

# Lohtu-projekti

## Projektisuunnitelma

Versiohistoria:

1.0	16.1.2003	Luonnos	Virve
1.1	24.1.2003	Korjailtu. Mukana riskienhallinta ja laatusuunnitelma	Mari, Kimmo, Virve
2.0	27.2.2003	Viimeiset korjaukset. Hyväksytty versio.	Virve
2.1	10.2.2003	1. määrittelyn katselmointi jouduttiin tekemään kaksi kertaa. Myös 1. suunnittelun katselmointi jaettiin kahteen osaan ja 1. suunnitteluvaihetta pidennettiin viikolla.	Virve
2.2	24.2.2003	1. suunnittelun toista katselmointia, toteutuksen päättymistä ja testauksen alkamista siirrettiin parilla päivällä.	Virve
2.3	3.3.2003	Päivitetty 2. syklin roolit.	Virve
2.4	7.4.2003	Päivitetty 3. syklin roolit sekä Gantt-kaavio sekä 2. että 3. syklin osalta.	Virve

Helsinki 7. huhtikuuta 2003

Kimmo Airamaa, Andreas Asuja, Mari Muuronen, Seppo Pastila, Virve Taivaljärvi

HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos  
Laudatur-projektityö, kevät 2003

# Sisältö

<b>1</b>	<b>Projektin tavoitteet</b> .....	<b>3</b>
<b>2</b>	<b>Projektin koko- ja aikatauluarviot</b> .....	<b>3</b>
<b>3</b>	<b>Riskien hallinta</b> .....	<b>4</b>
<b>4</b>	<b>Aikataulu</b> .....	<b>5</b>
	4.1 <i>Prosessimalli</i> .....	5
	4.2 <i>Aikataulu</i> .....	6
<b>5</b>	<b>Projektin resurssit</b> .....	<b>7</b>
	5.1 <i>Henkilöt</i> .....	7
	5.2 <i>Välineet ja menetelmät</i> .....	7
<b>6</b>	<b>Projektin organisaatio</b> .....	<b>8</b>
	6.1 <i>Projektiryhmän organisaatio</i> .....	8
	6.2 <i>Vastuualueet</i> .....	8
<b>7</b>	<b>Laatusuunnitelma</b> .....	<b>9</b>
	7.1 <i>Laatutavoitteet</i> .....	9
	7.2 <i>Ohjelmointistandardit ja tyylit</i> .....	10
<b>8</b>	<b>Projektin seuranta ja ohjaus</b> .....	<b>10</b>
	8.1 <i>Ryhmän sisäinen projektin seuranta ja ohjaus</i> .....	10
	8.2 <i>Ryhmän ulkopuolinen projektin seuranta</i> .....	11
<b>9</b>	<b>Muutoksenhallinta</b> .....	<b>11</b>
	<b>Liite 1. Aikataulu</b> .....	<b>12</b>

# 1 Projektin tavoitteet

Tietojenkäsittelytieteen laitoksen Ohjelmistotuotantoprojektien toteutusta uudistetaan siten, että projektien arviointimahdollisuuksia ja seurantaan parannetaan uuden työkalun avulla. Tämän työkalun tarkoitus on helpottaa projektien suunnittelua, projektin etenemiseen liittyvien tietojen keräämistä ja tehtyjen suunnitelmien toteutumista. Tavoitteena on myös kyetä työkalun avulla vertailemaan eri projektiryhmiä toisiinsa.

Lohtu-projektin tavoitteena on määritellä tuotettavan ohjelmiston ominaisuudet, rajata ja suunnitella projektin aikana toteutettavat ohjelmiston osat sekä toteuttaa ja testata valitut ominaisuudet. Projektin tavoitteena on lisäksi soveltaa laudatur-tasolla opittuja ohjelmistoprosessien laadunvarmistusmenetelmiä käytännössä. Näin ollen tavoitteena on täyttää valitut laatukriteerit sekä toteutettavan ohjelmiston että projektityöskentelyn osalta ja täten toteuttaa annettu tehtävä ”paremmin” ja ”oikeammin” kuin cum laude approbatur –tason Ohjelmistotuotantoprojekteissa on tapana.

