

# KIURU

Tietotekniikan sovellusprojekti

Toni Hilpinen  
Marko Koivuniemi  
Jussi Mäkinen  
Miika Nurminen

Projektisuunnitelma  
20.11.2002

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, puh.nro. (014) 260 4963, sähköpostilistojen osoitteet kiuru@korppi.jyu.fi ja kiurulaaja@korppi.jyu.fi.

**Työn nimi:** Kiuru-projekti, projektisuunnitelma

**Työ:** Projektisuunnitelma tietotekniikan sovellusprojektiin

**Tiivistelmä:** Kiuru-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitoksella toteutettuun Korppi-järjestelmään WWW-pohjaisen tilanvaraus- ja varausvahvistusosion. Projektisuunnitelmassa kuvataan projektin taustoja ja projektille asetettuja tavoitteita, projektiorganisaatiota sekä suunniteltua työnjakoa ja aikataulua.

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

## Versionhallinta

<b>Versio</b>	<b>Päiväys</b>	<b>Tehnyt</b>	<b>Muutokset</b>
0.1	25.9.2002	Jussi Mäkinen	
0.11	27.9.2002	Jussi Mäkinen	Pieniä muutoksia
0.2	2.10.2002	Jussi Mäkinen	Korjauksia J-P Santasen kommenttien pohjalta.
0.3	3.10.2002	Jussi Mäkinen	Lisätty aikataulusuunnitelma ja korjattu tekstiä
0.4	16.10.2002	Jussi Mäkinen	Korjauksia ja täydennyksiä
0.41	18.10.2002	Jussi Mäkinen	Dokumentti käännetty LaTeX-muotoon
0.42	30.10.2002	Jussi Mäkinen	Sivunumerointi korjattu
0.43	12.11.2002	Jussi Mäkinen	Pieniä muokkauksia
0.5	14.11.2002	Jussi Mäkinen	Korjauksia J-P Santasen kommenttien perusteella
0.51	20.11.2002	Jussi Mäkinen	Viimeisiä muutoksia

# Sisältö

<b>1 Johdanto</b>	<b>1</b>
<b>2 Termejä</b>	<b>1</b>
2.1 Aihepiirin yleisiä termejä . . . . .	2
2.2 Teknisiä termejä . . . . .	2
<b>3 Projektin taustaa ja tavoitteita</b>	<b>3</b>
3.1 Salivarausohjelmistojen taustaa . . . . .	3
3.2 Korppi-järjestelmän taustaa . . . . .	4
3.3 Projektin tavoitteet liittyen Korppi-järjestelmän varausmoduuliin . . . . .	4
3.4 Tietojen synkronointi . . . . .	5
3.5 Aihepiiriin liittyviä tulevaisuuden ideoita . . . . .	5
<b>4 Projektin resurssit ja organisaatio</b>	<b>5</b>
<b>5 Projektin vastualueet ja niiden jakautuminen</b>	<b>6</b>
<b>6 Projektin aikataulu</b>	<b>6</b>
<b>7 Riskien arviointi ja hallinta</b>	<b>7</b>
<b>8 Yhteenveto</b>	<b>7</b>
<b>Viitteet</b>	<b>8</b>
<b>Liite. Projektin aikataulu</b>	<b>9</b>

# 1 Johdanto

Kiuru-niminen tietotekniikan sovellusprojekti on jatkoa Kotka-, Korppi-, Kolibri- ja Koppelo-projekteille. Kiuru toteuttaa Kotka-järjestelmään tilanvarausmoduulin. Kiuru-ryhmän kirjoittama koodi julkaistaan GPL-lisenssin alaisuudessa.

Syksyn 2002 Kiuru-projektiryhmään kuuluvat tietotekniikan opiskelijat Toni Hilpinen, Marko Koivuniemi, Jussi Mäkinen ja Miika Nurminen. Tilaajana toimii Jyväskylän yliopistolta hallintovirasto ja tietotekniikan laitos. Projektiorganisaatioon kuuluvat henkilöt esitellään luvussa 4.

