

# KYYHKY-PROJEKTI

Timo Aarniovuori  
Teija Alasalmi  
Jaakko Hyvärinen  
Maunu Tuomainen

Projektisuunnitelma  
22.3.2004, 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:**

Kyyhky-projektin projektitila sijaitsee Agorassa, huoneessa C222.2. Huoneen puhelinnumero on 014-260 4963. 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, projektisuunnitelma

**Työ:** Projektisuunnitelma tietotekniikan Sovellusprojektiin

### **Tiivistelmä:**

Kyyhky-projekti toteuttaa päivyrirajapinnan 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. Projektisuunnitelmassa kuvataan projektin taustoja ja tavoitteita, projektiorganisaatiota, tehtävien työnjakoa sekä projektiin liittyviä riskejä ja suunniteltua aikataulua.

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

## VERSIOHISTORIA

| Versio | Päiväys   | Tehnyt            | Muutokset  |
|--------|-----------|-------------------|--|
| 0.1    | 21.2.2004 | Teija<br>Alasalmi | Ensimmäinen versio.  |
| 0.2    | 28.2.2004 | Teija<br>Alasalmi | Lisätty luku 8 sekä muutamia termejä lukuun 2, täydennystä lukuun 9.   |
| 0.3    | 2.3.2004  | Teija<br>Alasalmi | Ulkoasun muokkaus yhteneväksi muiden dokumenttien kanssa.  |
| 0.4    | 8.3.2004  | Teija<br>Alasalmi | Korjaukset Santasen kommenttien pohjalta (lisäykset johdantoon, termilistaukseen, tehtäviin, aikatauluun ja yhteenvetoon). |
| 0.5    | 9.3.2004  | Teija<br>Alasalmi | Lisäyksiä oppimistavoitteisiin, tehtäviin ja aikatauluun.  |
| 0.6    | 10.3.2004 | Teija<br>Alasalmi | Lisätty tehtävien tarkempi kuvaus sekä työmäärätaulukko lukuun 6. Lisätty aikataulukkaavio liitteeksi.                     |
| 0.7    | 15.3.2004 | Teija<br>Alasalmi | Kirjoitusvirheiden ym. korjaus Santasen kommenttien pohjalta.  |
| 1.0    | 22.3.2004 | Teija<br>Alasalmi | Viimeistelyä.  |

## SISÄLLYSLUETTELO

|           |  |           |
|-----------|--|-----------|
| <b>1</b>  | <b>JOHDANTO</b> .....  | <b>1</b>  |
| <b>2</b>  | <b>TERMEJÄ</b> .....   | <b>2</b>  |
| 2.1       | Korppi-järjestelmään liittyvät termit .....  | 2         |
| 2.2       | Käytettäviin työkaluihin, ohjelmistoihin sekä ohjelmointikieliin liittyvät termit..... | 2         |
| 2.3       | SyncML -tekniikkaan liittyvät termit.....  | 4         |
| <b>3</b>  | <b>PROJEKTIN TAUSTAA</b> .....   | <b>6</b>  |
| 3.1       | Sähköiset kalenterit .....   | 6         |
| 3.2       | Korppi-järjestelmä .....   | 7         |
| <b>4</b>  | <b>PROJEKTIN TAVOITTEET</b> .....  | <b>8</b>  |
| 4.1       | Tavoitteet sovelluksen osalta.....   | 8         |
| 4.2       | Oppimistavoitteet.....   | 9         |
| <b>5</b>  | <b>PROJEKTIN RESURSSIT JA ORGANISAATIO</b> .....                                       | <b>11</b> |
| 5.1       | Henkilöt .....   | 11        |
| 5.2       | Tilat, laitteet ja ohjelmistot .....   | 11        |
| <b>6</b>  | <b>PROJEKTIN JÄSENTEN TEHTÄVÄT JA NIIDEN JAKAUTUMINEN</b> .....                        | <b>12</b> |
| 6.1       | Projektipäällikkö.....   | 12        |
| 6.2       | Tehtäväalueet .....  | 12        |
| 6.3       | Tehtävien jakautuminen.....  | 14        |
| <b>7</b>  | <b>PROJEKTIN AIKATAULU</b> .....   | <b>15</b> |
| 7.1       | Projektin vaiheet .....  | 15        |
| 7.2       | Projektin aikataulu .....  | 16        |
| 7.3       | Projektin välitavoitteet.....  | 18        |
| <b>8</b>  | <b>RISKIEN ARVIOINTI JA HALLINTA</b> .....   | <b>18</b> |
| 8.1       | Aikatauluun liittyvät riskit .....   | 18        |
| 8.2       | Vaatimusten määrittelyyn liittyvät riskit.....   | 19        |
| 8.3       | Tekniset riskit.....   | 19        |
| 8.4       | Muut riskit .....  | 20        |
| <b>9</b>  | <b>DOKUMENTOINTI JA KOKOUSKÄYTÄNNÖT</b> .....  | <b>21</b> |
| 9.1       | Dokumentointikäytännöt .....   | 21        |
| 9.2       | Lähdekoodin kommentointi ja muuttujien nimeäminen.....                                 | 22        |
| 9.3       | Kokouskäytännöt .....  | 22        |
| <b>10</b> | <b>YHTEENVETO</b> .....  | <b>23</b> |
|           | <b>LÄHTEET</b> .....   | <b>25</b> |
|           | <b>LIITE. AIKATAULU JANAKAAVIONA</b> .....   | <b>26</b> |



## 1 JOHDANTO

Nykyään monilla ihmisillä on vanhojen paperikalentereiden ohella käytössään monenlaisia sähköisiä kalentereita. Sähköisiä kalentereita löytyy esimerkiksi taskumikroista, kämmenlaitteista, matkapuhelimista ja verkosta, ja hyvin usein niistä löytyy paperikalentereita tehokkaammat mahdollisuudet kalenteritietojen muokkaukseen, kalenteritapahtumien lisäämiseen ja poistamiseen. Lisäksi uudet, monipuoliset tiedonsiirtotekniikat tarjoavat kehittyneempiä tapoja siirtää ja päivittää tietoa sähköisten kalenterien välillä.

Kyyhky-projekti on Jyväskylän yliopiston tietotekniikan laitoksella keväällä 2004 toteutettava Sovellusprojekti. Projekti toteuttaa Korppi-nimiseen opintotietojärjestelmään päiväkirjapinnan, jonka avulla erilaisten mikrojen, matkapuhelinten ja kämmenlaitteiden kalenterisovellukset voidaan synkronoida Korpin kalenterin kanssa.

