

KYYHKY-PROJEKTI

Timo Aarniovuori
Teija Alasalmi
Jaakko Hyvärinen
Maunu Tuomainen

Projektiraportti
16.6.2004, versio 1.0

Jyväskylän yliopisto
Tietotekniikan laitos

KYYHKY-PROJEKTIN TIETOJA

Tekijät:

Timo Aarniovuori (taarnio@cc.jyu.fi)

Teija Alasalmi (teikku@cc.jyu.fi)

Jaakko Hyvärinen (japahyva@cc.jyu.fi)

Maunu Tuomainen (mttuomai@cc.jyu.fi)

Yhteystiedot:

Projektiorganisaation sähköpostilista on kyyhky@korppi.jyu.fi. Sen keskustelua voi seurata osoitteessa <https://korppi.it.jyu.fi/list-archive/kyyhky/>. Projektin kotisivu löytyy osoitteesta <http://kotka.it.jyu.fi/kyyhky/>.

Työn nimi: Kyyhky-projekti, projektiraportti

Työ: Projektiraportti tietotekniikan Sovellusprojektiin

Tiivistelmä:

Kyyhky-projekti toteutti päiväkirjapinnan Jyväskylän yliopiston tietotekniikan laitoksella toteutettuun Korppi-nimiseen opintotietojärjestelmään. Rajapinnan avulla pystytään synkronoimaan kalenteritapahtumia Korppi-järjestelmän kalenteriosion sekä erilaisten mikrojen, matkapuhelimien, taskumikrojen ja muiden kämmenlaitteiden kalenterien kesken. Projektiraportissa kuvataan projektin läpivientiä ja tavoitteiden toteutumista.

Avainsanat: synkronointi, sähköinen kalenteri, kalenterisovellus, SyncML, WWW-sovellus, Korppi-järjestelmä

VERSIOHISTORIA

Versio	Päiväys	Tehnyt	Muutokset
0.1	14.5.2004	Teija Alasalmi	Ensimmäinen versio.
0.2	24.5.2004	Teija Alasalmi	Korjauksia Santasen kommenttien pohjalta, lisätty ryhmän jäsenten omat kokemukset lukuun 9 sekä luku 8 riskien arviointi ja hallinta.
0.3	7.6.2004	Teija Alasalmi	Kirjoitusasuun ja lukujen 6, 7 ja 8 sisällön muokkausta.
0.4	8.6.2004	Teija Alasalmi	Korjauksia Santasen kommenttien pohjalta.
0.5	14.6.2004	Teija Alasalmi	Korjauksia ja lisäyksiä lukuun 6 ja 7. Lisätty liitteet.
0.6	16.6.2004	Teija Alasalmi	Pieniä korjauksia kirjoitusasuun Santasen kommenttien perusteella. Lisäyksiä lukuun 7 Santasen ehdotusten perusteella.
1.0	18.6.2004	Teija Alasalmi	Hyväksytty versio.

SISÄLLYSLUETTELO

1	JOHDANTO	1
2	TERMEJÄ	2
2.1	Korppi-järjestelmään liittyvät termit	2
2.2	Käytettäviin työkaluihin, ohjelmistoihin sekä ohjelmointikieliin liittyvät termit.....	2
2.3	SyncML -tekniikkaan liittyvät termit.....	4
3	PROJEKTIN TAUSTAA	6
3.1	Sähköiset kalenterit	6
3.2	Korppi-järjestelmä	7
4	PROJEKTIN TAVOITTEET JA NIIDEN TOTEUTUMINEN	9
4.1	Toteutettu sovellus	9
4.2	Oppiminen.....	10
5	PROJEKTIN RESURSSIT JA ORGANISAATIO	12
5.1	Henkilöt ja yhteystiedot	12
5.2	Laitteet ja ohjelmistot.....	13
6	PROJEKTIN TEHTÄVÄT JA NIIDEN JAKAUTUMINEN	13
6.1	Tehtäväalueet	13
6.2	Työnjako projektin alussa.....	13
6.3	Suunnittelu- ja toteutusvaihe.....	14
6.4	Projektipäällikön tehtävät	15
6.5	Viimeistelyvaihe	15
6.6	Dokumentointi.....	16
6.7	Suunnittelun tehtäväjaon onnistuminen.....	16
7	PROJEKTIN AIKATAULU	19
7.1	Aikataulun toteutuminen.....	19
7.2	Aikataulumuutosten pohdintaa	20
7.3	Työtunnit viikoittain ja tehtävätyypeittäin.....	22
8	RISKIEN ARVIOINTI JA HALLINTA	25
8.1	Aikatauluun liittyvät riskit	25
8.2	Vaatimusten määrittelyyn liittyvät riskit.....	25
8.3	Tekniset riskit.....	26
8.4	Muut riskit.....	27
8.5	Riskien vaikutus aikatauluun ja tavoitteisiin	27
9	HENKILÖKOHTAISET KOKEMUKSET PROJEKTISTA	28
9.1	Timo Aarniovuori.....	28
9.2	Teija Alasalmi.....	29
9.3	Jaakko Hyvärinen.....	31
9.4	Maunu Tuomainen.....	33
10	YHTEENVETO	36
	LÄHTEET	37
	LIITTEET	39
	Liite 1. Kyyhky-ryhmän yhteenlasketut työtunnit viikoittain ja tehtävittäin.....	40
	Liite 2. Timo Aarniovuoren työtunnit viikoittain ja tehtävittäin.....	41
	Liite 3. Teija Alasalmen työtunnit viikoittain ja tehtävittäin.....	42
	Liite 4. Jaakko Hyvärisen työtunnit viikoittain ja tehtävittäin.....	43
	Liite 5. Maunu Tuomaisen työtunnit viikoittain ja tehtävittäin	44

1 JOHDANTO

Kyyhky-niminen tietotekniikan Sovellusprojekti toteutti prototyypin synkronointirajapinnasta. Sen avulla erilaisten mikrojen, matkapuhelinten ja kämmenlaitteiden kalenterisovellukset voidaan synkronoida Korppi-opintotietojärjestelmän kalenterin kanssa.

Projektin puitteissa toteutuneita osia ovat Sync4j SyncServeriin liittyvä rajapinta sekä Korppi-opintotietojärjestelmään sisällytetty konfliktienhallintaosa. Sync4j-palvelinta hyödyntävän rajapinnan avulla pystytään siirtämään kalenteritapahtumia henkilökohtaisesta synkronoitavasta kalenterista Korpin kalenteriin ja päinvastoin. Korppi-opintotietojärjestelmään toteutetun konfliktienhallintaosan sivujen avulla voidaan käsitellä syntyneet konfliktit sekä määritellä synkronoitavat kalenterit.

Dokumentissa kuvataan Kyyhky-projektin läpivientä sekä arvioidaan suunnitelmien ja asetettujen tavoitteiden toteutumista. Sovelluksen rakennetta ja toteutuksessa käytettyjä ratkaisuja kuvataan erillisessä Sovellusraportissa. Aiemmin laadittuja dokumentteja ovat Projektisuunnitelma, Sovellussuunnitelma sekä Vaatimusmäärittely. Lisäksi laadittiin Asennus- ja käyttöohjeet. Korppikyselyn tulokset raportoitiin erillisessä Kyselyraportissa.

Luvussa 2 esitellään aiheeseen ja toteutukseen oleellisesti liittyviä käsitteitä. Luvussa 3 kuvataan taustaa ja luvussa 4 arvioidaan asetettujen tavoitteiden toteutumista. Luvussa 5 kuvataan projektin organisaatio ja resurssit. Luvussa 6 kuvataan toteutunutta työnjakoa projektin eri vaiheiden osalta. Luvussa 7 tarkastellaan projektille suunnitellun aikataulun toteutumista. Luku 8 kuvaa projektin arvioituja riskejä sekä niiden toteutumista, välttämistä ja hallintaa. Luvussa 9 projektin jäsenet kuvailevat henkilökohtaisia kokemuksiaan kevään 2004 sovellusprojektista.

2 TERMEJÄ

Luvussa kuvataan Kyyhky-projektin dokumentoinnissa käytettyä termistöä.

2.1 Korppi-järjestelmään liittyvät termit

Korppi-järjestelmään ja sitä kehittäneisiin projekteihin liittyviä termejä ovat seuraavat:

Kiuru	toteutti Korppi-järjestelmään salivaraussovelluksen.
Korppi	on Jyväskylän yliopistossa kehitetty opintotietojärjestelmä.
Kotka	on Korppi-järjestelmän tietokanta ja henkilötietojen hallintaosio.
Kolibri	toteutti Korppi-järjestelmään päiväyri- ja ajanvarausosion.

2.2 Käytettäviin työkaluihin, ohjelmistoihin sekä ohjelmointikieliin liittyvät termit

Projektin toteutuksessa käytettäviin ohjelmointi- ja kuvauskieliin liittyviä termejä ovat seuraavat:

CSS	eli Cascading Style Sheets on WWW-sivujen ulkoasun kuvaamiseen käytetty kieli.
HTML	(HyperText Markup Language) on WWW-sivujen sisältöä kuvaava kieli.
Java	on Sun Microsystemsin kehittämä laitteistoriippumaton olio-ohjelmointikieli.
Java-pavut	(engl. <i>JavaBeans</i>) ovat Java-ohjelmointikielellä toteutettuja uudelleen käytettäviä komponentteja, joita voidaan kutsua JSP-sivuilla.
JSP	eli Java Server Pages on skriptaustyylinen ohjelmointitekniikka, jossa HTML-koodin sekaan on mahdollista lisätä Java-kielellä kirjoitettua koodia.
Relaatiotietokanta	on tietokanta, jossa tiedot esitetään tauluina (engl. <i>table</i>) ja niitä yhdistävinä suhteina.

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 standardoitu kieli.
Tietokanta	on kokoelma loogisesti yhteenliittyvää tietoa.
XML	on rakenteinen dokumenttien määrittely- ja kuvauskieli.

Projektissa käytettävään kehitysympäristöön ja työkaluihin liittyviä termejä ovat seuraavat:

Apache	on vapaan lähdekoodin HTTP-palvelinohjelmisto.
CVS	(Concurrent Version System) on ohjelmistokehityksessä käytettävä versionhallintajärjestelmä.
HTTP	(HyperText Transfer Protocol) on WWW-tekniikassa käytettävä tiedonsiirtoprotokolla, jolla selain pyytää haluttuja sivuja WWW-palvelimelta.
JDBC	eli Java Database Connectivity on Java-tekniikan käyttämä tietosilta erilaisiin tietokantoihin.
J2EE	(Java 2 Enterprise Edition) on standardi, joka mahdollistaa palvelinsovellusten tuottamisen niin, että sovelluksia voi ajaa millä tahansa J2EE-palvelimella.
Palvelin	on WWW-sovellusten tapauksessa ohjelmisto, joka palvelee asiakkaana toimivien selainten hakupyynnöitä.
PostgreSQL	on vapaa relaatiotietokannan hallintajärjestelmä.
Protokolla	eli yhteiskäytäntö on sovittu liikennöintitapa, jota noudattaen laitteet tai ohjelmat voivat vaihtaa tietoa keskenään. Protokolla määrittelee siirrettävän tiedon esitystavan sekä mekanismin, jolla tieto siirretään.
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 tulkkaa HTML-kieliset sivut kuvaruudulla esitettävään muotoon.
SSL	(Secure Sockets Layer) on protokolla, jonka yli voidaan muodostaa yhteyksiä muilla protokollilla, kuten esimerkiksi HTTP:llä. Yhteys on salattu.
Tomcat	on vapaan lähdekoodin servletti- ja JSP-moottori.
Versionhallintajärjestelmä	on järjestelmä, johon voidaan mm. tallentaa tiedostoja siten, että tiedostojen kaikki versiot ovat palautettavissa.

2.3 SyncML -tekniikkaan liittyvät termit

Toteutustekniikkaan liittyviä termejä ovat seuraavat:

Avaintenkartoitus	(engl. <i>ID mapping</i>) kertoo, mitkä asiakkaan tunnisteet (LUID) ja palvelimen tunnisteet (GUID) vastaavat toisiaan. Tehtävän tekee Sync4j-palvelin.
GUID	(<i>Global Unique Identifier</i>) on tietueen yksilöivä numero SyncML-palvelimella.
Henkilökohtainen kalenteri	on pöytäkoneessa, matkapuhelimessa, taskumikrossa tai verkossa oleva sähköinen kalenteri.
Hidas synkronointi	(engl. <i>slow synchronization</i>) on kaksisuuntaisen synkronoinnin erikoismuoto, jossa kaikki asiakkaan tietoalkiot lähetetään palvelimelle. Palvelin vertaa jokaista tietoalkiota omiinsa ja tekee näille analyysin.
IMEI	(<i>International Mobile Equipment Identifier</i>) on mobiililaitteen yksilöivä 15-osainen numero.
Kaksisuuntainen synkronointi	(engl. <i>two-way synchronization</i>) on yleisin synkronointimenetelmä, jossa asiakas ja palvelin lähettävät toisilleen muuttuneet tietoalkionsa.
Konflikti	on synkronoinnin virhetilanne, jossa samaa tapahtumaa on muokattu synkronoitavissa kalenterisovelluksissa.
LUID	(<i>Locally Unique Identifier</i>) on tietueen yksilöivä numero asiakaslaitteessa.

MD5	(Message digest 5) on funktio, jota käytetään tiedon salaukseen. Se laskee annetusta syötteestä vasteen. Alkuperäisen syötteen palauttaminen vasteesta on mahdotonta.
SyncML	(Synchronization Markup Language) on synkronointiprotokolla henkilökohtaisen tiedon synkronointiin.
Synkronointi	(engl. <i>synchronization</i>) tarkoittaa saman tiedon pitämistä ajantasaisena kahden tai useamman tietokannan välillä.
Synkronointiankkuri	(engl. <i>synchronization anchor</i>) on aikaleima tai merkkijono, joka kuvaa synkronointitapahtumaa. Synkronointiosapuolet tallentavat toistensa synkronointiankkurit, ja siten voivat todeta toisissaan mahdollisesti tapahtuneet häiriötilat. SyncML määrittelee kaksi synkronointiankkuria: <code>Next</code> ja <code>Last</code> .
Sync4j	on Java-ohjelmointikielellä toteutettu vapaan lähdekoodin SyncML-palvelinohjelmisto.
Työpöytäsynkronointi	(engl. <i>desktop synchronization</i>) on työpöytäsovelluksen ja mobiililaitteen välillä tapahtuvaa synkronointia.
URI	(<i>Uniform Resource Identifier</i>) on universaali tapa yksilöidä resursseja.
vCalendar	on laitteisto- ja tiedonsiirtoriippumaton tiedostoformaatti kalenteritiedon tallentamiseen ja välittämiseen.
vCard	on formaatti henkilöön liittyvien yhteystietojen tallentamiseen ja välittämiseen.
Virkistysynkronointi	(engl. <i>refresh synchronization</i>) on synkronointimenetelmä, jossa lähettäjä lähettää kaikki tietoalkionsa vastaanottajalle, joka korvaa omat tietonsa lähettäjän tiedoilla. Menetelmä toimii molempiin suuntiin, eli joko asiakkaalta palvelimelle, tai palvelimelta asiakkaalle.