## 2 Projektin koko- ja aikatauluarviot

Tässä projektissa aikataulu muodostuu rajoittavaksi tekijäksi sille, miten suuri osa määritellystä ohjelmistosta ehditään toteuttamaan. Projektille on kurssimäärityksissä arvioitu työajaksi noin 160 tuntia kutakin projektiin osallistujaa kohden. Projekti kestää tammikuun puolivälistä toukokuun alkuun.

Toteutettavaksi valittavat ohjelmisto-osat tulisi siis rajata tämän mukaisesti. Koska aiempaa kokemusta ja kerättyä tietoa ryhmän työskentelystä ei juurikaan ole, on projektin aikana toteutettavan ohjelmiston kokoa tässä vaiheessa vaikea arvioida. Arvio tehdään ja tarkennetaan ohjelmamääritysten edetessä sekä projektisykliä vaihteissa.

### 3 Riskien hallinta

Projektin mahdollisia riskejä on listattu taulukossa 1. Nämä riskit on listattu erilaisten riskityyppien mukaiseen järjestykseen. Riskeistä mainitaan riskin todennäköisyys ja vaikutusaste.

Riskityypit:

- I Ryhmän kokemus ja käyttäytyminen
- II Projektin hallinto (asiakas?)
- III Kehitysympäristö ja tekniikka
- IV Tuotteen koko ja laatu
- V Prosessi

Vaikutusaste:

- 1 tuhoisa
- 2 vaarantava
- 3 heikentävä/hidastava

Taulukko 1. Projektin liittyviä riskejä

	<i>Riski</i>	<i>Riskityyppi</i>	<i>Tod. näk.</i>	<i>Vaikutus</i>
1.	Ryhmän jäsen sairastuu	I	70 %	3
2.	Tuote ei vastaa asiakastahojen odotuksia ennen projektin päättämistä	II	80 %	2
3.	Vaatimukset muuttuvat projektin kuluessa	II	65 %	1
4.	Tuotteesta saatava hyöty on kyseenalainen.	IV	50 %	2
5.	Tietojenkäsittelytieteen laitoksen laitteistossa on ongelmia, jotka häiritsevät/estävät projektin etenemistä	III	55 %	2
6.	Projektiryhmän jäsenillä ei ole tarpeeksi kokemusta käytettävistä menetelmistä	I	35 %	2
7.	Dokumentoinnista lipsutaan	I	50 %	3
8.	Aiemmin toteutetut ominaisuudet osoittautuvat rajoittaviksi	IV	40 %	2
9.	Aikarajat ovat epärealistisia	V	45 %	2
10.	Työ tai osa siitä katoaa/tuhoutuu	III	50 %	1
11.	Ryhmän työskentely ja päätöksenteko on tehotonta	I	30 %	1

Näille on laadittu seuraavat toteutumista ehkäisevät toimenpiteet:

1. Aikataulussa on pientä venymisen varaa. Ryhmät muut jäsenet voivat osin hoitaa sairastuneen tehtäviä, jotta työ etenisi aikataulussaan.
2. Pyritään varmistamaan (mm. asiakasryhmän jäseniltä kyselemällä) työn loppukäyttäjien toiveiden täyttyminen.
3. Pyritään selvittämään asiakastahojen toiveet mahdollisimman tyhjentävästi. Määrittämiä ja suunnitelmia voidaan korjata jokaisella iteraatiolla.
4. On varmistuttava siitä, että projektin tuotos on kelvollista. Ulkopuolista tahoa ei välttämättä saada tuotetta tutkiskelemaan ennen työn luovutusta, joten se on tehtävä itse.
5. Luetaan säännöllisesti ylläpidon ilmoituksia katkoista ja järjestetään näille ajoille osuva toiminta kotikoneille tai muuksi ajaksi. Ennakoimattomia katkoja varten ei voida paljoa tehdä.
6. Tutustutaan käytettäviin ohjelmiin ja tekniikoihin jo hyvin aikaisessa vaiheessa. Otetaan huomioon ennen suunnitelmien tekoa, mitä ryhmän jäsenet itse asiassa osaavat.
7. Varmistutaan siitä, että suunnitelmat ovat toteuttamiskelpoisia, jotta lipsumista ei tapahtuisi ainakaan huonon suunnittelun takia.
8. Huomioidaan lopullisen tuotteen tarpeet jokaisessa suunnitteluvaiheessa.
9. Aikataulun on oltava järkevä.
10. Kotitietokoneilla oleva tieto on hyvä varmuuskopioida tietojenkäsittelytieteen laitoksen koneille tai jonnekin muualle mahdollisten laiterikkojen takia. Laitoksen koneilla luotetaan säännöllisiin varmuuskopioihin, joista tiedot on palautettavissa. Ylläpidetään hyviä suhteita laitteistoylläpitoon. Ei tuhota tärkeitä tiedostoja ennen kuin kaikilla on projektista arvosana.
11. Ryhmän jäsenet mukautuvat rooleihinsa ja epäselvissä tilanteissa lopullinen päätöksenteko kuuluu sille, jonka vastuualueeseen aihe kuuluu.

