

KOPPELO

Tietotekniikan cum laude -työprojekti

Minna Hillebrand

Markus Silván

Antti Vanhanen

Marko Ylitalo

Projektisuunnitelma

11.3.2002

Jyväskylän yliopisto
Tietotekniikan laitos

Tiivistelmä

KOPPELO-projekti kehittää Jyväskylän yliopiston tietotekniikan laitokselle opinnäytteiden hallinnointisovelluksen. Sovellus toteutetaan olemassaolevan KOTKA-järjestelmän osioksi. Tämän dokumentin tavoitteena on antaa lukijalle kuva projektin resursseista, tavoitteista, tehtäväjaosta, aikataulusta ja riskeistä.

Tietoa KOPPELO-projektista

Tekijät:

Minna Hillebrand (mmhilleb@st.jyu.fi)

Markus Silván (mape@st.jyu.fi)

Antti Vanhanen (vanhanan@st.jyu.fi)

Marko Ylitalo (ylimark@st.jyu.fi)

Työ: Projektisuunnitelma tietotekniikan cum laude -työprojektiin

Tilaaaja ja teettäjä: Jyväskylän yliopiston tietotekniikan laitos

Jukka-Pekka Santanen toimii sekä tilaajan edustajana että laitoksen vastaavana ohjaajana. Teknisinä ohjaajina toimivat Heikki Uuksulainen ja Pauli Kujala.

Työtila: Agora, huone AgC222.2, puhelinnumero 014-2604963.

Avainsanat: tietotekniikan cum laude -työprojekti, opinnäytteet, JSP, KOTKA-järjestelmä, KORPPI-sovellus, KOLIBRI-sovellus

Sisältö

1	Johdanto	1
2	Termistöä	2
3	Projektin taustaa ja tavoitteet	4
3.1	KURKI- ja KOTKA-järjestelmät	4
3.2	KORPPI-sovellus ja HALKO-projekti	4
3.3	KOLIBRI-projekti	5
3.4	Projektin ja sovelluksen tavoitteet	5
3.5	Oppimistavoitteet	6
4	Organisaatio ja resurssit	7
4.1	Henkilöt ja yhteystiedot	7
4.2	Laitteet ja ohjelmistot	7
5	Riskit ja niiden hallinta	8
5.1	Projektin jäsenten poissaolot	8
5.2	Projektin jäsenten motivaatio	8
5.3	Suunnittelu- ja ohjelmointiongelmat	8
5.4	Laitteisto- ja ohjelmisto-ongelmat	9
5.5	Odottamattomat ongelmat	9
6	Dokumentointi	10
6.1	Dokumentointikäytäntö	10
6.2	Lähdekoodin kirjoittaminen	10
7	Työnjako ja aikataulu	11
7.1	Prototyypit	11
7.2	KOPPELO-sovellus	11
7.3	Raportointi	12
8	Yhteenveto	15

1 Johdanto

KOPPELO-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle opinnäytteiden hallinnointisovelluksen. KOPPELO-projekti on tietotekniikan laitoksen kevään 2002 cum laude -työprojekti.

KOPPELO-projekti on jatkoa KURKI-sovelluksen jatkokehittämisestä liikkeelle lähteneeseen opetus- ja tutkimushallintajärjestelmän kehittämiseen. Vuonna 1998 toteutettua KURKI-projektia ovat seuranneet syksyn 2000 KOTKA-projekti, kevään 2001 KORPPI-projekti ja syksyn 2001 KOLIBRI-projekti.

Olemassa oleva järjestelmä ja sen sisältämä tietokanta on nimeltään KOTKA. KORPPI-sovellus on KOTKA-järjestelmän kurssikirjanpito-osio. KOLIBRI-projekti toteutti päivyriohjelmiston, jota tullaan käyttämään useammassa KOTKA-järjestelmän osiossa.

KOPPELO on edelleen KOTKAan liitettävä opinnäytteiden hallintasovellus. Tällaisen sovelluksen tarve on syntynyt opiskelijamäärän kasvun myötä vähentämään ”paperisodan” tarvetta. Lisäksi KOPPELO-järjestelmän tavoitteena on helpottaa opinnäytteiden yleistä hallintaa, sillä toistaiseksi kukin ohjaaja pitää omaa kirjanpitoaan ohjaamistaan opinnäytteistä. KOPPELO-järjestelmän avulla esimerkiksi tietotekniikan laitoksen henkilökunta saa keskitetysti tietoa ehdotetuista, tekeilläolevista ja tehdyistä opinnäytetöistä.

URAANI-projekti toteuttaa kuluvan kevään aikana Avoimelle Yliopistolle opintoneuvonta ja urasuunnittelujärjestelmän, joka myös liittyy KOTKAan.

Tämän dokumentin tavoitteena on antaa lukijalle selvä kuva projektin resursseista, tavoitteista, tehtäväjaosta, aikataulusta ja riskeistä. Projektisuunnitelman lisäksi laaditaan suunnitteluvaiheessa sovellussuunnitelma sekä ennen projektin päättämistä kirjoitetaan sovellus- ja projektiraportti sekä käyttöohje. Lisää projektin aikana tuotettavista dokumenteista kerrotaan luvussa 6.

Lukuun 2 on kerätty projektissa ja sen dokumenteissa käytettäviä termejä. Luvussa 3 esitellään projektin taustoja ja tavoitteita. Projektiorganisaation rakenne ja projektin käytössä olevat resurssit on kuvattu luvussa 4. Mahdolliset riskit projektin onnistumiselle ja etenemiselle aikataulussa on esitelty luvussa 5. Dokumentointia käsitellään luvussa 6. Luku 7 käsittelee projektiryhmän jäsenten välistä työnjakoa sekä projektin aikataulua.