Projektisuunnitelmassa kuvataan projektin kulkuun liittyviä suunnitelmia. Projektisuunnitelmaan kuuluvat aikataulusuunnitelma, projektin tavoitteiden määrittely, projektiorganisaation kuvaus, projektin vastualueiden ja niiden jakautumisen suunnittelu sekä alkeellinen riskianalyysi.

Projektiryhmä laatii myös sovellussuunnitelman, jossa käsitellään tarkemmin sovelluksen rakennetta ja ominaisuuksia sekä vaatimusmäärittelyn, jossa käsitellään sovellukselle asetettuja vaatimuksia. Projektiraportti laaditaan projektin loppuvaiheessa ja siinä tarkastellaan projektin läpivientiä ja onnistumista. Sovellusraportti kuuluu myös projektin loppuvaiheen tuloksiin ja siinä käsitellään toteutetun sovelluksen onnistumista ja toimivuutta. Lisäksi ryhmä laatii sovelluksen käyttö- ja asennusohjeet. Projektiryhmä muokkaa myös aiemmin tehtyä Kotka-tietokantaraporttia.

Luvussa 2 esitellään aiheeseen oleellisesti liittyviä termejä sekä luvussa 3 projektin taustoja ja tavoitteita. Luku 4 esittelee projektin organisaation ja resurssit. Luvussa 5 käsitellään projektin vastualueita sekä niiden jakautumista projektiryhmän jäsenten kesken. Luku 6 kuvaa projektin suunnitellun aikataulun. Luvussa 7 arvioidaan projektiin liittyviä riskejä.

## 2 Termejä

Luvussa on kuvataan joitakin termejä, jotka tulee kaikkien projektiorganisaatioon kuuluvien henkilöiden ainakin jossain määrin sisäistää. Tekniset termit on pääosin listattu sellaisia henkilöitä varten, joilla ei ole ohjelmointitaustaa. Tällöin he voivat helpommin seurata keskustelua ja projektin laatimia dokumentteja.

## 2.1 Aihepiirin yleisiä termejä

Seuraavat termit liittyvät salinvarausprosessiin:

<b>Kotka</b>	on henkilötietojen hallintaosio ja tietokantarakenne.
<b>Korppi</b>	on Kotka-järjestelmän kurssikirjanpitomoduuli.
<b>Maksaja</b>	on henkilö tai organisaatio, joka maksaa varauksen.
<b>Oheisvarattava</b>	on liikuteltava esine, jonka vahtimestarit toimittavat tilaisuuden järjestäjän toiveiden mukaisesti haluttuun saliin haluttuna ajankohtana.
<b>Resurssi</b>	on johonkin saliin liittyvä ominaisuus tai kiinteästi saliin asennettu laite tai esine. Esimerkiksi salin paikkamäärä tai piirtoheittimen olemassaolo voivat olla resursseja.
<b>Resurssipyyntö</b>	kuten varauspyyntö, kuitenkin sillä erotuksella, että salin sijaan ilmoitetaan resurssien tarve.
<b>Sali</b>	on varattavissa oleva paikka, johon ihmiset voivat kokoontua (esimerkiksi luentosali tai kokoustila).
<b>Sali-järjestelmä</b>	on Jyväskylän yliopistossa tällä hetkellä käytössä oleva tekstipohjainen salienvarausjärjestelmä.
<b>Timmi</b>	on Yomi Applications Oy:n kehittämä tilanvaraussovellus.
<b>Vakiovaraus</b>	on sarja varauksia, jotka liittyvät toisiinsa ja toistuvat esimerkiksi viikoittain samana viikonpäivänä.
<b>Varauksen pyytäjä</b>	on henkilö, joka ehdottaa varattavaksi jotakin tilaa jonakin ajankohtana.
<b>Varauksen vahvistaja</b>	on henkilö, jolla on oikeus vahvistaa varaus tiettyyn tilaan.
<b>Varaus</b>	on käyttöoikeus tiettyyn tilaan tiettyinä ajankohtana.
<b>Varauspyyntö</b>	on varauksen pyytäjän tekemä ehdotus varaukseksi koskien tiettyä tilaa tiettyinä ajankohtana.