Yksisuuntainen synkronointi (engl. *one-way synchronization*) on synkronointimenetelmä, jossa muuttuneet tiedot lähetetään yhteen suuntaan, joko asiakkaalta palvelimelle tai palvelimelta asiakkaalle. Vastaanottaja ei lähetä omia muuttuneita tietojaan.

3 PROJEKTIN TAUSTAA

Kyyhky-projekti toteutti **päivyrirajapinnan** Jyväskylän yliopiston tietotekniikan laitoksella toteutettuun Korppi-nimiseen opintotietojärjestelmään. Rajapinnan avulla pystytään synkronoimaan tapahtumia Korppi-opintotietojärjestelmän kalenteriosion ja erilaisten mikrojen, matkapuhelimien ja muiden kämmenlaitteiden kalenterien kesken. Tarkemmin toiminnot on esitetty Vaatimusmäärittelyssä.

Luvussa käsitellään projektin lähtökohtia. Luvussa 3.1 kuvataan sähköisiä kalentereita sekä toteutetun synkronointisovelluksen tarpeita. Luvussa 3.2 kuvataan Korppi-opintotietojärjestelmää ja sitä kehittäneitä projekteja.

3.1 Sähköiset kalenterit

Vuosikausia käytössä olleiden paperikalentereiden rinnalle on viime vuosina tullut monenlaisia **sähköisiä kalentereita**, jotka tarjoavat paperikalentereita tehokkaamman mahdollisuuden henkilö- ja ryhmäkohtaiseen tapahtumien hallintaan. Tämän päivän langattomat tekniikat ja mobiililaitteiden ominaisuudet mahdollistavat aivan uudenlaisen tavan saada kalentereihin uutta tietoa. Uusilla langattomilla tekniikoilla paperikalentereiden perusominaisuudet on mahdollista saada jokaiseen kannettavaan laitteeseen. Lisäksi sähköisissä kalentereissa oleva tieto on helpommin muutettavissa ja päivitettävissä verrattuna paperiseen kalenteriin.

Nykyään yliopiston opiskelijoilla ja henkilökunnalla on käytössään erilaisia matkapuhelimia, kämmenlaitteita ja mikroja, joista löytyy sähköisiä kalentereita. Opiskelijalla tai henkilökuntaan kuuluvalla voi olla usealla mobiililaitteella tai työasemallaan

henkilökohtainen sähköinen kalenteri, joten kalenteritapahtumien ja kontaktitietojen tulisi tällöin näkyä samanlaisina kaikissa laitteissa ilman erillistä tapahtumien kirjausta. Muiden kalenterien ohella opiskelijalla tai henkilökuntaan kuuluvalla saattaa olla käytössään myös Korppi-opintotietojärjestelmässä oleva kalenteri. Kannettavien laitteiden yleistyttyä henkilökohtaista kalenteritietoa löytyy siis usealta eri laitteilta. Usean sähköisen kalenterin ajantasalla pitäminen on työlästä ja virhealtista.

Opiskelijoiden ja henkilökunnan **henkilökohtaisten sähköisten kalenterien synkronointi** Korppi-opintotietojärjestelmässä olevan kalenterisovelluksen kanssa tarjoaa järjestelmän käyttäjille monenlaista lisäarvoa. Ensinnäkin, käyttäjät saadaan helpommin tietoisiksi omista ja muiden menoista, sillä synkronointi mahdollistaa ryhmä- ja opetustapahtumien päivittämisen Korppi-opintotietojärjestelmän kalenterista henkilökohtaiseen kalenteriin sekä henkilökohtaisessa kalenterissa olevien henkilökohtaisten tapahtumien päivittämisen Korpin kalenteriin. Toiseksi, turha kalenteritapahtumien ja muiden henkilökohtaisten tietojen ja tapahtumien syöttäminen vähenee, koska kalenteritapahtumia tarvitsee syöttää vain yhteen kalenteriin synkronoinnin hoitaessa loput. Lisäksi synkronoinnin toteuttamisella voidaan mahdollisesti saavuttaa lisää käyttäjiä Korppi-opintotietojärjestelmään. Tarkemmin synkronointitekniikkaa kuvataan Vaatimusmäärittelyssä.

3.2 Korppi-järjestelmä

Korppi-opintotietojärjestelmää edelsi Kurki-niminen WWW-sovellus, joka oli ensimmäinen Jyväskylän yliopistossa toteutettu kurssikirjanpitojärjestelmä. Kurki-järjestelmä toteutettiin keväällä 1998 silloiselle matematiikan laitokselle tietotekniikan cum laude -työprojektina. Kyseisen järjestelmän tarkoituksena oli yhdenmukaistaa laitoksella käytetyt kurssikirjanpitojärjestelmät sekä laajentaa niiden käyttö verkkoon. Kurki-järjestelmässä esiintyneiden käytettävyysongelmien ja toteutustekniikan rajoittuneisuuden vuoksi aloitettiin järjestelmän laajamittaisempi kehitys.

Ensimmäinen Korppi-opintotietojärjestelmän kehittäjä oli syksyn 2000 Kotka-projekti. Se suunnitteli tietokannan rakenteen sekä määritteli uuden opintotietojärjestelmän omaisuudet.

Lisäksi Kotka-projekti toteutti järjestelmään henkilötietomoduulin, sekä kartoitti ja testasi sopivia toteutustekniikoita, ohjelmointikieliä ja ohjelmistoja.

Kotka-projektin työtä jatkoi Korppi-projekti toteuttaen järjestelmään kurssikirjanpito-osion keväällä 2001. Korppi-projektin kanssa Kotka-tietokantaa kehitti myös Halko-niminen projekti Heinolan kansalaisopiston tarpeisiin. Korppi-sovellus korvasi käytössä olleen Kurki-järjestelmän, ja kesästä 2001 lähtien Korppi-sovellusta jatkokehitettiin järjestelmässä havaittujen puutteiden ja saadun palautteen perusteella. Saman vuoden syksyllä Kolibri-niminen projekti jatkoi järjestelmän kehittämistä toteuttamalla päivyri- ja ajanvarausosion.

Sittemmin WWW-pohjaista Korppi-opintotietojärjestelmää on kehittänyt puolisen tusinaa opiskelijaprojektia. Näistä Koppelo-projekti kehitti keväällä 2002 opinnäytteiden hallintasovelluksen. Kiuru-projekti jatkoi järjestelmän laajentamista syksyllä 2002 toteuttamalla salivaraussovelluksen. Keväällä 2003 Kottarainen-projekti kehitti järjestelmään kyselyjen laatimis- ja hallintasovelluksen, jota jatkokehitti syksyllä 2003 Koskikara-projekti. Seuraava Korppi-opintotietojärjestelmää kehittänyt projekti, Käki, suunnitteli ja kehitti syksyllä 2003 WWW-pohjaisen henkilökohtaisten opintosuunnitelmien (HOPS) laadinta-, seuranta- ja raportointisovellusta.

Kyyhky-projektin lisäksi Korppi-opintotietojärjestelmän kehittämiseen osallistui keväällä 2004 Kuikka-projekti, joka jatkoi syksyn 2003 Käki-projektin työtä HOPS-sovellukseen liittyen. Lisätietoa Korppi-opintotietojärjestelmästä löytyy osoitteesta <https://korppi.jyu.fi/kotka/help/tietoja.jsp>. Tietotekniikan opiskelijaprojekteista löytyy lisätietoa osoitteesta <http://www.mit.jyu.fi/palvelut/sovellusprojektit/>.

4 PROJEKTIN TAVOITTEET JA NIIDEN TOTEUTUMINEN

Luvussa kuvataan projektin tavoitteiden toteutuminen sovelluksen ja projektiryhmän jäsenten henkilökohtaisen oppimisen osalta. Projektin tavoitteita kuvataan tarkemmin Projektisuunnitelman luvussa 4.

4.1 Toteutettu sovellus

Kyyhky-projekti toteutti Korppi-opintotietojärjestelmään rajapinnan, jonka avulla erilaiset pöytäkoneiden, taskumikrojen ja matkapuhelinten kalenterisovellukset voidaan synkronoida Korpin kalenterin kanssa. Sovelluksen avulla Korpin kalenterin käyttöä saataneen tulevaisuudessa laajennettua muita sähköisiä kalentereita käyttävien opiskelijoiden ja henkilökuntaan kuuluvien keskuuteen.

Sovelluksen avulla henkilökuntaan kuuluva voi päivittää Korppi-järjestelmässä olevia ryhmä- ja opetustapahtumia henkilökohtaisesta kalenterista käsin. Lisäksi sekä opetushenkilökunta että opiskelijat voivat päivittää henkilökohtaisia kalenteritapahtumiaan muista sähköisistä kalentereista Korpin kalenteriin ja päinvastaiseen suuntaan.

Toteutettu sovellus koostuu kahdesta osasta: konfliktienhallintaosasta ja Sync4j SyncServeriin liittyvästä rajapinnasta. Sync4j-palvelimen myötä toteutetun rajapinnan avulla pystytään siirtämään kalenteritapahtumia henkilökohtaisesta synkronoitavasta kalenterista Korpin kalenteriin ja päinvastoin. Konfliktien hallintaosan avulla voidaan Korpissa käsitellä syntyneet konfliktit sekä määritellä synkronoitavat kalenterit.

Toteutettavan sovelluksen edellytettiin tarjoavan useita tietoja ja toimintoja, jotka esitellään tarkemmin Vaatimusmäärittelyssä. Projektin puitteissa ryhmä sai toteutetuksi suurimman osan vaatimuksista ja tuloksena oli sovellus, jolla voidaan synkronoida Korpin kalenteri ja Outlookin henkilökohtainen kalenteri.

Projektin puitteissa ei saatu toimimaan synkronointia matkapuhelimen tai muiden kämmenlaitteiden kalenterien ja Korppi-opintotietojärjestelmän kalenterin välillä. **Toteutumattomia ominaisuuksia** olivat ryhmätapahtumien käsittely, autentikointi, synkronoitavan tiedon luottamuksellinen synkronointi (SSL-salaus) sekä toistuvien tapahtumien ehjänä säilyminen.

Sovelluksen rakennetta ja siihen toteutettuja ominaisuuksia kuvataan tarkemmin Sovellusraportin luvuissa 3, 4 ja 5. Tarkemmin toteutetun sovelluksen virheitä, puutteita sekä jatkokehitykseen jääneitä ominaisuuksia kuvataan Sovellusraportin luvussa 9.

4.2 Oppiminen

Projektiryhmän yhtenä ensisijaisena oppimistavoitteena oli **projektimuotoisen työskentelyn oppiminen**. Projektin jäsenet kokevat saaneensa hyvin totuudenmukaisen kuvan hieman laajemman sovelluksen toteuttamisesta ja kaikista vaiheista, joita projekti pitää sisällään.

Toisena projektiryhmän päällimmäisenä oppimistavoitteena oli **uusien tekniikoiden ja sovellusten käytön sisäistäminen** sekä Java-kielen, SyncML-tekniikan ja tietokantojen suunnittelun ja toteutuksen oppiminen. Uusista tekniikoista, kuten Javasta, JSP:stä, sync4j-palvelimesta ja synkronoinnista opittiin projektin kuluessa aina vain lisää. Jokainen kokee sisäistäneensä kyseiset tekniikat siinä määrin, että pystyisi tarpeen mukaan opettamaan niitä joissain määrin myös kyseisiä tekniikoita vähän tuntevalle.

Ryhmän yksi oppimistavoitteista oli **dokumentointiin perehtyminen**. Projektin jälkeen kukin projektiryhmän jäsen hallitsee tieteellisen tekstin kirjoittamisen tapoja prosessikirjoittamisesta pöytäkirjan laadintaan. Lisäksi kukin ryhmän jäsen on oppinut tieteellisessä kirjoittamisessa käytettäviä käytänteitä, dokumenttien sisällön rakenteen huomioimista, ulkoasun muotoilua ja lähdeviittaustekniikkaa.

Kokouksissa toimimisen oppiminen oli myös yksi oppimistavoitteista. Kokouksissa toimimista tuli harjoiteltua 14 palaverin ajan, joten jokaiselle jäi selkeä kuva kokouksissa toimimisen periaatteista sekä puheenjohtajan ja sihteerin tehtävistä.

Yksi projektiryhmän oppimistavoitteista oli **työtehtävien jakamisen oppiminen**. Työtehtäviä pyrittiin jakamaan kunkin projektin jäsenten tasaisesti, missä ryhmä onnistui kohtuullisen hyvin. Kukin ryhmän jäsen tutustui kaikkiin projektiin sisältyneisiin tehtäväalueisiin.

Projektin aikana yhtenä oppimistavoitteena oli **vieraskieliseen materiaaliin tutustuminen**. Vieraskieliseen materiaaliin perehdyttiin projektin aikana suhteellisen paljon, ja projektin jäsenet oppivat etsimään niistä olennaisimmat asiat.

Myös **ihmissuhteisiin, vuorovaikutukseen sekä yhteistyöhön liittyvien taitojen kehittäminen** olivat oppimistavoitteita. Ryhmän sisällä ei suurempia ristiriitoja noussut esiin. Asiat selviteltiin puhumalla ja keskustelemalla rakentavasti. Yhteistyö paitsi ryhmän, myös muiden projektiorganisaatioon kuuluvien henkilöiden kesken sujui hyvin. Projektin jälkeen kukin ryhmän jäsen on saanut arvokasta kokemusta yhteistyön tekemisestä erilaisten ihmisten kanssa. Projektin jäsenten erilaiset taustat eivät vaikuttaneet kuitenkaan projektin läpivientiin mitenkään erityisesti.

Yksi oppimistavoite oli **oma-aloitteisten ongelmanratkaisutaitojen kehittäminen**. Projektin edetessä esiinnousseita teknisiä ongelmia pyrittiin ratkaisemaan ensin ryhmän jäsenten kesken, ja tarpeen tullen konsultoitiin myös asiantuntijoita. Ryhmän jäsenet perehtyivät ohjauksen puutteen vuoksi itsenäisesti ja oma-aloitteisesti projektipäällikön tehtävien hoitamiseen, sekä tutustuivat syvällisemmin ennalta tuntemattomiin tekniikoihin ja työtapoihin. Projektin aikana jokaisen ryhmän jäsenen itsenäinen työskentely ja asioihin perehtyminen kehittyi suuresti.

