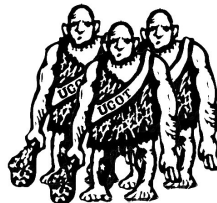


UCOT-Sovellusprojekti

Projektisuunnitelma

Ilari Liukko
Tuomo Pieniluoma
Vesa Pikki
Panu Suominen



Versio: 1.00
Julkinen
26. lokakuuta 2006

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2006		
Tilaaaja	__.__.2006		
Ohjaaja	__.__.2006		

Tietoa dokumentista

Tekijät:

- | | | |
|-------------------------|----------------------|-------------|
| • Ilari Liukko (IL) | ilanliuk@cc.jyu.fi | 050-4367494 |
| • Tuomo Pieniluoma (TP) | tujupien@cc.jyu.fi | 040-7202054 |
| • Vesa Pikki (VP) | vevijopi@cc.jyu.fi | 044-5288031 |
| • Panu Suominen (PS) | panu.suominen@iki.fi | 050-3458484 |

Dokumentin nimi: UCOT-projekti, Projektisuunnitelma

Sivumäärä: 27

Tiedosto: UCOT-projektisuunnitelma-1.00.tex

Tiivistelmä: Tämä dokumentti on projektisuunnitelma Jyväskylän yliopiston tietotekniikan laitoksen UCOT-sovellusprojektille. Suunnitelma kuvaa projektin aihetta, siihen käytettäviä resursseja ja läpivientiä. Suunnitelma sisältää aikataulujen, työtehtävien ja ajankäytön kuvauksen jokaisen ryhmän jäsenen osalta.

Avainsanat: UCOT, heuristiikka, käyttötapaus, olioluokka.

Versiohistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.01	18.9.2006	Luonnoksen tekeminen aloitettu.	VP,TP
0.02	22.9.2006	Yhdistetty johdanto ja tausta kappaleet. Muokattu tavoitteet-kappaleen ohjelmiston ja dokumenttien kuvausta.	VP,TP
0.03	25.9.2006	Yhteystiedot korjattu. Johdannon ensimmäiseen kappaleeseen lisätty Agora Centeristä ja Tekesistä. Tavoite luvun otsikot "Dokumentit" ja "muut tuotokset" yhdistetty. Testausuunnitelma ja testausraportti yhdistetty, sekä lisätty tietoa asennusohjeesta ja käyttöohjelmasta. Luvun 3.4 nimi vaihdettu. Organisaatio ja resurssit kappale kirjoitettu, Hallintatavat luku kirjoitettu. Kappale Tehtävät, työmäärät ja työnjako aloitettu.	VP,TP
0.04	27.9.2006	Dokumentin sisältö lisätty. Hallinta kappaletta täydennetty. Tehtävät, työmäärät ja työnjako -luku kirjoitettu.	VP,TP
0.05	29.9.2006	Lukua 6 aloitettu kirjoittamaan. 3 viikkopalaverissa esiin tulleet muutokset korjattu.	VP,TP
0.06	2.10.2006	Lukua 7 aloitettu kirjoittamaan. Ville Isomöttösen ehdottamat muutokset toteutettu.	VP
0.07	3.10.2006	Lukua 7 täydennetty. Luku 8 kirjoitettu.	VP,TP
0.08	4.10.2006	Lukua 6 ja 7 muokattu. Palaverien vuorolistaa muokattu. Liite B lisätty.	VP,TP
0.09	9.10.2006	Projektisuunnitelman osakatselmointiraportin muutokset toteutettu.	VP
1.00	26.10.2006	Allekirjoitettava versio	VP

Tietoa projektista

UCOT-projekti suunnittelee ja toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle ohjelmiston. Tällä ohjelmistolla voidaan heuristiikkoja käyttäen etsiä käyttötapauksista ohjelmistokehityksen analyysivaiheen olioluokkia.

Tekijät:

- | | | |
|-------------------------|----------------------|-------------|
| • Ilari Liukko (IL) | ilanliuk@cc.jyu.fi | 050-4367494 |
| • Tuomo Pieniluoma (TP) | tujupien@cc.jyu.fi | 040-7202054 |
| • Vesa Pikki (VP) | vevijopi@cc.jyu.fi | 044-5288031 |
| • Panu Suominen (PS) | panu.suominen@iki.fi | 050-3458484 |

Tilaaaja:

- | | | |
|--------------------|--------------------|-------------|
| • Tommi Kärkkäinen | tka@mit.jyu.fi | 040-5677854 |
| • Antti Hakala | anthakal@cc.jyu.fi | 040-7096224 |

Ohjaajat:

- | | | |
|---------------------|--------------------|-------------|
| • Ville Isomöttönen | vilisom@cc.jyu.fi | 014-2604976 |
| • Miika Nurminen | minurmin@cc.jyu.fi | 014-2602530 |

Tarkkailija:

- | | | |
|-------------------|--------------------|-------------|
| • Antti Hallamäki | antahall@cc.jyu.fi | 044-3555356 |
|-------------------|--------------------|-------------|

Yhteystiedot:

- | | |
|---------------------|---|
| • Sähköpostilistat: | ucot06@korppi.jyu.fi,
ucot_opetus@korppi.jyu.fi |
| • Projektiarkisto: | https://korppi.jyu.fi/list-archive/ucot06/ind.html |
| • Opetusarkisto: | https://korppi.jyu.fi/list-archive/ucot_opetus/ind.html |
| • Työhuone: | Ag C222.2 / 014-2604963 |

Sisältö

1	Johdanto	1
1.1	Taustaa	1
1.2	Dokumentin tarkoitus	2
1.3	Dokumentin sisältö	2
2	Tavoitteet	3
2.1	Sovellus	3
2.2	Dokumentit	4
2.3	Tulosten luovutus	5
2.4	Projektiryhmän oppimistavoitteet	5
3	Organisaatio ja Resurssit	6
3.1	Projektiryhmä	6
3.2	Koulutus	6
3.3	Resurssit	6
4	Hallintatavat	7
4.1	Hallinta	7
4.2	Dokumentit ja katselmoinnit	8
4.3	Palaverit	8
4.4	Versionhallinta	9
4.5	Tiedotus	9
5	Tehtävät, työmäärät ja työnjako	10
5.1	Vastuiden jako	11
5.2	1. Iteraation tehtävien jako	11
5.3	Dokumenttien vastuuhenkilöt	13
6	Prosessimalli ja aikataulu	14
6.1	Iteraatioiden aikataulut	15
7	Riskit ja niiden seuranta	16
7.1	Henkilöihin liittyvät riskit	16
7.1.1	Projektiryhmän jäsenten poissaolot	16
7.1.2	Kokemuksen puute	17
7.1.3	Viestinnän ongelmat	17

UCOT-projekti	Projektisuunnitelma 1.00	Julkinen
7.1.4	Ohjaajien poissaolo	18
7.1.5	Tilaaajan edustajien menot	18
7.2	Toteutukseen liittyvät riskit	19
7.2.1	Laitteisto- ja ohjelmisto-ongelmat	19
7.2.2	Ongelmat ohjelmoinnissa	19
7.2.3	Aikataulun suunnittelu	20
7.2.4	Vaatimusten muuttuminen	20
7.3	Sovellusaluekohtaiset riskit	20
7.3.1	Jäsentimen valinta	21
7.3.2	Heuristiikan soveltaminen	21
7.3.3	Käytettävät tallenusmuodot	22
8	Yhteenveto	23
	Liitteet	
A	Palaverin vuorolista	24
B	Termit	25

1 Johdanto

UCOT-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle ja Agora Centerille ohjelmiston, jolla analyysivaiheen olioluokkia voidaan muodostaa käyttötapauskuvauksista. Sovelluksen tarkoituksena on osittain automatisoida analyysivaiheen olioluokkien suunnittelua. Ohjelmisto toteutetaan Tekes-hankkeen rahoittamalle Tuotanto 2010 -tutkimusprojektille.