Dokumentissa kuvataan Kyyhky-projektin taustaa, tavoitteet, resurssit, tehtäviä ja niiden jakoa, aikataulua, riskejä sekä dokumentaatio- ja kokouskäytänteitä. Toteutettavalta sovellukselta vaadittavat toiminnalliset ja tekniset ominaisuudet on kuvattu erillisessä Vaatimusmäärittelyssä. Yksityiskohtaisempi suunnitelma itse sovelluksen rakenteesta ja toteutuksesta käy ilmi Sovellussuunnitelmasta.

Luvussa 2 esitellään aiheeseen oleellisesti liittyviä käsitteitä. Luvuissa 3 ja 4 kuvataan tarkemmin projektin taustaa ja tavoitteita. Luvussa 5 kuvataan projektin organisaatio ja resurssit. Luvussa 6 käsitellään projektin tehtäviä sekä niiden jakautumista projektin jäsenten kesken. Luku 7 kuvaa projektin aikataulusuunnitelmaa ja luvussa 8 arvioidaan projektiin liittyviä riskejä. Projektin dokumentointi- ja kokouskäytänteitä kuvataan luvussa 9.

## 2 TERMEJÄ

Luvussa kuvataan Kyyhky-projektin dokumentoinnissa käytettävää termistöä.

### 2.1 Korppi-järjestelmään liittyvät termit

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

|                |  |
|----------------|--|
| <b>Kiuru</b>   | toteutti Korppi-järjestelmään salivaraussovelluksen.               |
| <b>Korppi</b>  | on Jyväskylän yliopistossa kehitetty opintotietojärjestelmä.       |
| <b>Kotka</b>   | on Korppi-järjestelmän tietokanta ja henkilötietojen hallintaosio. |
| <b>Kolibri</b> | toteutti Korppi-järjestelmään päivyri- ja ajanvarausosion.         |

### 2.2 Käytettäviin työkaluihin, ohjelmistoihin sekä ohjelmointikieliin liittyvät termit

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

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



|                   |   |
|-------------------|---|
| <b>Servletti</b>  | on palvelimella sijaitseva sovelma (engl. <i>applet</i> ), joka toteuttaa HTTP-palvelimen pyynnöstä tietyn toiminnon. |
| <b>SQL</b>        | eli Structured Query Language on tietokantojen hallintaan kehitetty standardoitu kieli.                               |
| <b>Tietokanta</b> | on kokoelma loogisesti yhteenliittyvää tietoa.  |
| <b>XML</b>        | on rakenteinen dokumenttien määrittely- ja kuvauskieli.   |

Projektissa käytettävään kehitysympäristöön ja työkaluihin liittyviä termejä ovat seuraavat:

|                   |  |
|-------------------|--|
| <b>Apache</b>     | on vapaan lähdekoodin HTTP-palvelinohjelmisto.   |
| <b>CVS</b>        | (Concurrent Version System) on ohjelmistokehityksessä käytettävä versionhallintajärjestelmä.   |
| <b>HTTP</b>       | (HyperText Transfer Protocol) WWW-tekniikassa käytettävä tiedonsiirtoprotokolla, jolla selain pyytää haluttuja sivuja WWW-palvelimelta.  |
| <b>JDBC</b>       | eli Java Database Connectivity on Java-tekniikan käyttämä tietosilta erilaisiin tietokantoihin.  |
| <b>J2EE</b>       | (Java 2 Enterprise Edition) on standardi, joka mahdollistaa palvelinsovellusten tuottamisen niin, että sovelluksia voi ajaa millä tahansa J2EE-palvelimella.   |
| <b>Palvelin</b>   | on WWW-sovellusten tapauksessa ohjelmisto, joka palvelee asiakkaana toimivien selainten hakupyynnöitä.   |
| <b>PostgreSQL</b> | on vapaa relaatiotietokannan hallintajärjestelmä.  |
| <b>Protokolla</b> | eli yhteiskäytäntö on sovittu liikennöintitapa, jota noudattaen laitteet tai ohjelmat voivat vaihtaa tietoa keskenään. Protokolla määrittelee siirrettävän tiedon esitystavan sekä mekanismin, jolla tieto siirretään. |
| <b>Skriptaus</b>  | tarkoittaa WWW-sivujen tapauksessa ohjelmakoodin kirjoittamista HTML-dokumenttien sisään. Palvelinpuolen skriptauksessa koodi ajetaan palvelinkoneessa ja asiakaspään skriptauksessa selaimessa.                       |

|                                   |  |
|-----------------------------------|--|
| <b>Selain</b>                     | on ohjelma, joka käyttäjän koneella tulkkaa HTML-kieliset sivut kuvaruudulla esitettävään muotoon.   |
| <b>SSL</b>                        | (Secure Sockets Layer) on protokolla, jonka yli voidaan muodostaa yhteyksiä muilla protokollilla, kuten esimerkiksi HTTP:llä. Yhteys on salattu. |
| <b>Tomcat</b>                     | on vapaan lähdekoodin servletti- ja JSP-moottori.  |
| <b>Versionhallintajärjestelmä</b> | 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:

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

|                                    |   |
|------------------------------------|---|
| <b>MD5</b>                         | (Message digest 5) on funktio, joka laskee annetusta syötteestä vasteen. Alkuperäisen syötteen palauttaminen vasteesta on mahdotonta.   |
| <b>SyncML</b>                      | (Synchronization Markup Language) on synkronointiprotokolla henkilökohtaisen tiedon synkronointiin.   |
| <b>Synkronointi</b>                | (engl. <i>synchronization</i> ) tarkoittaa saman tiedon pitämistä ajantasaisena kahden tai useamman tietokannan välillä.  |
| <b>Synkronointiankkuri</b>         | (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> . |
| <b>Sync4j</b>                      | on Java-ohjelmointikielellä toteutettu vapaan lähdekoodin SyncML-palvelinohjelmisto.  |
| <b>Työpöytäsynkronointi</b>        | (engl. <i>desktop synchronization</i> ) on työpöytäsovelluksen ja mobiililaitteen välillä tapahtuvaa synkronointia.   |
| <b>URI</b>                         | ( <i>Uniform Resource Identifier</i> ) on universaali tapa yksilöidä resursseja.  |
| <b>vCalendar</b>                   | on laitteisto- ja tiedonsiirtoriippumaton tiedostoformaatti kalenteritiedon tallentamiseen ja välittämiseen.  |
| <b>vCard</b>                       | on formaatti henkilöön liittyvien yhteystietojen tallentamiseen ja välittämiseen.   |
| <b>Virkistysynkronointi</b>        | (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.   |
| <b>Yksisuuntainen synkronointi</b> | (engl. <i>one-way synchronization</i> ) on synkronointimenetelmä, jossa muuttuneet tiedot lähetetään  |

yhteen suuntaan, joko asiakkaalta palvelimelle tai palvelimelta asiakkaalle. Vastaanottaja ei lähetä omia muuttuneita tietojaan.

