Järjestelmä varaa määrätyn tilan Outlook kalenterista ja lähettää tilauksen Videralle sähköpostilla. Tilaaja saa aluksi sähköpostiviestin, jossa on tilauksen tiedot ja linkki tilauksen peruutukseen. Järjestelmä odottaa Videralta sähköpostitse tulevaa vahvistusta tilauksesta ja sen saatuaan lähettää vahvistuksen tilaajalle sekä ohjeet tilaajalle ja osallistujille.

Tilaus tallentuu järjestelmän tietokantaan, jossa se on tilaajan peruttavissa tai pääkäyttäjän muokattavissa. Tilausta peruttaessa vapautetaan varattu tila ja lähetetään sähköpostiviesti kaikille, joihin on oltu yhteydessä sähköpostitse kyseisestä tilauksesta. Tilausta muokattaessa niin, että aika tai paikka muuttuu, lähetetään tieto muutoksista osallistujille sähköpostilla.

Poikkeustilanteet

Jos Videralta ei tule vahvistusviestiä kolmen arkipäivän kuluessa, lähetetään siitä ilmoitus pääkäyttäjälle. Jos tilaus jää vahvistamatta 15 minuutiksi tilan varauksen jälkeen, perutaan mahdollinen tilan varaus ja ilmoitetaan käyttäjälle käyttöliittymässä tilauksen peruuntumisesta.

Kuvaus järjestelmästä harjoittelun lopussa

VNVaraus

Käyttöliittymä

Etusivu

Kaikilla sivuilla on yläreunassa navigointipalkki, josta pääsee etusivulle [Etusivu], kaikkien varausten sivulle [Varaukset], tukisivulle [Apua ja tukea] ja yhteystietoihin [Yhteystiedot]. Palkin oikeassa reunassa näkyy kirjautuneen käyttäjän tiedot.

Yksittäinen varaus otsikon alla olevasta [Varaa >>] napista pääsee tekemään uuden varauksen.

Toistuva varaus otsikon alla olevasta [Tilaa >>] napista pääsee tekemään toistuvan varauksen.

Uusi varaus

Sivulla on kentät varauksen tiedoille. Varauksen tekijän sähköpostiosoitteentään kirjoitetaan varauksen vastuuhenkilön sähköpostiosoite.

Henkilöiden määrä varatussa tilassa on niiden henkilöiden määrä, jotka tulevat varattuun tilaan paikanpäälle.

Neuvottelun alkuaika on aika, josta alkaen neuvottelu halutaan varata.

Neuvottelun kesto minuutteina on aika, jonka neuvottelu korkeintaan kestää.

Neuvottelulle varattu tila on tila, jonka neuvottelun varaaja on varannut neuvottelulle.

Neuvotteluun osallistuvien henkilöiden sähköpostiosoitteet pilkulla erotettuna kenttään annetaan kaikkien videoneuvotteluun etänä osallistuvien sähköpostiosoitteet.

Osoitteista lasketaan automaattisesti osallistujien määrä ja osallistujille lähetetään tiedot sekä ohjeet neuvottelusta. Osoitteet kirjoitetaan peräkkäin ja erotetaan pilkulla: osallistuja1@example.com, osallistuja2@example.com, osallistuja3...

Varaus tehdään painamalla tietojen alla olevaa [Tee varaus] nappia.

Tietokanta

Tietokantaan on talletettu postilaatikat, varaukset, osallistajat ja tilat. Jokaisessa varauksessa on ainakin varaajan sähköpostiosoite, varauksen aika ja -kesto. Varaukseen liittyy aina tila ja postilaatikko. Lisäksi tilaukseen voi liittyä osallistujia. Tila on pakollinen, koska tiloilla on omat sähköpostilaatikat ja varauksen sähköpostiliikenne tarvitsee sitä kautta saatavan laatikon.

Sähköpostiliikenne

Kun käyttöliittymällä tehdään uusi varaus täyttämällä lomake ja painamalla [Tee varaus], järjestelmä ottaa varauksen käyttöön postilaatikon varauksen päivämäärän osalta. Jokaisella tilalla on viisi postilaatikkoa, joista jokaista voi käyttää yhden kerran jokaisena päivänä. Järjestelmä lähettää laatikosta viestin videoneuvottelun toimittajan tilauspalveluun. Kun tilauspalvelusta vastataan vahvistusviestillä, järjestelmä lähettää vahvistuksen ohjeineen varaajalle ja osallistujille.

Testaus

Normaalin tilauksen kulku

Järjestelmän päätarkoitus on videoneuvottelun varaaminen. Uutta varausta tehdessä normaalitapauksessa pitäisi olla seuraavat askeleet:

1. Etusivulta painetaan uuden varauksen nappia.
2. Selain siirtyy uuden varauksen lomakkeelle.
3. Käyttäjä täyttää yhteystiedot ja varauksen tiedot.
4. Järjestelmä tarkastaa soveltuvilta osin kenttien oikeanmuotoisen sisällön (vanhemmissa selaimissa ei toimi kaikki tarkistukset).
5. Käyttäjä painaa [Tee varaus] nappia.
6. Järjestelmä luo tietokantaan varauksen ja liittää siihen osallistajat, tilan ja postilaatikon. Postilaatikko haetaan tilan perusteella ja tilan laatikoista tarkistetaan yksi kerrallaan saatavuus varauksen päivämäärälle, kunnes löytyy vapaa laatikko liitettäväksi.