## 2.2 Teknisiä termejä

Seuraavia ohjelmointiteknisiiä termejä käytetään puhuttaessa sovelluksen taustalla olevasta järjestelmästä:

<b>Apache</b>	on ilmainen HTTP-palvelinohjelmisto.
<b>CSS</b>	eli Cascading Style Sheets on WWW-sivujen ulkoasua kuvaava kieli.
<b>HTML</b>	on WWW-sivujen sisältöä kuvaava kieli.
<b>HTTP</b>	on WWW-arkkitehtuurin käyttämä tiedonsiirtoprotokolla.
<b>Jakarta-Tomcat</b>	on ilmainen servletti- ja JSP-moottori.
<b>Java</b>	on Sunin kehittämä laitteistoriippumaton olio-ohjelmointikieli.
<b>Java-pavut</b>	(engl. <i>JavaBeans</i> ) ovat Java-ohjelmointikielellä luotuja komponentteja, joita voidaan kutsua JSP-sivuilla.
<b>JDBC</b>	eli Java Database Connectivity on Java-tekniikan käyttämä rajapinta erilaisiin tietokantoihin.
<b>JSP</b>	eli Java Server Pages on skriptaustyylinen ohjelmointikieli.
<b>Keksi</b>	(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.
<b>Käyttötapaus</b>	(engl. <i>use case</i> ) on käyttäjän tai sovelluksen toimintoa tietyn tehtävän suorittamiseksi kuvaava dokumentti.
<b>Moduuli</b>	on tarkkaan rajattu toiminnallinen ohjelmakokonaisuus.
<b>Poolman</b>	on ohjelma, joka kontrolloi tietokantayhteyksiä.
<b>PostgreSQL</b>	on ilmainen tietokannanhallintajärjestelmä.
<b>Skripti</b>	on yksinkertainen ohjelmanpätkä, joka useimmiten on tehty tulkattavalla kielellä.
<b>Selain</b>	on ohjelma, joka käyttäjän koneella tulkaa HTML-kieliset sivut esitettävään muotoon.
<b>Servletti</b>	(engl. <i>servlet</i> ) on palvelimella sijaitseva sovellus, joka toteuttaa HTTP-palvelimen pyynnöstä tietyn toiminnon.
<b>SQL</b>	eli Structured Query Language on relaatiotietokantojen hallintaan kehitetty standardi kieli.

### 3 Projektin taustaa ja tavoitteita

Luvussa esitellään projektin taustoja ja projektille asetettuja tavoitteita.

#### 3.1 Salivarausohjelmistojen taustaa

Jyväskylän yliopistossa on tällä hetkellä käytössä keskitetty Sali-niminen tekstipohjainen varaussovellus. Järjestelmällä hoidetaan periaatteessa kaikkien yliopiston tilojen varaus.

Yomi Oyj:n Timmi-järjestelmä piti ottaa käyttöön 23.10.2002. Se korvanee tähän asti käytössä olleen Sali-järjestelmän. Timmi-järjestelmän merkittävin etu on automaattinen salien

laskutuksesta huolehtiminen, mitä nykyinen Sali-järjestelmä ei tue. Timmi on Windows-sovellus, johon on nykyään saatavilla myös WWW-käyttöliittymä.

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

## **3.2 Korppi-järjestelmän taustaa**

Tietotekniikan laitoksella on ollut 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 sovellusten tarvitsemia muutoksia [1].

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 opinnäytetyömoduulin.

## **3.3 Projektin tavoitteet liittyen Korppi-järjestelmän varausmoduuliin**

Projekti toteuttaa järjestelmän, jonka avulla salien varaus kurssien opetustapahtumille voidaan hoitaa alusta loppuun asti Korppi-järjestelmällä. Varaukset tulisi pystyä hoitamaan keskitetysti kaikkien laitoksen kurssien osalta (yleensä ennen lukukauden alkua, kun päätetään järjestettävistä kursseista) tai yksittäisten opetustapahtumien tarkkuudella (muutokset kurssin järjestelyissä lukukauden aikana tai järjestettäessä uutta kurssia kesken vuoden). Opetustapahtumien lisäksi saleja ja tiloja voidaan varata muuhunkin käyttöön. Tällaisia tapahtumia ovat esimerkiksi kokoukset, konferenssit ja maksulliset tapahtumat.