### **3 PROJEKTIN TAUSTAA**

Luvussa esitellään projektin taustaa.

#### **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 monenlaisia 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 kalentereiden ohella opiskelijalla tai henkilökuntaan kuuluvalla saattaa olla käytössään myös Korppi-järjestelmässä oleva kalenteri. Kannettavien laitteiden yleistyttyä henkilökohtaista kalenteritietoa löytyy siis usealta eri laitteelta. Usean sähköisen kalenterin ajan tasalla pitäminen on työlästä ja virhealtista.

Opiskelijoiden ja henkilökunnan henkilökohtaisten sähköisten kalenterien synkronointi Korppi-järjestelmässä 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-järjestelmän 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-järjestelmään. 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 käytettävyyso Ongelmien ja toteutustekniikan rajoittuneisuuden vuoksi aloitettiin järjestelmän laajamittaisempi kehitys.

Ensimmäinen Korppi-järjestelmän kehittäjä oli syksyn 2000 Kotka-projekti. Se suunnitteli tietokannan rakenteen sekä määritteli uuden opintotietojärjestelmän ominaisuudet. 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-järjestelmää 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-järjestelmää kehittänyt projekti, Käki, suunnitteli ja kehitti syksyllä 2003 WWW-pohjaisen henkilökohtaisten opintosuunnitelmien (HOPS) laadinta-, seuranta- ja raportointisovellusta.

Kyyhky-projektin lisäksi Korppi-järjestelmän kehittämiseen osallistuu keväällä 2004 Kuikka-projekti, joka jatkaa syksyn 2003 Käki-projektin työtä HOPS-sovellukseen liittyen. Lisätietoa Korppi-järjestelmästä ja sitä kehittäneistä tietotekniikan opiskelijaprojekteista löytyy osoitteesta <http://sovellusprojektit.it.jyu.fi/>.

## **4 PROJEKTIN TAVOITTEET**

Luvussa esitellään projektin tavoitteet toteutettavan sovelluksen sekä projektiryhmän jäsenten henkilökohtaisen oppimisen suhteen.

### **4.1 Tavoitteet sovelluksen osalta**

Kyyhky-projekti liittyy Jyväskylän yliopistossa käytössä olevaan Korppi-nimiseen opintotietojärjestelmään. Projekti toteuttaa järjestelmään rajapinnan, jonka avulla erilaiset pöytäkoneiden, taskumikrojen ja matkapuhelinten kalenterisovellukset voidaan synkronoida Korpin kalenterin kanssa.

Kyyhky-projekti toteuttaa sovelluksen, jolla Korpin kalenterin käyttö saadaan laajennettua muita sähköisiä kalentereita käyttävien opiskelijoiden ja henkilökuntaan keskuuteen. Sovelluksen avulla henkilökuntaan kuuluva voi päivittää, muokata ja poistaa 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 eli Korpin kalenterista henkilökohtaisiin kalentereihin.

Toteutettavan sovelluksen edellytetään tarjoavan useita tietoja ja toimintoja, jotka esitellään tarkemmin Vaatimusmäärittelyssä. Sovelluksen rakennetta ja siihen toteutettavia ominaisuuksia kuvataan tarkemmin Sovellussuunnitelmassa.

## **4.2 Oppimistavoitteet**

Projektiryhmän ensisijaisia oppimistavoitteita ovat projektimuotoisen työskentelyn oppiminen, uusien tekniikoiden ja sovellusten käytön sisäistäminen, Java-kielen, SyncML-tekniikan sekä tietokantojen suunnittelun ja toteutuksen oppiminen. Lisäksi tavoitteena on vieraskieliseen materiaaliin tutustuminen ja dokumentointiin perehtyminen sekä kokouksissa toimimisen ja työtehtävien jakamisen oppiminen. Muita oppimistavoitteita ovat ihmissuhteisiin, vuorovaikutukseen, palautteen vastaanottamiseen, yhteistyöhön ja ongelmanratkaisuun liittyvien taitojen kehittäminen sekä ajankäytön hallinnan oppiminen.

Projektiryhmän jäsenten oppimisen näkökulmasta katsottuna päällimmäisenä tavoitteena on saada kokemusta ja kokonaiskuva projektimuotoisesta työskentelytavasta. Projektin jälkeen ryhmän jäsenillä tulisi olla selkeä kuva projektin vaiheista, hallinnasta ja läpiviennistä. Lisäksi projektin kautta ryhmän jäsenet tutustuvat ja saavat kokemusta suuremman sovelluksen määrittelystä, suunnittelusta, toteutuksesta ja testauksesta sekä ohjelmakoodin kunnollisesta kommentoinnista.

Yhtenä oppimisen kannalta keskeisenä tavoitteena on uusien tekniikoiden ja sovellusten käytön oppiminen ja sisäistäminen. Tavoitteena on, että projektin jälkeen kukin ryhmän jäsen hallitsee käytössä olleiden työkalujen ja ohjelmistojen sujuvan käytön sekä osaa tarvittaessa opettaa työkalujen ja ohjelmistojen käyttöä myös muille. Projektin aikana käytössä olleisiin työkaluihin perehtyminen antaa ryhmän jäsenille valmiudet omaksua muiden samantyyppisten työkalujen käyttö suhteellisen vähällä vaivalla. Lisäksi projektiryhmän jäsenten tavoitteena projektin aikana on sisäistää Java-ohjelmointikieli, tietokantojen suunnitteluun ja toteutukseen liittyvät vaiheet sekä SyncML-tekniikka.

Projektin aikana ryhmän jäsenet joutuvat perehtymään monenlaisiin dokumentaatioihin ja materiaaleihin, jotka hyvin usein on kirjoitettu englanniksi. Yksi projektiryhmän oppimistavoitteista on oppia lukemaan englanninkielisiä tieteellisiä sekä teknisiä termejä sisältäviä oman alan tekstejä. Toinen materiaaliin perehtymiseen liittyvä keskeinen oppimistavoite on käsillä olevien materiaalien kriittinen arviointi ja suhteuttaminen muihin samaa aihetta käsitteleviin materiaaleihin.

