

KIURU

Tietotekniikan sovellusprojekti

Toni Hilpinen
Marko Koivuniemi
Jussi Mäkinen
Miika Nurminen

Projektiraportti
21.1.2003

Jyväskylän yliopisto
tietotekniikan laitos

Kiuru-projektin tietoja

Tekijät:

Toni Hilpinen (hilpinen@cc.jyu.fi)
Marko Koivuniemi (koimark@cc.jyu.fi)
Jussi Mäkinen (jusmaki@cc.jyu.fi)
Miika Nurminen (minurmin@cc.jyu.fi)

Yhteystiedot: Työtila AgC222.2, sähköpostilistojen osoitteet kiuru@korppi.jyu.fi ja kiurulaaja@korppi.jyu.fi.

Työn nimi: Kiuru-projekti, projektiraportti

Työ: Projektiraportti tietotekniikan sovellusprojektiin

Tiivistelmä: Kiuru-projekti toteutti Jyväskylän yliopiston tietotekniikan laitoksella toteutettuun Korppi-järjestelmään WWW-pohjaisen tilanvaraus- ja varausvahvistusosion. Projektiraportissa kuvataan projektin taustoja ja projektille asetettuja tavoitteita, projektiorganisaatiota sekä suunnitelmien toteutumista ja projektin onnistumista.

Avainsanat: Salivaraus, kurssien opetustapahtumat, resurssivaraus, WWW-sovellus, Korppi-järjestelmä, Timmi-järjestelmä

Versionhallinta

Versio	Päiväys	Tehnyt	Muutokset
0.1	2.1.2003	Jussi Mäkinen	Ensimmäinen versio
0.2	13.1.2003	Jussi Mäkinen	Pieniä muutoksia
0.3	14.1.2003	Jussi Mäkinen	Muokattu lukua 3
0.4	15.1.2003	Miika Nurminen	Lisätty omia kommentteja
0.5	16.1.2003	Toni Hilpinen	Lisätty omia kommentteja
0.6	20.1.2003	Marko Koivuniemi	Virheiden korjailua ja lisää tekstiä
0.7	21.1.2003	Jussi Mäkinen	Pieniä muutoksia

Sisältö

1 Johdanto	1
2 Termejä	2
2.1 Aihepiirin yleisiä termejä	2
2.2 Teknisiä termejä	3
3 Projektin taustaa ja tavoitteita	4
3.1 Salivarausohjelmistojen taustaa	4
3.2 Korppi-järjestelmän taustaa	4
3.3 Projektille asetettujen tavoitteiden toteutuminen	5
3.4 Oppimistavoitteet ja niiden toteutuminen	5
3.5 Aihepiiriin liittyviä tulevaisuuden ideoita	6
4 Projektin resurssit ja organisaatio	7
5 Projektiryhmän tehtävät ja niiden jakautuminen	8
6 Projektin aikataulu	8
7 Projektin riskit ja ongelmat	9
8 Henkilökohtaisia arvioita projektista	10
8.1 Toni Hilpinen	10
8.2 Marko Koivuniemi	11
8.3 Jussi Mäkinen	12
8.4 Miika Nurminen	13
9 Yhteenveto	13
Lähteet	14
Liite Projektin toteutunut aikataulu	15

1 Johdanto

Kiuru-niminen tietotekniikan sovellusprojekti oli jatkoa Kotka-, Korppi-, Kolibri- ja Koppelo-projekteille. Kiuru toteutti Korppi-järjestelmään salinvarausmoduulin ja rajapintayhteyden Yomi Oyj:n Timmi-järjestelmään. Kiuru-ryhmän koodi on julkaistu GPL-lisenssin alaisuudessa.

Syksyn 2002 Kiuru-projektiryhmään kuuluivat tietotekniikan opiskelijat Toni Hilpinen, Marko Koivuniemi, Jussi Mäkinen ja Miika Nurminen. Tilaajana toimi Jyväskylän yliopistolta hallintovirasto ja tietotekniikan laitos. Muut projektiorganisaatioon kuuluneet henkilöt esitellään luvussa 4.

Projektiraportissa kuvataan projektin läpivientiin liittyneitä suunnitelmia ja niiden toteutumista. Raportissa tarkastellaan projektin aikataulua, tavoitteiden toteutumista, resursseja, vastuualueita ja niiden jakautumista sekä arvioitujen riskien ja toteutuneiden ongelmien ratkaisemista.

Projektiryhmä laatii myös Sovellusraportin, jossa käsitellään tarkemmin sovelluksen rakennetta ja ominaisuuksia. Projektiryhmä muokkasi myös aiemmin laadittua Kotka-tietokantaraporttia. Projektiryhmä on laatinut jo aiemmin Projektisuunnitelman, Sovellussuunnitelman, Vaatimusmäärittelyn ja Haastatteluraportin.

Luvussa 2 esitellään aiheeseen oleellisesti liittyviä termejä. Luvussa 3 käsitellään projektin taustoja sekä tavoitteita ja niiden toteutumista. Luku 4 esittelee projektin organisaation ja resurssit. Luvussa 5 käsitellään projektin vastuualueita sekä niiden jakautumista projektiryhmän jäsenten kesken. Luku 6 kuvaa projektin toteutuneen aikataulun. Luvussa 7 arvioidaan projektiin liittyneitä riskejä, ongelmia ja niiden ratkaisuja. Luvussa 8 projektiryhmän jäsenet kertovat omista kokemuksistaan liittyen projektiin.

2 Termejä

Luvussa kuvataan termejä, jotka auttavat ymmärtämään dokumentteja paremmin. Tekniset termit on pääosin listattu sellaisia henkilöitä varten, joilla ei ole ohjelmointitaustaa.