1.1 Taustaa

Oliokeskeisen ohjelmistotuotannon kaksi ensimmäistä vaihetta ovat vaatimusmäärittely ja -analyysi. Vaatimusmäärittelyvaiheessa asiakas ja järjestelmää kehittävä osapuoli määrittelevät järjestelmän tarkoituksen ja analysoivat ongelman, jota ratkaisemaan järjestelmä kehitetään. Tuloksena on vaatimusmäärittely ja myös käyttötapauskuvaukset, mikäli käytetään käyttötapauspohjaista kehitysmenetelmää. Käyttötapauskuvauksista voidaan löytää kirjoitettua tekstiä tutkimalla tärkeimmät sovellusalueen käsitteet, joista tehdään analyysivaiheen olioluokkia. Analyysivaiheen tarkoituksena on tuottaa sovellusalueen tarkasti kuvaava malli, jonka pohjalta järjestelmä voidaan suunnitella ja toteuttaa.

Projektin läpiviennin kannalta oleelliseksi muodostuu sopivan jäsentimen valinta. Tämä ilmenee siinä, ettei käytettävää rajapintaa voi tarkkaan määrittellä, jollei jäsentimen tarkkuutta tiedetä. Rajallisten resurssien takia ei jäsenintä voida projektin puitteissa toteuttaa, eikä valmista jäsenintä ole projektiin ennalta valittu. Jäsentimen toimintaa sovelluksessa kuvataan tarkemmin luvussa 2.

Projekti toteutetaan neljän hengen ryhmässä. Ryhmään kuuluu Ilari Liukko, Tuomo Pieniluoma, Vesa Pikki ja Panu Suominen. Projektipäällikkönä toimii Vesa Pikki. Projektin vastaavana ohjaajana toimii Ville Isomöttönen ja teknisenä ohjaajana Miika Nurminen. Projektiryhmästä ainoastaan Vesa Pikillä on aiempaa työkokemusta ohjelmistoalalta.

Projekti toteutetaan opintojakson määrittelemässä 4 kuukauden ajassa käyttäen iteraatiivista prosessimallilla, jossa ohjelmisto toteutetaan useassa iteraatioissa. Jokaiselle iteraatiolle asetetaan toteutettavat vaatimukset ja aikataulu. Iteraatioita toteutetaan noin 2-3 viikon pituisina. Prosessimallia kuvataan tarkemmin luvussa 5.

1.2 Dokumentin tarkoitus

Tämän dokumentin tarkoituksena on toimia projektin läpiviennin suunnitelmana. Projektisuunnitelma kuvaa projektin aiheen ja tavoitteet, käytössä olevat resurssit, läpiviennin ja riskit. Läpiviennin osalta projektisuunnitelma sisältää prosessimallin, hallinnan, vastuualueiden, työtehtävien ja ajankäytön kuvauksen jokaisen ryhmän jäsenen osalta.

1.3 Dokumentin sisältö

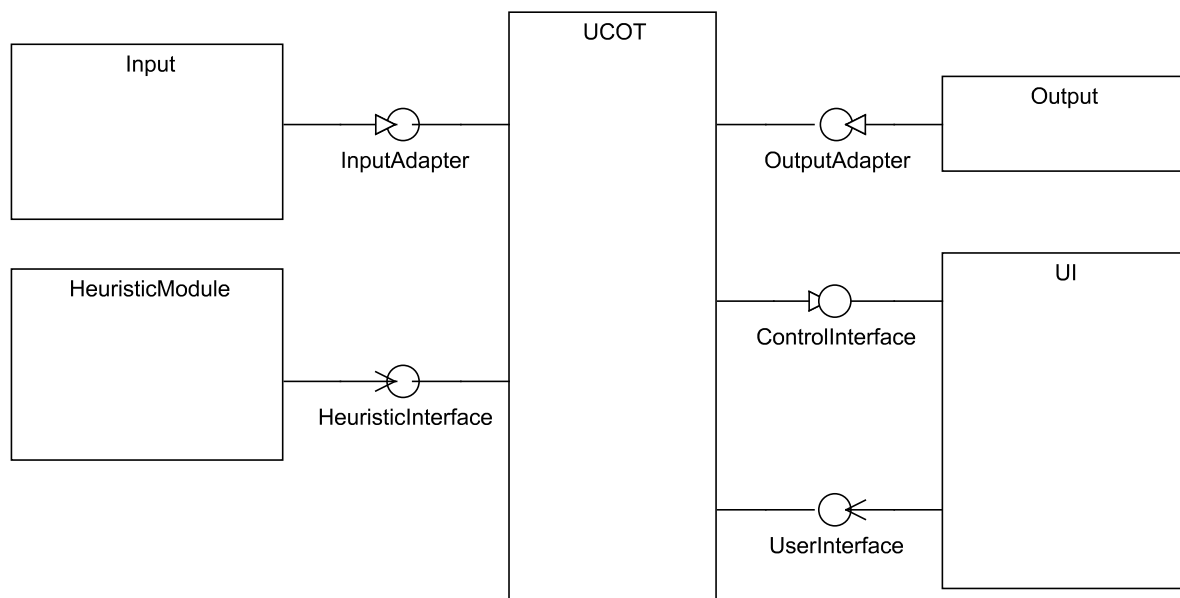
Luvussa 2 kuvataan projektin keskeisimmät tavoitteet. Luvussa 3 esitellään projektiorganisaatio ja sen käytössä olevat resurssit. Luvussa 4 kuvataan projektin hallintatapoja. Luvussa 5 käsitellään projektin tehtävänjakoa ja tehtäväkokonaisuuksia sekä arvioidaan eri tehtävien työmääriä. Luvussa 6 kuvataan projektissa käytettävä prosessimalli ja sen aikataulu. Luvussa 7 käsitellään projektiin liittyviä riskejä ja arvioidaan niiden vaikutuksia projektin läpivientiin.

2 Tavoitteet

Tässä luvussa kuvataan projektin keskeisimmät tavoitteet.

2.1 Sovellus

Sovelluksen tavoitteena on luoda syötteenä saadusta käyttötapauskuvauksesta ohjelmistokehityksen analyysivaiheen olioluokkia. Tämä saadaan aikaiseksi soveltamalla Abbottin-heuristiikka jäsennetylle käyttötapauskuvaukselle. Kuvassa 2.1 on esitelty sovelluksen arkkitehtuuri.



Kuva 2.1: Arkkitehtuurisuunnitelma

Kuvassa 2.1 on määritelty sovelluksen oleelliset komponentit ja niiden väliset rajapinnat. Jokainen komponentti *core*-osaa lukuun ottamatta toimii rajapinnan kautta, mikä mahdollistaa niiden vaihtamisen. Halutessa voidaan pienillä muutoksilla ottaa käyttöön useita jäsentimiä tai heuristiikkoja.