Projektin aikana opitut asiat helpottivat projektin läpivientiä ja yhteistyötä. Kaiken kaikkiaan projektin lopussa ryhmän jäsenillä on jokseenkin selkeä kuva projektin vaiheista, hallinnasta ja läpiviennistä. Lisäksi projektin kautta ryhmän jäsenet tutustuivat ja saivat kokemusta suuremman sovelluksen määrittelystä, suunnittelusta, toteutuksesta ja testauksesta sekä ohjelmoinnista ja ohjelmakoodin kunnollisesta kommentoinnista.

5 PROJEKTIN RESURSSIT JA ORGANISAATIO

Luvussa kuvataan projektin käytössä olleita resursseja sekä organisaatioon kuuluneet henkilöt ja heidän suhteensa projektiin.

5.1 Henkilöt ja yhteystiedot

Projektiryhmään kuuluivat tietotekniikan opiskelijat Timo Aarniovuori (tietotekniikka), Teija Alasalmi (englanti, tietotekniikka), Jaakko Hyvärinen (tietojärjestelmätiede) ja Maunu Tuomainen (matematiikka). Sovelluksen tilaajia olivat yliopiston organisaatioista tietotekniikan laitos edustajanaan Vesa Lappalainen sekä virtuaaliyliopistohanke, jota edusti Antti Auer.

Kyyhky-projektin vastaavana ohjaajana toimi Jukka-Pekka Santanen. Teknisenä ohjaajana toimi Tiina Pöyhönen. Myös seuraavia asiantuntijoita konsultoitiin projektin edetessä:

- Jani Kurhista SyncML-tekniikan osalta,
- Minna Hillebrandia ja Jonne Itkosta Java-kielen, JSP-tekniikan ja Kotka-tietokannan osalta,
- Heikki Kainulaista JBossin ja J2EE-arkkitehtuurin osalta,
- Pauli Kujalaa Kotka-tietokannan ja tietokantasuunnittelun osalta,
- Sami Murtolahtea Novellin GroupWise-kalenteriin liittyvissä asioissa,
- Mika Raentoa vapaan lähdekoodin SyncML-palvelimien käyttökokemusten osalta sekä
- Harri Tuomea Linuxiin liittyvissä asioissa.

Kyyhky-projektin dokumentaatio on luettavissa projektin kotisivuilla osoitteessa <http://sovellusprojektit.it.jyu.fi/kyyhky/>. Projektilla oli käytössä sähköpostilista kyyhky@korppi.jyu.fi, jolle lähetettyjä viestejä voi tarkastella WWW-osoitteessa <https://korppi.it.jyu.fi/list-archive/kyyhky/>.

5.2 Laitteet ja ohjelmistot

Projektiryhmä sai käyttöönsä tietotekniikan laitokselta Agorasta huoneen C222.2, jossa oli asennettuina kaksi Linux-konetta ja kaksi Windows XP -konetta. Linux-koneille oli asennettu Fedora Core 1 -levityspaketti kaikkine ohjelmineen, sekä JBuilder 9 Enterprise -sovelluskehitin. Windows-koneille oli asennettu MS Office 2000 -ohjelmistopaketti, MS Project sekä MS Visio.

Linux-koneisiin oli asennettu HTTP-palvelimeksi Apache 2.0.48, sekä Tomcatin versio 4.1 testausta varten. Lisäksi Linux-koneisiin oli asennettuina tietokannanhallintajärjestelmäksi PostgreSQL. Toiselle Linux-koneelle oli asennettuina Sync4j-palvelimen versio 4.0 sekä JBoss-sovelluspalvelimen versio 4.0.1. Samoilla Linux-koneilla oli joidenkin tietojen osalta siivottu versio Kotka-tietokannasta sekä Korppi-järjestelmän kehitysversio 2004.05.24.12.11. Projektiryhmä käytti projektin edetessä myös CVS-versionhallintajärjestelmän versiota 1.11 sekä tietokannanhallintaohjelmistoa nimeltä phpPgAdmin (versio 3.2.1).

6 PROJEKTIN TEHTÄVÄT JA NIIDEN JAKAUTUMINEN

Luvussa kuvataan toteutunutta työnjakoa projektin eri vaiheiden osalta.

6.1 Tehtäväalueet

Projektiin liittyviä tärkeimpiä tehtäväalueita olivat aiheeseen ja käytettäviin työkaluihin perehtyminen, dokumentointi, projektipäällikkyys, projektin WWW-sivujen ylläpito, vaatimusten määrittely, tietokanta-, Sync4j- ja käyttöliittymäsuunnittelu, ohjelmointi sekä testaus. Kokonaisuudessaan jokainen ryhmän jäsen osallistui kaikkiin edellä mainittuihin tehtäväalueisiin. Näin kaikki saivat arvokasta kokemusta ohjelmistoprosessin eri vaiheista ja niihin sisältyvistä tehtävistä.

6.2 Työnjako projektin alussa

Projektin alussa projektiryhmän jäsenet tutustuivat yleisesti aihealueeseen, SyncML- ja vCalendar-materiaaleihin sekä käytettäviin työkaluihin ja ohjelmistoihin. Vastuualueet jaettiin aiheeseen perehtymisvaiheessa siten, että projektipäällikkönä toimi Teija Alasalmi.

Ryhmän laatimien dokumenttien HTML-muotoon muuttamisesta sekä projektin WWW-sivujen luonnista ja päivityksestä vastasi Maunu Tuomainen. SyncML:ään ja erilaisiin palvelintoteutuksiin tarkemmin perehtyi Jaakko Hyvärinen.

Määrittelyvaiheessa tehtäväalueita tarkennettiin siten, että Hyvärinen ja Tuomainen laativat Vaatimusmäärittelyn. Korppi-kyselyn suunnittelusta ja toteutuksesta sekä Sovellussuunnitelmasta vastasi Timo Aarniovuori. Projektisuunnitelman laatimisesta vastasi Teija Alasalmi.

6.3 Suunnittelu- ja toteutusvaihe

Suunnittelu- ja toteutusvaiheen alussa tehtävät jaettiin projektijäsenten kesken vastuualueisiin siten, että kukin projektiryhmän jäsen perehtyi omaan osa-alueeseensa muita ryhmän jäseniä tarkemmin. Toteutusvaiheen aikana jaetut tehtävät säilyivät pääasiallisesti suunnitelmaa vastaavina.

Käytännössä toteutusvaiheessa projektiryhmä jakaantui kahteen osaan. Alasalmi ja Aarniovuori keskittyivät sovelluksen käyttöliittymän suunnitteluun, JSP-tekniikkaan sekä Korppi-järjestelmän tietokantaan. Lisäksi Aarniovuori laati Sovellussuunnitelman. Hyvärinen ja Tuomainen vastasivat Sync4j-palvelimeen liittyvästä suunnittelusta ja toteutuksesta. Tietokanta suunniteltiin kunkin ryhmän jäsenen voimin Minna Hillebrandin, Jonne Itkosen ja Pauli Kujalan erinomaisella avustuksella.

Jokaisen projektiryhmän jäsenen tuli olla perillä myös toisten tekemistä osuuksista ja laadituista dokumenteista. Aina ei toinen puolikas projektiryhmästä ollut kuitenkaan täysin selvillä toisen puolikkaan tekemistä osuuksista. Tämä johtui paljolti siitä, että käytettävissä oleva aika oli rajallinen.

Ohjelmointitehtävät pyrittiin jakamaan mahdollisimman tasapuolisesti kaikkien ryhmän jäsenten kesken, jotta jokainen saisi kokemusta ohjelmoimisesta. Tässä ryhmä onnistui erittäin hyvin, sillä jokainen sai rutkasti kokemusta ohjelmoimisesta etenkin, kun toteutusvaihe päästiin aloittamaan hiukan suunniteltua aikaisemmin. Lisäksi pareittain

ohjelmoitaessa oli omat etunsa, sillä näin ohjelmoitaessa välttyttiin tekemästä suurempia virheitä kooditasolla. Tällöin myös toteutettuja tietorakenteita mietittiin perusteellisesti.

6.4 Projektipäällikön tehtävät

Projektipäällikkönä toimi koko projektin ajan Teija Alasalmi. Projektipäällikkö laati projektisuunnitelman sekä yksityiskohtaisen aikataulun projektin läpiviennille. Projektipäällikkö arvioi käytettäviin tehtäviin kuluva työmäärää, suunnitteli projektiin liittyvien tehtävien työnjaon ja valvoi töiden edistymistä. Tarvittaessa projektipäällikkö tarkensi ja organisoi ryhmän jäsenten tehtäviä uudelleen. Projektipäällikön vastuulla oli myös projektipalavereiden esityslistojen laatiminen sekä asioista tiedottaminen projektiryhmälle ja -organisaatiolle.

Toteutusvaiheessa projektipäällikön olisi pitänyt keskittyä hieman tarkemmin projektipäällikön tehtäviin. Suuri osa ajasta kului kuitenkin ohjelmointiin, josta johtuen itse projektin hallinta jäi vähemmälle. Tämä oli varmasti yksi osasy s siihen, että projekti viivästyi.

Projektipäällikön tehtävien hoitoon ei saanut myöskään minkäänlaista ohjausta projektin aikana. Projektipäällikön kokemattomuudesta ja ohjauksen puutteesta johtuen tehtäviä ei aina osattu hoitaa oikeassa järjestyksessä.

6.5 Viimeistelyvaihe

Viimeistelyvaiheessa ohjelmakoodi sekä Sovellus- ja Projektiraportit viimeisteltiin luovutuskuntoon. Lisäksi Alasalmi analysoi Aarniovuoren laatiman Korppi-kyselyn, jonka tulokset ovat luettavissa erillisessä Korppi-kyselyraportissa.

Viimeistelyvaiheessa toteutettua sovellusta myös testattiin, koska erillinen testausvaihe jäi käytännössä puuttumaan ajanpuutteen vuoksi.

6.6 Dokumentointi

Projektiryhmä dokumentoi huolellisesti jokaisessa projektin vaiheessa. Pöytäkirjojen laatimisesta huolehti kukin ryhmän jäsen sihteerinvuoron osuessa kohdalle. Kaiken kaikkiaan dokumentointi jakaantui suhteellisen tasaisesti kunkin projektiryhmän jäsenen kesken. Koska projektilla ei ollut erillistä sihteeriä, jaettiin dokumentointi ryhmän jäsenten kesken kunkin vastuualueen mukaisesti.

Määrittely- ja suunnitteluvaiheiden dokumentointi jakautui siten, että Projektiraportista vastasi Alasalmi, Vaatimusmäärittelystä Hyvärinen ja Tuomainen sekä Sovellussuunnitelmasta Aarniovuori. Viimeistelyvaiheessa dokumentoinnista vastasivat Projektiraportin osalta yhdessä muiden jäsenten kanssa Alasalmi ja Sovellusraportin osalta Aarniovuori. Korppikyselyn tulokset raportoitiin erillisessä Kyselyraportissa.

Kaikki suuremmat dokumentit laadittiin käyttäen Kyyhky-projektin luomaa dokumenttipohjaa. Dokumentointi laadittiin MS Wordilla ja tallennettiin RTF-muotoon. Esityslistojen, pöytäkirjojen ja opponointiraporttien dokumentoinnissa käytettiin omaa asiakirjamallia, ja edellä mainituista dokumenteista tehtiin RTF- ja tekstimuotoiset versiot.

Projektiryhmä asetti laatimansa dokumentit projektin kotisivulle. Kotisivulla dokumentit ovat saatavissa RTF-muodossa (sekä Sovellussuunnitelma myös PDF-muodossa) osoitteessa <http://sovellusprojektit.it.jyu.fi/kyyhky/>. Dokumentit toimitettiin tilaajan edustajien ja ohjaajien kommentoitaviksi ja hyväksyttäväiksi.

6.7 Suunnittelun tehtäväjaon onnistuminen

Suunniteltu tehtäväjako onnistui ryhmän mielestä kohtuullisen hyvin. Projektin alussa työtehtävät pyrittiin jakamaan projektiryhmän jäsenten kiinnostuneisuuden sekä tieto- ja taitotasojen mukaan. Määrittelyvaiheen tehtäväalueiden tarkennus onnistui hyvin, koska jokainen ryhmän jäsen sai vastuulleen selkeän kokonaisuuden.

Suunnitteluvaiheessa tehtäviä tarkennettiin siten, että ryhmä jakaantui kahteen osaan. Projektipäällikön näkemyksen mukaan Hyvärisen ja Tuomaisen oli helpointa jatkaa kahdestaan Sync4j:hin liittyvän ohjelmoinnin toteutusta, koska he olivat perehtyneet vaatimuksia määriteltäessä Aarniovuorta ja Alasalmea tarkemmin Sync4j-palvelimen toimintaan. Aarniovuori ja Alasalmi puolestaan perehtyivät JSP-tekniikkaan ja Korppi-opintotietojärjestelmään sekä sen eri osiin, joten heidän oli helpointa jatkaa Korppi-opintotietojärjestelmään toteutetun konfliktienhallintaosan ohjelmoimista.

Toteutuneen kahtiajaon suurimpana syynä oli aiheen laajuus. Ryhmän kahtiajako osoittautui hyväksi valinnaksi, joskin ajanpuutteen vuoksi osapuolet eivät aina olleet täysin selvillä toistensa tekemisistä. Käytännössä kuitenkin ryhmän kahtiajako oli selkein tapa viedä suunnittelu- ja toteutusvaiheet läpi, koska vähäinen käytettävissä oleva aika rajoitti ryhmän jäsenten mahdollisuuksia perehtyä tasapuolisesti sync4j- ja konfliktienhallintaosatoteutuksen ohjelmoimiseen.

Projektin tavoitteena oli, että kaikki ryhmän jäsenet saisivat kokemusta kaikista tehtäväalueista. Tämä tavoite toteutui, sillä kaikki osallistuivat jokaiseen tehtäväalueeseen. Yksittäisten tehtävien suunnitellut ja todelliset vastuuhenkilöt on esitelty taulukossa 1.