Dokumentointi on yksi iso osa projektia, ja siten myös yksi projektimme oppimistavoite. Tavoitteena on, että projektin jälkeen kukin projektiryhmän jäsen hallitsee tieteellisen tekstin kirjoittamisen tapoja prosessikirjoittamisesta pöytäkirjan laadintaan. Lisäksi kukin ryhmän jäsen oppii tieteellisessä kirjoittamisessa käytettäviä käytänteitä, dokumenttien ulkoasun muotoilua ja lähdeviittaustekniikkaa.

Projektiryhmän tavoitteena on myös oman työpanoksen esille tuonnin, tehtävien jaon sekä kokouksessa toimimisen oppiminen. Projektin jälkeen jokaisen tulisi olla selvillä siitä, miten suuressa projektissa tehtävät voidaan jakaa pienimmiksi kokonaisuuksiksi. Jokaisella projektiryhmään kuuluvalla on tällöin oma vastuualueensa, johon liittyvää työpanosta hän osaa tuoda esille. Projekti antaa myös mahdollisuuden harjoittaa yleisiä kokouksissa tarvittavia taitoja. Siksi jokainen projektiryhmän jäsen tutustuu sihteerin ja puheenjohtajan tehtäviin.

Muita huomionarvoisia oppimistavoitteita ovat työelämässäkkin erityisen tarpeelliset ihmissuhde-, vuorovaikutus-, yhteistyö- ja ongelmanratkaisutaidot. Sovellusprojekti tarjoaa erinomaisen mahdollisuuden harjoitella ja kehittää edellä mainittuja taitoja. Lisäksi projektiryhmän jäsenten tavoitteena on oppia vastaanottamaan palautetta kirjoittamistaan teksteistä sekä muista omista töistään. Rakentavan palautteen antamisen oppiminen muille ryhmän jäsenille, ohjaajille ja tilaajan edustajille on myös tärkeää. Lisäksi ajankäytön hallinta ja sovitusta aikataulusta kiinni pitäminen ovat oppimistavoitteita, jotka ryhmä itselleen asettaa.



## 5 PROJEKTIN RESURSSIT JA ORGANISAATIO

Luvussa kuvataan projektin käytössä olevia resursseja sekä organisaatioon kuuluvat henkilöt ja heidän suhteensa projektiin.

### 5.1 Henkilöt

Projektiryhmän muodostavat opiskelijat Timo Aarniovuori (tietotekniikka), Teija Alasalmi (englanti), Jaakko Hyvärinen (tietojärjestelmätiede) ja Maunu Tuomainen (matematiikka).

Tilaaajan edustajina toimivat lehtori Vesa Lappalainen tietotekniikan laitokselta ja koordinaattori Antti Auer virtuaaliyliopistohankkeesta. Projektin vastaava ohjaaja on lehtori Jukka-Pekka Santanen. Teknisenä ohjaajana toimii tietotekniikan opiskelija Tiina Pöyhönen. Muita projektiorganisaatioon kuuluvia henkilöitä ovat Minna Hillebrand, Jonne Itkonen ja Jani Kurhinen, jotka neuvovat tarvittaessa.

### 5.2 Tilat, laitteet ja ohjelmistot

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

Linux-koneisiin on asennettu kehitettävän sovelluksen kehitys- ja testausympäristöjä varten Apache, Tomcat 4.1 ja PostgreSQL. Toiselle Linux-koneelle on asennettuna Sync4j-palvelin (versio 1.2) sekä JBoss -sovelluspalvelin, jota Sync4j käyttää. Projektiryhmä käyttää projektin edetessä CVS-versionhallintajärjestelmää, Bugzilla, Korppi-järjestelmää sekä joidenkin tietojen osalta siivottua versiota Kotka-tietokannasta.

## **6 PROJEKTIN JÄSENTEN TEHTÄVÄT JA NIIDEN JAKAUTUMINEN**

Luvussa kuvataan projektiin liittyviä tehtäväalueita ja projektin jäsenten välistä työnjakoa.

### **6.1 Projektipäällikkö**

Projektipäällikkönä koko projektin ajan toimii Teija Alasalmi. Projektipäällikkö laatii projektille yksityiskohtaisen aikataulun sekä arvioi käytettäviin tehtäviin kuluvaan työmäärää. Lisäksi projektipäällikkö kirjoittaa projektisuunnitelman, jossa kuvataan projektin aihealue, tausta, tavoitteet, resurssit, tehtäväjako, dokumentointi- ja kokouskäytännöt ja riskit.

Projektipäällikkö suunnittelee projektiin liittyvien tehtävien työnjaon ja valvoo töiden edistymistä. Tarvittaessa projektipäällikkö tarkentaa ja organisoii ryhmän jäsenten tehtäviä uudelleen. Projektipäällikön vastuulla on myös asioista tiedottaminen projektiryhmälle ja -organisaatiolle niin, että kaikki asianomaiset ovat ajan tasalla. Lisäksi projektipäällikön tulee olla selvillä meneillään olevista ja tulevista tehtävistä, mahdollisista ongelmista sekä toteutettavan sovelluksen tilasta.

### **6.2 Tehtäväalueet**

Projektiin liittyviä tärkeimpiä tehtäväalueita ovat aiheeseen ja käytettäviin työkaluihin perehtyminen, dokumentointi, projektin WWW-sivujen ylläpito, vaatimusten määrittely, tietokanta-, Sync4j- ja käyttöliittymäsuunnittelu, ohjelmointi sekä testaus. Projektiin liittyvät tehtävät, niiden työmäärä sekä vastuuhenkilöt esitetään taulukossa 1.