Sovellus pyytää *InputAdapter*-rajapinnan kautta *Input*-komponenttia jäsentämään käyttötapauskuvauksen. *Input*-komponentti jäsentää käyttötapauskuvauksen sanat sanaluokittain ja palauttaa sen *InputAdapter*-rajapinnan kautta takaisin sovellukselle *InputAdapter*-rajapinnan määrittämässä muodossa. Tämän jälkeen *AbbottsHeuristic* muodostaa jäsennetyistä käyttötapauskuvauksesta analyysi-

vaiheen olioluokkia. Output muotoilee analyysivaiheen olioluokista myöhemmin määriteltävän tulosteen.

2.2 Dokumentit

Projektin aikana tuotetaan seuraavat dokumentit:

Projektisuunnitelma selventää yleisellä tasolla, mistä projektissa on kyse. Projektisuunnitelmassa esitellään projektin läpivienti, projektiorganisaatio, ympäristö, resurssit, käytänteet ja aikataulut.

Vaatusmäärittely kertoo millaisia teknisiä ja toiminnallisia vaatimuksia valmiilta sovellukselta edellytetään. Vaatusmäärittely päivitetään iteraatiokohtaisesti.

Projektiraportti kertoo, kuinka projektisuunnitelma ja aikataulut ovat toteutuneet. Projektiraportti esittää myös projektin työtehtävien jakautumisen projektiryhmän jäsenten kesken.

Sovellusraportti on perusteltu kuvaus toteutetusta sovelluksesta. Sovellusraportti sisältää kuvauksia sovelluksen osien toiminnasta, luokkajaoista, rajapinnoista ja tietorakenteista.

Testausraportti määrittelee, kuinka sovellusta testataan kehitysvaiheessa. Siihen kirjataan myös testauksen tulokset iteraatiokohtaisesti. Dokumenttia päivitetään projektin läpiviennin ajan.

Asennusohje on lyhyt ohje sovelluksen asentamiseen käyttäjän tietokoneelle.

Käyttöohje opastaa sovelluksen käyttämisessä.

Ajankäyttöraportit jokaisen iteraation osalta selvittävät kunkin projektiryhmän jäsenen ajankäyttöä kyseisessä iteraatiossa.

Sähköpostiarkistot keräävät kaikki projektin sähköpostilistojen viestit.

Esittelyraportit laaditaan väliesittelyjen pohjalta.

2.3 Tulosten luovutus

Projektin lopputuloksiin kuuluvat sovelluksen ja edellä mainittujen dokumenttien lisäksi dokumentoidut lähdekoodit sekä kaikki muut projektin aikana tuotetut asiakirjat. Muihin dokumentteihin kuuluu muun muassa sovelluksen oma käyttöta-pauskuvaus. Tuotoksiin myös kuuluu palavereihin keskustelun tueksi tehtyjä erillisiä suunnitelmia ja asiakirjoja.

Kaikki projektin dokumentit, lähdekoodit ja itse sovellus arkistoidaan kahteen **projek-tikansioon**, jonne liitetään myös digitaalinen versio projektin tuloksista CD-levyllä. Toinen projektikansioista luovutetaan asiakkaalle ja toinen jää laitokselle. Tämän lisäksi laitoksen arkistoon toimitetaan yksi tulos-CD.

2.4 Projektiryhmän oppimistavoitteet

Projektin yksi päätavoitteista on saada käytännön kokemusta projektityöskentelystä, harjoitusta ryhmätyöskentelystä ja käytettävän prosessimallin läpiviennistä. Dokumentoinnin harjoittelu, työtehtävien ajankäytön ja huolellisen toteutuksen suunnittelu kuuluvat myös projektin tavoitteisiin. Projektin aikana opetellaan myös uusien työkalujen käyttöä ja hankitaan lisää ohjelmointikokemusta.

3 Organisaatio ja Resurssit

Tässä luvussa kuvataan projektin organisaatio ja sen käytettävissä olevat resurssit.

3.1 Projektiryhmä

UCOT-projektiryhmä koostuu 4 tietotekniikan opiskelijasta. Ilari Liukko on 5. vuoden mobiilijärjestelmien opiskelija. Vesa Pikki on 5. vuoden ohjelmistotekniikan ja mobiilijärjestelmien opiskelija ja toimii projektipäällikkönä. Tuomo Pieniluomo on 4. vuoden tieteellisen laskennan ja tietoliikenteen opiskelija. Panu Suominen on 4. vuoden ohjelmistotekniikan opiskelija.

3.2 Koulutus

Projektiryhmä saa sovellusprojektiohjelmistojen jaksoon kuuluvan koulutuksen lisäksi koulutusta Bugzillan käytöstä. Projektiryhmän koulutus on esitelty projektin osapuolille jaetussa yhteystietodokumentissa.

3.3 Resurssit

Projektiryhmällä on käytössä Agoran toisessa kerroksessa sijaitseva lukittava projektitila, huone C222.2. Projektitilassa on käytössä neljä tietokonetta, joihin on asennettu projektin vaatimat ohjelmistot. Projektiryhmä hyödyntää laitoksen ATK-tukea ohjelmistojen asentamisessa ja mahdollisissa laitteisto-ongelmissa. Projektiryhmällä on käytössä kaksi ohjaajaa, joista toiselta saadaan teknistä ohjausta ja toiselta ohjausta projektin läpivientiin. Ohjauksen pyytäminen on projektiryhmän vastuulla.

Ryhmän käytettävissä on muiden projektien kanssa yhteiset projektitilat, joissa sijaitsee myös kokoustila. Ryhmä voi halutessaan lainata laitokselta kirjallisuutta, ja sen lisäksi lainattavissa on kaksi kannettavaa tietokonetta, digisanelin, minidisc-tallennin, digikamera ja kaksi videoprojektoraa.

4 Hallintatavat

Tässä kappaleessa kuvataan projektin hallintatapoja.

4.1 Hallinta

Projektin etenemistä seurataan iteraatioiden taitoskohtiin sijoittuvissa palaverissa. Niissä käydään läpi nykyisen iteraation tavoitteiden toteutuminen ja esitetään asiakkaalle suunnitelma seuraavan iteraation tehtävistä. Tämän lisäksi projektipäällikkö seuraa projektin etenemistä ryhmän sisäisissä palaverissa. Näitä palaveria järjestetään vähintään kerran viikossa ennen viikoittaista palaveria. Ryhmän sisäisissä palaverissa valmistaudutaan viikkopalaveriin selvittämällä viikon aikana toteutetut tuotokset, selvitetään mahdolliset ongelmakohdat ja kootaan asiakkaalle esitettävä aineisto. Projektiryhmä on oikeutettu tarvittavaan ohjaukseen, mutta on veloitettu itse pyytämään sitä.

Excel-ajankäyttösovelluksen avulla projektipäällikkö voi seurata tehtäväkohtaisesti ryhmän jäsenten suunniteltujen työtuntien toteutumista.

Projektia hallitaan myös jakamalla vastuualueet ja työtehtävät ryhmän jäsenille. Näitä kuvataan tarkemmin luvussa 5. Vastuualueiden jakamisen myötä projektipäällikkö pystyy erityisesti seuraamaan tehtäväkokonaisuuksien toteutumista ja keskittymään asiakasrajapintaan.