Tehtävä	Suunniteltu vastuuhenkilö	Todellinen vastuuhenkilö
Projektin hallinta (projektipäällisyys)	Teija	Teija
Projektikansion ylläpito	Teija	Teija
WWW-sivujen päivitys	Maunu	Maunu
Sync4j-palvelin	Jaakko	Jaakko, Maunu
Korppi-kyselyn laatiminen ja yhteenveto	Timo	Timo, Teija
Perehdytykset	Kaikki	Kaikki
Aiheeseen tutustuminen	Kaikki	Kaikki
JBuilderiin ja CVS:ään tutustuminen	Kaikki	Kaikki
SyncML-tekniikkaan tutustuminen	Jaakko, Maunu	Kaikki
Korppiin sekä Kotka-tietokantaan tutustuminen	Kaikki	Kaikki
JSP-tekniikkaan tutustuminen	Timo, Teija	Timo, Teija
Viikoittaiset palaverit	Kaikki	Kaikki
Ryhmän sisäiset palaverit	Kaikki	Kaikki
Pöytäkirjat	Kaikki	Kaikki
Projektisuunnitelma	Teija	Teija
Vaatimusmäärittely	Jaakko, Maunu	Jaakko, Maunu
Sovellussuunnitelma	Timo	Timo
Testaussuunnitelma	Maunu	Ei toteutettu
Sync4j:hin liittyvä toteutus	Jaakko, Maunu	Jaakko, Maunu
Käyttöliittymään ja Korppiin liittyvä toteutus	Timo, Teija	Timo, Teija
Yksittäisten ohjelmanosien testaukset	Kaikki	Kaikki
Koko sovelluksen testaus	Kaikki	Jaakko, Maunu
Käyttöohje	Jaakko	Jaakko, Maunu
Sovellusraportti	Timo	Timo
Projektiraportti	Teija	Teija
Viimeistely	Kaikki	Kaikki
Loppuesittely ja sen valmistelu	Kaikki	Kaikki

Taulukko 1. Tehtävien suunnitellut ja todelliset vastuuhenkilöt.

7 PROJEKTIN AIKATAULU

Luvussa tarkastellaan projektille suunnitellun aikataulun toteutumista.

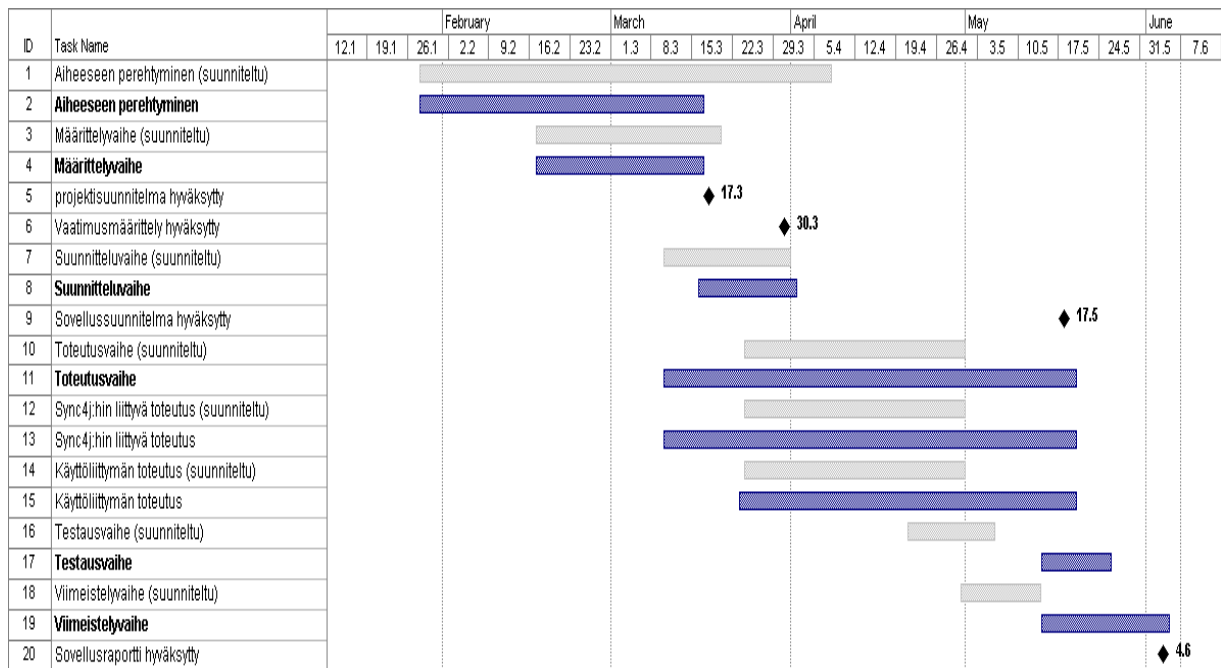
7.1 Aikataulun toteutuminen

Alkuperäisen suunnitelman mukaan projektin oli tarkoitus kestää arviolta 16 viikkoa. Todellisuudessa projekti kuitenkin venyi, ja sen kokonaiskesto oli noin 19 viikkoa. Projektiin käytettävissä oleva aika oli jaettu kuuteen osaan: aiheeseen perehtymiseen, määrittelyyn, suunnitteluun, toteutukseen, testaukseen ja viimeistelyyn. Taulukossa 2 esitetään projektin eri tarkastuspisteille suunniteltu ja toteutunut aikataulu.

Vaihe	Tulos	Toteutunut pvm	Suunniteltu pvm
Määrittely	Projektisuunnitelma	17.3.2004	19.3.2004
Määrittely	Vaatimusmäärittely	30.3.2004	19.3.2004
Suunnittelu	Sovellussuunnitelma	17.5.2004	7.4.2004
Viimeistely	Sovellusraportti	4.6.2004	14.5.2004
Viimeistely	Projektiraportti	15.6.2004	14.5.2004

Taulukko 2. Projektin eri tarkastuspisteiden toteutuminen.

Kuvassa 1 on esitetty projektin suunniteltu ja toteutunut aikataulu janakaaviona. Alkuperäinen suunniteltu aikataulu löytyy Projektisuunnitelman liitteestä. Kuvassa 1 esitetään projektille suunniteltujen vaiheiden kestot vaaleammalla ja toteutuneet kestot tummemmalla värillä.



Kuva 1. Suunniteltu ja toteutunut aikataulu.

Kuvasta 1 ja taulukosta 1 voidaan todeta, että projektin eri vaiheista aiheeseen tutustuminen, määrittely ja suunnittelu kestivät hieman odotettua vähemmän. Toteutusvaihe sen sijaan kesti huomattavasti suunniteltua kauemmin, mikä pitkitti testaus- ja viimeistelyvaiheiden aloittamista.

Suunnitteluvaiheesta on huomattava, että vaikka itse vaihe kestitkin odotettua vähemmän aikaa, oli sen tuloksena syntynyt Sovellussuunnitelma reilusti myöhässä. Syynä tähän oli toteutusvaiheen aloittaminen, jonka vuoksi aikaa dokumentoinnille ei ollut riittävästi.

7.2 Aikataulumuutosten pohdintaa

Projekti lähti hyvin käyntiin materiaaliin ja työkaluihin perehtymisellä. Alussa aikaa tosin veivät hyvin paljon CVS-versionhallinnan säätäminen ja Korppi-järjestelmän pystytys. Niiden käyttöohjeet eivät olleet ajan tasalla, eikä ryhmän jäsenillä ollut aiempaa kokemusta kyseisistä järjestelmistä. Lopulta päädyttiin siihen, että CVS-versionhallintaa käytettiin suhteellisen vähän.

Projekti oli tarkoitus saada päätökseen toukokuun puoleen väliin mennessä. Pelivaraa kuitenkin jätettiin varmuuden vuoksi toukokuun loppuun asti. Tämä osoittautui hyväksi valinnaksi, sillä projekti viivästyi parisen viikkoa. Projektin viivästymiseen vaikutti se, että toteutusvaiheen aikataulu oli hieman liian optimistinen. Lisäksi suunnitteluvaiheessa aloitettu Sovellussuunnitelma viivästyi yli kuukaudella, koska sitä ei jäädytetty tilaajan ja vastaavan ohjaajan toimesta riittävän ajoissa.

Projektin kuluessa huomattiin, että aihe oli oletettua selvästi laajempi. Sovelluksen toteutus viivästyi, sillä aiheeseen perehtyminen vei paljon aikaa. Tietotekniikan laitokselta ei löytynyt henkilöä, joka olisi tiennyt SyncML-tekniikasta sen verran, että olisi pystynyt ohjaamaan ja neuvomaan ryhmää ongelmia kohdattaessa.

Konfliktienhallintaosan toteutus viivästyi, koska käyttöliittymän toteutukselle varattuun aikataulu ei oltu sisällytetty Korpin koodiin ja muiden projektiryhmien koodiin tutustumiseen vievää aikaa. Lisäksi konfliktien käsittelyosaan tulevien osien ohjelmoiminen osoittautui odotettua työläämmäksi.

Sync4j-rajapinnan toteutus viivästyi, koska aihe oli selvästi odotettua laajempi ja siihen perehtyminen vei aikaa, jotta kaikki tarvittavat osat saatiin toteutettua. Asiantuntevan Sync4j:hin liittyvän ohjauksen puute vaikeutti ryhmän työtä, sillä ryhmä ei voinut kysyä keneltäkään neuvoa Sync4j:hin liittyvissä asioissa. Lisäksi valitun Sync4j-palvelimen versio 1.0 sisälsi paljon virheitä, eikä uusikaan versio toiminut aluksi toivotulla tavalla.

Myös JSP ja Korpin koodausstandardit sekä Sync4j:hin liittyvät tekniikat olivat ryhmän jäsenille uusia, joten niihin jouduttiin paneutumaan tarkemmin ennen varsinaisen ohjelmoinnin aloittamista. Aikataulun venymistä aiheutti myös Sync4j-palvelimen huono dokumentointi ja toimintaan liittyvät virheet. Ryhmä joutuikin selvittämään palvelimen toimintaa tarkoin, ennen kuin ohjelmointi pääsi hyvin käyntiin. Toteutusvaiheen viivästymisen vuoksi testausvaihe jäi puutteelliseksi, sillä viimeistelyvaiheen lähestyessä ryhmä kirjoitti aikaavievät Sovellus- ja Projektiraportit.

7.3 Työtunnit viikoittain ja tehtävätyypeittäin

Kyyhky-projektiryhmä käytti projektin aikaa yhteensä noin 1739 tuntia. Aarniovuorelle tunteja kertyi 441, Alasalmelle 454, Hyväriselle 430 ja Tuomaiselle 412. Taulukossa 3 kuvataan ryhmän jäsenten eri tehtäväalueisiin käytettyä kokonaisaikaa projektin osalta. Taulukossa 4 puolestaan kuvataan jäsenten kurssiin käyttämä kokonaisaika. Kokonaistuntimäärästä suurin osa käytettiin projektin läpiviemiseen, ja vain reilu 200 tuntia käytettiin projektin ohessa suoritettuun kurssiin.

Taulukoissa 3 ja 4 esitetyissä yhteenlasketuissa tuntimäärissä on pientä epätarkkuutta ajankäytön arvioinnin takia. Esimerkiksi kurssiin käytetyn dokumenttien kirjoitusasuun muokkaukseen kulunut aika oli ryhmän keskuudessa suurilta osin arvioitua, sillä oli mahdotonta tietää, kuinka paljon kirjoitusasuun kiinnitettiin tarkalleen huomiota dokumenttien kirjoittamisen ohessa. Lisäksi kirjauskäytännöt vaihtelivat ryhmän sisällä ainakin perehdytysten ja projektienhallinnan osalta.

Projektiin liittyvä tehtäväalue	Käytetty aika (h)
Suunnittelu	92
Dokumentointi	222
Määrittely	20
Ohjelmointi	629
Testaus	44
Materiaaliin tutustuminen	244
Palaverit	154
Projektienhallinta	57
Perehdytys	60
Yhteensä (h)	1522

Taulukko 3. Projektiin käytetyt työtunnit.

Kurssiin liittyvä tehtäväalue	Käytetty aika (h)
Ryhmätyöt	8
Luennot	67
Muut tehtävät	3
Dokumenttien kirjoitusasu	96
Palautteen läpikäyminen	6
Opponoinnit ja niiden valmistelu	37
Yhteensä (h)	217

Taulukko 4. Kurssiin käytetyt työtunnit.

Liitteessä 1 on kuvattu Kyyhky-projektiryhmän jäsenten yhteenlasketut työtunnit ja niiden jakautuminen viikoittain. Ryhmä käytti eniten aikaa ohjelmointiin, dokumentointiin ja materiaaliin tutustumiseen, koska nämä olivat suurimmat tehtäväalueet. Lisäksi materiaaliin tutustumisen vaatima työmäärä oli oletettua suurempi projektin aiheen laajuuden vuoksi. Vähiten aikaa käytettiin määrittelyyn ja testaukseen. Määrittelyyn käytettyjen työtuntien vähyys johtuu kirjaamistavasta. Testaukseen käytettiin vähän aikaa, sillä projektin lähestyessä loppuaan aikaa testaamiselle ei jäänyt viimeistelyn suorittamisen vuoksi. Viikoittaiset työtunnit vaihtelevat jonkun verran projektin aikana. Viikoilla 15 ja 16 tunteja on selvästi vähemmän pääsiäisloman vuoksi.

Liitteissä 2-5 on kuvattu ryhmän jäsenten henkilökohtaiset työtunnit ja niiden jakautuminen viikoittain sekä tehtävittäin. Työtunteja tehtävittäin kuvaavissa kaavioissa on eroavaisuuksia ryhmän jäsenten välillä. Tämä johtuu osittain siitä, että työtuntien merkkaukseen ei ollut yhtenäistä käytäntöä. Ryhmän jäsenet ovat saattaneet tehdä samaa asiaa, mutta saattaneet merkata suorittamansa tehtävän eri tehtäväryhmään. Tästä johtuu esimerkiksi materiaaliin tutustumisen ja perehdytysten tuntimäärissä näkyvät ryhmän jäsenten väliset erot.

Projektin aikana kaikki ryhmän jäsenet osallistuivat kaikkiin tehtäväalueisiin, mikä näkyy myös liitteiden 2-5 piirakkagraafeissa. Alasalmi tutustui muita vähemmän materiaaliin, koska projektipäällikön tehtävien hoito vei aikaa. Projektipäällikön tehtävien hoidosta johtuen Alasalmi käytti muita projektiryhmän jäseniä enemmän aikaa projektienhallintaan.

Liitteissä 1-5 esitetyistä piirakkagraafeista käy ilmi tehtäväjaosta johtuvat ohjelmoinnin ja dokumentoinnin työtuntien erot. Hyvärinen ja Tuomainen käyttivät ohjelmointiin enemmän aikaa kuin Aarniovuori ja Alasalmi, jotka puolestaan käyttivät työtunteja dokumentointiin ohjelmointia enemmän.