| Tehtävä  | Työmäärä tunteina | Vastuuhenkilö   |
|--|-------------------|-----------------|
| Projektin hallinta                             | 38                | Teija           |
| Projektikansion ylläpito                       | 14                | Teija           |
| WWW-sivujen päivitys                           | 18                | Maunu           |
| Sync4j-palvelin                                | 40                | Jaakko          |
| Korppi-kyselyn laatiminen                      | 12                | Timo            |
| Perehdytykset                                  | 52                | Kaikki          |
| Aiheeseen tutustuminen                         | 40                | Kaikki          |
| JBuilderiin ja CVS:ään tutustuminen            | 60                | Kaikki          |
| SyncML-tekniikkaan tutustuminen                | 40                | Jaakko ja Maunu |
| Korppiin sekä Kotka-tietokantaan tutustuminen  | 20                | Kaikki          |
| JSP-tekniikkaan tutustuminen                   | 40                | Timo ja Teija   |
| Viikoittaiset palaverit                        | 112               | Kaikki          |
| Ryhmän sisäiset palaverit                      | 32                | Kaikki          |
| Pöytäkirjat                                    | 42                | Kaikki          |
| Projektisuunnitelma                            | 40                | Teija           |
| Vaatimusmäärittely                             | 40                | Jaakko ja Maunu |
| Sovellussuunnitelma                            | 40                | Timo            |
| Testaussuunnitelma                             | 20                | Maunu           |
| Sync4j:hin liittyvä toteutus                   | 200               | Jaakko ja Maunu |
| Käyttöliittymään ja Korppiin liittyvä toteutus | 200               | Timo ja Teija   |
| Yksittäisten ohjelmanosien testaukset          | 40                | Kaikki          |
| Koko sovelluksen testaus                       | 20                | Kaikki          |
| Käyttöohje                                     | 12                | Jaakko          |
| Sovellusraportti                               | 40                | Timo            |
| Projektiraportti                               | 40                | Teija           |
| Viimeistely                                    | 36                | Kaikki          |
| Loppuesittely ja sen valmistelu                | 12                | Kaikki          |
| <b>Yhteensä</b>                                | <b>1300</b>       |                 |

**Taulukko 1.** Projektiin liittyvät tehtävät, niiden työmäärä sekä vastuuhenkilöt.

### 6.3 Tehtävien jakautuminen

Projektin alussa projektiryhmän jäsenet tutustuivat yleisesti aihealueeseen, SyncML- ja vCalendar -materiaaleihin sekä käytettäviin työkaluihin ja ohjelmistoihin. Vastuualueet jakaantuvat sovelluksen vaatimusten määrittelyn ja suunnittelun osalta ryhmän jäsenten kesken seuraavasti. Ryhmän laatimien dokumenttien HTML-muotoon muuttamisesta sekä projektin WWW-sivujen luonnista ja päivityksestä vastaa Maunu Tuomainen. SyncML:ään ja erilaisiin palvelintoteutuksiin tarkemmin perehtyy Jaakko Hyvärinen. Hyvärinen ja Tuomainen hoitavat vaatimusmäärittelyn laatimisen. Korppi-kyselyn suunnittelusta ja toteutuksesta sekä sovellussuunnitelmasta vastaa Timo Aarniovuori. Projektilla ei ole erillistä sihteeriä, vaan jokainen ryhmän jäsen vastaa määritettyjen vastuualueiden mukaisesti dokumentoinnista.

Suunnitteluvaiheen lopussa tehtävät jaetaan projektijäsenten kesken vastuualueisiin siten, että kukin projektiryhmän jäsen perehtyy omaan osa-alueeseensa muita ryhmän jäseniä tarkemmin. Jokaisen projektiryhmän jäsenen tulee olla perillä myös toisten tekemistä osuuksista sekä heidän laatimistaan dokumenteista. Ohjelmointitehtävät pyritään jakamaan mahdollisimman tasapuolisesti kaikkien ryhmän jäsenten kesken, jotta jokainen saa kokemusta ohjelmoimisesta.

Erityiset vastuualueet toteutukseen liittyen on jaettu alustavasti seuraavasti. Alasalmi ja Aarniovuori keskittyvät sovelluksen käyttöliittymän suunnitteluun, JSP-tekniikkaan sekä Korppi-järjestelmän tietokantaan. Hyvärinen ja Tuomainen vastaavat Sync4j-palvelimeen liittyvästä suunnittelusta ja toteutuksesta. Tietokanta tullaan suunnittelemaan yhdessä ryhmän jäsenten kesken. Tehtävien vaatima arvioitu työmäärä sekä vastuuhenkilöt on esitetty taulukossa 1.

## 7 PROJEKTIN AIKATAULU

Luvussa kuvataan projektille suunniteltu aikataulu sisältäen eri toteutusvaiheiden aikajänteet sekä vastaavat tarkastuspistepäivämäärät. Luvun 7.2 taulukossa 2 on esitetty projektin eri vaiheille määritellyt välitavoitteet esittelypäivämäärinä, joihin mennessä kyseisten vaiheiden tuloksena syntynyt dokumentti tulee olla esittelykunnossa. Projektin aikataulu on esitetty janakaaviona liitteessä.

### 7.1 Projektin vaiheet

Projekti kestää arviolta 16 viikkoa. Projekti pyritään saamaan päätökseen toukokuun puoleen väliin mennessä. Pelivaraa on jätetty varmuuden vuoksi toukokuun loppuun asti, jos projekti kohtaa odottamattomia ongelmia tai muuten viivästyy. Projektin vaiheita ovat aiheeseen perehtyminen, määrittely, suunnittelu, toteutus, testaus ja viimeistely.

Projekti aloitetaan alustavalla aiheeseen perehtymisellä. Aiheeseen tutustutaan tarkemmin kuitenkin koko projektin ajan, koska aihealue on projektin jäsenille täysin uusi. Alustavan aiheeseen perehtymisen jälkeen aloitetaan määrittelyvaihe. Tarkan kehitettävän sovelluksen vaatimusten määrittelyn jälkeen aloitetaan sovelluksen käyttöliittymän, tietokantarakenteen ja ominaisuuksien toteutuksen suunnittelu.

Huolellisen suunnittelun jälkeen aloitetaan varsinainen sovelluksen ohjelmointi. Toteutusvaiheen jälkeen sovellusta testataan monipuolisesti. Viimeistelyvaiheessa sovellus- ja projektiraportit sekä sovelluksen prototyyppi viimeistellään esityskuntoon. Edellä mainituista vaiheista eniten aikaa vievät määrittely, toteutus sekä aiheeseen perehtyminen, jota siis tapahtuu koko projektin ajan.

## 7.2 Projektin aikataulu

Projektin aikatauluun liittyvät vaiheet ja niiden aikataulu on seuraava:

**28.1.2004** alkanut projektin aloitus sisälsi seuraavia ryhmän tehtäviä:

- Projekti aloitettiin sovellusprojektien aloitusluennolla.
- Hahmoteltiin projektin toteutusta yleisellä tasolla.
- Tutustuttiin yleisellä tasolla projektin aihealueeseen ja SyncML-toteutustekniikkaan.
- Tutustuttiin edellisten projektien projektikansioihin.