Projektin dokumentit ja muut tuotokset sijoitetaan projektin WWW-sivuille, josta asiakas voi seurata projektin etenemistä. Dokumenttien kaikki eri versiot sijoitetaan WWW-sivuille. Projektin WWW-sivuilta löytyy seuraavat dokumentit ja tuotokset:

- Projektin liittyvät dokumentit, jotka kuvattiin luvussa 2.2
- Palaverien esityslistat ja pöytäkirjat
- Ajankäytön raportointi sisältää kuvauksen projektin ryhmän jäsenten käyttämistä työtunneista
- Sovelluksen lähdekoodi
- Projektiryhmän tuottamat muut dokumentit

WWW-sivujen lisäksi projektin tiedotus tapahtuu pääasiallisesti sähköpostilistojen avulla. Opetussähköpostilista on projektiryhmän ja ohjaajien kesken, jota käytetään teknisten asioiden ja läpiviennin käsittelyyn. Projektisähköpostilistaa käytetään asiakkaan kanssa kommunikointiin ja tiedottamiseen. Sen avulla voidaan ilmoittaa muun muassa palavereista ja aikataulumuutoksista.

4.2 Dokumentit ja katselmoinnit

Dokumentit ladotaan $\text{\LaTeX}2_{\epsilon}$ -ladontaohjelmalla käyttäen laitoksen tarjoamaa valmiista dokumenttipohjaa. Dokumentit tallennetaan PDF-muodossa. Dokumentit säilytetään projektin verkkolevyllä. Dokumentit ovat nähtävissä projektin WWW-sivuilla. Dokumenttien nimeämisessä käytetään seuraavaa tyyliä: `UCOT-dokumentin_nimi-versio`. Tiedostonimi on jaettu kolmeen erilliseen osioon, jotka erotetaan toisistaan välilyönnillä. Välilyönnit korvataan alaviivalla. Versionumerointia kuvataan myöhemmässä kappaleessa.

Dokumenttien katselmointi tapahtuu palaverien yhteydessä. Katselmoitavat dokumentit tulee lähettää katselmoinnin osapuolille kaksi työpäivää ennen itse palaveria. Katselmoinnin puheenjohtajana toimii dokumentista vastaava ryhmän jäsen. Katselmoinnin sihteerinä toimii silloisen palaverin sihteeri. Palaverissa dokumentti käydään läpi sivu sivulta katselmoinnin sihteerin kirjatessa ylös ehdotetut muutokset. Katselmoinnissa valitaan myös dokumentille tarkastajat. Katselmoinnin jälkeen dokumentista vastaava ryhmän jäsen kirjoittaa ehdotetut muutokset ja lähettää dokumentin tarkastajille. Dokumentti merkitään hyväksytyksi, kun valitut tarkastajat ovat hyväksyneet sen. Osapuolet vahvistavat hyväksynnän allekirjoituksella.

Projektin dokumenteista vain seuraavat allekirjoitetaan: projektisuunnitelma, projektiraportti ja sovellusraportti.

4.3 Palaverit

Projektipalavereita järjestetään ainakin projektin alkuvaiheessa viikoittain. Palaverin esityslista toimitetaan sähköpostilla osallistujille viimeistään palaveria edeltävän työpäivän aikana. Jokainen palaveriin osallistuva on oikeutettu ehdottamaan muutoksia esityslistaan. Mikäli muutoksia ei vastusteta, ennen palaveria ehdotetut muutokset voidaan lisätä esityslistaan.

Palaverien puheenjohtajan ja sihteerin tehtävät kiertävät projektin ajan jäseniltä toiselle. Vuorot selviävät liitteenä olevasta taulukosta A.1. Puheenjohtajan vastuulle jää kokouksen johtamistehtävien lisäksi esityslistan valmistaminen ja lähettäminen palaveriin osallistujille edellisenä työpäivänä. Sihteeri laatii palaverin jälkeen pöytäkirjan ja toimittaa sen palaveriin osallistujille viimeistään kolme työpäivää palaverin jälkeen.

4.4 Versionhallinta

Projektissa käytetään CVS-versionhallintajärjestelmää sekä dokumenttien että lähdekoodin versioiden hallintaan. Dokumenteissa ja lähdekoodissa käytetään seuraavanlaista versionumerointia: versio 0.x tarkoittaa keskeneräistä versiota, missä x on kokonaisluku väliltä 01-99. Esimerkiksi "0.05". Versio 1.00 on hyväksytty versio. Tarvittaessa numerointi jatkuu edellä mainitulla tavalla.

4.5 Tiedotus

Projektin tiedotuksesta vastaa projektipäällikkö. Tiedottamista tapahtuu viikoittain joko sähköpostitse, puhelimitse tai palavereissa. Erityisesti ongelmatilanteista on tiedotettava viipymättä projektin osapuolille.

5 Tehtävät, työmäärät ja työnjako

Projektin kuluessa projektin jäsenet suorittavat oheiskurssin, joka lasketaan osaksi projektia. Oheiskurssin lisäksi projektin osakokonaisuuksia ovat projektin hallinta, esitutkimus, vaatimusmäärittely, sovelluksen suunnittelu ja toteutus.

Projektin hallinta kattaa projektisuunnitelman laatimisen, tiedottamisen, raportoinnin ja aiheeseen sekä käytettäviin työkaluihin perehtymisen. Projektin hallinta kattaa myös palaverit, niihin valmistautumisen ja pöytäkirjojen laatimisen. Esitutkimus kattaa projektin aihealueeseen tutustumisen ja käytettävien komponenttien kartoittamisen. Vaatimusmäärittely kattaa itse vaatimusmäärittelyn laatimisen lisäksi käyttötapauskuvausten laatimisen. Sovelluksen suunnittelu käsittää varsinaisen sovelluksen eri osa-alueiden suunnittelun. Toteutus kattaa ohjelmoinnin ja lähdekoodin dokumentoinnin lisäksi sovelluksen kriittisten osien testaamisen. Oheiskurssi sisältää projektiin liittyvät luennot ja väliesittelyt.

Projektin viimeistely on aikataulutettu siten, että projektia voidaan viimeistellä ennen loppuesittelyä, ja sen jälkeen mikäli loppuesittelyssä ilmenee siihen tarvetta.

Iteratiivisen prosessimallin takia työvaiheet ja ajankäyttö suunnitellaan iteraatiokohtaisesti. Luvussa 6 esitellään ajankäytön suunnitelma. 22.9.2006 alkaneen ensimmäisen iteraation työvaiheet on esitelty taulukossa 5.2. Seuraavista iteraatioista laaditaan vastaava tehtävänjako kunkin iteraation taitekohdassa.

Työnjako projektissa toteutetaan siten, että kaikki ryhmän jäsenet osallistuvat ainakin osittain kaikkiin tehtäviin. Projekti on jaettu vastuualueisiin, jotka on jaettu projektiryhmän jäsenten kesken. Projektiryhmän jäsenten vastuualueet on kuvattu tarkemmin taulukossa 5.1.