Määrittelyyn käytettiin projektin aikana aikaa suhteellisen vähän, sillä suurin osa määrittelystä kirjattiin dokumentointiin Vaatimusmäärittelyn kirjoittamisen muodossa. Suunnitteluun käytetyissä työtunneissa on eroja ryhmän jäsenten kesken. Tämä johtuu suurimmaksi osaksi kirjaamiseroista. Esimerkiksi Hyvärinen ja Tuomainen kirjasivat osan sync4j-kokeiluista suunnitteluun. Alasalmi puolestaan kirjasi suunnitteluun projektin aikataulun suunnittelua sekä kaavioiden piirtämistä, jotka selkeästi olisivat kuuluneet projektinhallintaan. Muuten jäsenet kirjasivat suunnitteluun lähinnä konfliktienhallintaosan, sync4j-rajapinnan ja tietokannan suunnittelut.

Perehdytykseen kirjattujen työtuntien erot johtuvat siitä, että Korpin pystytykseen käytetyt työtunnit on kirjattu osin perehdytykseen joidenkin jäsenten osalta. Lisäksi osa jäsenistä kirjasi perehdytykseen materiaaliin tutustumista. Kurssiin kirjattujen työtuntien erot johtuvat dokumenttien kirjoitusasuun kulutetusta tuntimäärästä, joka Aarniovuorella ja Alasalmella oli hieman korkeammat Hyväristä ja Tuomaista suuremman dokumentointivastuun vuoksi.

8 RISKIEN ARVIOINTI JA HALLINTA

Luvussa kuvataan projektin arvioituja riskejä sekä niiden toteutumista, välttämistä ja hallintaa.

8.1 Aikatauluun liittyvät riskit

Projektin suurin riski oli sovitussa aikataulussa pysyminen. Aikatauluun liittyviä riskejä pyrittiin välttämään siten, että projektin alussa aikataulu suunniteltiin huolellisesti, ottaen huomioon myös projektin mahdollinen viivästyminen sekä odottamattomat ongelmat.

Aikataulu kuitenkin petti sovelluksen valmistumisen ja testauksen osalta. Yksi syy tähän oli huonosti määritellyt tavoitteet projektin alussa. Lisäksi projektille suunniteltu aikataulu oli kohtuullisen tiukka. Etukäteen määrättyt välitavoitteet ja tarkastuspisteet auttoivat projektin alussa määrittelyvaiheessa aikataulussa pysymistä, mutta suunnittelu- ja toteutusvaiheessa aikataulu venyi ohjelmoinnin vuoksi. Ryhmän keskittyessä ohjelmointiin myöskään dokumenttien laatimiselle ei jäänyt riittävästi aikaa.

Yksi suurimmista syistä aikataulun venymiseen oli se, ettei projektin aiheen laajuutta ymmärretty tarpeeksi ajoissa. Sekä tilaaja, että ohjaajat aliarvioivat projektin aiheen laajuuden ja vaativuuden. Myöskään kaikilla ryhmän jäsenillä ei projektin alussa ollut tarpeeksi vahvaa käsitystä projektin vaativuudesta.

8.2 Vaatimusten määrittelyyn liittyvät riskit

Vaatimusten määrittely muodosti toisen riskitekijän. Riskiä pyrittiin välttämään siten, että toteutettavalta sovellukselta edellytettävät vaatimukset rajattiin riittävän hyvin. Vaatimusten määrittely ja rajaaminen veivät kuitenkin aikaa, koska kaikki erikoistapaukset (esimerkiksi ryhmä- tai toistuvat tapahtumat) oli otettava tarkoin huomioon. Vaatimukset priorisoitiin, jotta sovellus toteuttaisi ainakin välttämättömimmät ja tärkeimmät ominaisuudet. Tiukasta aikataulusta ja työmäärän rajallisuudesta johtuen kaikkia ideoita, ominaisuuksia ja

vaatimuksia ei kuitenkaan pystytty toteuttamaan tämän projektin puitteissa, joten sovellus ei kokonaisuudessaan vastaa määritettyjä vaatimuksia.

Myös tilaajan edustajien, ohjaajien ja projektin jäsenten eriävät näkemykset toteutettavasta järjestelmästä toivat mukanaan omat riskinsä. Riskiä pyrittiin pienentämään laatimalla sekä kattavat että selkeät pöytäkirjat ja dokumentit koko projektin ajan. Lisäksi projektipalavereissa pyrittiin keskustelemaan vaatimusten määrittelyyn liittyvistä seikoista mahdollisimman kattavasti. Onneksi tilaajat osallistuivat varsinkin alkuvaiheessa aktiivisesti palavereihin, joissa asiat saatiin puitua selviksi. Eri termeistä ja asioista käytiin osin kiivaitakin keskusteluita. Vaatimuksista päästiin kuitenkin kaikkia osapuolia tyydyttävään lopputulokseen riittävällä vaatimusmäärittelyjen läpikäynnillä ja tarkentamisella projektipalavereiden aikana.

8.3 Tekniset riskit

Käytettävään tekniikkaan liittyvät riskit muodostivat kolmannen suuren riskitekijän. Projektin kuluessa huomattiin, että aihe oli oletettua selvästi laajempi. Tietotekniikan laitoksella ja informaatioteknologian tiedekunnasta ei löytynyt henkilöitä, jotka olisivat tienneet tarpeeksi projektin aihealueesta ja tekniikasta pystyäkseen neuvomaan projektiryhmän jäseniä käytettyihin tekniikoihin liittyvissä kiperimmissä ongelmatilanteissa. Riskiä pyrittiin hallitsemaan perusteellisella ja monipuolisella aiheeseen perehtymisellä sekä oma-aloitteisella ja aktiivisella tiedonhankinnalla ennen varsinaisen sovelluksen suunnittelua ja toteuttamista sekä niiden aikana.

Toteutettavan sovelluksen testiversio oli määrä saada valmiiksi hyvissä ajoin, jotta testauksessa saatua palautetta ja kokemuksia olisi voitu hyödyntää sovelluksen viimeistelyssä. Sovellusta ei kuitenkaan saatu testauskuntoon tarpeeksi ajoissa, koska sovelluksen ensimmäinen kunnollinen versio valmistui pari viikkoa myöhässä aikataulustaan. Tästä syystä testausvaihe jäi suunniteltua lyhyemmäksi, eikä sovellusta saatu testattua tarpeeksi kattavasti.

Käytettävät työkalut olivat suurelta osin projektiryhmän jäsenille uusia, joten niihin tutustuminen vei oman aikansa. Riskiä pyrittiin pienentämään siten, että projektiryhmän jäsenet osallistuivat tietotekniikan laitoksen tarjoamiin monipuolisiin perehdyttämistilaisuuksiin. Myös eri alan teknisiä asiantuntijoita konsultoitiin projektin edetessä.

8.4 Muut riskit

Muihin riskeihin kuuluivat projektin rinnalla suoritettavat muut opinnot, sairastumiset, kokemuksen puute sekä ryhmädynamiikka. Projektin aikana ryhmän jäsenet eivät sairastuneet tai kokeneet muutoksia elämäntilanteessaan, joten niiltä osin ei aikataulua tarvinnut joustaa.

Projektiryhmän jäsenillä oli projektin lisäksi meneillään muita opintoja, joiden suorittaminen ei kuitenkaan aiheuttanut suurempia riskejä tai esteitä projektin etenemiselle ja läpiviennille. Henkilökohtaisten aikataulujen ja muiden menojen osalta kukin projektiryhmän jäsen pyrki joustamaan, ja jokainen ilmoitti suuremmista menoistaan myös muille.

Projektiryhmän jäsenten kokemuksen puute projektimuotoisesta työskentelystä, projektin läpiviennistä ja projektinhallinnasta ei vaikeuttanut projektin läpivientä, sillä jokainen oli valmis toimimaan yhteistyössä projektin parhaaksi ja ongelmien ratkaisemiseksi. Lisäksi ryhmän kesken pidettiin palavereita, joissa jaettiin tehtäviä, suunniteltiin tulevaa, ja keskusteltiin toteutettavasta sovelluksesta. Projektiryhmän keskinäinen vuorovaikutus, toisen kuunteleminen ja positiivinen kannustaminen auttoivat ryhmän (vähäisten) sisäisten ristiriitojen selvittelyssä ja yhteisymmärrykseen pääsemisessä.

8.5 Riskien vaikutus aikatauluun ja tavoitteisiin

Sovelluksen valmistuminen viivästyi pari viikkoa suunnitellusta. Tämä johtui siitä, että aihe oli huomattavasti ensivaikutelmaa laajempi. Lisäksi toteutusvaihe venähti liian pitkäksi. Aikataulun pettämiseen vaikuttivat kaikki toteutuneet riskit yhdessä.

Aikatauluongelmia pyrittiin ratkomaan neuvottelemalla projektipalavereissa projektin puitteissa toteutettavista ja jatkokehitykseen jäävistä ominaisuuksista tilaajien ja ohjaajien kanssa. Näin saatiin rajattua sovellus tämän projektin osalta järkeviin raameihin.

Jotta projekti ei olisi viivästynyt, olisi aikataulun toteutumista pitänyt valvoa paremmin projektipäällikön osalta. Myös jokaisen projektin jäsenen olisi pitänyt omalta osaltaan huolehtia tarkemmin siitä, että dokumentit ja muut vaaditut työt olisivat valmistuneet määräajassa. Suurimpaan osaan ongelmista oli kuitenkin syynä jokaisen ryhmän jäsenen kokemattomuus projektinhallinnan ja läpiviennin osalta.

9 HENKILÖKOHTAISET KOKEMUKSET PROJEKTISTA

Luvussa projektin jäsenet kuvailevat henkilökohtaisia kokemuksiaan kevään 2004 sovellusprojektista.

9.1 Timo Aarniovuori

Projekti meni osaltani melko hyvin, vaikka en ollut aivan tyytyväinen omaan työpanokseeni. Tunteja kertyi kyllä paljon, mutta mielestäni minun olisi pitänyt saada aikaan enemmän. Projekti oli työläs, jota osasin odottaakin. Itselleni projekti oli kuitenkin henkisesti raskas ja aiheutti stressiä melko paljon.

Oppimisen kannalta projekti oli hyvä. Ennen projektia en tiennyt esimerkiksi, mikä on Tomcat. Projekti oli mielestäni todella vaativa. Varsinkin alussa tilanne tuntui kummalliselta, kun kukaan ohjaajista ei tiennyt SyncML-tekniikasta yhtään mitään. Mitään erityisiä tavoitteita minulla ei projektin suhteen ollut, kunhan saisin sen suoritettua. Ryhmän asettamat tavoitteet projektin alussa saavutettiin hyvin.

Ryhmä toimi mielestäni hyvin ja kaikki tulivat toimeen keskenään. Ohjaajilta ja henkilökunnalta sai apua ja he olivat auttavaisia ja ystävällisiä.

Ohjelmistoprojektin vaiheet eivät itselleni tulleet kovin hyvin selville. Tämä varmaan johtui aiheesta, jossa kokoajan tuli muutoksia ja yllätyksiä. Samoin työkalujen kanssa tuli yllättäviä ongelmia, joita ei oikein kukaan osannut ratkaista. Mielestämme selvisimme kuitenkin hyvin vaativasta aiheesta.

9.2 Teija Alasalmi

Hakeuduin tietotekniikan Sovellusprojektiin, koska halusin kokea käytännössä, millaista olisi työskentely normaalia luentokurssia suuremman projektin puitteissa. Lisäksi olin jo aikaisemmin suunnitellut suorittavani kyseisen opintojakson, mutta vasta nyt keväällä 2004 löytyi sen suorittamiseen vaadittavaa aikaa. Koska olin viime vuodet keskittynyt englannin ja muiden kielten opiskeluun toisen pääaineeni puitteissa, ei tietotekninen osaamiseni ennen projektia ollut kovinkaan laaja. Lisäksi moni ohjelmointiin ja tietokantoihin liittyvä asia oli unohtunut. Nelisen vuotta sitten olin käynyt joitakin ohjelmointikursseja C++:lla sekä yhden Java-ohjelmointikurssin. Lisäksi aineopinnoista puuttui vielä jokunen kurssi. Asiat, jotka koin hallitsevani hyvin, olivat WWW-sivujen luonti ja päivittäminen sekä erilaisten ohjelmistojen käyttö.

Kevään 2004 alussa alkoi sitten se kauan odotettu Sovellusprojekti. Alussa jännitti, pystynkö toimimaan yhtenä ryhmän jäsenenä, vaikka tietotekniikan opinnot olivat pahasti kesken eikä ohjelmoinnistakaan ollut paljoa kokemusta. Ensimmäisellä projektiluennolla Sovellusprojektin laajuus ja siihen vaadittava työpanos alkoivat valjeta. Myös projektiryhmien kokoonpano selvisi. Minä osallistuisin Korppia kehittävään projektiin, ja aiheeseen liittyisi jotenkin myös matkapuhelimet ja niissä olevat sähköiset kalenterit.

Osaltani projekti alkoi aiheeseen perehtymisellä, sekä Java-kielen ja tietokantojen itsenäisellä kertaamisella. Alussa tuntui, etten ymmärtänyt aiheesta yhtään mitään, koko synkronointi oli suuri musta aukko. Enhän ollut aiemmin edes käyttänyt kännykkää, jossa olisi kalenteri. Tilannetta ei helpottanut yhtään se, että kaiken päässäni olevien asioiden sekamelskan keskellä minusta tuli projektipäällikkö.

Materiaaliin, taustakirjallisuuteen ja aikaisempien projektin tuotoksiin perehtymällä projektimme aihe alkoi itselleni vähitellen selkiintyä. Projektin kuluessa sen eri vaiheet selkiintyivät ja jäsenyivät yhä paremmin, kuten myös oma tehtäväni ja vastualueeni projektissa. Projektipäällikön toimen hoitamisen lisäksi vastasin muun muassa yhdessä Timon kanssa konfliktienhallintaosan suunnittelusta ja toteutuksesta.

Alussa projektipäällikön tehtävät tuntuivat vaikeilta. En oikein tiennyt, mitä kaikkea homma pitää sisällään, ja kuinka tehdä asioita heti alusta alkaen oikein. Mitään ohjausta projektipäällikön tehtävienhoitoon ei saanut, joten kaikki joutui opettelemaan kantapään kautta. Näin jälkempäin ajateltuna tekisin monia asioita toisin.

Opin sen, että projektipäällikön tehtävien hoito on suhteellisen vaikeaa, vaativaa ja ajoitin myös stressaavaa. Asioita ilman aikaisempaa kokemusta saati kunnan ohjausta voi oikein hoitaa niin, että kaikki asiat tulisi otettua huomioon. Kaikesta huolimatta sain projektin aikana erinomaista kokemusta projektipäällikkyydestä. Virheistä ja onnistuneista ratkaisuista opin paljon.