**4.2.2004** alkanut aihealueeseen perehtymisvaihe sisälsi seuraavia tehtäviä:

- Kerättiin SyncML:ään liittyvää materiaalia.
- Pidettiin projektin aloituspalaveri.
- Saatiin alustava kuva projektista.
- Valittiin käytettävät työkalut ja tutustuttiin niihin.
- Tarkennettiin ja rajattiin projektiaihetta.
- Aloitettiin projektisuunnitelman laadinta.

**17.2.2004** alkanut määrittelyvaihe sisältää seuraavia tapahtumia:

- Projektin tavoitteet ovat selkiintyneet.
- Vaatimusmäärittelyn laatiminen aloitettiin.
- Vaiheen tuloksena on Projektisuunnitelma.
- Sync4j-ohjelmointikokeiluja aloitettiin.
- Tarkennetaan ryhmän jäsenten vastuualueita.

**10.3.2004** on vaatimusmäärittelyvaiheen tarkastuspiste, ja alkaa suunnitteluvaihe

- Sovellussuunnitelman laatiminen aloitetaan.
- Vaatimusmäärittelyä on tarkennettu.
- Käyttöliittymän suunnittelu aloitetaan.

**19.3.3.2004** määrittelyvaiheen päätös

- Projektisuunnitelma ja Vaatimusmäärittely on hyväksytty.

**24.3.2004** suunnitteluvaiheen tarkastuspiste, alkaa toteutusvaihe

- Sovellussuunnitelma on laadittu.
- Sovelluksen toteuttaminen alkaa.

**31.3.2004** suunnitteluvaiheen päätös

**7.4.2004** on toteutusvaiheen tarkistuspiste, ja päättyy aiheeseen perehtymisvaihe

- Sovellussuunnitelma on hyväksytty.
- Ryhmän jäsenet tarkastavat ohjelmakoodin.
- Projektin aiheeseen ollaan perehdytty monipuolisesti niin, että aiheesta voidaan kertoa muille projektiin kuulumattomille henkilöille.

**21.4.2004** alkaa testausvaihe, jolloin

- sovelluksen valmiina olevia toimintoja testataan.

**30.4.2004** päättyy toteutusvaihe, sisältäen seuraavat tapahtumat:

- Järjestelmä on toiminnallisesti valmis.
- Kaikki projektiryhmän jäsenet testaavat koko sovellusta.
- Tilaajan edustajat ja ohjaajat aloittavat sovelluksen testaamisen.
- Loppuraporttien laatiminen aloitettu.

**3.5.2004** viimeistelyvaihe alkaa, jolloin

- viimeistellään dokumentit esityskuntoon ja
- viimeistellään sovellus luovutuskuntoon.

**5.5.2004** testausvaiheen päätös

- Sovellusta on testattu perusteellisesti, tuloksena valmis prototyyppi.
- Sovellus on saatu projektin osalta päätökseen.

14.5.2004 projekti päättyy tulosten luovutukseen tilaajille. Tällöin

- dokumentointi on viimeistelty ja
- sovellus on valmis.
- Lisäksi tilaajan edustajat ja ohjaat hyväksyvät tulokset.

30.5.2004 on projektin ehdoton päättymispäivä.

### 7.3 Projektin välitavoitteet

Projektin eri vaiheille määritellyt välitavoitteet ja esittelypäivämäärät esitetään taulukossa 2.

| Vaihe       | Dokumentti          | 1. Esittelypäivämäärä |
|-------------|---------------------|-----------------------|
| Määrittely  | Projektisuunnitelma | 25.2.2004             |
| Määrittely  | Vaatimusmäärittely  | 25.2.2004             |
| Suunnittelu | Sovellussuunnitelma | 31.3.2004             |
| Toteutus    | Testaussuunnitelma  | 21.4.2004             |
| Toteutus    | Ohjelmakoodi        | 21.4.2004             |
| Viimeistely | Käyttöohje          | 5.5.2004              |
| Viimeistely | Sovellusraportti    | 5.5.2004              |
| Viimeistely | Projektiraportti    | 12.5.2004             |

**Taulukko 2.** Projektin välitavoitteet ja niiden esittelypäivämäärät.

## 8 RISKIEN ARVIOINTI JA HALLINTA

Luvussa kuvataan projektin läpiviemiseen liittyviä riskejä ja niiden hallintakeinoja.

### 8.1 Aikatauluun liittyvät riskit

Suurin projektin läpivientiin liittyvä riski on mitä ilmeisemmin **sovitussa aikataulussa pysyminen**. Koska projekti toteutetaan yhden lukukauden aikana, on projektille suunnitellun aikataulun oltava kohtalaisen tiukka. Aikataulu on suunniteltava huolellisesti, jotta projekti saadaan vietyä läpi sille varatussa ajassa. Etukäteen määrätyt välitavoitteet ja



tarkastuspisteet auttavat projektin aikataulussa pysymistä. Jokaisen projektiryhmän jäsenen on omalta osaltaan huolehdittava siitä, että dokumentit ja muut vaaditut työt valmistuvat määräajassa.

## 8.2 Vaatimusten määrittelyyn liittyvät riskit

Vaatimusten määrittely muodostaa toisen riskitekijän. Toteutettavalta sovellukselta edellytettävät **vaatimukset tulee rajata riittävän hyvin**, jotta niiden toteuttaminen kurssin asettamissa rajoissa on mahdollista. Suhteellisen tiukasta aikataulusta ja työmäärän rajallisuudesta johtuen kaikkia ideoita, ominaisuuksia ja vaatimuksia ei välttämättä pystytä toteuttamaan tämän projektin puitteissa. Riittävän tarkka vaatimusten rajaus suunnitteluvaiheessa onkin edellytys sille, että olennaisimmat ominaisuudet pystytään myös toteuttamaan. Vaatimukset olisi myös hyvä pystyä priorisoimaan, jotta sovellus toteuttaisi ainakin välttämättömimmät ja tärkeimmät ominaisuudet.