Tehtävät tunnistetaan projektissa iteraatiokohtaisesti. 13.9.2006 alkaneessa noin viikon mittaisessa 0-iteraatioissa valittiin prosessimalli, tutustuttiin aiheeseen ja asennettiin projektissa tarvittavat ohjelmistot projektin tietokoneisiin sekä suunniteltiin alustavasti ohjelmiston arkkitehtuuria. Ensimmäisissä iteraatioissa käyttötapauskuvausten suunnittelu ja vaatimusmäärittely saa huomattavasti enemmän painoarvoa, keskivaiheilla sovelluksen suunnittelu ja toteutus ja lopussa viimeistely.

5.1 Vastuiden jako

Projektin jäsenten vastualueet on listattu taulukossa 5.1.

Vastuualue	Vastuuhenkilö
Projektin hallinta	Vesa
Esitutkimus	Tuomo
Sovelluksen osakokonaisuuksien toteutus	
rajapinnat	Tuomo
syöte ja jäsenin	Ilari
käyttöliittymä	Ilari
tuloste	Panu
heuristiikka	Panu
ydin	Panu
Lähdekoodi	
dokumentointi	Ilari
yhdenmukaisuus	Ilari
viimeistely	Vesa
Vaatimusten hallinta	Tuomo
Sovelluksen suunnittelu	Vesa
Käyttötapauskuvaukset	Panu
Testaus	Tuomo
WWW	Tuomo

Taulukko 5.1: Projektin tehtävien vastuiden jako.

5.2 1. Iteraation tehtävien jako

Ryhmän jäsenten tehtävien jako 1. Iteraation kohdalta on esitetty taulukossa 5.2.

Tehtävä	Ilari	Tuomo	Vesa	Panu	Yhteensä
Projektin hallinta	7	27	39	7	80
Seuranta ja tiedotus	0	2	11	0	13
ajankäytön hallinta	0	2	2	0	4
projektin suunnittelu (projektisuunnitelma)	0	15	15	0	30
viikkopalaveri	4	4	4	4	16
-valmistelu	0	2	4	0	6
-pöytäkirja	3	0	3	3	9
www-sivujen toteutus	0	2	0	0	2
Esitutkimus	0	8	0	0	8
jäsentimeen perehtyminen	0	8	0	0	8
Vaatimusmäärittely	27	0	0	27	54
kirjoitus	10	0	0	10	20
käyttötapauskuvausten esitutkimus	5	0	0	5	10
käyttötapauskuvausten palaveri	2	0	0	2	4
käyttötapauskuvausten kirjoitus	10	0	0	10	20
Sovelluksen suunnittelu	0	2	0	2	4
arkkitehtuurisuunnitelma	0	0	0	2	2
rajapintojen suunnittelu	0	2	0	0	2
Toteutus	7	4	0	5	16
toteutuksen osavaiheet	7	4	0	5	16
testaus	0	0	0	0	0
Oheisluennot	4	4	6	4	18
Bugzilla-luento	2	2	2	2	8
CVS-luento	2	2	2	2	8
projektipäälliköiden tapaaminen	0	0	2	0	2
Yhteensä	45	45	45	45	180

Taulukko 5.2: 1. Iteraation tehtäväjako.

5.3 Dokumenttien vastuuhenkilöt

Dokumenttien vastuuhenkilöt on esitetty listassa 5.3.

Dokumentit	Ryhmän jäsen
Testausraportti	Tuomo
Projektisuunnitelma	Vesa
Projektiraportti	Vesa
Vaatimusmäärittely	Panu
Sovellusraportti	Ilari

Taulukko 5.3: Dokumenttien vastuuhenkilöt.

6 Prosessimalli ja aikataulu

Projekti toteutetaan käyttäen iteratiivista prosessimallia. Jokaisessa iteraatiossa on tarkoitus kehittää edellisessä iteraatiossa aikaansaanut tuotosta ja näin varmistaa, että projekti etenee tasaisesti ja tilaaja pääsee mahdollisimman pian seuraamaan projektin lopputuotteen eli sovelluksen kehitystä. Iteratiivisella prosessimallilla minimoidaan myös projektin kannalta kriittisiä riskejä ja voidaan helpommin vastata uusiin vaatimuksiin.

Käytettävässä prosessimallissa projektiryhmän ja asiakkaan välinen kommunikointi on keskeisessä asemassa. Kommunikoinnin helpottamiseksi järjestetään projektiryhmän ja asiakkaan välisiä palavereja viikoittain tai tarpeen mukaan useammin. Iteraatioiden taitekohdat sijoittuvat viikkopalaveriihin.

Projektin tehtävät ja ajankäyttö suunnitellaan iteraatiokohtaisesti iteraatioiden taitekohtiin sijoittuvissa palavereissa. Niissä projektiryhmä arvioi asiakkaan kanssa uudelleen projektin prioriteetit ja päättää seuraavan iteraation tehtävistä. Projektin läpiviennin aikana toteutetaan viisi iteraatiota.

13.9.2006 alkaneessa noin viikon mittaisessa 0-iteraatioissa valittiin prosessimalli, tutustuttiin aiheeseen ja asennettiin projektissa tarvittavat ohjelmistot projektin tilan tietokoneisiin sekä suunniteltiin alustavasti ohjelmiston arkkitehtuuria. Ensimmäisissä iteraatioissa käyttötapauskuvausten suunnittelu ja vaatimusmäärittely saa huomattavasti enemmän painoarvoa. Projektin keskivaiheilla keskitytään sovelluksen suunnitteluun ja toteutukseen. Projektin lopussa keskitytään enimmäkseen viimeistelyyn.

6.1 Iteraatioiden aikataulut

Taulussa 6.1 on esitetty iteraatioiden suunnitellut taitekohdat ja tärkeimmät tavoitteet. Iteraatioiden välisellä aikajaksolla tarkastetaan asiakkaan kanssa edellisen iteraation tuotokset ja hiotaan seuraavan iteraation suunnitelmaa. Tämä *siirtymävaihe* kestää kahdesta päivästä viikkoon, riippuen tehtävien määrästä.

Iteraatio	Kesto	Tärkeimmät tehtävät
0	1 Viikko	Aiheeseen tutustuminen, arkkitehtuurisuunnittelun aloitus, esitutkimus
1	2 Viikkoa	Vaatimusmäärittely, projektisuunnitelma, sovelluksen "ketjun"-todennus
Siirtymävaihe	1 Viikko	Edellisen iteraation tulosten hyväksyminen, seuraavan iteraation suunnitelmien tarkentaminen
2	2 Viikkoa	Sovelluksen ytimen laajennos, käyttöliittymän prototyyppi, testausraportin aloitus, testaus
Siirtymävaihe	1 Viikko	Edellisen iteraation tulosten hyväksyminen, seuraavan iteraation suunnitelmien tarkentaminen
3	2 Viikkoa	Rajapintojen toteutus, toimiva käyttöliittymä
Siirtymävaihe	1 Viikko	Edellisen iteraation tulosten hyväksyminen, seuraavan iteraation suunnitelmien tarkentaminen
4	2 Viikkoa	Käyttöliittymän viimeistely, rajapintojen viimeistely, testaus, sovellus- ja projektiraportin aloitus
Siirtymävaihe	1 Viikko	Edellisen iteraation tulosten hyväksyminen, seuraavan iteraation suunnitelmien tarkentaminen
5	3 Viikkoa	Projektin tulosten viimeistely ja hyväksyminen

Taulukko 6.1: Iteraatioiden suunnitellut taitekohdat.