## **4 Aikataulu**

### **4.1 Prosessimalli**

Projektissa seurataan niin kutsuttua lisäävää prosessimallia. Prosessi on jaettu kolmeksi sykliksi, ja yhden syklin sisällä suoritetaan hieman muokattuna lineaarisen eli vesiputousmallin mukaiset työvaiheet. Syklisyyteen päädyttiin, koska sen avulla voidaan projektityöskentelyssä syntyneitä virheitä seurata ja parantaa toimintaa syklissä opittujen asioiden myötä siten, että viimeinen sykli lopulta toivon mukaan tuottaa realistisesti arvioidun ja tarkkaan seuratun työskentelyn tuloksena laadukkaan lopputuotteen.

Syklisen sisäinen työskentely jakautuu seuraaviin vaiheisiin:

1. *Määrittely ja suunnittelu*: Kartoitetaan ohjelmiston vaatimukset: ominaisuudet, sidosryhmät, liittymät ja rajoitteet. Kartoitus tapahtuu pääasiassa ensimmäisen syklin alussa. Määrittelyä tarkennetaan UML:n käyttötapausten ja luokkakaavioiden avulla. Ensimmäisen syklin aikana tavoitteena on luoda yleiskuva koko sovelluksesta ja tämän jälkeen kunkin syklin aluksi tarkennetaan syklissä toteutettavan ohjelmisto-osan suunnittelua. Vaiheen lopuksi muodostetaan ohjelmiston suunnitteludokumentti, jota täydennetään sykleittäin. Käyttötapauksista muodostetaan testitapaukset, jotka toimivat pohjana testauksen suunnittelulle.
2. *Toteutus*: Ohjelmisto toteutetaan suoraviivaisesti suunnitteludokumentin mukaan. Toteutuksen lomassa tehdään testausta testaussuunnitelman mukaan. Toteutus dokumentoidaan ohjelmakoodiin.
3. *Testaus*: Valmista ohjelmaa testataan mahdollisimman kattavasti testaussuunnitelman mukaan. Testaustulokset kootaan erilliseen testausdokumenttiin. Mahdolliset virheet korjataan seuraavan syklin kuluessa ja testausdokumenttia päivitetään sykleittäin.
4. *Post mortem*: Kunkin syklin lopuksi kirjataan ylös syklin aikana opitut asiat ja heränneet ajatukset. Projektin lopuksi nämä kootaan yhteen loppuraportiksi, johon liitetään myös yhteenveto työtunneista sekä muista projektiin liittyvistä asioista. Syklin vaihtuessa suoritetaan lisäksi projektiryhmässä uusi roolijako ja määritellään seuraavan syklin tehtävät.

Yllämainittujen dokumenttien lisäksi projektin aikana kirjoitetaan toteutettavan ohjelmiston käyttöohje. Lisäksi prosessin seurannasta arkistoidaan projektiryhmän pöytäkirjat ja muistiot sekä työtuntilistat. Näitä dokumentteja tuotetaan läpi koko projektin.

## 4.2 Aikataulu

### Projektin aikataulu:

Projekti alkaa	15.1.2003
Projektisuunnitelma valmistuu	28.1.2003
Ohjelmiston kokonaisuuden määrittelyn katselmointi	3.2.2003
1. syklin postmortem, 2. syklin aloitus	10.3.2003
2. syklin post mortem, 3. syklin aloitus	1.4.2003
3. syklin post mortem	5.5.2003
Demotilaisuus ja loppuraportin palautus	viikolla 19

Aikataulua tarkennetaan projektin edetessä siten, että jokaisen työvaiheen jälkeen ositetaan seuraavan vaiheen tehtävät. Projektin työvaihekohtainen aikataulu on esitetty Gantt-kaaviona liitteessä 1.