2 Termistöä

Taulukoissa 1 ja 2 on esitelty toteutusympäristöön ja projektiin liittyvää termistöä.

TERMI	SELITYS
Apache	on ilmainen HTTP-palvelinohjelmisto.
CSS	eli Cascading Style Sheets on WWW-sivujen ulkoasua kuvaava kieli.
CVS	eli Concurrent Versions System on versionhallintaan tarkoitettu ohjelmisto.
Delphi	on Inprisen ohjelmistonkehitysympäristö, jossa ohjelmointikielenä käytetään Object Pascal -kieltä.
GNUJSP	on ilmainen JSP-moottori.
HTML	on WWW-sivujen rakennetta kuvaava kieli.
HTTP	on WWW-arkkitehtuurin käyttämä tiedonsiirtoprotokolla.
Java	on Sunin kehittämä laitteistoriippumaton olio-ohjelmointikieli.
Java-pavut	(engl. <i>Java Beans</i>) ovat Java-ohjelmointikielillä luotuja komponentteja, joita voidaan kutsua JSP-sivuilla.
JDBC	eli Java Database Connectivity on Java-teknologian käyttämä tietosilta erilaisiin tietokantoihin.
Jserv	on ilmainen Java-servlettimoottori.
JSP	eli Java Server Pages on skriptaustyylinen ohjelmointikieli.
Keksi	(tai eväste, engl. <i>cookie</i>) on Netscapen luoma, mutta nykyään jo standardoitu menetelmä saada WWW-sivuista vuorovaikutteisempia. Palvelin lähettää pieniä tietopaketteja selaimelle, jonka perusteella käyttäjä ase- tuksineen tunnistetaan jatkossa.
Poolman	on ohjelma, joka kontrolloi tietokantayhteyksiä.
PostgreSQL	on ilmainen tietokannanhallintajärjestelmä.
Skripti	on ohjelmointikieltä muistuttava ja usein hieman yksinkertaistettu tulkattava kieli.
Skriptaus	tarkoittaa WWW-sivujen tapauksessa ohjelmakoodin kirjoittamista HTML-dokumenttien sisään. Palvelinpuolen skriptauksessa koodi ajetaan palvelinkoneessa ja asiakaspään skriptauksessa selaimessa.
Selain	on ohjelma, joka käyttäjän koneella tulkaa HTML-kieliset sivut esitettävään muotoon.
Palvelin	on ohjelmisto, joka WWW-sovellusten tapauksessa palvelee asiakkaana toimivien selainten hakupyynnöjä.
Servletti	on palvelimella sijaitseva sovelma (engl. <i>applet</i>), joka toteuttaa HTTP-palvelimen pyynnöstä tietyn toiminnon.
SQL	eli Structured Query Language on tietokantojen hallintaan kehitetty standardi kieli.
Tomcat	on ilmainen servletti- ja JSP-moottori.

Taulukko 1: Toteutusympäristöön liittyvää termistöä.

TERMI	SELITYS
HALKO KOTKA	on Heinolan kansalaisopistolle toteutettu opetuksenhallintajärjestelmä. on Jyväskylän yliopiston tietotekniikan laitoksella kehitetty opetus- ja tutkimushallintajärjestelmä.
KORPPI KOLIBRI KURKI	on KOTKA-järjestelmän kurssikirjanpitomoduuli. on KOTKA-järjestelmän kalenteriosio. on Jyväskylän yliopistossa käytössä oleva Intrabuilderilla toteutettu WWW-pohjainen kurssikirjanpitojärjestelmä.
KUUKKELI	on Antti Viinikan Jyväskylän yliopistossa kehittämä järjestelmä ohjauksaajien ajanvaraukseen ja hallintaan.
Käyttötapa	(engl. <i>use case</i>) on käyttäjän tai sovelluksen toimintoa tietyn tehtävän suorittamiseksi kuvaava dokumentti.
Moduuli	on Kotka-järjestelmässä tietyn tehtäväjoukon suorittamiseen kehitetty sovellus ja käyttöliittymä.

Taulukko 2: KOTKA-järjestelmään liittyvää termistöä.

Taulukossa 3 esitetään projektin eri dokumenteissa käytettävää opinnäytteisiin liittyvää termistöä.

TERMI	SELITYS
Aihe	on aihe-ehdotuksesta mahdollisesti tietylle linjalle hyväksytty aihe opinnäytteeksi.
Aihe-ehdotus Opinnäyte	on opinnäytteen aiheeksi esitetty idea. on opiskelijan työn alle ottama opintokokonaisuus eli hyväksytty aihe, jolle on kiinnitetty tekijä. Tarkoittaa esimerkiksi graduja, seminaareja ja työharjoitteluja.
Vastaava ohjaaja	on opinnäytteen pääasiallinen ohjaaja. Vastaavalta ohjaajalta voi käydä varaamassa aiheen itselleen, ja yleensä vastaava ohjaaja myös ohjaa opinnäytettä sen edetessä.
Ohjaaja Tarkastaja Kurssi	Muu opinnäytteen ohjaaja. arvostelee opinnäytteen. KOTKA-tietokannan käsite yleiselle kurssinimikkeelle. Jokainen opinnäyte liittyy johonkin kurssiin, jotta opinnäytteeseen liittyisi yksikäsitteinen kurssikoodi.
Kurssi-instanssi	KOTKA-tietokannan käsite tiettyinä lukukautena tai muuna ajankohtana järjestettävälle kurssille.
Opinnäytteen vaihe	Kuvaa opinnäytteen elinkaarta ja ohjausprosessia. Niitä ovat mm. ehdotettu aiheeksi, hyväksytty aiheeksi, tekeillä, hyväksytty ja keskeytetty.
Tapahtuma	Opinnäytteeseen liittyvä raportointi- tai ohjaustapahtuma. Jokaiseen tapahtumaan voi edelleen liittyä oma työvaihe.
Työtehtävä	Esimerkiksi työprojektin tapauksessa opinnäytteen suorituksesta voidaan erottaa projektisuunnitelman, sovellussuunnitelman ja varsinaisen ohjelmoinnin suorittaminen. Nämä työtehtävät voidaan edelleen luokitella esimerkiksi suunnitteluun, ohjelmointiin ja dokumentointiin.