7 Riskit ja niiden seuranta

Tässä luvussa kuvataan projektin läpivientiin liittyviä riskejä, niiden todennäköisyyttä ja vaikutusta projektiin. Riskit on listattu taulukossa 7.1 käyttäen pieni / kohtalainen / suuri -asteikkoa todennäköisyydelle ja vaikutukselle.

Riski	Todennäköisyys	Vaikutus
Henkilöihin liittyvät riskit		
Projektiryhmän jäsenten poissaolot	Pieni	Kohtalainen
Ohjaajien poissaolo	Pieni	Kohtalainen
Tilaajan edustajien poissaolo	Pieni	Suuri
Kokemuksen puute	Suuri	Kohtalainen
Viestinnän ongelmat	Kohtalainen	Suuri
Toteutukseen liittyvät riskit		
Laitteisto- ja ohjelmisto-ongelmat	Pieni	Suuri
Ongelmat ohjelmoinnissa	Pieni	Kohtalainen
Aikataulun suunnittelu	Kohtalainen	Suuri
Vaatimusten muuttuminen	Kohtalainen	Kohtalainen
Sovellusaluekohtaiset riskit		
Jäsentimen valinta	Kohtalainen	Suuri
Heuristiikan soveltaminen	Kohtalainen	Suuri
Käytettävät tallenusmuodot	Kohtalainen	Kohtalainen

Taulukko 7.1: Projektin läpivientiin liittyviä riskejä

7.1 Henkilöihin liittyvät riskit

Tässä luvussa kerrotaan projektin henkilöihin liittyviä riskejä.

7.1.1 Projektiryhmän jäsenten poissaolot

Projektiryhmän jäsenillä voi olla lomaa tai muita poissaoloja henkilökohtaisten syiden takia.

Ennakointi: Projektiryhmän jäsenet ovat velvoitettuja ilmoittamaan etukäteen tulevista menoista.

Seuranta: Projektiryhmän jäsenten ajankäyttöä seurataan Excel-taulukolla.

Reagointi: Poissaolevan projektiryhmän jäsenen työt jaetaan läsnä olevien jäsenten kesken.

7.1.2 Kokemuksen puute

Projektiryhmän jäsenille ei ole UCOT-projektia edeltävää kokemusta täysimittaisesta projektityöskentelystä ohjelmistoalalla. Vesa Pikki on projektiryhmästä ainut, joka on työskennellyt ohjelmistoalalla, mutta hänellä ei ole kokemusta projektin johtamisesta.

Ennakointi: Projektiryhmän jäsenet tutustuvat sovellusprojektissa jaettuun materiaaliin, jossa projektin käytänteitä on selitetty. Tämän lisäksi hyödynnetään projektin ohjaajien opastusta.

Seuranta: Projektin läpiviennin aikana tunnistetaan projektiryhmän jäsenten heikoudet ja tuetaan taitojen kehitystä.

Reagointi: Projektiryhmän jäsenillä on aina jokaisessa tehtävässä toinen projektiryhmän jäsen avustamassa kyseisen tehtävän kanssa. Tämän lisäksi hyödynnetään ohjausta välittömästi.

7.1.3 Viestinnän ongelmat

Projektiorganisaation henkilöiden erilaisten taustojen takia kommunikoinnissa voi terminologian kanssa tulla ongelmia. Organisaation koon takia kommunikointi voi muodostua ongelmaksi.

Ennakointi: Projektin läpiviennin ajan pyritään välittömään ja kasvotusten tapahtuvaan kommunikointiin. Tämän tueksi ylläpidetään listaa projektin osa-alueiden termeistä. Sekä hoidetaan yleinen kommunikointi sähköpostilistan kautta. Kaikki projektiorganisaation jäsenet kuuluvat siihen. Tämän lisäksi kaikkien projektiorganisaatioon kuuluvien puhelinnumerot on listattu projektisuunnitelmaan.

Seuranta: Kaikki projektiorganisaation jäsenet ovat aktiivisesti mukana projektissa viestinnän osalta.

Reagointi: Projektiorganisaation eri osapuolten välisiä viestintäongelmia pyritään

ratkomaan tarpeen mukaan joko puhelimitse, sähköpostitse, kasvokkain viikkopalavereissa tai mahdollisuuksien mukaan ylimääräisissä palavereissa.

7.1.4 Ohjaajien poissaolo

Ohjaajien muiden kiireiden ja mahdollisten sairastumisten takia, he eivät aina ehdi opastamaan projektin jäseniä lyhyellä varoitusajalla.

Ennakointi: Projektin läpivienti pyritään pääsääntöisesti pitämään projektiryhmän vastuulla niin, ettei projektin eteneminen pysähdy ohjaajien mahdollisiin poissaoloihin. Projektin hallintaan ja suunnitteluun on pyydetty paljon ohjausta projektin alkuvaiheessa.

Seuranta: Projektiryhmä sopii tapaamiset ohjaajien kanssa etukäteen ja ohjaajat tiedottavat ryhmän jäsenille mahdollisista esteistä.

Reagointi: Tarvittavaa ohjausta siirretään seuraavaan mahdolliseen ajankohtaan tai ohjausta voidaan hakea tarvittaessa laitokselta.

7.1.5 Tilaajan edustajien menot

Tilaajan edustajat eivät välttämättä aina muilta menoiltaan ehdi viikkopalaveriin tai vastaamaan heille lähetettyihin sähköposteihin.

Ennakointi: Tärkeimmät asiat käsitellään palavereissa, joiden ajankohta on sovittu hyvissä ajoin. Palaverit eivät ole päätösvaltaisia, ellei ainakin toinen tilaajan edustajista ole paikalla. Tilaajan edustajaan otetaan yhteyttä puhelimitse tärkeiden asioiden selvittämiseksi, mikäli asia hidastaa projektin etenemistä.

Seuranta: Tilaajan edustajat ilmoittavat etukäteen projektiryhmälle sopivaksi katsomallaan tavalla mahdollisista esteistä. Projektiryhmä pitää yllä ryhmän sisäistä kysymyslistaa viikkopalaveria varten.

Reagointi: Pyritään tavoittamaan tilaajan edustaja puhelimitse. Jos päätöksiä ei voida projektin kannalta tehdä ilman tilaajan edustajan läsnäoloa, niin tällöin sovitaan erikseen tilaajan edustajien kanssa käytettävästä menettelystä.

7.2 Toteutukseen liittyvät riskit

Tässä luvussa käsitellään projektiin liittyviä teknisiä ja toteutuksellisia riskejä.

7.2.1 Laitteisto- ja ohjelmisto-ongelmat

Projektiryhmän käytössä oleva laitteisto saattaa jostakin satunnaisesta syystä vioittua, jokin käytettävistä ohjelmistoista saattaa osoittautua epävakaa tai projektiryhmälle syntyy tarve käyttää sellaista ohjelmistoa, jota ei projektitilan laitteille ole vielä asennettu.

Ennakointi: Pyritään hankkimaan tarvittavat ohjelmistot etukäteen ja hajautetaan tiedostot mahdollisten laitteisto-ongelmien varalta.

Seuranta: Käytetään laitteistoa ja ohjelmistoja aktiivisesti.