#### Poissaolot:

Andreas Asuja on lomalla viikon 12.

Seppo Pastila on lomalla viikon 13.

Virve Taivaljärvi on lomalla viikon 16.

Kaikista poissaoloista ja tehtävien viivästymisestä on ilmoitettava projektiryhmälle mahdollisimman aikaisessa vaiheessa.

## **5 Projektin resurssit**

### **5.1 Henkilöt**

#### Projektin osapuolet:

Kurssivastaava: Inkeri Verkamo

Projektiryhmä: Kimmo Airamaa

Andreas Asuja

Mari Muuronen

Seppo Pastila

Virve Taivaljärvi

Asiakas: Turjo Tuohiniemi, Tietojenkäsittelytieteen laitoksen  
Ohjelmistotuotantoprojektin vastuuhenkilö

### **5.2 Välineet ja menetelmät**

Projektin kehitysympäristönä toimii Helsingin yliopiston Tietojenkäsittelytieteenlaitoksen laiteympäristö. Ympäristö, kehitystyökalut ja –kielet tarkentuvat paremmin määrittelyvaiheen yhteydessä.

Projektin dokumentit tuotetaan Word-tekstinkäsittelyohjelmalla ja julkaistaan pdf-muodossa projektin kotisivuilla osoitteessa <http://www.cs.helsinki.fi/group/lohtu/>. Dokumentit säilytetään

projektin kotihakemiston alla olevassa Dokumentit-hakemistossa Word-muodossa. Dokumentit nimetään yhtenäisellä formaatilla DokumentinNimi\_kkpp, jossa pp tarkoittaa dokumentin valmistuspäivää ja kk kuukautta. Tämä pätee lähinnä projektin pöytäkirjoihin. Ohjelmakoodiin lisätään javadoc-dokumentointi.

Lähdekoodin ja dokumenttien versiohallinnassa käytetään tietojenkäsittelytieteen laitoksen Linux-koneista löytyvää Concurrent Version System (CVS) –ohjelmistoa.

## 6 Projektin organisaatio

### 6.1 Projektiryhmän organisaatio

Projektiryhmän organisaationa toimii demokraattinen, hajautettu malli. Ryhmätyöskentely on keskeistä ja ongelmanratkaisu sekä päätöksenteko tapahtuu yhteistyössä. Projektiryhmä tuntee nimen Lohtu.

### 6.2 Vastuualueet

Projektiryhmän roolijako uudistetaan jokaisen projektin syklin vaihteessa. Kunkin syklin roolitus päivitetään alla esitettävään taulukkoon syklin alkaessa.

Projektiryhmän roolijako:

Rooli	1. sykli	2. sykli	3. sykli
projektipäällikkö	Virve	Seppo	Kimmo
ohjelmistopäällikkö	Andreas	Kimmo	Seppo
suunnittelupäällikkö	Seppo	Mari	Mari
prosessi/laatupäällikkö	Mari	Virve	Virve
tukipalvelupäällikkö	Kimmo	Andreas	Andreas

Roolien tehtävät perustuvat Team Software Process (TSP) –metodin rooli- ja tehtäväjakoon.

Vastuualueet jakautuvat roolien kesken seuraavasti:

- Projektipäällikkö vastaa ryhmän johtamisesta, puheenjohtajuudesta sekä ryhmän edustamisesta ulospäin.
- Ohjelmistopäällikkö vastaa tuotteen suunnittelusta, toteutuksesta, testauksesta ja käyttöohjeesta.
- Suunnittelupäällikkö vastaa aikatauluista ja työn etenemisen seurannasta ja raportoinnista.



- Prosessi/laatupäällikkö vastaa tuotteen ja työprosessin laadusta. Hän vetää katselmukset ja kirjoittaa kokouspöytäkirjat.
- Tukipalvelupäällikkö vastaa projektissa tarvittavista työkaluista ja ympäristöstä sekä hallinnoi riskejä ja uudelleenkäyttöä.