Taulukko 3: KOPPELO-projektiin liittyvää termistöä.

3 Projektin taustaa ja tavoitteet

Luvussa käsitellään projektin lähtökohtia sekä sen tavoitteita.

3.1 KURKI- ja KOTKA-järjestelmät

Pauli Kujala oli keväällä 1998 työprojektissa mukana kehittämässä KURKI-järjestelmää kurssitietojen ylläpitämistä varten. Pauli Kujala on jatkanut KURKI-järjestelmän ylläpitoa ja kehitystä näihin päiviin saakka.

WWW-pohjaisessa KURKI-järjestelmässä on erilliset käyttöliittymät opettajille ja oppilaille. Opettajien käyttöliittymän puolella voidaan luoda kurssista ilmentymä ja liittää kurssiin muun muassa pääteohjauksia ja demotilaisuuksia. Jokaiseen oppilaaseen voidaan liittää kurssikohtaista tietoa. Opiskelijat voivat omassa käyttöliittymässään ilmoittautua kurssille ja valitsemaansa demoryhmään. Oppilas näkee reaaliajassa kurssin liittyvät suoritusmerkintänsä, esimerkiksi kurssin demoissa saamansa pisteet.

KURKI-järjestelmää on arvosteltu sen käytön hankaluudesta. Käytettävyyden parantamista ja systeemin laajentamista rajoittaa lisäksi toteutukseen käytetty tekniikka (Borland Intrabuilder).

KURKI-järjestelmä edustaa muutaman henkilön näkemystä järjestelmän tarpeista ja ominaisuuksista. Käyttäjäkunnan koko ajan laajennuttua järjestelmään on yritetty lisätä uusia ominaisuuksia, mutta ensimmäisistä versioista asti seuranneet heikko käytettävyys ja hitaus vaivaavat järjestelmää edelleen. Järjestelmässä on myös yhteensopivuusongelmia tiettyjen selainten kanssa (erityisesti Internet Explorerin kohdalla). Lisäksi Borland ei ole vuoden 1998 jälkeen tukenut Intrabuilder-työkalua, jolla KURKI-järjestelmä on toteutettu.

KOTKA-projekti suunnitteli ja toteutti uuden järjestelmän tietorakenteen, kartoitti käytettävät työkalut sekä toteutti myöhempien projektiryhmien työtä tukemaan henkilötietojen selaus- ja muokkausmoduulin. KOPPELO-projekti käyttää myös KOTKA-projektin valitsemissa työkaluja.

Lisää KOTKA-järjestelmästä ja sen dokumentaatiosta löytyy Internetistä osoitteesta <http://kotka.it.jyu.fi/>.

3.2 KORPPI-sovellus ja HALKO-projekti

KORPPI-projekti toteutti uuteen KOTKA-järjestelmään kurssikirjanpidon osuuden korvaten vanhan KURKI-järjestelmän. Samalla KORPPI-sovellus paikkasi vanhaan KURKI-järjestelmään verrattuna tietoturva-aukkoja ja paransi käytettävyyttä.

Tietokantaan tarvittavat muutokset toteutettiin osaksi yhdessä HALKO-projektin kanssa. HALKO-projekti toteutti kansalaisopistoille ja muille oppilaitoksille opetushallinto-ohjelman, käyttäen osittain samoja kehitystyökaluja ja -ympäristöä kuin KORPPI-projekti.

Lisää tietoa KORPPI-järjestelmästä löytyy osoitteesta <http://kotka.it.jyu.fi/korppi>.

3.3 KOLIBRI-projekti

KOLIBRI-projekti toteutti yksinkertaisen ja helppokäyttöisen päivyriohjelman Jyväskylän yliopiston henkilökunnan ja opiskelijoiden käyttöön. Kalenteriosioon käyttäjä voi merkitä sekä yksityiset että julkiset opetukseen ja erilaisiin ryhmiin liittyvät tapahtumansa, joista saa nähtäväkseen päivä-, viikko-, kuukausi- ja vuosinäkymät. Sovelluksessa voi merkitä tapahtuman jollekin hallitsemalleen ryhmälle niin, että näkee kaikkien ryhmän jäsenten kalenterit päällekkäin.

Tarkempaa tietoa KOLIBRI-projektista löytyy osoitteesta <http://kotka.it.jyu.fi/kolibri/>.

3.4 Projektin ja sovelluksen tavoitteet

KURKI-järjestelmän käyttöliittymästä on tullut runsaasti negatiivista palautetta. Tästä johtuen myös KOPPELO-sovelluksen käyttöliittymän suunnittelussa tulee ottaa huomioon eritasoiset käyttäjät. Sovelluksen käyttäminen ei siis saa olla liian monimutkaista henkilöille, joilla ei ole kokemusta tietokoneiden käytöstä. Uuden sovelluksen tulisi olla kaikin puolin helppokäyttöinen, selkeä ja nopea. Lisäksi sovelluksen suunnittelussa tulee ottaa huomioon kokeneempien käyttäjien käyttötavat.