7. Järjestelmä lähettää neuvottelun toimittajalle sähköpostin, joka on muodostettu varauksen ajan, keston ja osallistujien määrän perusteella.
8. Palveluntoimittaja vastaa (ei aina) viestiin automaattisella viestillä, joka sisältää referenssikoodin tilauksen asioiden hoitoon.
9. Järjestelmä käy säännöllisesti tarkistamassa viestit kaikista laatikoista ja automaattisen viestin tullessa vastaan, järjestelmä liittää referenssikoodin varaukseen.
10. Palveluntoimittaja lähettää vahvistusviestin tilatun neuvottelun käsittelystä.
11. Järjestelmä noutaa seuraavalla tarkistuskerralla vahvistusviestin ja etsii otsikkokentän päivämäärän ja vastaanottaneen laatikon perusteella oikean varauksen.
12. Järjestelmä lähettää varauksen tekijälle vahvistusviestin ohjeineen.
13. Järjestelmä lähettää osallistujille ohjeviestin.

Testauksessa huomioitavaa

Tässä versiossa (0.6.1) Palveluntoimittajan osoitteena on vn-tilaukset ja admin osoitteena vn-tapahtumat, joihin järjestelmä lähettää viestejä. vn-tilaukset laatikosta vastaan järjestelmälle vahvistusviestillä, jolloin viestiliikenne lähtee tilaajalle ja osallistujille. vn-tilaukset on myös järjestelmän omana osoitteena, eli tilauspalvelun pääosoitteena on testiversiossa sama kuin neuvottelun toimittajaa edustava testiosoite.

Vastatessa järjestelmän tilausviestiin, jonka pitäisi olla lähtöisin tilan perusteella valitusta osoitteesta ja päätyä vn-tilaukset laatikkoon, käytetään seuraavaa muotoa: Järjestelmä yrittää etsiä viestiin liittyvän tilauksen kun viesti on lähetetty toimittajan osoitteesta (tässä vn-tilaukset). Testiviestiin täytyy vastata replyllä, jolloin se päätyy oikeaan laatikkoon, ja viestin otsikossa täytyy olla sama päivämäärä tilausviestistä. Automaattivastaus tunnistetaan viestin rungossa olevan ”on automaattinen viesti” (ilman lainausmerkkejä) tekstin perusteella.

Järjestelmän pikaohje

Mikä on Videoneuvottelujen varausjärjestelmä

Videoneuvottelujen varausjärjestelmä, eli VNVaraus, on työkalu videoneuvottelujen varaukseen. Järjestelmän käyttöliittymä on nettisivulla. Käyttöliittymällä täytetään lomake halutun neuvottelun tiedoilla ja järjestelmä varaa kyseiseksi ajaksi videoneuvotteluresurssin. Varauksessa määritetyille osallistujille lähetetään sähköposti ja kalenterivaraus Exchangella. Viesteissä on varauksen tiedot ja soitto-ohjeet. Varaus tallennetaan myös tietokantaan.

Järjestelmä käyttää Exchange yhteyteen AD-tunnusta, jonka kalentereista löytyy kalenteri jokaiselle videoneuvotteluhuoneelle. Näissä kalentereissa on tieto varatuista videoneuvotteluista.

Sovelluksen käyttö

Sovelluksen käyttöliittymä on osoitteessa vnvaraus.medbit.fi ja sitä pääsee käyttämään VSSHP sisäverkosta. Järjestelmään kirjautuminen tapahtuu Windows-tunnistautumisella ilman käyttäjän erillisiä toimenpiteitä.

Etusivulta valitaan joko neuvottelun varaaminen tai toistuvan neuvottelun varaaminen. Normaalin neuvottelun varaaminen ohjaa lomakkeelle, joka tekee kaikki varauksen toimenpiteet automaattisesti. Toistuva varaus vie lomakkeelle, joka lähettää lomakkeen sisällön sähköpostilla manuaalikäsittelyyn.

Sivun yläreunassa on linkit etusivulle, yhteystietoihin ja aikaisemmin varattujen varausten listaan. Varaustilassa on tieto neuvottelun ajankohdasta ja osallistujista kaikista tietokannassa olevista, kirjautuneen käyttäjän varauksista.

Varauksen tekeminen generoi vahvistusviestin osoitteeseen, jota käytettiin ennen nykyistä järjestelmää neuvottelujen tilaukseen. Vahvistusviesti ehkäisee päällekkäisiä varauksia järjestelmän käyttöönottoaiheessa.

Mahdolliset ongelmatilanteet

Jos järjestelmässä tapahtuu virhe, käyttäjälle näytetään yleismuotoinen virheilmoitus ja kehoitetaan ottamaan yhteys tukeen ongelman toistuessa.

Vanhoilla Internet Explorer selaimilla pystyy mahdollisesti täyttämään lomakkeen kenttiin virheellistä tietoa, joka aiheuttaa virheen. Yksittäisen käyttäjän kohdalla toimimaton järjestelmä johtuu todennäköisesti vanhalla selaimella virheellisesti täytetystä lomakkeesta.

Mahdollisia teknisiä ongelmia saattaa tulla tietokantayhteydessä ja Exchange-yhteydessä. Ongelmia ei ole esiintynyt julkaisuversion tietokantayhteydessä ja Exchangen ongelmat ovat liittyneet vanhentuneeseen järjestelmän käytössä olevaan AD-käyttäjätiliin. Molemmat vaikuttavat koko sovelluksen toimintaan ja kaikkiin käyttäjiin.

Järjestelmä lähettää administraattorille tarkoitetut viestit osoitteeseen, johon ylläpidolla on pääsy. Tähän osoitteeseen tulee ilmoitus mahdollisista ongelmatilanteista, jotka eivät välttämättä kuitenkaan tarkoita varauksen epäonnistumista.