2.1 Aihepiirin yleisiä termejä

Seuraavat yleiset aihepiirin termit liittyvät olennaisesti Jyväskylän yliopiston salinvarausprosessiin:

Korppi	on Jyväskylän yliopistossa joidenkin laitosten käyttämä opetus- ja tutkimushallintajärjestelmä.
Kotka	on Korppi-järjestelmän tietokanta.
Maksaja	on henkilö tai organisaatio, joka maksaa varauksen.
Oheisvarattava	on liikuteltava esine, jonka vahtimestarit toimittavat tilaisuuden järjestäjän toiveiden mukaisesti haluttuun saliin haluttuna ajankohtana.
Resurssi	on johonkin saliin liittyvä ominaisuus tai kiinteästi saliin asennettu laite tai esine. Esimerkiksi salin paikkamäärä tai piirtoheittimen olemassaolo voivat olla resursseja.
Resurssipyyntö	vastaa varauspyyntöä, kuitenkin sillä erotuksella, että salin sijaan ilmoitetaan resurssien tarve.
Sali	on varattavissa oleva paikka, johon ihmiset voivat kokoontua (esimerkiksi luentosali tai kokoustila).
Sali-järjestelmä	on Jyväskylän yliopistossa aiemmin käytössä ollut tekstipohjainen salinvarausjärjestelmä.
Timmi	on Yomi Oyj:n kehittämä salinvaraussovellus.
Vakiovaraus	on sarja varauksia, jotka liittyvät toisiinsa ja toistuvat esimerkiksi viikoittain samana viikonpäivänä.
Varauksen pyytjä	on henkilö, joka ehdottaa varattavaksi jotakin salia jonakin ajankohtana.
Varauksen vahvistaja	on henkilö, jolla on oikeus vahvistaa varaus tiettyyn saliin.
Varaus	on käyttöoikeus tiettyyn saliin tiettyinä ajankohtana.
Varauspyyntö	on varauksen pyytäjän tekemä ehdotus varaukseksi koskien tiettyä salia tiettyinä ajankohtana.

2.2 Teknisiä termejä

Seuraavat ohjelmointitekniset termit liittyvät sovelluksen taustalla olevaan järjestelmään:

AJP12	on tiedonsiirtoon Apache- ja Tomcatpalvelimien välillä käytetty protokolla.
Apache	on ilmainen HTTP-palvelinohjelmisto.
CSS	eli Cascading Style Sheets on WWW-sivujen ulkoasua kuvaava kieli.
HTML	on WWW-sivujen sisältöä kuvaava kieli.
HTTP	on WWW-arkkitehtuurin käyttämä tiedonsiirtoprotokolla.
Jakarta-Tomcat	on ilmainen servletti- ja JSP-moottori.
Java	on Sunin kehittämä laitteistoriippumaton olio-ohjelmointikieli.
JavaDoc	on Java-kehitysympäristön mukana tuleva dokumentointityökalu.
Java-pavut	(engl. <i>JavaBeans</i>) ovat Java-ohjelmointikielellä luotuja komponentteja, joita voidaan kutsua JSP-sivuilla.
JDBC	eli Java Database Connectivity on Java-teknologian käyttämä rajapinta erilaisiin tietokantoihin.
JSP	eli Java Server Pages on skriptaustyylinen ohjelmointikieli.
Keksi	eli eväste(engl. <i>cookie</i>) on menetelmä saada tallennettua tietoja WWW-sivuilta käyttäjien koneille. Palvelin lähettää pieniä tietopaketteja selaimelle, jotka tallennetaan käyttäjän koneelle. Myöhemmin selain lähettää tiedot palvelimelle ja palvelin voi käyttää niitä esimerkiksi istuntojen tallentamiseen.
Käyttötapaus	(engl. <i>use case</i>) on käyttäjän tai sovelluksen toimintoa tietyn tehtävän suorittamiseksi kuvaava dokumentti.
Moduuli	on tarkkaan rajattu toiminnallinen ohjelmakokonaisuus.
Poolman	on ohjelma, joka kontrolloi tietokantayhteyksiä.
PostgreSQL	on ilmainen tietokannanhallintajärjestelmä.
Skripti	on yksinkertainen ohjelmanpätkä, joka useimmiten on tehty tulkattavalla kielellä.
Selain	on ohjelma, joka käyttäjän koneella tulkaa HTML-kieliset sivut esitettävään muotoon.
Servletti	(engl. <i>servlet</i>) on palvelimella sijaitseva sovellus, joka toteuttaa HTTP-palvelimen pyynnöstä tietyn toiminnon.
SQL	eli Structured Query Language on relaatiotietokantojen hallintaan kehitetty standardi kieli.

3 Projektin taustaa ja tavoitteita

Luvussa esitellään projektin taustoja sekä projektille asetettuja tavoitteita ja niiden toteutumista.

3.1 Salivarausohjelmistojen taustaa

Jyväskylän yliopistossa oli aiemmin käytössä keskitetty Sali-niminen tekstipohjainen varaussovellus. Järjestelmällä hoidettiin periaatteessa kaikkien yliopiston salien varaus.

Yomi Oyj:n Timmi-niminen salivarausjärjestelmä piti ottaa käyttöön 23.10.2002, mutta käyttöönotto kuitenkin viivästyi joulukuulle 2002. Timmi-järjestelmän merkittävin etu aikaisempaan Sali-järjestelmään verrattuna on automaattinen salien laskutuksesta huolehtiminen, mitä vanha järjestelmä ei tukenut. Timmi on Windows-sovellus, johon on saatavilla myös WWW-käyttöliittymä. Tosin WebTimmiä ei ole tarkoitus hyödyntää Jyväskylän yliopistossa.