Opetus- ja muita tapahtumia varattaneen jatkossa Timmi-järjestelmällä, mistä poikkeuksena laitoksen omassa käytössään olevat tilat. Tällaisia tiloja ovat mm. kokoustilat ja laboratoriot. Tällaisia tiloja ei oteta Timmi-järjestelmän piiriin. Näiden tilojen varaus on mahdollista liittää Korppi-järjestelmään.

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



### **3.4 Tietojen synkronointi**

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

Käytännössä synkronointi hoidetaan HTTP-protokollaa käyttäen XML-tiedostoilla. Jokaisesta Timmiin tehdystä lisäys-, muokkaus- tai poistotietokantatapahtumasta laaditaan uusi tiedosto. Tiedostot numeroidaan juoksevasti, tietokantatapahtumien keskinäisen järjestyksen varmistamiseksi. Tiedoston sisällä kuvataan tapahtumaan liittyvät tiedot (tarkempi formaatti toimitetaan Yomi Oyj:n taholta myöhemmin ilmoitettavana ajankohtana). Myös kyselyt ja tietojen päivitykset Korppi-järjestelmästä Timmiin suoritetaan vastaavan muotoisilla tiedostoilla.

Yksittäisten tapahtumailmoitusten lisäksi Timmistä tarkastetaan varausten nykytilanne säännöllisin väliajoin. Tilannetiedot saadaan muuten samalla tiedostoformaattilla kuin tapahtumailmoitukset, mutta nyt tiedoston sisällä voi olla useampia tapahtumia.

Synkronoinnissa mahdollisena ongelmana on Kotkan ja Timmin välillä siirrettävien tietojoukkojen yhtenäisyys, erityisesti varaavien henkilöiden tunnistamisen osalta. Kotka-järjestelmään on määritelty henkilöille yksilölliset tunnukset, mutta Timmissä niitä ei ole. Yhtenäisyysongelman ratkaisusta päätetään tarkemmin ennen kuin tiedostoformaatti lyödään lukkoon ko. järjestelmien välillä.

### **3.5 Aihepiiriin liittyviä tulevaisuuden ideoita**

Kotka-tietokantaan ja opettajan käyttöliittymään lisätään mahdollisuus tilatarpeiden merkitsemiseen. Tilatarpeiden merkitseminen hoidetaan tällä hetkellä siten, että kurssin opettaja toimittaa tilatarpeet henkilölle (esim. amanuenssi), joka suunnittelee kootusti laitoksen kurssien opetustapahtumien sijoittamisen käytössä oleviin saleihin. Tilatarpeet sisältävät vaatimuksia salin paikkamäärään ja tarvittaviin resursseihin (esim. videoprojektori, liitutaulu). Resurssit voivat liittyä saliin kiinteästi tai ne voivat olla erikseen varattavia. Tietokannassa huomioidaan tilatarpeiden suunnittelu ja muokkaus, mutta varsinaista suunnitteluohjelmaa projektin resurssien puitteissa ei ehditä toteuttamaan.

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

## **4 Projektin resurssit ja organisaatio**

Projektiryhmän muodostavat 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, Ta-

ru Väisänen ja Jukka Valkonen ovat Timmi-järjestelmän kehittäneen Yomi Oyj:n edustajia. Jukka-Pekka Santanen on projektin vastaava ohjaaja. Teknisiä ohjaajia ovat Markku Vire (vastuualueenaan tietokannat) ja Kirsi Koponen (vastuualueenaan Java Server Pages).

Projektiryhmä sai käyttöönsä tietotekniikan laitokselta Agorasta huoneen C222.2, jossa on asennettuina kolme Linux-työasemaa ja yksi Windows 2000 -työasema. Käyttöjärjestelmäksi Linux-työasemille on asennettuna Redhat 7.3 -levitysversio. Käyttöjärjestelmään kuuluu useita erilaisia sovelluspaketteja.