Projekti toteuttaa helppokäyttöinen opinnäytteiden hallintasovelluksen, joka liitetään osaksi KOTKA-järjestelmää. Sovelluksessa huomidaan käyttäjien erilaiset käyttöoikeudet. Opiskelija pääsee katsomaan vapaana olevia opinnäytteiden aiheita ja tarkastelemaan tarkempia tietoja omista opinnäytteistään.

Opinnäyteprosessiin liittyy useita vaiheita alkaen aiheen ehdotuksesta opinnäytteen hyväksymiseen. Ensiksi aihe-ehdotuksen hyväksyy opinnäytteestä vastaava henkilö, jonka jälkeen seuraa suunnittelu-, toteutus- ja raportointivaiheet. Prosessin kulussa opinnäyteellä on mm. seuraavia vaiheita: vapaa (=hyväksytty aiheeksi), tekeillä, hyväksytty, hylätty, keskeytetty, poistettu. Opinnäytetyön lajista riippuen työvaiheita voi olla enemmän tai vähemmän edellä esitetystä.

Jokaisen laitoksen henkilökunnalla on oikeus ehdottaa aihetta ko. laitoksen opinnäyteeksi. Rajatulla käyttäjäjoukolla on lisäksi oikeus hyväksyä ehdotettuja aiheita to-

teutettaviksi opinnäytteiksi. Opinnäytteen ohjaajalla on oikeus kiinnittää työ opiskelijoille ja kirjata muun muassa ohjaustapahtumiin liittyviä tietoja. Samalla sovellus toimii eräänlaisena opinnäytteeseen liittyvänä päiväkirjana sekä opiskelijalle että ohjaajalle.

Kaikilla sovelluksen käyttäjillä on oikeus hakea tietoa valmistuneista opinnäytteistä ilman tarvetta kirjautua sisään KOTKA-järjestelmään. Opinnäytteistä näytetään nimi, yleinen kuvaus, tekijä- ja ohjaajatiedot sekä hyväksymispäivämäärä.

Sovelluksen ei ole tarkoitus kilpailla kirjaston ”opinnäytepankin” kanssa, sillä sovellus ei ole tarkoitettu tutkielmien ja julkaisujen hallintaan ja levittämiseen. Sovellus kuitenkin osaa tarvittaessa ohjata käyttäjän joidenkin valmiiden opinnäytteiden verkkosivulle, jos kyseinen osoite on kirjattu järjestelmään.

Lisäksi järjestelmän tietokannan hajauttamista voisi ottaa huomioon jatkokehitystä suunniteltaessa. Sovelluksen toiminnallisuuden tarkempi määrittely on kuvattu sovellussuunnitelmassa.

3.5 Oppimistavoitteet

Projektin tavoitteena on antaa projektilaisille kuva luentokurssien harjoitustöitä laajemman ohjelmistoprojektin läpiviennistä ja yleensäkin perehdyttää projektimuotoiseen ryhmätyöskentelyyn. Samalla projektiryhmän jäsenet oppivat uuden ohjelmointikielen ja -tekniikoita.

Työnjako pyritään toteuttamaan projektiryhmän jäsenten osaamisen vahvuusalueiden ja kiinnostusten mukaan. Kuitenkin ryhmän jäsenet joutuvat myös paikkaamaan heikkouksiaan.

Projektiryhmällä on tavoitteena saada kokonaiskuva WWW-pohjaisen sovelluksen tuottamisesta suunnittelusta toteutukseen saakka. Toimivan, helppokäyttöisen ja hyvin kommentoidun ohjelman tuottamisen lisäksi ryhmä perehtyy yleiskäyttöisen ohjelman toteuttamisen esimerkiksi huomioimalla Java-pavut jo suunnitteluvaiheessa.

Koska dokumentit on päätetty kirjoittaa L^AT_EXilla, joutuvat ryhmän jäsenet opettelemaan uuden tavan dokumenttien laatimiseen. Osa ryhmästä joutuu siirtymään totutusta ulkoasukeskeisestä ajattelutavasta dokumentin rakennetta korostavaan lähestymistapaan.

4 Organisaatio ja resurssit

Luvussa esitellään projektiin osallistuvien henkilöiden tiedot ja projektiryhmän käytössä olevat resurssit.

4.1 Henkilöt ja yhteystiedot

KOPPELO-projektiryhmään kuuluvat Minna Hillebrand (mmhilleb@st.jyu.fi), Markus Silván (mape@st.jyu.fi), Antti Vanhanen (vanhanan@st.jyu.fi) ja Marko Ylitalo (ylimark@st.jyu.fi).

Tilaaajana toimivan Jyväskylän yliopiston tietotekniikan laitoksen edustajana ja projektin vastaavana ohjaajana toimii Jukka-Pekka Santanen (santanen@mit.jyu.fi). Teknisinä ohjaajina toimivat Pauli Kujala (pjkujala@it.jyu.fi) ja Heikki Uuksulainen (heuuksul@st.jyu.fi).