Jyväskylän yliopiston hallintovirastossa on tarkoitus hankkia myöhemmin Timmin rinnalle myös yleiskäyttöisempi tilahallintasovellus, jolla hallittaisiin kaikkien yliopiston tilojen (rakennukset, osastot ja huoneet) vuokrausta ja laskutusta. Tulevalla järjestelmällä ei kuitenkaan ole todettu olevan suoraa vaikutusta Timmi- tai Korppi-järjestelmiin.

3.2 Korppi-järjestelmän taustaa

Tietotekniikan laitoksella oli käytössä Kurki-järjestelmä vuosina 1998-2002. Järjestelmän avulla opiskelijat pystyivät ilmoittautumaan kurssille ja selaamaan tietojaan eri kursseilta. Opettajat pystyivät hallinnoimaan kursseja, pääteohjauksia ja demotilaisuuksia.

Syksyllä 2000 Kotka-projekti kehitti tietokantarakenteen ja henkilötietohallintamoduulin uuden opetus- ja tutkimushallintajärjestelmän pohjaksi. Keväällä 2001 Korppi-projekti kehitti järjestelmään kurssikirjanpito-osion, jolloin vanha Kurki-järjestelmä alkoi käydä tarpeettomaksi. Korppi-projektin kanssa yhteistyötä teki Halko-projekti, joka teki kansalaisopistoille opetuksenhallintajärjestelmän. Yhdessä nämä projektit tekivät Kotka-tietokantaan ryhmien toteuttamien sovelusten tarvitsemia muutoksia [6].

Syksyllä 2001 järjestelmän kehitystyötä jatkoi Kolibri-projekti, joka kehitti Korppi-järjestelmään kalenteriosion. Kalenteriosio tukee mm. henkilöryhmiä, joille voi määrittellä yhteisiä tapahtumia. Keväällä 2002 Koppelo-projekti toteutti Korppi-järjestelmään opinnäyttemoduulin.

3.3 Projektille asetettujen tavoitteiden toteutuminen

Projektiryhmä toteutti järjestelmän, jonka avulla kurssien opetustapahtumille voidaan varata salit Korppi-järjestelmällä. Varaukset on hoidetaan yksittäisten opetustapahtumien tarkkuudella. Opetustapahtumien lisäksi saleja ja tiloja voidaan varata muuhunkin käyttöön. Tällaisia tapahtumia ovat esimerkiksi kokoukset, konferenssit ja maksulliset tapahtumat.

Opetus- ja muita tapahtumia varattaneen jatkossa myös Timmi-järjestelmällä, mistä poikkeuksena ovat laitoksien omassa käytössään olevat salit. Tällaisia saleja ovat mm. kokoustilat ja laboratoriot, joita ei oteta Timmi-järjestelmän piiriin. Näiden salien varaus on mahdollista liittää Korppi-järjestelmään varsin helposti.

Korppi-järjestelmään lisättiin mahdollisuus varata yksittäisiä saleja tiettyä tapahtumaa varten. Tällöin opettaja voi pyytää salin varausta ja varauksista vastaava vahvistaa sen, jolloin tiedot siirtyvät automaattisesti Timmiin.

Projektiryhmä toteutti Korppi-järjestelmään Kiuru-osion lisäksi synkronoinnin Timmi-järjestelmän kanssa. Korppi-järjestelmän kautta tiedot varauksista välittyvät Timmiin ja Timmiin tehdyt salivaraukset välittyvät myös Kotkan tietokantaan.

Projektin puitteissa jäi toteuttamatta joitakin yksityiskohtia sovelluksesta ja varsinainen testausvaihe puuttui kokonaan. Kokonaan toteuttamatta jäivät mm. resurssivarauksien sekä oheisvarattavien käsittely.

Yleistä testausta ei projektin puitteissa ollut mahdollista toteuttaa tarvittavissa määrin, mutta jokainen komponentti kuitenkin testattiin nopeasti, jotta pahimmat viat saatiin korjattua. Lisäksi Vesa Lappalainen ja Jukka-Pekka Santanen testailivat käyttöliittymää ja toiminnallisuutta ansiokkaasti.

Sovelluksen ja Timmi-rajapinnan välistä toimivuutta ei ehditty kunnolla testaamaan, sillä rajapintaa projektiryhmä pääsi testaamaan vasta viikolla 3 vuonna 2003, jolloin projektin olisi pitänyt olla jo valmis. Tämän vuoksi synkronoinnin testaaminen jäi todella vähälle projektin aikana.

Kiuru-sovellus toimii tarvittaessa ilman Timmi-järjestelmää. Tällöin kaikki Timmi-rajapintaan liittyvät ohjelmakomponentit ovat tarpeettomia.

Sovellukselle asetettujen tavoitteiden toteutumista käsitellään tarkemmin Sovellusraportissa. Syitä siihen, miksi kaikkia määriteltyjä ja toivottuja ominaisuuksia ei pystytty toteuttamaan, pohditaan tarkemmin luvussa 7.

3.4 Oppimistavoitteet ja niiden toteutuminen

Projektin tavoitteena oli antaa projektilaisille kuva projektityöskentelystä käytännössä. Lisäksi tavoitteena oli opettaa projektiryhmäläisille uusia teknologioita ja niiden toteuttamiseen soveltuvien sovellusten käyttöä.