Lisäksi sovelluspalvelimeksi asennettiin Jakarta Tomcat 3.2.3. Sovelluskehittimenä toimii NetBeans. Windows-työasemaan on asennettu mm. WinTimmi, Microsoft Visio 2000 ja MS Project.

Projektin dokumentointi on luettavissa projektin kotisivuilta

<http://kotka.it.jyu.fi/kiuru>. Projektilla on käytössä sähköpostilistat [kiuru@korppi.jyu.fi](mailto:kiuru@korppi.jyu.fi) ja [kiurulaaja@korppi.jyu.fi](mailto:kiurulaaja@korppi.jyu.fi). Sähköpostilistojen keskustelua voi lukea WWW-selaimella osoitteesta <http://korppi.it.jyu.fi/list-archive/kiuru/> ja <http://korppi.it.jyu.fi/list-archive/kiurulaaja>.

## 5 Projektin vastualueet ja niiden jakautuminen

Ryhmä jakaa tehtäviä melko monipuolisesti siten, että kaikki tekevät (ainakin jossain vaiheessa projektia) kutakin eri tehtävätyyppiä : suunnittelua, toteutusta ja testausta. Näin toimien kukin projektiryhmän jäsen saa kokemusta kaikista ohjelmistoprosessin vaiheista.

Kuitenkin alustavasti ryhmän jäsenet ovat sopineet keskenään kunkin päävastuualueesta. Koska ryhmän JSP-osaaminen on projektin alussa lähinnä Jussi Mäkisellä, hän on ryhmän JSP-vastuuhenkilö. Jussi on myös (ainakin projektin alussa) projektipäällikkö. Miika Nurminen vastaa pääosin tietokannan suunnittelusta. Marko Koivuniemi vastaa projektin kirjoittamien dokumenttien viimeistelystä. Toni Hilpinen vastaa pääosin XML-lukijan toteuttamisesta.

Muilta osin (mm. käyttöliittymän suunnittelu) vastualueet jaetaan myöhemmin. Dokumentointi hoidetaan koko ryhmän voimin, jotta kukaan ei kyllästyisi dokumentointityöhön. Dokumentointia toteutetaan koko projektin ajan tasaisesti, jotta kaikki oleellinen saadaan kirjattua. Toisaalta tällöin projektin loppuvaiheessa ei tule paniikkia raporttien loppuunhiomisen kanssa.

## 6 Projektin aikataulu

Projekti kestää 14 viikkoa ja kyseinen aikajänne on jaettu kolmeen päävaiheeseen: analyysiin ja suunnitteluun, toteutukseen sekä testaukseen. Analyysi ja suunnittelu -vaihe kestää 5 viikkoa. Toteutusvaihe kestää 5 viikkoa. Testaukselle ja loppuhiomiselle on varattu 4 viikkoa.

Ensimmäisen vaiheen tuloksia ovat projektisuunnitelma, sovellussuunnitelma ja vaatimusmäärittely. Toisen vaiheen tuloksia ovat valmis ohjelmakoodi (JavaBeanit ja JSP-sivut) sekä

koodin dokumentaatio. Kolmannen vaiheen tuloksia ovat testaussuunnitelma, testausraportti, projektiraportti ja sovellusraportti. Vaiheet on esitelty taulukossa. Tarkemmin aikataulutusta käsitellään liitteessä 1.

<b>Vaihe</b>	<b>Tulos</b>	<b>Esittely</b>
Suunnittelu 1	Projektisuunnitelma	04.10.2002
Sunnittelu 2	Sovellussuunnitelma	17.10.2002
Toteutus	Ohjelmakoodi ja dokumentaatio	13.12.2002
Testaus	Testausraportti, sovellusraportti ja projektiraportti	20.12.2002

Taulukko 6.1: Projektin vaiheet ja tarkistuspisteet

## 7 Riskien arviointi ja hallinta

Projektin suurin riski on sovitussa aikataulussa pysyminen. Projektiin kuuluvien henkilöiden henkilökohtaisten aikataulujen osalta pyritään joustamaan, jotta projekti saadaan toteutettua aikataulussa. Projektipäällikkö yrittää myös huolehtia aikataulussa pysymisestä.