Projektille on perustettu sähköpostilista osoitteella koppelo@mit.jyu.fi. Sähköpostilistaan kuuluvat kaikkien edellä mainittujen henkilöiden lisäksi ainakin Kari Kärkkäinen (ktkar@mit.jyu.fi), Tommi Lahtonen (tjlahton@mit.jyu.fi), Vesa Lappalainen (vesal@mit.jyu.fi), Timo Männikkö (mannikko@mit.jyu.fi), Tapani Ristaniemi (riesta@mit.jyu.fi) ja Harri Tuomi (hamitu@it.jyu.fi).

Jyväskylän yliopiston tietotekniikan laitos on antanut projektin työtilaksi huoneen AgC222.2. Työtilan puhelinnumero on 014-2604963.

4.2 Laitteet ja ohjelmistot

Jyväskylän yliopisto on antanut ryhmälle käyttöön neljä tietokonetta. Kolmessa tietokoneessa on käyttöjärjestelmänä Red Hat Linux ja yhdessä Windows 2000. HTTP-palvelimena käytetään Apache-palvelinohjelmistoa, JSP- ja servlettimoottorina TomCat-ohjelmistoa sekä tiedot tallennetaan PostgreSQL-tietokantaan. Poolman käsittelee tietokantayhteydet TomCat-servlettimoottorin ja PostgreSQL-tietokannan välillä toimien samalla välimuistina.

Dokumenttien kirjoittamiseen käytetään \LaTeX -kuvauskieltä. Tarvittavat kaaviot piirretään Dia-nimisellä kaavioiden piirto-ohjelmalla. Versionhallintaan käytetään CVS-ohjelmistoa. Käytetyistä ohjelmistoista kerrotaan tarkemmin projektin sovellussuunnitelmassa.

5 Riskit ja niiden hallinta

Luvussa kuvataan mahdollisia riskejä, jotka voivat estää projektin onnistumisen tai etenemisen aikataulun mukaisesti.

5.1 Projektin jäsenten poissaolot

Projektin suurin riski on sovitussa aikataulussa pysyminen. Projektiin kuuluvien henkilöiden henkilökohtaiset aikataulut on pyritty luomaan joustaviksi, jotta projekti saadaan toteutettua aikataulussa. Loma- tai työmatkoja ei ole ennalta tiedossa projektiryhmän jäsenten osalta, mutta niitä saattaa ilmaantua. Myös sairastapaukset saattavat viivästyttää projektin etenemistä.

Matkojen tai sairastapausten ilmaantuessa muut projektiryhmän jäsenet jakavat työmäärän. Samoin menetellään, jos työmäärä on jostain syystä jakautunut epätasaisesti projektiryhmän jäsenten kesken.

Projektin aikana osa projektiryhmästä on töissä myös projektin ulkopuolella. Minna Hillebrand, Markus Silván ja Antti Vanhanen työskentelevät enintään 20 tuntia viikossa, joten sen ei pitäisi haitata projektin edistymistä. Marko Ylitalo aloittanee kesätyöt toukokuun alusta. Tämänkään ei oleteta tuovan ongelmia, sillä mikäli projekti valmistuu aikataulussa, toukokuun työmäärät ovat vähäisiä. Kuitenkin työkiireiden ja projektin asioiden kasautuessa voi ongelmia ilmetä.

5.2 Projektin jäsenten motivaatio

Ryhmän jäsenten motivaatio saattaa myös olla ongelma. Muun muassa loputon dokumentointi saattaa laskea ryhmän motivaatiota. Projektipäällikkö pitää ohjat käsissään ja huolehtii projektin etenemisestä aikataulun mukaisesti.

5.3 Suunnittelu- ja ohjelmointiongelmat

Kaikissa ohjelmistoprojekteissa on mahdollista, ettei luotava ohjelmisto täytä sille asetettuja vaatimuksia. Myös järjestelmän laajennettavuus voi loppua kesken. Projektiryhmällä on jonkin verran kokemusta ohjelmistojen suunnittelusta, mutta ei vastaavan laajuisen järjestelmän toteuttamisesta. Mahdollisia suunnittelun ja vaatimusmäärittelyn puutteita pyritään paikkaamaan prototyyppien tekemisellä jo ennen varsinaisen järjestelmän toteuttamista.

Ohjelmointipuolen ongelmia pyritään välttämään rationaalisella ja yhdenmukaisella ohjelmointitavalla. Ohjelmakoodi kirjoitetaan selkeiksi komponenteiksi, jotka jokainen

voidaan testata ja korjata erikseen. Myös lähdekoodin luettavuuteen kiinnitetään huomiota. Käytetyt ohjelmointikielet ja -tekniikat ovat projektiryhmälle ennalta tuntemattomia. Mahdollisissa ohjelmointiongelmassa käännytään teknisten ohjaajien puoleen.

5.4 Laitteisto- ja ohjelmisto-ongelmat

Laitteiston rikkoutuminen tai muut puutteet voivat haitata projektin edistymistä. Laitteistorikkoihin varaudutaan kopiomalla projektin tuottama materiaali europa-levypalvelimelle, jonka tiedot varmuuskopioidaan päivittäin.

Projektin tuotosten tallentamisessa siirrytään CVS-versionhallintapalvelimen käyttöön heti, kun tämä on mahdollista. Varsinaisen CVS-palvelimen pystyttämiseen saakka KOPPELO-projekti käyttää Markus Silvánin kotikoneessa sijaitsevaa CVS-palvelintä.

Käytetyt ohjelmistot ovat osalle projektiryhmästä ennestään tuntemattomia, joten ne saattavat aiheuttaa ongelmia. Ohjelmistojen käyttöön liittyvissä ongelmassa käännytään teknisten ohjaajien puoleen. Varsinaisissa ohjelmisto-ongelmassa auttaa tiedekunnan ATK-ylläpito.