Projekttilaiset saivat varmasti hyvin totuudenmukaisen kuvan hieman laajemman sovelluksen toteuttamisesta. Määräpäivät tulivat nopeasti vastaan ja aina tuntui olevan hieman kiire. Lisäksi projekttilaiset saivat kokemusta alihankkijan kanssa toimimisesta. Tätä kokemusta eivät muut projektit ole aikaisemmin saaneet.

Projektiryhmä yritti ensin itse hakea ratkaisut ongelmiin. Jos ongelma ei ratkennut ryhmän sisällä, ryhmä tiedusteli teknisiltä ohjaajilta ratkaisua ongelmaan. Teknisten ohjaajien ja projektiorganisaatiosta löytyneiden asiantuntijoiden apu oli välillä tarpeen ja hyödyllistä, mutta suurimmaksi osaksi selvittiin omin voimin.

Projektiryhmän jäsenet tutustuivat myös uusiin teknologioihin ja sovelluksiin. Java Server Pages -teknologia ja SQL-kyselykieli tulivat tutuiksi kaikille projektiryhmäläisille. Uusia työkaluja projektilaisille olivat mm. versionhallintaohjelma CVS ja ohjelmointiympäristö NetBeans. Näistä työkaluista jäi ryhmälän jäsenille hyvä kuva ja he aikovat käyttää niitä myös jatkossa. Lisäksi ryhmän jäsenet oppivat \LaTeX in käytön, koska pääsääntöisesti kaikki ryhmän tekemät dokumentit ladottiin \LaTeX -ohjelmalla.

Projektiryhmä sai myös arvokasta kokemusta ryhmätyöskentelystä. Myös ryhmäläisten viestintätaidot kehittyivät. Projektissa oli monta osapuolta, joiden kaikkien piti olla ajan tasalla projektin etenemisestä. Ryhmän jäsenet oppivat myös työnjakoa ja organisointia.

3.5 Aihepiiriin liittyviä tulevaisuuden ideoita

Opettajan käyttöliittymään tulisi lisätä mahdollisuus resurssitarpeiden merkitsemiseen. Resurssitarpeiden merkitseminen hoidetaan tällä hetkellä siten, että kurssin opettaja toimittaa tarpeet henkilölle (esim. amanuenssi), joka suunnittelee kootusti laitoksen kurssien opetustapahtumien sijoittamisen käytössä oleviin saleihin. Resurssitarpeet sisältävät vaatimuksia salin paikkamäärään ja tarvittaviin resursseihin (esim. videoprojektori, liitutaulu). Resurssit voivat liittyä saliin kiinteästi tai ne voivat olla erikseen varattavia. Tietokannassa huomioidaan resurssitarpeiden suunnittelu ja muokkaus, mutta varsinaista suunnitteluohjelmaa projektin resurssien puitteissa ei ehditä toteuttamaan.

Kurssien sijoittumisen suunnittelusta vastaavat henkilöt tarvitsisivat ohjelmaa, jonka avulla voisi sijoittaa kurssien ajankohdat siten, että opiskelijoilla ja opettajilla olisi mahdollisimman vähän päällekkäisyyksiä. Ohjelman tulisi ainakin ilmoittaa, jos esimerkiksi saman vuositason tai aineen kurseja on menossa samaan aikaan. Ihanteellinen tilanne olisi, jos ohjelma osaisi sijoitella kurssit valmiiksi lukujärjestykseen eri osapuolia tyydyttävällä tavalla.

Projektin toteuttamia ominaisuuksia tulisi testata myös perusteellisemmin, jotta mahdollisista virheistä voisi päästä eroon. Lisäksi projektiryhmältä jäi toteuttamatta yksittäisiä ominaisuuksia, joita käsitellään tarkemmin Sovellusraportissa.

Alustavan sopimuksen mukaan projektiryhmä jatkaa Kiuru-sovelluksen kehitystyötä erikoistytöinä.

4 Projektin resurssit ja organisaatio

Projektiryhmän muodostivat Toni Hilpinen, Marko Koivuniemi, Jussi Mäkinen ja Miika Nurminen. Tilaajan edustajina toimivat Pauli Kujala ja Vesa Lappalainen tietotekniikan laitokselta sekä Esko Korhonen ja Pekka Lampinen yliopiston hallinnosta. Jani Asikainen, Taru Väisänen ja Jukka Valkonen olivat Timmi-järjestelmän kehittäneen Yomi Oyj:n edustajia. Jukka-Pekka Santanen oli projektin vastaava ohjaaja. Teknisiä ohjaajia olivat Kirsi Koponen (vastuualueenaan Java Server Pages) ja Markku Vire (vastuualueenaan tietokannat).

Projektiryhmä sai käyttöönsä tietotekniikan laitokselta Agorasta huoneen C222.2, jossa oli asennettuina kolme Linux-työasemaa ja yksi Windows 2000 -työasema. Käyttöjärjestelmäksi Linux-työasemille oli asennettuna Redhat 7.3 -levitysversio. Käyttöjärjestelmään kuului useita erilaisia sovelluspaketteja. Lisäksi sovelluspalvelimeksi asennettiin Jakarta Tomcat 3.2.3. Sovelluskehittimenä toimi NetBeans. Sovelluksen testaamiseen ryhmä käytti NetBeansin mukana tulevaa Jakarta Tomcat 4.0:aa. Windows-työasemaan oli asennettu mm. WinTimmi, Microsoft Visio 2000 ja MS Project.

Projektin dokumentointi on kokonaisuudessaan luettavissa projektin kotisivuilla <http://kotka.it.jyu.fi/kiuru>. Projektilla oli käytössään sähköpostilistat kiuru@korppi.jyu.fi ja kiurulaaja@korppi.jyu.fi. Sähköpostilistojen keskustelua voi lukea WWW-osoitteista <http://korppi.it.jyu.fi/list-archive/kiuru/> ja <http://korppi.it.jyu.fi/list-archive/kiurulaaja>.