Kokonaisuudessaan Sovellusprojekti yllätti paitsi työmäärän laajuudessaan, myös siinä, ettei se ollutkaan pelkkää ohjelmointia. Siihen sisältyi myös kaikkea muutakin: kokouksissa istumista, laajaa etukäteissuunnittelua, asioihin ja aiheeseen perehtymistä, ryhmätyötä, esittelyjä ja opponointeja sekä dokumentointia ja testausta. Projektin lähestyessä loppuaan voin todeta, että vaikka opintojaksojakso on rankka ja aikaavievä, on se myös kurssien Ohjelmointi++, Graafisten käyttöliittymien ohjelmointi ja Virtuaalisten oppimisympäristöt ohella yksi tähänastisesti käymistäni tietotekniikan opintojaksoista hyödyllisin.

Olen erittäin tyytyväinen ryhmämme työmoraaliin ja sisäisiin suhteisiin. Kaikki tekivät hommansa tunnollisesti. Suuri kiitos kuuluu ryhmäni jäsenille paitsi loistavasti sujuneesta yhteistyöstä, myös erinomaisesti tehdystä työstä projektin hyväksi.

9.3 Jaakko Hyvärinen

Opiskelen tietojenkäsittelytieteiden laitoksella ja siellä ohjelmistotekniikan linjalla. Linjamme kurssivalikoima on suhteellisen pieni, ja näin ollen suurin osa linjamme opiskelijoista ottaa luonnollisesti kurseja myös ”alakerran” laitoksen tarjonnasta. Ystäväni Jari Axenin kanssa päätimme viime syksynä hakeutua Lappalaisen ohjelmointikurssien innoittamana tietotekniikan laitoksen projektiopintoihin. Olimme halukkaita kasvattamaan ohjelmointiosaamistamme.

Ennen joulua 2003 kävimme neuvottelemassa asiasta tietotekniikan laitoksen projekteista vastaavan opettajan, Jukka-Pekka Santasen, kanssa. Santanen lupasi harkita meidän ottamista projektiin, mutta silmin nähden hieman epäilevästi. Olisimme kuulemma ensimmäiset ”yläkerran” laitoksen opiskelijat, jotka projektin tulisivat suorittamaan.

Joulun jälkeen saimme kuulla päässeemme projektiin. Tosin toivomuksistamme poiketen pääsimme eri ryhmiin. Minut oli sijoitettu projektiryhmään, jonka tarkoituksena oli toteuttaa Korpin kalenteriin synkronointirajapinta. Ensimmäinen ajatukseni tästä oli: ”Mitä ihmettä? Rajapinnan toteutuksenhan tuli olla jo sovittu minulle ja Axenille erikoistyöksi.” Lappalaisen kanssa olimme jo syksyllä sopineet, että minä ja Axen toteutamme ko. rajapinnan erikoistyönä. Nyt olinkin toteuttamassa sitä projektityönä täysin vieraiden ihmisten kanssa (joka ei ollut yhtään huono ratkaisu). Perusteluina minun sijoittamiseksi kalenterirajapintaryhmään johtui kiinnostuksestani mobiilialueeseen. Erikoistyöstä ei tämän jälkeen enää puhuttu.

Työskentely käynnistyi projektimme osalta ripeästi. Minun tehtäväni oli kartoittaa mahdollisia toteutusratkaisuja. Ymmärsin Santasen puheista, että kaikki projekteille suunnatut aiheet ovat tarkoin pureksittuja ja mietittyjä. Aiheemme kohdalla kyseenalaistan tämän. Mielestäni olin ainoa, joka oli aiheeseen etukäteen tutustunut. Tutustumisen aiheeseen tein miettiessäni erikoistyölle järkevää toteutusta. Tosin olin itsekkin ”speksailnut” aihetta vain teknisten toteutusratkaisujen osalta, en oppimistavoitteiden. Tilaaajilla tai ohjaajilla ei ollut aavistustakaan kuinka tällainen synkronointi tulee toteuttaa. Ainut

konkreettinen saamme apu oli Lappalaisen jo syksyllä mainitsema SyncML. Tämä oli kuulemma joku sähköisiin kalentereihin liittyvä ”juttu”.

Kaksi ensimmäistä viikkoa kulutin aiheen sulattelemiseen, ja speksailin mielessäni erilaisia ratkaisumalleja toiminnan käynnistämiseksi. Tarpeeksi kauan mietittyäni sain aiheesta käsitteellisellä tasolla riittävän kuvan. Ajatukseni tyrmättiin aluksi huonoina, koska joidenkin Korppi-kehittäjien ajatusmaailmaan ei mahtunut se, että Kotka-tietokantaa käsittelee Korpin kannalta ulkoinen toimija. En luovuttanut, vaan käännyin Lappalaisen puoleen. Vesa hyväksyikin idean toteuttamiskelpoisena, kuultuaan perusteluni.

Projektimme jakaantui selkeästi kahteen toisistaan erilliseen pienryhmään. Teija ja Timo toteuttivat Korppiin synkronointiin liittyvää käyttöliittymää. Minä ja Maunu toteutimme Sync4j:in liittyvää toteutusta. Sync4j:n liittyvä ohjelmointi oli selkeästi vaikeaa. Tämän vaikeuden aiheutti ohjelmakoodin puutteellinen kommentointi, monimutkaisuus ja dokumentaation vaillinaisuus. Oman vaikeutensa tuotti myös J2EE-tekniikkaan liittyvät seikat. Asioiden selvittäminen oli aikaa vievää ja välillä jopa turhauttavaa. Pienen ja vähäpätöisen asian selvittämiseen saattoi kuluu helposti viikko.

Työskentelyni Maunun kanssa oli antoisaa ja hedelmällistä. Agorassa viettämästämme ohjelmointiin liittyvästä yhteisajasta suurin osa oli pariohjelmointia. Tulin toki muunkin ryhmän kanssa tutuiksi, ja juttua tuntui välillä riittävän.

Viikkojen kuluessa Sync4j:hin liittyvä tekniikka ja toteutusratkaisut alkoivat selvitä. Projektimme osoitti minulle hyvin sen, että asiat, joita ei osaa yhtään, on mahdollista ottaa haltuun, jos on vain tarpeeksi sitkeä ja tarmokas. Asiat, joiden luulee olevan vähäpätöisiä, ovatkin niitä jotka vievät eniten aikaa.

Projekti varmisti minulle tosiasian, jonka paikkansapitävyyttä olen usein pohtinut. Ohjelmoijan työstä loppujen lopuksi ohjelmakoodirivien määrä on häviävän pieni suhteessa käytettyihin työtunteihin. Käytetystä ajasta suurin osa kuluu muuhun kuin varsinaiseen ohjelmointiin.

Sovellusprojektien osalta en voi ymmärtää, miten tietyt asiat ovat niin hankalia toteuttaa. Esimerkiksi, miten voi olla mahdollista, että Korppi-projektin onnistuneeseen kääntämiseen menee projektiryhmältä kolme viikkoa, vaikka tekninen tuki on paikalla. Tästä johtuen osa projektin jäsenistä joutuu ohjelmoimaan vain yhdellä työasemalla, koska ainoastaan siinä Korppi kääntyy. Lisäksi tehtäessä kehitystyötä, on mielestäni kohtuutonta asettaa niin natsimaista tietoturvapoliittikkaa työasemiin, että se ajaa ihmiset työn vaikeuttamisen vuoksi käyttämään omia kannettavia tietokoneitaan.

Huolimatta kohtaamistamme eriasteisista hankaluuksista, Sovellusprojekti on antoisa kokemus ja onnistunut tavoitteissaan.

9.4 Maunu Tuomainen

Osallistuin Sovellusprojektiin tavoitteenani saada tuntumaa siihen, mitä ohjelmistojen tuottaminen todella on. Aiemmat tiedot pohjautuivat lähinnä luennolla kuultuun, kirjasta luettuun ja aiempien kurssien harjoitustöistä muodostuneisiin (harha)kuvitelmiin.

Päällimmäisenä ajatuksena projektista on sen suuri työmäärä. Tottuneena korkeintaan parin luennon päivätahtiin tuntuivat yli 30-tuntiset työviikot varsin raskailta. Kuitenkin työ oli mielenkiintoista, joten pidemmätkin työpäivät sujuivat ongelmitta. Oppimista tapahtui mielestäni jatkuvasti ja enemmän kuin normaaleilla opintojaksoilla. Yksittäiset tiiviit luennot eri aiheista, joita projektin oheiskurssi tarjosi, toivat tehokkaasti hyödyllistä lisätietoa. Jäin kuitenkin kaipaamaan vielä perehdytystä esimerkiksi yksikkötestaukseen ja sen työkaluihin.

Haasteita projektin aikana tuli vastaan enimmäkseen itselle aivan uuden SyncML-tekniikan ja Sync4j-palvelinohjelmiston muodoissa. Palvelimeen tutustuminen, sen toiminnan ihmetteleminen ja ohjelmakoodiin perehtyminen vei valtavan siivun aikaa. Toisaalta myös JBuilder-kehitysympäristö ja CVS oikkuineen sekä erityisesti edellä mainittujen kummallinen yhteistyö ja puutteellinen ohjeistus Korpin ajoon saivat repimään hiuksia päästä. Kaikesta kuitenkin selvittiin, joskin tehokasta työaikaa paloi mielestäni liiankin

paljon. Jossain vaiheessa projektia siirryin itse JBuilderin ja sen CVS-kalun käytöstä Eclipse-ympäristöön ja CVS:n komentorivikäyttöön, joilla ongelmat suurimmaksi osaksi katosivat.

Projektin vaatimusmäärittelyvaihe oli verrattain helppo. Valmiin sovelluksen oleelliset vaatimukset pystyi tiivistämään yhteen lauseeseen, joskin tietysti yksityiskohtien kanssa sai hieman painiskella. Nämä selvisivät tilaajan kanssa keskustelemalla hyvin.

Suunnitteluvaiheessa huomasin ainakin tietokantoihin liittyvät taitoni hieman puutteellisiksi. Tietokantaratkaisujen pätkäily olikin haastavaa ja mielenkiintoista työtä ja luulen tuntevani aiheen merkittävästi paremmin kuin ennen projektia. Synkronointipalvelimen toimintaan liittyvät suunnitteluratkaisut tulivat lähes kaikki saneltua suoraan valmiin palvelintoteutuksen toimesta, joten oma työ tältä osalta jäi vähäiseksi.

Itse toteutusvaihe - koodaus - oli ajallisesti merkittävin vaihe projektissa. Java oli kielenä ennestään tuttu, joten alkuun päästiin mukavasti. Ongelmia tuottivat lähinnä puutteellisesti dokumentoidut ratkaisut synkronointipalvelimen ohjelmakoodissa. Projektiryhmä jakautui kahteen osaan. Timo ja Teija toteuttivat Korpin konfliktienhallintaan liittyvää käyttöliittymäosiota sekä minä ja Jaakko keskityimme palvelimen ja Korpin toiminnan yhteen liittämiseen. Ohjelmoinnin teimme lähes kokonaan parityöskentelynä, jonka havaitsin erinomaiseksi työskentelymuodoksi. Hieman työn nopeus kaksin tehdessä kärsii kahteen itsenäisesti työskentelevään verrattuna, mutta toisaalta virheiden määrä tippuu merkittävästi kahden silmäparin tarkastellessa jatkuvasti työn tuloksia.

Yksi parhaista projektin anneista oli ohjelmakoodin katselmointi. Tuskin koskaan olen muutaman tunnin aikana oppinut yhtä paljon uutta kuin katselmointitilaisuudessa. Suuri kiitos Jonnelle panoksestaan tähän liittyen.

Projekti kokonaisuutena oli mielestäni onnistunut, joskin aika loppui kesken. Jos meillä olisi ollut muutama viikko lisää työskentelyaikaa, niin sovellus olisi ehkä pystytty viimeistelemään ja testaamaan tuotantokuntoon. Minulle jäi epäselväksi, miksi opintojaksoa

ei aloitettu heti vuoden alusta, jolloin olisi ollut mahdollisuus joko hieman väljempään työtahtiin tai pieneen lisäpanostukseen sovelluksen valmistumiseksi.

10 YHTEENVETO

Kyyhky-projekti toteutti synkronointirajapinnan Jyväskylän yliopiston tietotekniikan laitoksella toteutettuun Korppi-nimiseen opintotietojärjestelmään. Rajapinnan avulla pystytään synkronoimaan kalenteritapahtumia Korppi-järjestelmän kalenteriosion sekä erilaisten mikrojen, matkapuhelimien, taskumikrojen ja muiden kämmenlaitteiden kalenterien kesken. Projektiraportissa kuvataan projektin läpivientiä ja tavoitteiden toteutumista.

Sovelluksen prototyyppi myöhästyi suunnitellusta aikataulusta kaksi viikkoa, eikä täyttänyt kaikkia sille alun perin asetettuja vaatimuksia. Jatkokehitykseen jäivät käyttäjän autentikointi ja konfliktien käsittely Sync4j-palvelimen puolelta. Korppi-opintotietojärjestelmään luodun konfliktienhallintaosan osalta jatkokehitykseen jäi ryhmätapahtumien ristiriitojen ratkaisu.

Projektin läpivienti oli todella vaativaa työtä, sekä se vei odotettua enemmän aikaa ja resursseja. Projekti myöhästyi kaksi viikkoa suunnitellusta aikataulusta kahdesta syystä. Ensinnäkin suurempien dokumenttien hiominen vei odotettua kauemmin aikaa, ja toiseksi ohjelmointi ei ollut aivan niin yksinkertaista, kuin aluksi luulimme. Myös aiheeseen perehtyminen vei paljon aikaa. Kaikesta huolimatta projektiryhmä teki parhaansa ja pääsi omasta mielestään hyvään lopputulokseen, joka toivottavasti tyydyttää myös tilaajan edustajia.

Kokonaisuutena projekti oli tarkoituksensa mukainen. Projektiryhmän jäsenet oppivat sovellusprojektin aikana monia hyödyllisiä taitoja, ja saivat ennen kaikkea kokemusta projektimuotoisesta työskentelystä.