5.5 Odottamattomat ongelmat

On mahdollista, ettei projektiryhmä ole osannut varautua kaikkiin mahdollisiin riskeihin. Odottamattomien ongelmien ilmaantuessa toimitaan tilanteen edellyttämällä tavalla.

6 Dokumentointi

Tässä luvussa kuvataan kirjoitettavat dokumentit, dokumentointiin käytettävät työkalut sekä dokumenttien ja lähdekoodin ulkoasuun liittyviä seikkoja.

6.1 Dokumentointikäytäntö

Projektin puitteissa kirjoitetaan lukematon määrä erilaisia dokumentteja. Jokaiseen palaveriin laaditaan esityslista ja palaverissa käsitellyistä asioista ja päätöksistä kirjoitetaan palaveripöytäkirja. Toteutettavan sovelluksen todennäköisten käyttäjien haastatteluista kirjoitetaan haastatteluraportit. Näiden lisäksi kirjoitetaan projekti-, sovellus- ja testaussuunnitelma sekä projektin päätteeksi projekti- ja sovellusraportti. Kaikki dokumentit tullaan keräämään projektikansioon.

Projektin dokumenttien pohjana käytetään tietotekniikan laitoksen pro gradu -tutkielmien L^AT_EX-pohjaa. Kokouspöytäkirjat ja esityslistat kirjoitetaan käyttäen samaa pohjaa, mutta levitetään sähköpostitse pelkistettynä ASCII-tekstinä. Kaikissa dokumenteissa pyritään säilyttämään yhtenäinen ulkoasu.

6.2 Lähdekoodin kirjoittaminen

KOPPELO-projektin tuottaman lähdekoodin kommentoinnissa tullaan käyttämään KOTKA-, KORPPI- ja KOLIBRI-projektien kanssa yhtenäistä kommentointityyliä. Jokaisen tiedoston alussa on tiedoston sisältöä kuvaava kommenttilohko. Lähdekoodin muotoilussa noudatetaan yhtenäistä käytäntöä. Ennen jokaisen funktion alkua kuvataan ko. funktion tehtävä ja toimintatapa. Lisäksi hankalasti ymmärrettäviin kohtiin lisätään selventäviä kommentteja.

Jatkokehityksen helpottamiseksi lähdekoodi ja kommentit kirjoitetaan englanniksi, vaikkakin dokumentit kirjoitetaan suomeksi. Muuttujan tyyppi kuvataan nimen alussa, esim. `string strFirstName` ja `class Person`.

Kommentoinnista kerrotaan enemmän KOPPELO-projektin sovellussuunnitelmassa.

7 Työnjako ja aikataulu

Projektin aloittaminen, aiheeseen tutustuminen ja sovelluksen suunnittelu on ollut ryhmätyötä. Uusiin työkaluihin pyritään tutustumaan aluksi ryhmässä ja myöhemmin itseksensä.

Kaikki projektiryhmän jäsenet osallistuvat Harri Tuomen ja Heikki Uuksulaisen pitämiin perehdyttämisluentoihin koskien uusia työkaluja. KOTKA-tietokantaan perehdyttämisen hoitaa Markku Vire sekä CSS-tyytilomakkeiden käytön Tommi Lahtonen. Projektin palaverien sihteeriksi ja puheenjohtajaksi valitaan kukin ryhmän jäsenistä kerrallaan.

Tehtävät jaetaan koko projektin ajan ryhmän kesken tilanteen mukaan päivä- ja viikkokohtaisesti, mutta ryhmän jäsenille sovitaan kuitenkin omat vastuualueensa. Poikkeuksellisesti projektinpäällikkönä toimii jokainen projektiryhmän jäsen vuorollaan. Tarkempaa tietoa projektipäällikkyydestä löytyy kuvassa 1 esitetystä aikataulusta.

7.1 Prototyypit

Marko Ylitalo toteuttaa Delphi-ohjelmointiympäristöllä demon sovelluksen käyttöliittymästä. Demon avulla pyritään helpottamaan käyttöliittymän suunnittelua ja siten välttämään pahoja epäloogisuuksia. Tämän demon pohjalta Antti Vanhanen toteuttaa alkeellisen WWW-toteutuksen. Markus Silván keskittyy alkuvaiheen dokumentointiin ja JSP-tekniikan tutkimiseen. JSP-tekniikan opettelun yhteydessä Markus aloittaa prototyypin toteuttamisen. Tarkempaa tietoa prototyyppien aikataulusta löytyy kuvan 2 toteutusaikataulusta.

Minna Hillebrand toteuttaa tuntikirjanpitosovelluksen PHP-kielillä projektien käyttöön. Myöhemmin kyseinen sovellus siirretään opettelumielessä JSP-sivuiksi. Dokumentointi pyritään hoitamaan tasavertaisesti kaikkien projektiryhmän jäsenten kesken. Tarkemmin työnjakoa käsitellään sovellussuunnitelmassa.

7.2 KOPPELO-sovellus

Ryhmä tutkii tietokannalta vaadittavia ominaisuuksia ja laatii niistä ER-kaavion. Sen pohjalta päätetään projektin kokouksissa, miten tietokanta tai sen muutokset toteutetaan. KOTKAN tietokannasta löytyy opinnäytteitä koskevia tauluja valmiiksi suunniteltuna, joita ryhmä tarkastelee oman suunnitelmansa valossa. Ryhmä esittelee mielestään parhaimman vaihtoehdon.