5 Projektiryhmän tehtävät ja niiden jakautuminen

Ryhmä jakoi tehtäviä melko monipuolisesti siten, että kaikki tekivät (ainakin jossain vaiheessa projektia) kutakin eri tehtävätyyppiä eli suunnittelua, toteutusta ja dokumentointia. Tällä tavalla kukin projektiryhmän jäsen sai kokemusta kaikista ohjelmistoprosessin vaiheista ja tehtävistä.

Ryhmän projektipäällikkönä ja JSP-vastaavana toimi Jussi Mäkinen. Tietokantavastaavana toimi Miika Nurminen. LaTeX-dokumenttien pohjat ja XML-dokumenttien hallinnan toteutti Marko Koivuniemi. Marko Koivuniemi vastasi myös projektin kirjoittamien dokumenttien viimeistelystä. Toni Hilpinen vastasi pääosin Timmi-rajapinnan kanssa tarvittavista luokista.

Muilta osin (mm. käyttöliittymän suunnittelu) vastuualueet jaettiin kunkin projektin jäsenien tilanteen mukaan projektin edetessä. Dokumentointi hoidettiin koko ryhmän voimin, jotta kukaan ei olisi kyllästynyt dokumentointityöhön. Dokumentointia yritettiin toteuttaa koko projektin ajan tasaisesti, mutta projektin edetessä dokumentoinnin määrä kasvoi.

Päävastuualueita ei tarvinnut jakaa uudestaan projektin aikana, mutta kukin projektiryhmän jäsen osallistui myös muiden työtehtävien toteuttamiseen. Tehtävät selkiytyivät projektin edetessä hyvin työstettäviksi ja kaikille riitti töitä projektin loppuun saakka.

6 Projektin aikataulu

Projektin oli tarkoitus kestää 14 viikkoa ja kyseinen aikajänne oli jaettu kolmeen päävaiheeseen: analyysiin ja suunnitteluun, toteutukseen sekä testaukseen. Analyysi- ja suunnitteluvaiheiden oli suunniteltu kestävän 5 viikkoa, mutta todellisuudessa niihin kului aikaa 7 viikkoa. Toteutusvaiheen oli tarkoitus kestää 5 viikkoa mutta se kesti 6 viikkoa. Testaukselle ja loppuhiomiselle oli varattu 4 viikkoa, mutta testausta ei ehditty omana vaiheenaan toteuttamaan ollenkaan. Loppuhiomiseen kului 4 viikkoa, joista 3 viikkoa vuoden 2003 puolella.

Ensimmäisen vaiheen tuloksia olivat Projektisuunnitelma, Sovellussuunnitelma ja Vaatimusmäärittely. Toisen vaiheen tuloksia olivat valmis ohjelmakoodi (JavaBeanit ja JSP-sivut), Testaussuunnitelma sekä koodin dokumentaatio. Kolmannen vaiheen tuloksia olivat Projektiraportti ja Sovellusraportti. Vaiheet on esitelty taulukossa 6.1.

Timmi-rajapinnan määrittely venyi sovitusta aikataulusta huomattavasti, joka viivästytti myös rajapintakomponenttien kehitystä. Määrittelyvaiheen tuloksena saatu määrittelydokumentti saatiin lopulliseen muotoonsa 12.12.2002. Tämän seurauksena varsinainen rajapinta, jota testauksen ja kehityksen yhteydessä olisi tarvittu, ei valmistunut ajoissa eikä projektin yhteydessä voitu rajapinnan kanssa

kommunikoivia komponentteja testata lähes ollenkaan.

Vaihe	Tulos	Toteutunut	Suunniteltu
Suunnittelu 1	Projektisuunnitelma	20.11.2002	14.10.2002
Suunnittelu 2	Sovellussuunnitelma	7.11.2002	17.10.2002
Rajapinta	Timmi-rajapintamäärittely	12.12.2002	28.10.2002
Rajapinta	Timmi-rajapinta	10.1.2003	28.11.2002
Toteutus	Ohjelmakoodi ja dokumentaatio	10.1.2003	13.12.2002
Testaus	Sovellusraportti ja Projektiraportti	15.1.2003	20.12.2002

Taulukko 1: Projektin vaiheet ja tarkistuspisteet.

Projektin aikataulun viivästyminen usealla rintamalla johtuneen suurelta osin siitä, ettei projektin laajuutta kyetty ennakoimaan alussa. Vaatimusmäärittelyvaiheessa olisi pitänyt pystyä rajaamaan asioita tarkemmin ja aloittamaan kehitystyö nopeammin. Timmi-rajapintaan liittyvän kehitystyön hitauden lähtökohtana oli toimiminen yrityksen kanssa tavalla, joka oli uutta koko Jyväskylän yliopiston sovellusprojektien historiassa. Lisäksi rajapinnan määrittelyvaiheen venyminen huomattavasti alkuperäistä pidemmäksi aiheutti varsinaisen rajapinnan viivästyksen lisäksi ongelmia kehitysvaiheeseen.

Monimutkaisen tietokantarakenteen vuoksi myös tietokannan suunnittelu vei paljon aikaa. Suunnitteluvaiheessa tarvittiin teknisten ohjaajien apua ja korjauskierroksia ja lisäparannuksia tarvittiin useita.

Osasyynä projektin viivästyminen on varmasti myös sovellusprojektin jäsenten kokemattomuus vastaavista projekteista sekä osalla projektilaisista myös raajan ohjelmointitaidon puutteista. Alkukahakuksien jälkeen projektin loppuvaiheessa työ alkoi edistyä erittäin hyvin ja näkyvää jälkeä alkoi syntyä.