Reagointi: Otetaan yhteys laitoksen ATK-tukeen, jonka tehtävänä on ratkaista ongelmatilanteet. Lopuksi varmistetaan, että ATK-tuki on poistanut ongelman.

7.2.2 Ongelmat ohjelmoinnissa

Sovellusta suunniteltaessa tai ohjelmoitaessa saattaa ilmaantua teknisesti hyvin moinimutkaisia tehtäviä, joiden toteuttaminen on projektiryhmän jäsenille vaikeaa ja tulos epävakaa.

Ennakointi: Suunnitelmat validoidaan teknisellä ohjaajalla. Tuotettu ohjelmakoodi testataan. Perehdytyksillä varmistetaan, että projektiryhmän jäsenillä on tarvittavat esitiedot.

Seuranta: Projektin siirtymävaiheessa ohjelmakoodi tarkistetaan teknisen ohjaajan kanssa.

Reagointi: Ongelmatilanteet ratkotaan mahdollisuuksien mukaan projektiryhmän jäsenten kesken tai tarvittaessa ohjaajien ja tilaajan kanssa. Pyydettyäessä on mahdollisuus saada laitokselta asiantuntija-apua erittäin haastaviin tehtäviin.

7.2.3 Aikataulun suunnittelu

Projektin eri työtehtävien työmäärien arviointi voi epäonnistua ja jokin työtehtävä voi osoittautua arvioitua vaativammaksi. Tämä voi aiheuttaa iteraatioiden pitkittymistä tai pahimmassa tapauksessa viivästymistä koko projektille.

Ennakointi: Aikataulut ja tehtävät suunnitellaan iteraatiokohtaisesti, jolloin työtehtävien työmäärien arviointi helpottuu ja aikataulut saadaan suunniteltua joustavasti.

Seuranta: Projektiryhmän jäsenet pitävät kirjaa toteutuneista työtunneista ja työtehtävistä.

Reagointi: Aikataulujen suunnittelun epäonnistuminen kompensoidaan mahdollisimman pian siirtämällä joko vapaata tai pienemmän prioriteetin tehtävien aikaa ajallisesti vaativampiin kokonaisuuksiin. Tarvittaessa toteutettavien osakokonaisuuksien ominaisuuksia voidaan asiakkaan suostumuksella karsia ja näin pysyä aikataulussa.

7.2.4 Vaatimusten muuttuminen

Projektin edetessä on mahdollista ja todennäköistä, että tilaajalle tulee uusia vaatimuksia sovelluksen toiminnallisuuksien suhteen tai vanhat vaatimukset tai niiden tärkeysjärjestys muuttuu oleellisesti.

Ennakointi: Alussa vaatimukset kartoitetaan pääpiirteittäin ja niitä lähdetään toteuttamaan mahdollisimman korkealla tasolla. Projektin loppua kohti vaatimukset tarkentuvat ja niitä toteutetaan iteraatioissa mahdollisuuksien mukaan.

Seuranta: Käytettävän prosessimallin ansiosta asiakas voi helposti seurata vaatimusten toteutumista. Kun sovellusta kehitetään inkrementaalisesti useassa iteraatiossa, niin kunkin vaiheen valmistuttua voidaan valmiit tuotokset esitellä asiakkaalle ja asiakas voi halutessaan tarkentaa vaatimuksia.

Reagointi: Aikataulut suunnitellaan uudestaan uusiin vaatimuksiin sopivammiksi.

7.3 Sovellusaluekohtaiset riskit

Tässä luvussa kuvataan projektin sovellusaluekohtaiset riskit.

7.3.1 Jäsentimen valinta

Projektin läpiviennin kannalta oleelliseksi muodostuu sopivan jäsentimen valinta. Ilman jäsenintä rajapintojen suunnittelu ja sovelluksen testaus vaikeutuu. Mikäli jäsenin valitaan myöhäisessä vaiheessa projektia, voi jäsentimien yleistä rajapintaa joutua suunnittelemaan uudelleen. Tämän lisäksi myös jäsentimen integrointi vie aikaa. Nämä molemmat vaikuttavat muiden sovelluksen osa-alueiden toteuttamiseen ja aikatauluihin.

Ennakointi: Pyritään tutkimaan olemassa olevia jäsentimiä ja sen avulla suunnittelemaan jäsentimien rajapinta huolellisesti. Tämän lisäksi pyydetään tilaajan edustajia ilmoittamaan jäsentimen valinnasta mahdollisimman aikaisessa vaiheessa.

Seuranta: Käytettävä prosessimalli mahdollistaa nopean reagoinnin muutoksiin, jonka avulla voidaan paremmin suhtautua jäsentimen valinnan tuomiin muutoksiin. Tämän lisäksi tiedustellaan tilaajan edustajilta jäsentimen valinnasta.

Reagointi: Valittuun jäsentimeen tutustutaan ja tutkitaan mihin sovelluksen osa-alueisiin se vaikuttaa. Tarvittaessa suunnitellaan silloisen iteraation tehtävien aikataulut ja prioriteetit uudelleen.

7.3.2 Heuristiikan soveltaminen

Sovellukselle oleellista on, että käytettävä heuristiikka tuottaa haluttuja tuloksia. Heuristiikka saattaa myös vaihtua projektin läpiviennin aikana.

Ennakointi: Suunnitellaan heuristiikoille mahdollisimman yleinen rajapinta.

Seuranta: Rajapintaa tarkennetaan projektin läpiviennin aikana. Iteraatioiden taitekohdissa tilaajan edustajat näkevät heuristiikan tuottamat tulokset ja voivat tarkentaa vaatimuksia.

Reagointi: Vaatimusten tarkennuttua suunnitellaan tarvittaessa iteraation aikataulut ja prioriteetit uudelleen. Suunnitellaan heuristiikan toteutus ja tutkitaan mihin sovelluksen osa-alueisiin mahdollinen muutos vaikuttaa.

7.3.3 Käytettävät tallenusmuodot

Sovelluksen tulosten, käyttötapauskuvausten ja keskeneräisen analyysivaiheen oliomallin tallenusmuotoja ei ole vielä päätetty. Päätöksen pitkittyminen muodostuu projektin kannalta riskiksi, sillä niiden tutkimiseen ja toteuttamiseen tulee voida varata riittävästi aikaa. Testaamisen kannalta päätös on myös oleellinen, koska muutoin sovelluksen koko suoritusketjua ei voida todentaa.

Ennakointi: Sovellus suunnitellaan siten, että tallennus- ja latauskomponentti voidaan vaihtaa, sekä ettei muu toteutus esty edellä mainittujen komponenttien puutteen takia.

Seuranta: Soveltuvia tallennusmuotoja kartoitetaan ja tutkitaan. Tämän lisäksi projektin alkuvaiheessa toteutetaan tiedoston lataamiseen ja tallentamiseen testikomponentteja, joiden avulla voidaan tarkentaa käytettävää rajapintaa.

Reagointi: Valittuun tallennusmuotoon tutustutaan ja sen toteuttamiseen varataan aikaa.

8 Yhteenveto

UCOT-projekti toteuttaa sovelluksen, jolla voidaan muodostaa käyttötapauskuvauksista analyysivaiheen olioluokkia. Tämän dokumentin oleellisimpiin tuloksiin kuuluu ajankäytön, iteraatioiden työtehtävien suunnitelmat ja riskianalyysi. Tässä dokumentissa esitellyt suunnitelmat auttavat projektin läpiviennissä.