LÄHTEET

- [1] Fornari Stefano, "Sync4j Coding Standards", saatavilla HTML-muodossa <URL: <http://sync4j.funambol.com/project/development/sync4j-coding-standards.html>>, Sync4j, 2002.
- [2] Gamma E., Helm R., Johnson R., and Vlissides J., "Design Patterns - Elements of Reusable Object-Oriented Software", Addison-Wesley, 1995.
- [3] Haikala Ilkka ja Märijärvi Jukka, "Ohjelmistotuotanto", Satku, 2000.
- [4] Heinonen Petri ja Lahtonen Tommi, "Tietoverkot työvälineenä 2001", Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [5] Hillebrand Minna, "Korppi, Jsp ja Java", perehdytyksen materiaali, Jyväskylän yliopisto, tietotekniikan laitos 23.2.2004.
- [6] Hillebrand Minna, Itkonen Jonne, Kujala Pauli, Lahtonen Tommi, Lappalainen Vesa ja Nurminen Miika, "Korpin koodausstandardi, ohjeita ja vinkkejä", Jyväskylän yliopisto, Tietotekniikan laitos, 2003.
- [7] Hillebrand Minna, Silván Markus, Vanhanen Antti ja Ylitalo Marko, "Koppelo-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [8] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika, "Kiuru-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2002.
- [9] Horppu Ismo, Mielityinen Markku ja Vire Markku, "Kotka-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2000.
- [10] Huttunen Sami, Lamminmäki Tatu, Lappi Juha ja Pelkkikangas Eija, "Käki-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2003.
- [11] Internet Mail Consortium, "vCalendar V1.0 Specification", saatavilla PS-muodossa <URL: <http://www.imc.org/pdi/vcal-10.ps>>, viitattu 26.3.2004.
- [12] Jaakkola Mia, Juutinen Sanna, Lupari Matti ja Nieminen Mikko, "Kolibri-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2001.
- [13] Koikkalainen Pasi, "Ohjelmistotuotanto 2000 -kurssimateriaali", saatavilla HTML-muodossa <URL: <http://erin.it.jyu.fi/pako/kurssit/ot2000/>>, Jyväskylän yliopisto, tietotekniikan laitos, 2000, viitattu 19.3.2004.
- [14] Kosonen Pekka, "WWW-sovellusten rakentaminen Javalla", erikoistyö, saatavilla HTML-muodossa <URL:

<http://www.mit.jyu.fi/luk/toteutettuja/WWWsovelluksiaJavalla/>,
 Jyväskylän yliopisto, tietotekniikan laitos, 2003.

[15] Lesonen Minna, Pekkanen Hannu, Tawast Tuukka ja Uuksulainen Heikki, "Korppi-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2001.

[16] Lintunen Sampsa, Pöyhönen Tiina, Vähä-Ruka Teemu ja Ylönen Timo, "Kottarainen-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2003.

[17] Lipitsäinen, Arvo "JSP – JavaServer Pages", saatavilla HTML-muodossa <URL: <http://myy.helia.fi/~atka2d/jsp/jsp.html>>, Helsingin Liiketalouden Ammattikorkeakoulu, viitattu 11.3.2004.

[18] MS Project 98 projektinhallinta, Teknolit, 1999.

[19] Open Mobile Alliance, "SyncML Sync Protocol Specification, version 1.1", saatavilla PDF-muodossa <URL: http://www.openmobilealliance.org/syncml/docs/syncml_sync_protocol_v11_20020215.pdf>, viitattu 19.3.2004.

[20] Paadar Juha Pekka, "SyncML:n käyttö henkilökohtaisen tiedon synkronointiin", pro gradu-tutkielma, saatavilla PDF-muodossa <URL: <http://www.cc.jyu.fi/~ppaadar/gradu.pdf>>, Jyväskylän yliopisto, tietotekniikan laitos, 2003.

[21] Santanen Jukka-Pekka, "Opinnäytteiden kirjoittaminen, lyhyt oppimäärä", saatavilla HTML-muodossa <URL: <http://www.mit.jyu.fi/opetus/opinnayte/kirjoittamisesta.html>>, Jyväskylän yliopisto, tietotekniikan laitos, 2000.

[22] Santanen Jukka-Pekka, "Tietotekniikan Sovellusprojektien ohje", saatavilla HTML-muodossa <URL: <http://www.mit.jyu.fi/opetus/sovellusprojektit/projohje.html>>, Jyväskylän yliopisto, tietotekniikan laitos, 2004.

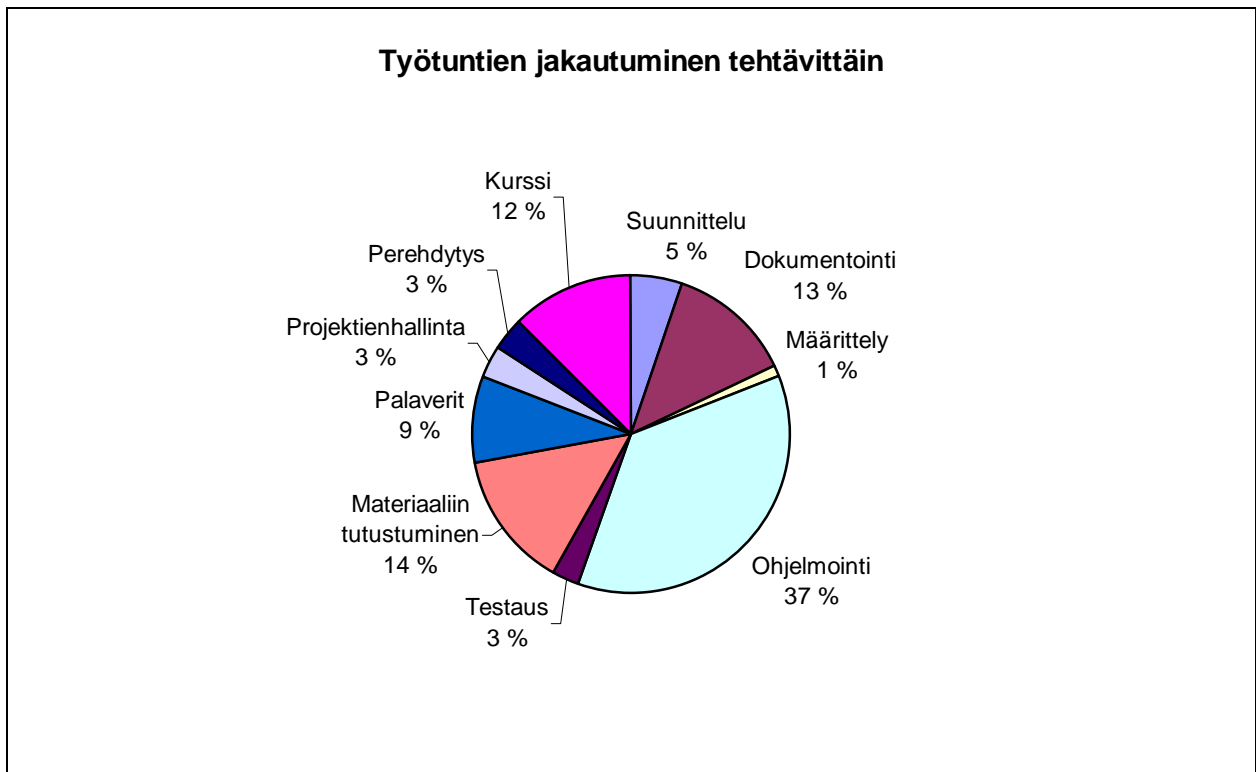
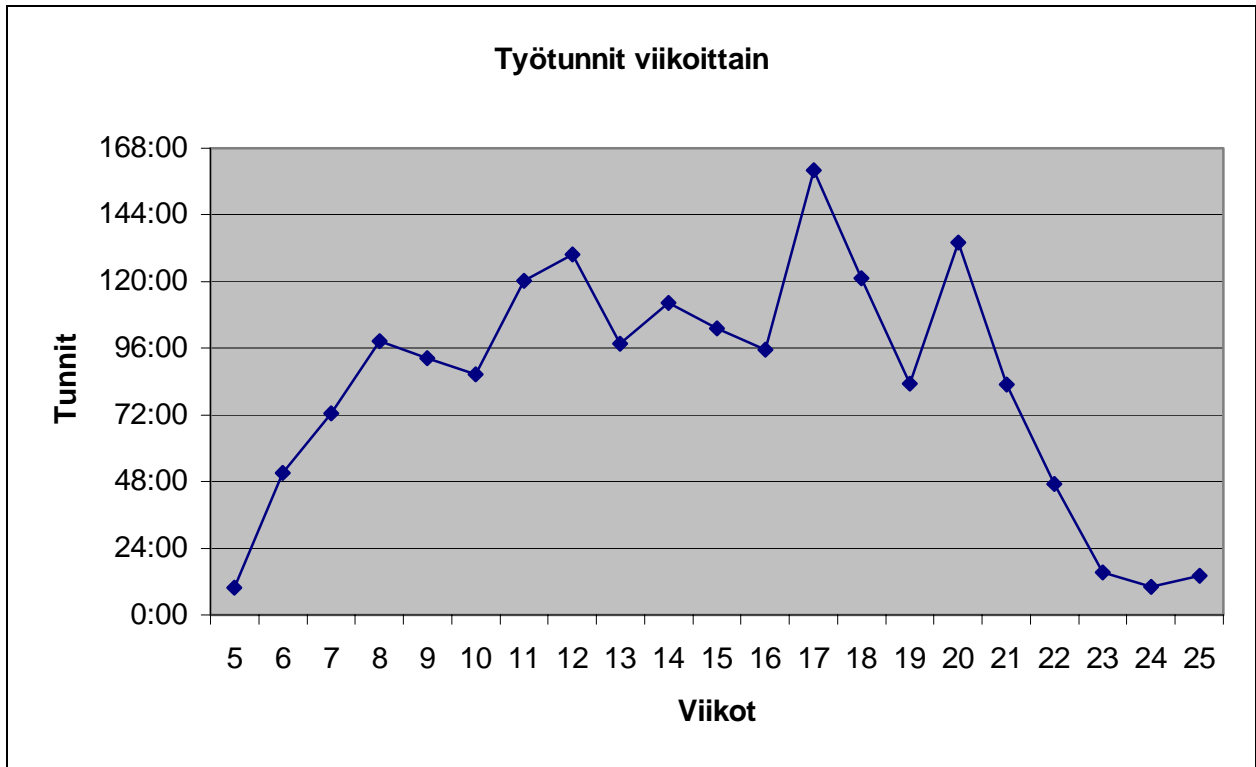
[23] Vahvanen Arto, "Henkilökohtainen mobiilikalenteri", pro gradu-tutkielma, saatavilla PDF-muodossa <URL: http://www.cs.uta.fi/research/theses/masters/Vahvanen_Arto.pdf>, Tampereen yliopisto, tietojenkäsittelytieteiden laitos, 2002.

[24] Videnoja Mika, "Ssh-yhteyden ottaminen etäkoneeseen käyttäen DSA Public Key autentikointia", saatavilla HTML-muodossa <URL: <http://www.mit.jyu.fi/atktuki/ssh.html>>, Jyväskylän yliopisto, 2004.

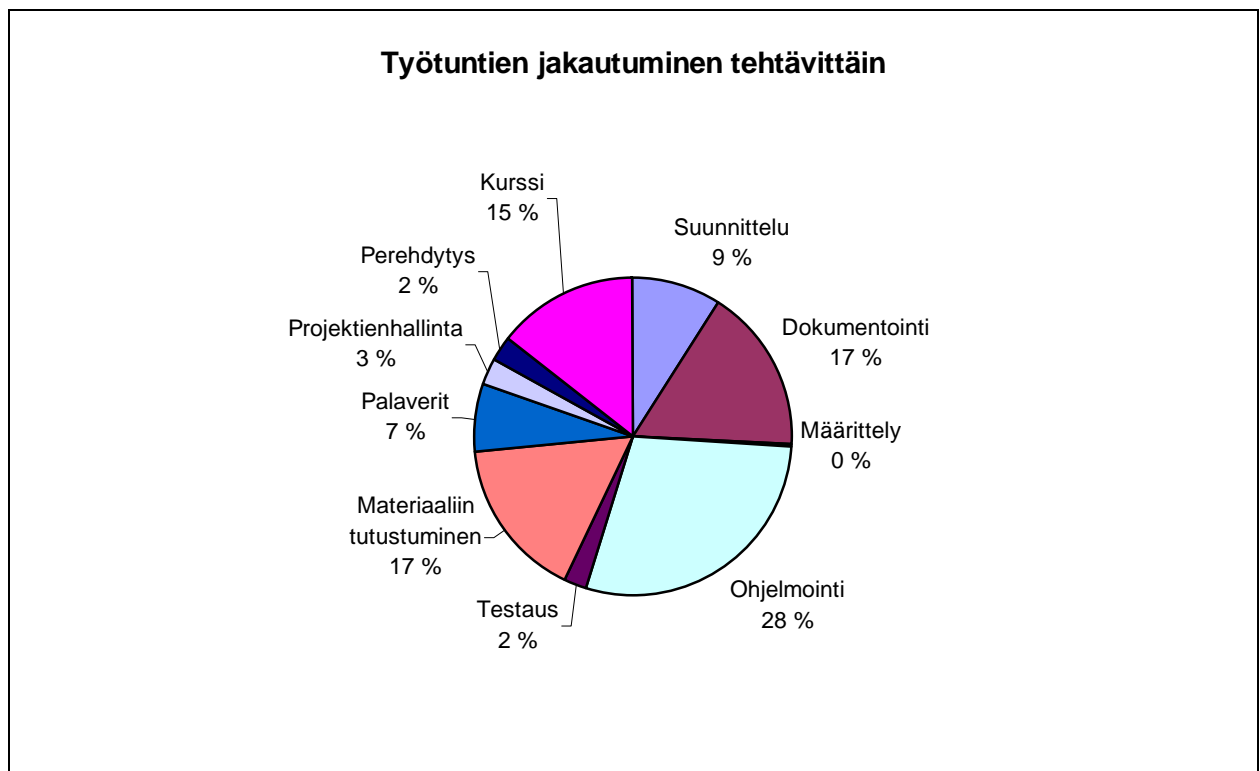
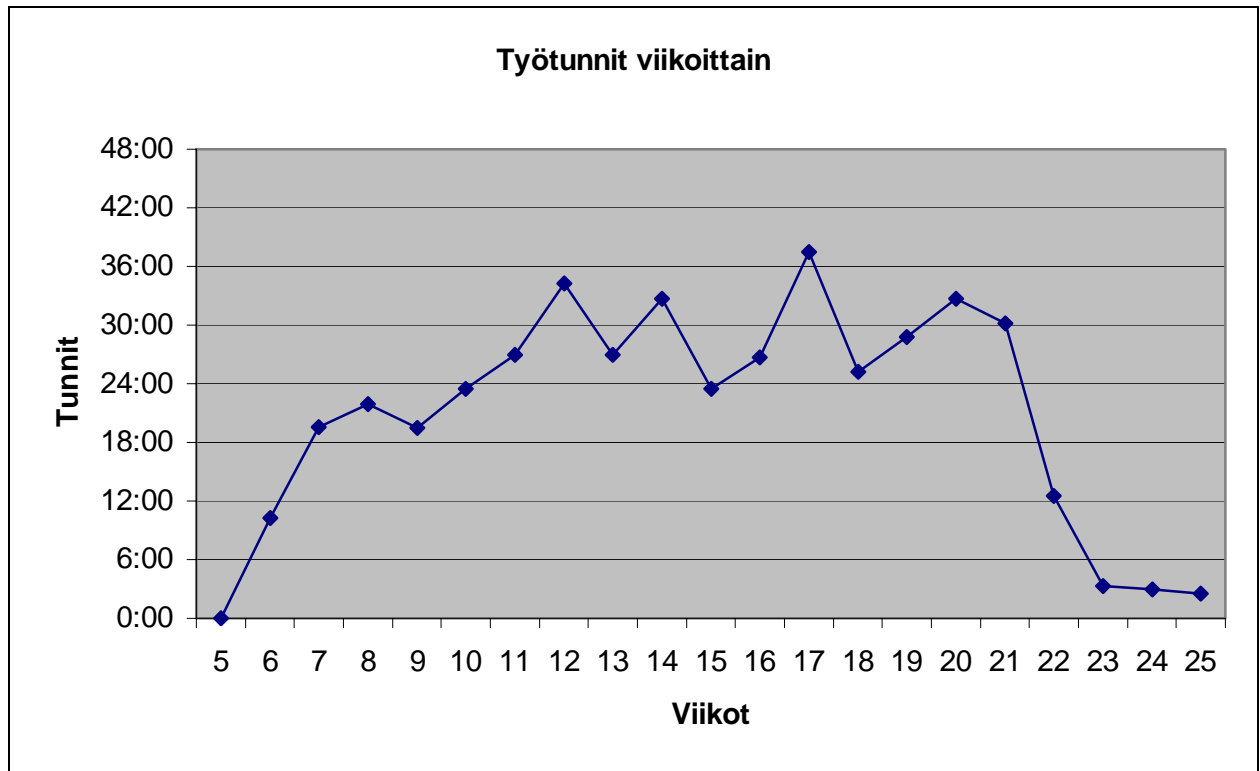
LIITTEET