7 Projektin riskit ja ongelmat

Projektin suurin ongelma oli sovitussa aikataulussa pysyminen. Yksi syy tähän oli huonosti määritellyt tavoitteet projektin alussa. Toinen syy oli se, että kaksi projektin jäsentä työskenteli samaan aikaan muualla ja tämä hidasti projektin etenemistä jonkin verran. Loppuvaiheessa jokainen projektin jäsen käytti kaiken mahdollisen ajan projektin loppuunsaattamiseen.

Toinen suurehko ongelma oli Timmi-sovelluksen ja Kiuru-sovelluksen välinen viestintä ja yhteistoiminta. Timmi ei ollut projektin aikana vielä aktiivisessa käytössä salien varausprosessissa Jyväskylän yliopistolla, mikä aiheutti epävarmuutta projektiin. Rajapintamäärittelyn laatimiseen hukkui liian paljon aikaa ja Yomi sai rajapintakomponentin noin kuukauden myöhässä sovitusta aikataulusta.

Myös eriävät näkemykset ja mielikuvat toteutettavasta sovelluksesta toivat mukanaan omat haasteensa. Kaikkia esitettyjä toiveita ei ollut mahdollista toteuttaa projektin puitteissa, osin siitä syystä, että toiveet olivat toisensa poissulkevia. Riskiä pyrittiin pienentämään laatimalla sekä kattavat että selkeät pöytäkirjat ja dokumentit koko projektin ajan.

Projektin aikana esiintunut riski liittyen olemassaolevien ohjelmien lisensseihin ei muodostunut kuitenkaan ongelmaksi. Korppi-järjestelmä on toteutettu GPL-lisenssin alla ja ainoastaan Timmi-rajapintadokumentti sovittiin salaiseksi. Tästä projektiorganisaatio pääsi yhteisymmärrykseen.

Sovelluksen testiversio pyrittiin saamaan valmiiksi hyvissä ajoin, jotta testauksessa saatua palautetta olisi voitu hyödyntää sovelluksen loppuhiomisessa. Aikataulu kuitenkin petti ja sovelluksen ensimmäinen kunnollinen versio valmistui pari viikkoa myöhässä aikataulustaan. Myöskään sovelluksen testausta ei ehditty kunnolla toteuttaa.

8 Henkilökohtaisia arvioita projektista

Sovellusprojekti oli kaikille jäsenilleen monipuolinen ja paljon antava kokemus, joka kuitenkin oli myös haastava ja rankka. Projektin jäsenten yhteishenki ei kuitenkaan rakoillut ja ennemminkin projektissa ilmenneet ongelmat muodostuivat jäseniä yhdistäviksi tekijöiksi. Kaikki projektilaiset saivat uutta oppia niin dokumentoinnin kuin ohjelmoinninkin suhteen ja uusia mielenkiintoisia työkaluja opittiin käyttämään. Projektilaiset saivat nähdä "oikean työelämän" vaikeuksia, kuten aikataulun venymistä itsestä riippumattomista sekä itsestä riippuvistakin syistä. Projektityöskentelyn hyviä ja huonoja puolia tuli katseltua uudesta näkökulmasta ja tätä tietämystä varmasti projektin jäsenet tulevat tarvitsemaan jatkossakin.

8.1 Toni Hilpinen

Rauhallisen kesäloman jälkeen oli taas aika jatkaa opintoja. Olin kesällä ilmoitautunut Sovellusprojektiin, josta aivan uudenaikaisena kurssina minulla ei ollut paljoakaan tietoa. Projekti, jossa tehdään sovellus, päättelin. Odotuksia kurssin suhteen ei ollut, joten ei auttanut kun käydä hommaan käsiksi ja katsoa, mitä tuleman piti. Ensimmäisellä luennolla sain sätkyn, en niinkään projektiin liittyvistä asioista, vaan niiden paljoudesta. Tunsin olevani aivan hukassa, mutta ei auttanut. Ken elää, se oppii. Muutaman ensimmäisen viikon aikana alkoi asiat selvenemään huomattavasti ja projektiryhmä oppi tuntemaan toisiaan paremmin. Silloin alkoi jo tuntumaan, että kyllähän tämä tästä.

Omalta kohdaltani projektin eteneminen tuntui rytmiltään kovin pumppaavalta. Välillä tuntui, ettei mistään tule mitään, kun taas toisinaan tunnelma oli katossa

ja homma eteni mukavasti. Mutta kuten sanotaan ”Ei tuskaa, ei tulosta”. Tämä toteutui myös projektissa. Kaikki edellä mainittu kuitenkin vaikutti siihen, että en ole koskaan aikaisemmin oppinut kurssin puitteissa yhtä paljon asioita kuin projektin aikana. Tuli opittua Javaa, SQL:ää, JSP:tä ja varsinkin paljon dokumentointiin liittyvää asiaa. Myös projektiluonnoilta jäi toivottavasti jotain mieleen.

Oppiminen ei jäänyt kuitenkaan tekniikkaan ja tietokoneen käyttöön vaan myös sosiaalista kanssakäymistä oli runsain mitoin, kuten haastattelut, monenlaiset palaverit ja tietysti projektiryhmän kanssa työskentely. Näiden ajattelin jo etukäteen olevan ongelmakohtia, hiljainen kun olen. Pelkäsinkin joillekin koko tiimin jäsenistä jäävän minusta ehkä hieman saamattoman ihmisen kuvan.