Tuotettua sovellusta ei tämän projektin aikana voida toteuttaa täysin valmiiksi, joten sen jatkokehitys on välttämätöntä ennen varsinaista käyttöönottoa. Jatkokehitystä voidaankin tämän projektin jälkeen tehdä monella eri sovelluksen osa-alueella, mukaan lukien uusien jäsentimien sovittaminen sovellukseen, käyttöliittymän tai liittymien parantelu sekä erilaisten tallennusmuotojen lisääminen sovellukseen.

A Palaverin vuorolista

Taulukossa A.1 on esitetty UCOT-projektin palaverien vuorolista.

Palaveri	Ilari	Panu	Tuomo	Vesa
1	Sihteeri	Puheenjohtaja	Sihteeri	
2	Puheenjohtaja	Sihteeri		
3	Sihteeri			Puheenjohtaja
4	Sihteeri		Puheenjohtaja	
5		Puheenjohtaja	Sihteeri	
6	Puheenjohtaja	Sihteeri		
7	Sihteeri			Puheenjohtaja
8			Puheenjohtaja	Sihteeri
9		Puheenjohtaja	Sihteeri	
10	Puheenjohtaja	Sihteeri		
11	Sihteeri			Puheenjohtaja
12			Puheenjohtaja	Sihteeri
13		Puheenjohtaja	Sihteeri	
14	Puheenjohtaja	Sihteeri		
15	Sihteeri			Puheenjohtaja
16			Puheenjohtaja	Sihteeri
17		Puheenjohtaja	Sihteeri	
18	Puheenjohtaja	Sihteeri		
19	Sihteeri			Puheenjohtaja

Taulukko A.1: Palaverien vuorolista.

B Termit

Alkuperäinen käyttötapaus	on lähteen sisältämä käyttötapaus.
Core	on sovelluksen ydin, joka ohjaa ohjelman muiden komponenttien toimintaa.
Entiteetti	on vaatimusmäärittelyssä esiintyvä toimija tai toimenpiteen kohde. Käytännössä mikä tahansa substantiivi voi olla entiteetti.
HeuristicCollection	on luokka, joka säilöö <code>HeuristicModule</code> ja.
HeuristicInterface	on rajapinta, joka määrittää miten <code>HeuristicModule</code> en kanssa kommunikoidaan.
HeuristicModule	on luokka, joka suorittaa heuristiikan sille annetulle jäsenetylle käyttötapaukselle ja palauttaa käsitellin.
InputAdapter	on luokka, joka lataa URL:llä osoitetun lähteen sisältämät alkuperäiset käyttötapaukset ja palauttaa ne jäsentämättöminä käyttötapauksina.
InputInterface	on rajapinta, joka määrittää miten <code>InputAdapter</code> in kanssa kommunikoidaan.
InputCollection	on luokka, joka säilöö input adaptereita.
Iteraatio	tarkoittaa yleisesti jonkin asian toistamista uudelleen siten, että edellisen suorituskerän tulos on seuraavan kerän syöte. Sovelluskehityksessä iteraatiolla tarkoitetaan projektin suorittamista pienissä paloissa edellisen iteraation tulosten toimissa seuraavan iteraation toteutuksen pohjana. Tuloksilla tässä tapauksessa tarkoitetaan kaikkea ohjelmiston kehityksen tuottamaa materiaalia eikä vain lähdekoodia. Peräkkäiset iteraatiot eivät välttämättä käytä ollenkaan samaa lähdekoodia vaan koodi voidaan välillä kirjoittaa uudestaan.

Jäsennetty käyttötapaus	on käyttötapaus, jolle on suoritettu morfologinen jäsenitys.
Jäsentämätön käyttötapaus	on käyttötapausten suoritusaskeleet tekstimuodossa. Käyttötapaus on jo otettu sisään järjestelmään, mutta sitä ei ole vielä toimitettu parserille.
Käsitemalli	on heuristiikan muodostama malli jäsenetystä käyttötapauksesta.
Käyttötapaus	on kuvaus järjestelmän ja sen käyttäjän välisestä vuorovaikutuksesta tietyn tuloksen aikaansaamiseksi.
Käyttötapausten muoto	kertoo, mitä attribuutteja ja missä järjestyksessä sekä muodossa yksittäisen käyttötapausten kuvaus sisältää. Näitä attribuutteja ovat mm. tiedot pääaktorista ja muista aktoreista, tietoa järjestelmän tilasta ennen ja jälkeen käyttötapausten toiminnan sekä käyttötapausten suoritusaskeleet.
LaTeX 2ϵ	on ladontaohjelmisto, millä tämäkin dokumentti on tehty.
Moduuli	on ohjelman osa, joka piilottaa varsinaisen toiminnan toteutuksen sisäänsä. Hyvin kirjoitetun moduulin sisäistä toteutusta on helppo muuttaa. Yleensä moduuli toteuttaa jonkin rajapinnan vaatiman toiminnallisuuden. Esimerkiksi UCOT-ohjelmistossa eri läheteistä tullutta dataa voidaan lukea kun vain datan lukemista varten on toteutettu moduuli, joka täyttää UCOT-ohjelmiston syöterajapinnan määrittelyn.
Output	on luokka, joka hoitaa heuristiikan tuottaman käsitemallin esittämisen/tallentamisen.
OutputCollection	on luokka, joka säilöö Outputteja
OutputInterface	on rajapinta, joka määrittää outputin kanssa kommunikoidaan.

Parser	tarkoittaa morfologista jäsenointiä.
ParserAdapter	on luokka joka toteuttaa <code>ParserInterface</code> ja kommunikoi parserin kanssa. Ottaa vastaan jäsentämättömän käyttötapauksen ja palauttaa jäsenetyn käyttötapauksen.
ParserCollection	on luokka, joka säilöo <code>ParserAdapter</code> ereita.
ParserInterface	on rajapinta, joka määrittää miten <code>ParserAdapter</code> in kanssa kommunikoidaan.
Projekti	tarkoittaa tämän dokumentin yhteydessä sovellusprojektiä.
Rajapinta	erottaa kaksi toisistaan erillistä ohjelman osaa toisistaan siten, että osat tietävät vain osan toisen toiminnallisuudesta. Tällöin rajapinnan takan olevaa osaa voidaan vaihtaa toisen osan häiriintymättä.
Sovellusprojekti	on tietotekniikan laitoksen opintojakso.
Syöte	on ohjelman vastaanottama data.
Syötemoduuli	on ohjelman osa, joka lukee ohjelmalle tarkoitetun syöteen ja palautta ohjelmalle jäsenetyn käyttötapauksen. Rakentuu <code>InputAdapter</code> ista ja <code>ParserAdapter</code> ista.
Tuloste	on ohjelman tuottama data.
UI	tarkoittaa käyttöliittymää (<i>user interface</i>). Tarkentuu myöhemmissä iteraatioissa.
UIInterface	on käyttöliittymän rajapinta.
UCOT	on tämän sovellusprojektin toteuttava ryhmä.
Vaikutussuhde	on kahden entiteetin välillä vallitseva suhde, jossa toinen käyttää toista.