Liitteessä 1 kuvataan projektiin käytetyt projektiryhmän jäsenten yhteenlasketut työtunnit viikoittain ja tehtävittäin. Liitteissä 2-5 on kuvattu jokaisen projektiryhmän jäsenen henkilökohtaiset työtunnit viikoittain ja tehtävittäin.

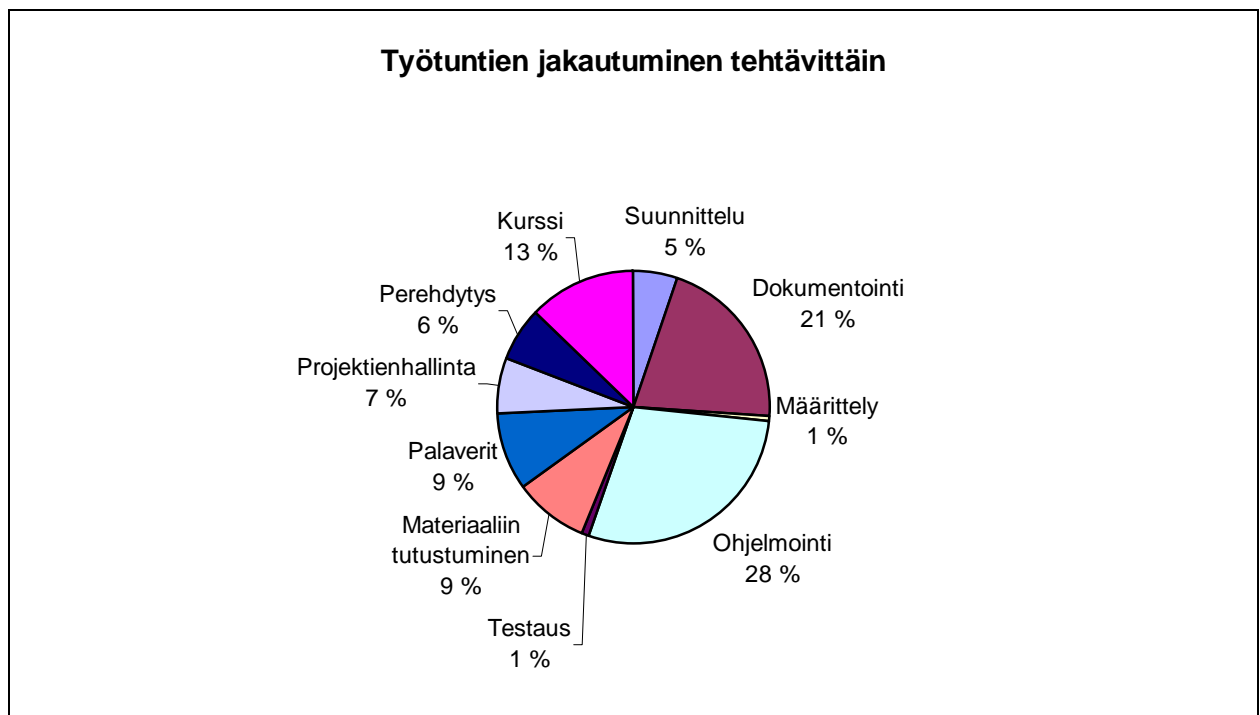
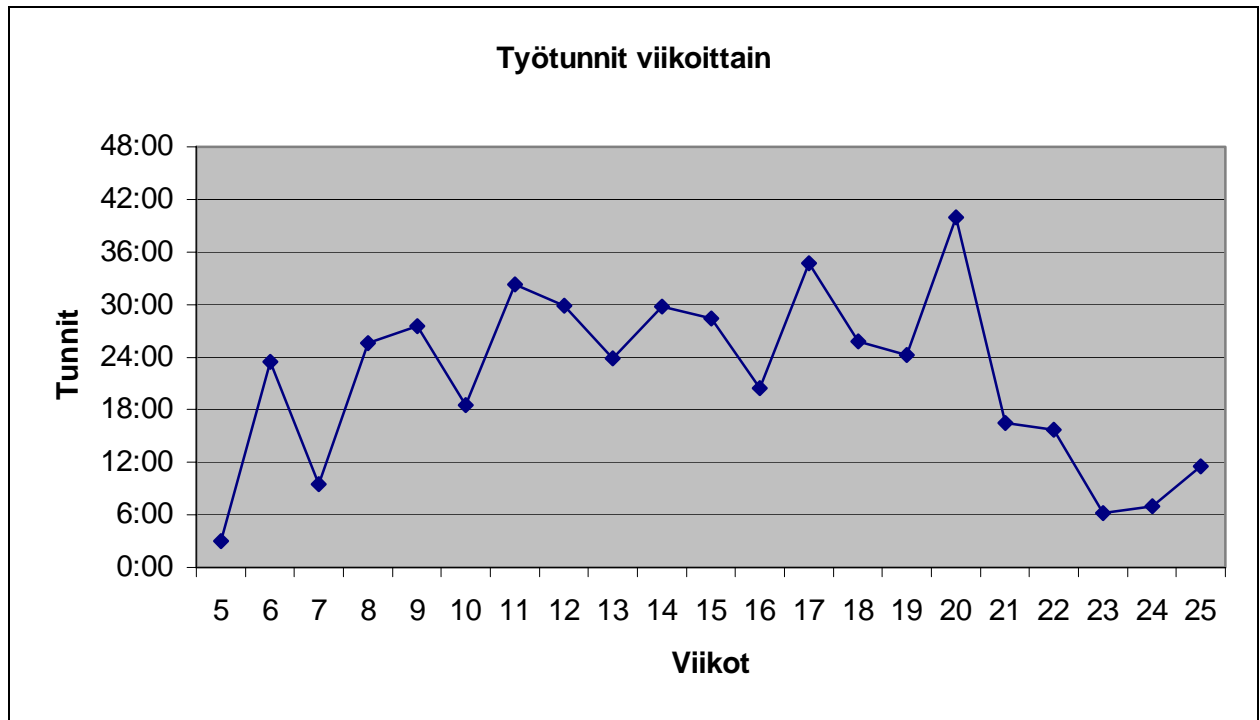
Liite 1. Kyyhky-ryhmän yhteenlasketut työtunnit viikoittain ja tehtävittäin



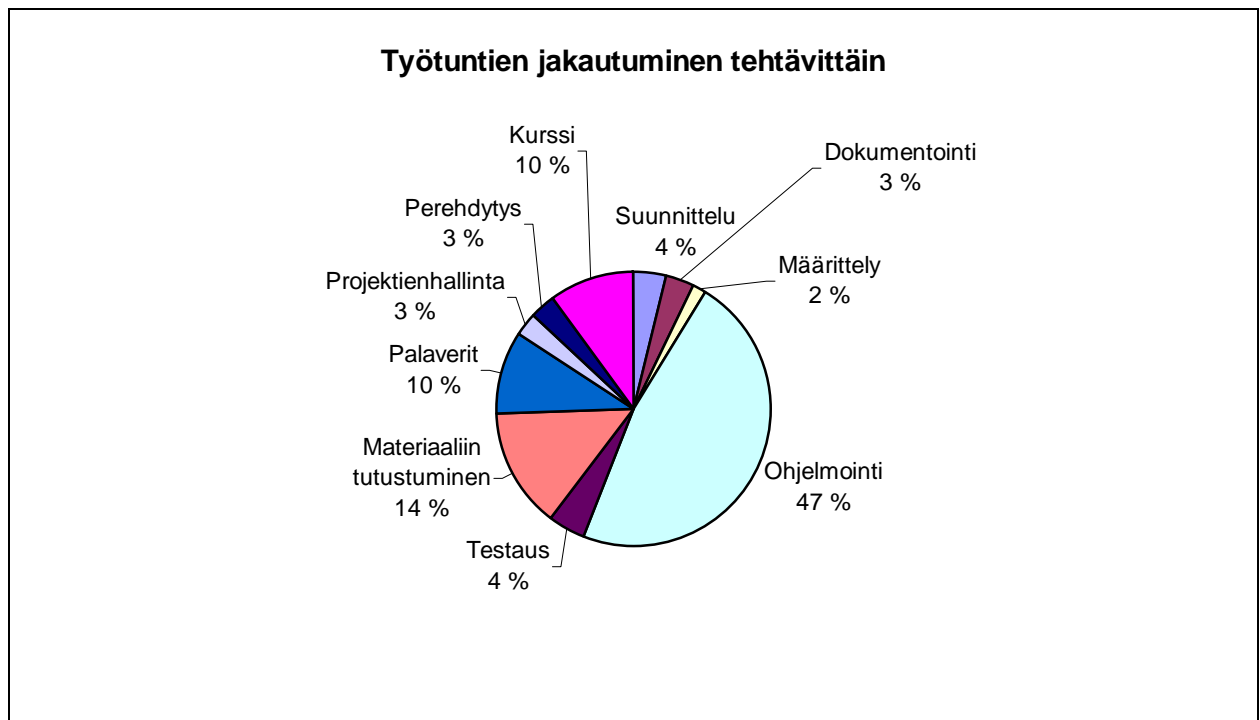
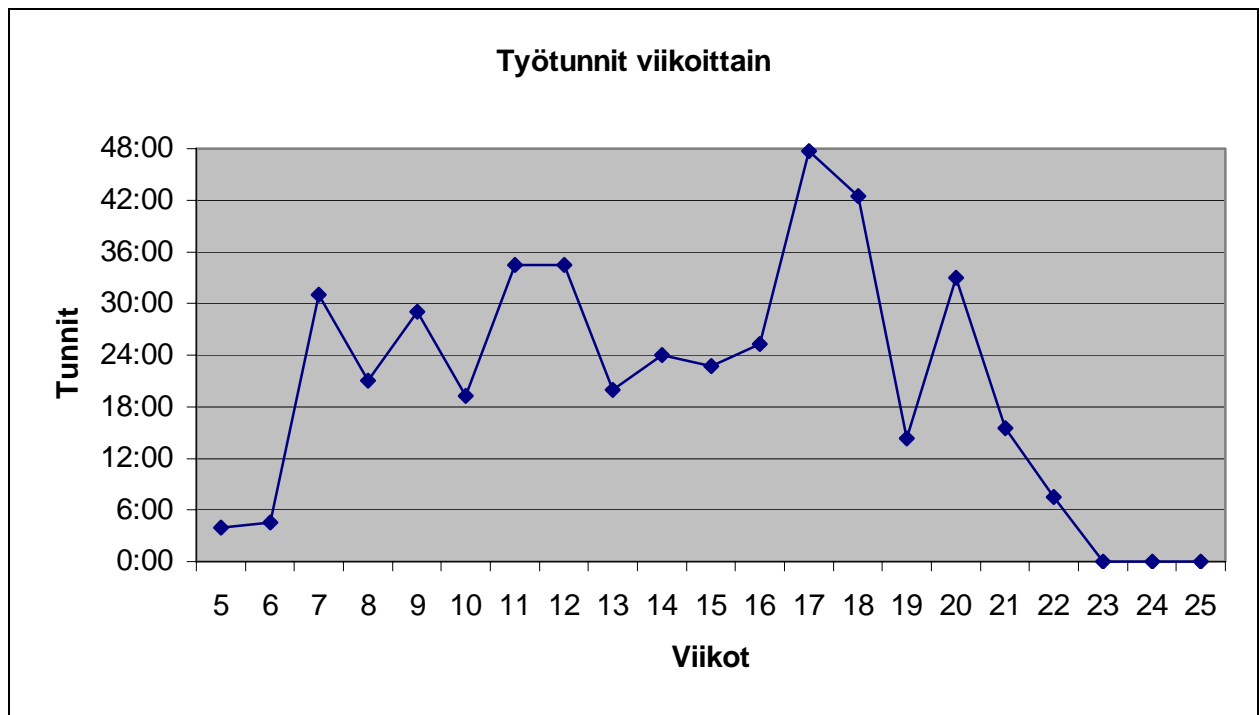
Liite 2. Timo Aarniovuoren työtunnit viikoittain ja tehtävittäin



Liite 3. Teija Alasalmen työtunnit viikoittain ja tehtävittäin



Liite 4. Jaakko Hyvärisen työtunnit viikoittain ja tehtävittäin



Liite 5. Maunu Tuomaisen työtunnit viikoittain ja tehtävittäin