Koodauksessa tehtäviini kuului lähinnä Timmi-luokkien tekeminen. Nämä myöhästyivät kuitenkin aikataulusta sekä Timmi-rajapinnan valmistumisen myöhästymisen että oman koodauksen aloittamisen myöhästymisen takia. Yleensä ottaen en ole omaan panokseeni täysin tyytyväinen, tosin en ole koskaan aikaisemmin ollut. No, ainakin on jotain tavoiteltavaa jatkossakin.

Kokonaisuudessaan projekti oli kuitenkin oikein mielenkiintoinen ja mukava kokemus. Vaikeuksista huolimatta voin sanoa jopa nauttineeni siitä hetkellisesti ja ainakin arvokasta kokemusta kerääntyi runsain mitoin. Projektitiimi toimi oikein mukavasti, eikä yhteishengessä ollut valittamista. Siispä... Kiitos kaikille mukana olleille!

8.2 Marko Koivuniemi

Syksyllä 2002 olin vielä Loviisassa kesätöiden parissa, kun huomasin, että sovellusprojekti olisi tarjolla. Kun sain tietää, että kesätyöni uhkaavasti venyivät syyskuun loppuun asti, olin hieman epävarmoin mielin, että pääsenkö projektiin ollenkaan. Aloitusluennoilla saimme kouraan listan erinäköisistä projekteista, jotka vaikuttivat todella kiinnostavilta. Rehellisesti sanottuna pikainen vilkaisu näytti ainoastaan yhden projektin, joka vaikutti vähemmän kiinnostavalta. Murphyn lakikin jo sanoo, että mihin projektiin sitten pääsin.

Projektiryhmä vaikutti kuitenkin hyvältä, olihan joukossamme kattavaa asiantuntemusta sekä JSP:stä että tietokannoista jo entuudestaan. Yritysedustajan kanssa järjestettävä rajapintakuvaus ja sen pohjalta tehtävä rajapinta muodostui jo projektin alussa isoksi kysymysmerkiksi, eikä projektiryhmän luottamus tämän tärkeimmän osa-alueen toimivuuteen ollut kovin suuri. Rajapintamäärittely kokonaisuudessaan antoi kuitenkin hyvää tietämystä määrittelyvaiheen vaikeuksista ja haasteista.

Projektin alkuvaihe oli tutustumista uusiin työkaluihin. Jo pelkästään Linuxin käyttö perustyökaluna oli hieman hapuilevaa. Projektin aikana tutustuimme uusiin työkaluihin ja etenkin \LaTeX -opinnot antavat paljon myös tulevaisuutta silmällä pitäen. CVS oli myös täysin uusi tuttavuus, joka helpotti projektin työtä huomatta-

vasti. Dokumentointi, palaverointi ja muut pakolliset projektiin liittyvät byrokra-
tiat tuntuivat välillä raskailta, mutta niistäkin sai varmasti paljon uutta tietoa ja
kokemusta.

Oma panokseni huoletti välillä minun lisäksi myös JP Santasta - tunneis-
sakin näkyi se, että minä olin välillä hieman sivustaseuraaja. Yritin kompensoi-
da parhaani mukaan osallistumalla dokumentointiin, mutta kireminen jäi hieman
puolitiehen. Syyksi tähän voisin mainita päällimmäisen, eli oman saamattomuuden
lisäksi sen, että vastuualueeni komponentit riippuvat ensisijaisesti rajapinnasta,
jonka määrittelyvaihe venyi puhumattakaan testattavast versiosta. En oikein tien-
nyt mistä lähteä liikkelle, kun vielä seisoi tyhjän päällä tietonamme vain
"pahoittelemme viivästystä".

Koodauksessa vastuualueenani oli Timmi-rajapinnan kanssa kommunikoivat
komponentit ja XML-parsinta. Alkuvaiheessa XML-parsinta vaatii erittäin pitkä-
kestoista tutustumista dokumentteihin ja dokumentaatioon, mutta sen jälkeen var-
sinainen parserointi sujuikin paremmin. Rajapinnan kanssa keskustelevien kom-
ponenttien työstäminen oli hankalaa, sillä rajapintamäärittelykin valmistui liian
myöhässä. Kun määrittely oli lyöty lukkoon, voitiin kehitystä aloittaa, mutta kom-
ponentin testausta ei voinut suorittaa.

Kaiken kaikkiaan projektissamme oli hyvä ja kannustava ryhmähenki, vaikka
kenties työnjako oli hieman epätasainen johtuen ehkä myös ryhmän osaamistason
vaihtelusta. Muiden projektin jäsenten osaamisesta toivottavasti muutama hippu-
nen jäi minunkin päähäni, sillä heiltä oli hyvä ammentaa apua omaan hieman
hapanoivaan toimintamalliin Java-ohjelmoinnin saralla.

Kiitän projektin jäseniä niistä yhteisistä hetkistä, joita vietimme Kiurun paris-
sa.

8.3 Jussi Mäkinen

Sovellusprojektin alku oli jännittävää ja oikein odotti, että pääsee koodaamaan
jotain hienoa ja suurta. Alkuvaiheessa tuntui, että aikaa oli jopa liikaa varattu pro-
jektille.

Parin viikon jälkeen karu totuus paljastui; tehtävää oli paljon ja kokonaiskuva
oli kaikilla hieman sekava. Lisäksi aiempien projektien materiaaleihin tutustumi-
nen vei paljon aikaa, koska dokumentteja oli paljon. Toisaalta myös paljon puuttui
dokumentaatiota, joka olisi ollut tarpeellista.

Opin projektin aikana projektityöskentelyä, organisoimista ja viestintää. Myös
dokumentointiin joutui paneutumaan koko projektin ajan. Välillä kyseenalaistin
joidenkin dokumenttien relevanttiutta. Opin käyttämään joitakin uusia työkaluja,
mistä on paljon hyötyä myös jatkossa.