Tehtävänjako suoritetaan kussakin tilanteessa jäsenten kiinnostusten, vahvuusalueiden ja henkilökohtaisten aikataulujen perusteella. Suunnittelussa kaikki ryhmän jäsenet

ovat tasapuolisesti mukana tuoden omia mielipiteitään julki. Toteutusvaiheen alkaessa ohjelmointitehtävät jaetaan pienempiin kokonaisuuksiin, joita voidaan rakentaa itsenäisesti. Osion valmistuttua se testataan. Kun toimivuus on todettu, toteutetut osiot voidaan yhdistää kokonaisuudeksi. Tämän jälkeen kokonaisuus testataan. Tällä tavoin koko sovellus vähitellen muotoutuu.

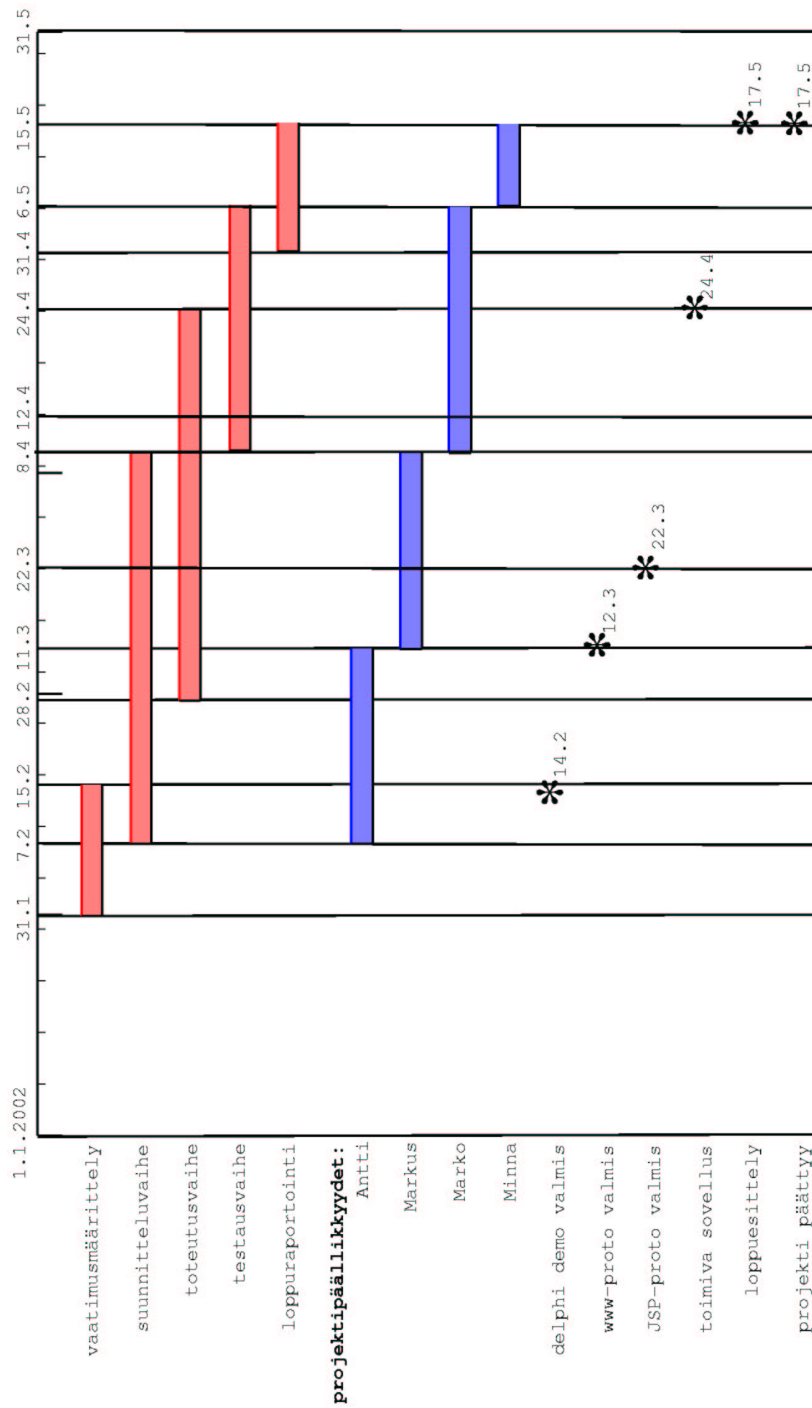
Toteutusvaihe on jaettu lyhyempiin jaksoihin (kuva 2). Vaatimusmäärittelyn valmistettua suunnitellaan käyttöliittymää ja tietokantaa. Sovelluksen toiminnallisuus rakentuu tietokannan valmistuttua huhtikuun aikana. Koko ryhmä osallistuu toiminnallisuuden rakentamiseen.

Kun sovellus on saatu toteutettua riittävälle toimivuuden tasolle, voidaan aloittaa testausvaihe. Tähän osallistuu koko ryhmä. Tehtäviä testauksessa voisi olla esimerkiksi käyttöliittymän toimivuuden tarkastelu, tietokannan toimivuuden tarkastelu ja kokonaisuuden testaus. Sovellusta testautetaan myös muiden projektien jäsenillä.