Toinen melkein yhtä suuri riski on Timmi-sovelluksen ja Kiuru-sovelluksen välinen viestintä ja yhteistoiminta. Timmi ei myöskään ole vielä aktiivisessa käytössä salien varausprosessissa, mikä aiheuttaa epävarmuutta varausprosessiin. Riskit pienenevät jonkin verran, jos Yomi Oyj saa rajapintamäärittelyt valmiiksi mahdollisimman aikaisessa vaiheessa.

Myös mahdolliset eriävät näkemykset ja mielikuvat toteutettavasta sovelluksesta tuovat mukanaan omat haasteensa. Kaikkia esitettyjä toiveita ei ole mahdollista toteuttaa projektin puitteissa. Tarvittaessa projektioorganisaation kesken sovitaan ominaisuuksista, jotka jätetään myöhempien jatkokehittäjien toteutettaviksi. Riskiä pystytään pienentämään myös laatimalla sekä kattavat että selkeät pöytäkirjat ja dokumentit koko projektin ajan.

Uusin ilmennyt riski, jota ei kyetty havaitsemaan ajoissa, on olemassaolevien ohjelmien lisensseihin liittyvä riski. Korppi-järjestelmä on toteutettu GPL-lisenssin alla ja toisaalta Timmi on kaupallinen, suljettu ohjelma.

Sovelluksen testiversio täytyy saada valmiiksi hyvissä ajoin, jotta testauksessa saatua palautetta voidaan hyödyntää sovelluksen loppuhiomisessa. Järjestelmää yritetään testata mahdollisimman kattavasti testaussuunnitelman mukaisesti, jotta ongelmat saadaan poistetuksi ennen järjestelmän luovuttamista tilaajalle.

Pääosa riskeistä kohdistuu siis yhteistoimintaan muiden sovellusten kanssa ja varsinaiseen sovelluskehitykseen, jota tarkastellaan tarkemmin projektiryhmän myöhemmin laatimassa sovellussuunnitelmassa.

## 8 Yhteenveto

Projektiryhmä toteuttaa tilanvarausmoduulin Korppi-järjestelmään. Koska Kiuru-projekti liittyy läheisesti aikaisemmin toteutettuihin Kotka-, Korppi-, Halko-, Kolibri-, ja Koppelo-pro-

jekteihin sekä Sali- ja Timmi-sovelluksiin, tulee ryhmän olla ainakin jossain määrin selvillä myös kaikista projektien tuloksista.

Projektin läpivienti on vaativaa työtä varsinkin, kun projektilaisilla ei ole mainittavaa kokemusta suurten järjestelmien kehittamisestä. Osa projektilaisista on kuitenkin kehittänyt joi-takin pienempiä sovelluksia ennen tätä projektia. Kukin projektinjäsen tekee parhaansa ja toivon mukaan se riittää takaamaan kaikkia osapuolia tyydyttävän lopputuloksen.

## Lähteet

- [1] Hillebrand Minna, Silván Markus, Vanhanen Antti ja Ylitalo Marko, ”Koppelo-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [2] Jaakkola Mia, Juutinen Sanna, Lupari Matti ja Nieminen Mikko, ”Kolibri-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [3] Lesonen Minna, Pekkanen Hannu, Tawast Tuukka ja Uuksulainen Heikki, ”Korppi-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [4] Horppu Ismo, Mielityinen Markku ja Vire Markku, ”Kotka-projektin projektikansio”, Jyväskylän yliopisto, tietotekniikan laitos, 2000.
- [5] Santanen Jukka-Pekka, ”Tietotekniikan Sovellusprojektien ohje”, Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [6] Santanen Jukka-Pekka, ”Opinnäytteiden kirjoittaminen, lyhyt oppimäärä”, Jyväskylän yliopisto, tietotekniikan laitos, 2000.