Projektista jäi kokonaisuudessaan ihan mukava kuva. Kiitän kaikkia projektin
jäseniä hyvästä työstä sekä ohjaajia palautteesta ja vinkeistä.

8.4 Miika Nurminen

Kiuru-projekti onnistui projektina melko hyvin ja oppimiskokemuksena erinomaisesti. Olin jo ennakkoon asennoitunut, että odotettavissa on tähän mennessä työläin opintokokonaisuus, mutta työmäärän laajuus yllätti silti. Projektiinhan kuuluu ”itsestäänselvien” koodauksen ja dokumentoinnin lisäksi mm. palavereja, suunnittelua, asioista sopimista, tiedotusta ja tietenkin monta dokumenttien korjauskierrosta...

Projektin lopputulos ei ollut lähellekään tavoitteiden mukainen, mutta ainakin saimme tehtyä perusteellisen vaatimusmäärittelyn ja suunnitelman, josta on toivottavasti hyötyä myös tuleville projekteille. Myös osa valmistuneesta koodista on toivon mukaan uudelleenkäytettävää.

Vaikeinta oli projektimuotoisen työtavan omaksuminen. Kiuru-projekti oli itselleni ensimmäinen ryhmätyönä tehtävä ohjelmistoprojekti, mistä johtuen työtapojen löytäminen vei aikansa. Tehtävien jako ei aina selkiytynyt ja asioista sopiminen kesti yllättävän kauan. Edellisten Kotka-järjestelmän kehitykseen liittyvien projektien tietojen omaksuminen oli vaativaa. Teknisistä ohjaajista oli tässä suuresti apua.

Projekti lähti liikkeelle liian hitaasti. Rajapinnasta sopiminen Yomi Oy:n kanssa vei lähes kaksi kuukautta kauemmin kuin alunperin oli suunniteltu ja pääsimme testaamaan sitä vasta juuri ennen loppuesittelyä. Osittain tästä, osittain myös alkuperäisen tehtävänannon puutteellisuudesta johtuen määrittely- ja suunnitteluvaihe venyivät kohtuuttoman pitkiksi. Parempi olisi ollut määrittää osalle ryhmäläisistä koodustehtäviä jo suunnitteluvaiheen aikana.

Kehitystyössä käytettävät työkalut ja tekniikat olivat useimmille ryhmän jäsenistä uusia. Tämä hidasti hieman työskentelyä projektin alkupuolella, mutta oli hyvin opettavaista. Projektimuotoisen työtavan omaksumisen ohella projektin hyödyllisintä antia olikin ehkä uusien tekniikoiden oppiminen: CVS-versionhallinta, L^AT_EX-ladontaohjelma, Linux-skriptit sekä tietenkin Java ja JSP tulivat tutuiksi. Jukka-Pekka Santasen huomiot dokumenttien muodon ja sisällön suhteen olivat aluksi perusteellisuudessaan masentavia, mutta kiistämättä auttoivat täsmällisen ilmaisuuden kehittämisessä.

Erytishuomion ansaitsee vielä syksyn aikana kehittynyt hyvä ryhmähenki, joka osaltaan mahdollisti projektin läpiviennin. Kiitokset kaikille projektiin osallistuneille!

9 Yhteenveto

Projektiryhmä toteutti ensimmäisen version Korppi-järjestelmän salinvarausmoduulista. Kiuru-sovelluksen avulla voi tehdä sekä varauspyyntöjä että varauksia

Jyväskylän yliopiston saleihin halutulle tapahtumalle. Kiurun kautta voi myös vahvistajat vahvistaa muiden tekemiä varauspyyntöjä.

Projekti onnistui lähtökohdat huomioonottaen hyvin. Jos projekti toteutettaisiin uudestaan, niin luultavasti muuttaisimme aikataulua ja prosessimallia. Suunnitteluun käytettyä aikaa voisi hieman supistaa ja kokeilla prototyypin kautta tapahtuvaa sovelluskehitystä.

Lisäksi Timmin rajapintakuvaus olisi pitänyt saada paljon aikaisemmin valmiiksi. Tämä olisi hoitunut pitämällä suunnittelupäivä Taru Väisäsen kanssa paljon aiemmin. Välikäsien kautta tapahtunut kommunikointi hidasti määrittelyä liikaa.

Lähteet

- [1] Celko Joe ”SQL For Smarties”, Morgan Kaufmann Publishers, San Francisco, 2000.
- [2] Geary David M. ”Advanced JavaServer Pages”, Sun Microsystems - Prentice Hall PTR, Upper Saddle River, 2001.
- [3] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika ”Kiuru-projektin projektisuunnitelma”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [4] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika ”Kiuru-projektin sovellussuunnitelma”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [5] Hillebrand Minna, Silván Markus, Vanhanen Antti ja Ylitalo Marko, ”Koppelo-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [6] Jaakkola Mia, Juutinen Sanna, Lupari Matti ja Nieminen Mikko, ”Kolibri-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [7] Lesonen Minna, Pekkanen Hannu, Tawast Tuukka ja Uuksulainen Heikki, ”Korppi-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [8] Horppu Ismo, Mielityinen Markku ja Vire Markku, ”Kotka-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2000.
- [9] Santanen Jukka-Pekka, ”Tietotekniikan Sovellusprojektien ohje”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.

[10] Santanen Jukka-Pekka, ”Opinnäytteiden kirjoittaminen, lyhyt oppimäärä”,
Jyväskylän yliopisto, tietotekniikan laitos, 2000.