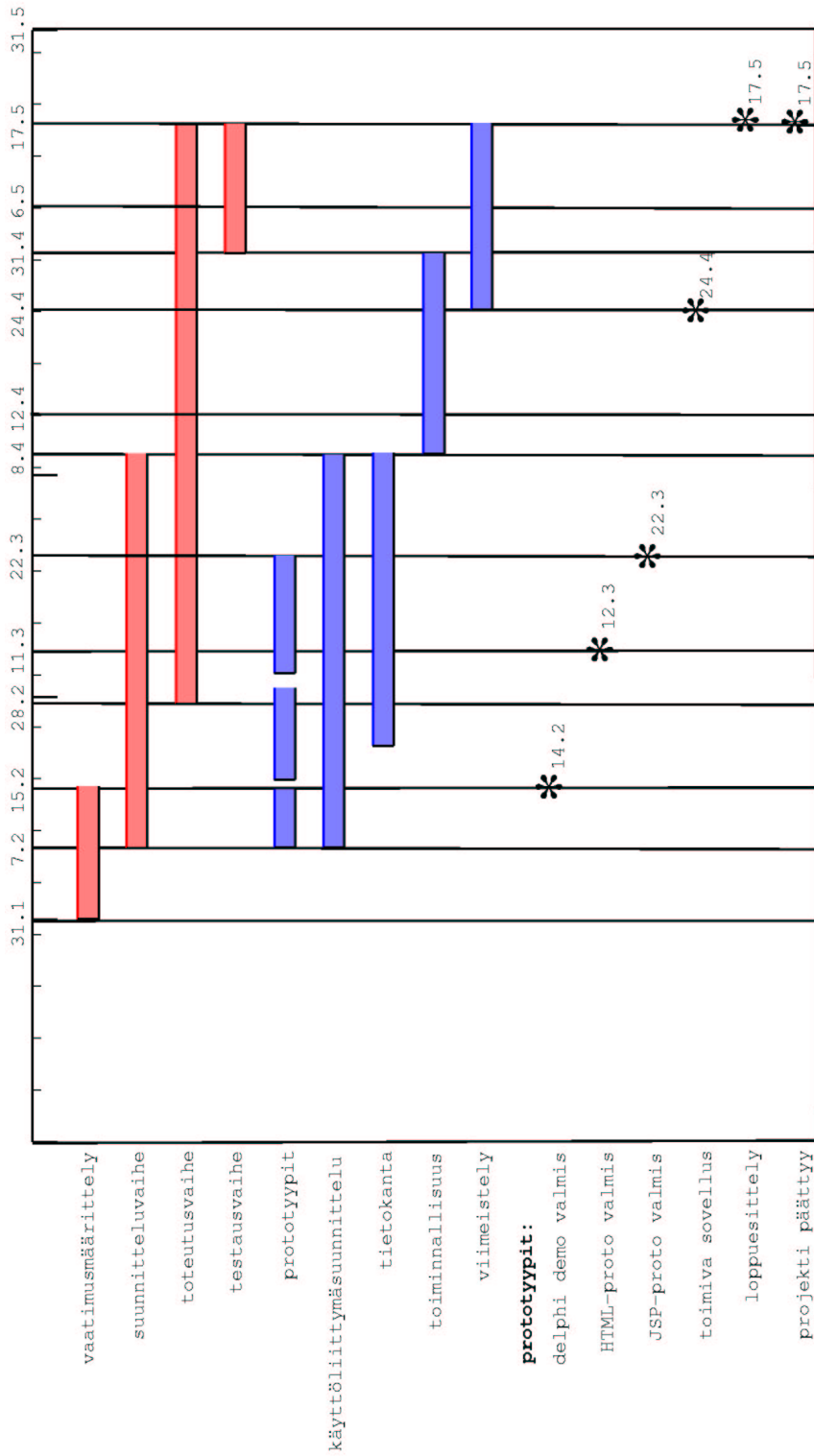
7.3 Raportointi

Raportointia tapahtuu palaverien muodossa koko projektin ajan. Loppuraportointi aloitetaan testausvaiheen päätyttyä, kuitenkin viimeistään toukokuun alussa. Loppuraportointi toteutetaan yhdessä. Lisäksi jokainen ryhmän jäsen tekee itsearviointin projektista.

Raportointi on valmiina kevään 2002 työprojektien loppuesittelyyn mennessä, joka on toukokuun puolessa välissä. Tilaajan mahdollisesti havaitsemia puutteita voidaan vielä täydentää tämän jälkeen, mutta tavoitteena on saada kaikki dokumentaatio valmiiksi loppuesittelyyn mennessä.



Kuva 1: KOPPELO-projektin suunniteltu aikataulu.



Kuva 2: Toteutuksen tarkempi aikataulu.

8 Yhteenveto

Tässä dokumentissa on kuvattu KOPPELO-nimisen tietotekniikan cum laude -työprojektin taustaa, tavoitteita, resursseja ja aikataulua. Projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle opinnäytteiden hallinnointisovelluksen olemassaolevan KOTKA-järjestelmän osaksi. Sovelluksen on tarkoitus vähentää opinnäytetöiden ohjaajien ja tekijöiden työmäärää helpottamalla ehdotettujen aiheiden sekä tekeillä olevien ja hyväksytyjen opinnäytteiden hallinnointia ja raportointia. Lisäksi sovelluksen avulla saadaan kätevästi tietoa sekä tekeillä olevista että valmistuneista opinnäytetöistä.

Projektin tavoite on saada huhtikuun puoleen väliin mennessä toimiva prototyyppi ohjelmasta, huhtikuun loppuun mennessä ensimmäinen toimiva versio ja toukokuun puoleen väliin mennessä projekti olisi kokonaisuudessaan valmis. Projektin edetessä ryhmän jäsenet oppivat projektityöskentelyä, ryhmätyöskentelyä, ohjelmiston suunnittelua, ohjelmointia ja dokumentointia. Työnjako toteutetaan siten, että jokainen selviää omasta osuudestaan.