Myös tilaajan edustajien, ohjaajien ja projektin jäsenten **eriävät näkemykset** toteutettavasta järjestelmästä tuovat mukanaan omat riskinsä. Tästä johtuen projektiryhmällä ja tilaajilla tulisi olla samanlaiset käsitykset siitä, millaista sovellusta ollaan toteuttamassa. Toteutettavan sovelluksen rakenteesta päästään yksimielisyyteen riittävällä vaatimusmäärittelyjen läpikäynnillä ja tarkentamisella sekä projektiorganisaation jäsenten välisillä keskusteluilla.

## 8.3 Tekniset riskit

**Käytettävään tekniikkaan** liittyvät riskit muodostavat kolmannen suuren riskitekijän. SyncML on tekniikkana kohtuullisen tuore, eikä siitä ole juurikaan saatavissa kattavaa materiaalia. Kalenteritietojen synkronointi on myös aiheena ryhmän jäsenille kohtalaisen tuntematon. Lisäksi tietotekniikan laitoksella ja informaatioteknologian tiedekunnassa ei juurikaan ole henkilöitä, jotka tietäisivät tarpeeksi projektin aihealueesta pystyäkseen neuvomaan kiperimmissä ongelmatilanteissa. Perusteellinen ja monipuolinen aiheeseen perehtyminen, oma-aloitteisuus ja aktiivinen tiedonhaku ennen varsinaisen sovelluksen

suunnittelua ja toteuttamista sekä niiden aikana edesauttavat projektin etenemistä ja läpivientä.

Toteutettavan sovelluksen **testiversio tulee saada valmiiksi hyvissä ajoin**, jotta testauksessa saatua palautetta ja kokemuksia voitaisiin hyödyntää sovelluksen viimeistelyssä. Sovellusta pyritään testaamaan mahdollisimman kattavasti testaussuunnitelman mukaisesti, jotta esiin tulevat ongelmat saadaan ratkaistua ennen sovelluksen luovuttamista tilaajan edustajalle.

**Käytettävät työkalut** ovat suurelta osin projektiryhmälle uusia, joten niihin tutustuminen vie oman aikansa. Tämä voi viivästyttää sovelluksen toteuttamisen aloittamista. Työkaluihin ja käytettävään tekniikkaan tutustuminen hyvissä ajoin ennen toteutusvaiheen aloittamista vähentää projektin viivästymisen riskiä. Lisäksi projektiryhmän jäsenet pyrkivät osallistumaan tietotekniikan laitoksen tarjoamiin monipuolisiin perehdyttämistilaisuuksiin.

#### 8.4 Muut riskit

Muihin riskeihin kuuluvat projektin rinnalla suoritettavat muut opinnot, sairastumiset, kokemuksen puute, ryhmädynamiikka sekä odottamattomat ongelmat.

Projektiryhmän jäsenillä on projektin lisäksi meneillään  **muita opintoja**, joiden suorittaminen voi viedä paljon aikaa. Tämä voi viivästyttää projektin etenemistä. Henkilökohtaisten aikataulujen ja muiden menojen osalta kukin projektiryhmän jäsen pyrkii joustamaan, jotta projekti saadaan toteutettua sovitussa aikataulussa.

Projektiryhmän jäsenen **sairastuminen tai elämäntilanteen muutos** voivat viivästyttää projektia. Sairauden tai muun äkillisen tapauksen sattuessa aikataulua pyritään joustamaan kyseessä olevan projektiryhmän jäsenen osalta. Tällöin myös muiden on kannettava hiukan suurempaa vastuuta kuitenkin siten, ettei kenenkään työmäärä kasva kohtuuttomaksi.

Projektiryhmän jäsenillä **ei ole juurikaan kokemusta** laajamittaisesta projektityöskentelystä, mikä osaltaan voi vaikeuttaa projektin läpivientä. Projektityöskentelyyn tarjotaan kuitenkin monenlaista apua ATK-tuen sekä teknisen ja vastaavan ohjaajan tahoilta.

Lisäksi **ryhmän sisäiset suhteet** saattavat kiristyä projektin edetessä hitaasti tai ryhmän kohdatessa odottamattomia ongelmia. Tällöin jokaisen projektiryhmän jäsenen on puhallettava yhteen hiileen ja toimittava yhteistyössä projektin parhaaksi ja ongelmien ratkaisemiseksi. Projektiryhmän keskinäinen vuorovaikutus, toisen kuunteleminen ja positiivinen kannustaminen auttavat ryhmän sisäisten riitojen selvittelyssä ja yhteisymmärrykseen pääsemisessä. Odottamattomien ongelmien haittavaikutuksia puolestaan voidaan lieventää ennakoimalla tulevaa tarpeeksi ajoissa. Lisäksi aikataulun on oltava suhteellisen tiukka, jotta mahdolliset odottamattomat ongelmat eivät aiheuta projektin viivästyistä.

## **9 DOKUMENTOINTI JA KOKOUSKÄYTÄNNÖT**

Luvussa kuvataan dokumentointikäytäntöjä, dokumentointiin käytettävät työkalut sekä dokumenttien ulkoasuun liittyvät seikat. Lisäksi luvussa kuvataan projektin kokouskäytännöt. Lähdekoodin kommentoimista ja muuttujien nimeämistä sivutaan yleisellä tasolla, sillä tarkemmin aihetta käsitellään Sovellussuunnitelmassa.

### **9.1 Dokumentointikäytännöt**

Projektin aikana laaditaan dokumentteina ainakin projekti- ja sovellussuunnitelmat, vaatimusmäärittely sekä projekti- ja sovellusraportit. Mahdollisesti laaditaan myös sovelluksen käyttöohje. Lisäksi kutakin kokousta varten laaditaan yksityiskohtainen esityslista. Jokaisessa kokouksessa käsitellyt asiat ja päätökset dokumentoidaan erilliseen pöytäkirjaan.

Projekti- ja sovellussuunnitelmat, vaatimusmäärittely sekä projekti- ja sovellusraportit laaditaan Microsoft Word -tekstinkäsittelyohjelmalla rtf-tiedostomuotoon. Dokumenttipohjana käytetään ryhmän suunnittelemaa dokumenttipohjaa. Kokousten esityslistat ja pöytäkirjat laaditaan käyttäen pohjana erillistä, muista dokumenteista eroavaa pöytäkirjapohjaa. Kaikkien dokumenttien ulkoasuun tullaan kiinnittämään erityistä

huomiota, ja jokainen dokumentti pyritään tarkastamaan huolellisesti ennen niiden toimittamista tilaajan edustajille ja ohjaajille.

Kaikki ryhmän laatimat dokumentit sekä projektin sähköpostilistalle lähetetyt viestit tulostetaan paperille ja kerätään projektikansioon, josta ne ovat helposti saatavilla. Projektin dokumentointi on myös luettavissa verkossa projektin kotisivuilla osoitteessa <http://sovellusprojektit.it.jyu.fi/kyyhky/>. Jokaisesta dokumentista tullaan tekemään rtf-muodon lisäksi myös HTML-versio. Projektin sähköpostilistalle lähetetyt sähköpostit ovat luettavissa osoitteessa <https://korppi.it.jyu.fi/list-archive/kyyhky/>.

## **9.2 Lähdekoodin kommentointi ja muuttujien nimeäminen**

Ohjelmakoodin kommentoinnissa ja muuttujien nimeämisessä tullaan noudattamaan kahdenlaista käytäntöä. Sync4j-palvelimeen liittyvät koodit kommentoidaan käyttäen valmista Sync4j-standardia, kun taas Korppi-järjestelmään liittyviin koodeihin ja muuttujien nimeämiseen sovelletaan yleisiä Korppi-järjestelmän lähdekoodikäytänteitä.

Vaikeasti ymmärrettäviin ohjelmakoodin kohtiin lisätään koodin lukemista helpottavat kommentit. Jatkokehityksen helpottamiseksi lähdekoodi ja kommentit kirjoitetaan englanniksi. Lähdekoodin kommentointia ja muuttujien nimeämistä kuvataan yksityiskohtaisemmin Sovellussuunnitelmassa.

## **9.3 Kokouskäytännöt**

Kyyhky-projektin kokouksissa puheenjohtajan ja sihteerin tehtäviä kierrätetään projektiryhmän jäsenten kesken siten, että jokainen esiintyy vuorollaan kummassakin toimijan roolissa. Ryhmän jäsenet toimivat kokouksissa pareittain siten, että parillisina viikkoina sihteerin ja puheenjohtajan tehtäviä hoitavat Timo Aarniovuori ja Teija Alasalmi sekä parittomina viikkoina em. tehtävien hoidosta vastaavat Jaakko Hyvärinen ja Maunu Tuomainen.

Sihteerin vastuulla on tarkkojen muistiinpanojen tekeminen sekä varsinaisen pöytäkirjan laatiminen ja puhtaaksikirjoittaminen. Puheenjohtaja valmistelee kokouksen poimimalla edellisen palaverin pöytäkirjasta olennaisimmat esille tuotavat asiat. Puheenjohtajan tulee viedä kokous sujuvasti läpi esityslistan mukaisessa järjestyksessä. Mahdollisimman pian kokouksen jälkeen sihteeri laatii pöytäkirjan, tarkistuttaa sen puheenjohtajalla tai vaihtoehtoisesti muulla ryhmän jäsenellä ja lähettää tarkastetun version projektin sähköpostilistalle.

Projektipäällikkö laatii kokouksen yksityiskohtaisen esityslistan hyvissä ajoin ja lähettää sen viimeistään vuorokautta ennen seuraavaa kokousta projektin sähköpostilistalle. Lisäksi esityslistaan liitetään edellisessä kokouksessa laadittu tehtävälista, johon on kirjattu kunkin projektiin kuuluvan seuraavat tehtävät.

Jokaisen kokouksen alussa esitetään ryhmän jäsenten ajankäyttövihkot tarkistettaviksi. Sen jälkeen tarkastetaan edellisen kokouksen pöytäkirja. Jos edelliseen pöytäkirjaan ei tarvita muutoksia, se hyväksytään sellaisenaan, muussa tapauksessa kirjataan ylös pöytäkirjaan sovitut muutokset. Seuraavaksi käydään läpi edellisessä kokouksessa laadittu tehtävälista. Tämän jälkeen käydään läpi käsiteltävät asiat esityslistan mukaisessa järjestyksessä. Lopuksi päätetään kokous.

## **10 YHTEENVETO**

Kevään 2004 aikana toteutettava Kyyhky-projekti kehittää sovelluksen Korppi-järjestelmän kalenteriosion tietojen vaihtamiseen erilaisten pöytämikrojen, kämmenlaitteiden ja matkapuhelinten kalenterien kanssa. Sovelluksen avulla sähköisiä kalentereita käyttävät opiskelijat ja yliopiston henkilökuntaan kuuluvat voivat muokata, lisätä ja poistaa Korppi-järjestelmässä olevia kalenteritapahtumia henkilökohtaisesta kalenteristaan käsin. Sovelluksen avulla henkilökohtaisten kalenteritapahtumien päivitys muista sähköisistä kalentereista Korpin kalenteriin ja Korpin kalenterista henkilökohtaisiin kalentereihin onnistuu sujuvasti.

Dokumentissa on kuvattu Kyyhky-projektin taustaa, tavoitteita sekä projektin läpivientiin liittyviä keskeisimpiä seikkoja. Projektin toteuttaman sovelluksen vaatimuksia esitellään Vaatimusmäärittelyssä. Sovelluksen toteutusta puolestaan kuvataan Sovellussuunnitelmassa.

Projektin läpiviennissä on omat riskinsä, jotka on otettu huomioon myös aikataulun suunnittelussa. Projektin läpivienti on vaativaa työtä jo pelkästään siksi, että useimmilla projektin jäsenistä ei ole kokemusta suurten järjestelmien kehittämisestä. Projektiryhmänä teemme parhaamme viedäksemme projektin kunnialla läpi ja päästäksemme kaikkia osapuolia tyydyttävään lopputulokseen.

## LÄHTEET

[1] Hillebrand Minna, Silván Markus, Vanhanen Antti ja Ylitalo Marko, "Koppelo-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2002.

[2] Hilpinen Toni, Koivuniemi Marko, Mäkinen Jussi ja Nurminen Miika, "Kiuru-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos 2002.

[3] Horppu Ismo, Mielityinen Markku ja Vire Markku, "Kotka-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2000.

[4] Huttunen Sami, Lamminmäki Tatu, Lappi Juha ja Pelkkikangas Eija, "Käki-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2003.

[5] Jaakkola Mia, Juutinen Sanna, Lupari Matti ja Nieminen Mikko, "Kolibri-projektin projektikansio", Jyväskylän yliopisto, tietotekniikan laitos, 2001.

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

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

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

[9] Santanen Jukka-Pekka, "Tietotekniikan Sovellusprojektien ohje", Jyväskylän yliopisto, tietotekniikan laitos, 2004.

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

## LIITE. AIKATAULU JANAKAAVIONA