## 7 Laatusuunnitelma

### 7.1 Laatutavoitteet

Laatusuunnitelman tarkoituksena on asettaa projektiryhmän prosessille sekä tuotetulle tuotteelle laatuvaatimukset, joihin pyrkimällä projektiryhmä saavuttaa laadukkaan tuotteen ja laadukkaan prosessin vaatimukset.

Sovelluksen tulee täyttää seuraavat laatukriteerit: Sovelluksesta löydetyistä virheistä 80% löytyy ennen yksikkötestausta. Systemitestausvaiheessa tuotteesta löytyy 0 virhettä. Valmiista tuotteesta on löydyttävä kaikki määrittelyvaatimukseen kirjatut toiminnallisuudet. Lisäksi laadukas tuote on tuotettu siten, että työkirjan laatusuunnitelmaan kirjatut vaatimukset täyttyvät (lomake SUMQ). Lisäksi vaaditaan *TSPi laatukriteerit* -lomakkeen kohdat täyttyväksi (Introduction to the Team Software Process, table 5.8).

Projektiryhmän prosessi on laadukas, kun seuraavat ehdot täyttyvät. Ohjelman koon arvion ja toteutuneen arvon välinen ero on vähemmän kuin 20%. Myöskin tuntien arvion ja toteutuneiden tuntien välinen on oltava vähemmän kuin 20%. Laadukkaassa ohjelmistoprosessissa kaikki ryhmän jäsenet ovat keränneet ja kirjanneet vaaditut tiedot työkirjaan 100%:sti. Projektiryhmässä mitattavia suureita ovat aika, tuotosten koko ja virheiden tiedot. Prosessin aikana ryhmän jäsenet voivat parantaa prosessin toimintaa täyttämällä prosessin parannus lomakkeita (lomake PIP).

Projekti on suoritettu laadukkaasti mikäli se valmistuu suunnitellun aikataulun mukaisesti. Aikataulun laadinnan voidaan katsoa onnistuneen, mikäli projekti valmistuu vaaditussa ajassa, neljän päivän heitto molempiin suuntiin on sallittu.

Laatua valvotaan ja sitä pyritään parantamaan seuraamalla laatusuunnitelmaan toteutuneita arvoja (lomake SUMQ). Lisäksi ryhmän jäsenten on noudatettava projektissa käytössä olevia yhteisiä standardeja. Tällaisia ovat ohjelmointistandardit, nimeämisstandardit ja koodirivien laskenta.

## **7.2 Ohjelmointistandardit ja tyylit**

Ohjelmoinnissa pyritään hyvään ohjelmointityyliin. Java-osioissa noudatetaan Sunin Java-tyylioppaan (<http://www.java.sun.com/docs/codeconv/>) määräyksiä soveltuvin osin.

Java Servletit toteutetaan Java Servlet 2.3 ja Java Server Pages 1.2 –standardien mukaisesti.

HTML-kieli noudattelee World Wide Web Consortiumin määrittelemää HTML Specification 4.01 Strict –standardia (<http://www.w3.org/>) soveltuvin osin.

Ohjelmistoarkkitehtuurin ja tietokannan rakenteiden mallinnuksessa noudatetaan Unified Modelling Language (UML) –standardia.

## **8 Projektin seuranta ja ohjaus**

### **8.1 Ryhmän sisäinen projektin seuranta ja ohjaus**

Projektiryhmä kokoontuu maanantaisin klo 8-10 tietojenkäsittelytieteen laitoksen huoneessa B451. Näissä tapaamisissa seurataan projektin etenemistä ja keskustellaan projektin ja päätöksenteon kannalta tärkeistä asioista. Projektin etenemisen seuraamista varten kukin ryhmän jäsen päivittää omat TSP-tietonsa käytössä olevaan TSP-työkaluun kunkin viikon perjantaiamuun mennessä.

Lisäksi määrittelystä ja suunnitteluvaiheista järjestetään Formal Technical Review (FTR) –kokouksia, joissa tarkistetaan systemaattisesti vaiheessa tuotetun dokumentin laatu. Projektin aikana tuotettavan ohjelmiston testaus määritellään erillisessä testaussuunnitelmassa, jonka laatu myös varmistetaan FTR-kokouksessa.

Projektin aikana tuotettava ohjelmakoodi katselmoidaan siten, että jokainen projektiryhmäläinen tarkistaa jonkun toisen ryhmäläisen tuottaman koodin. Tämä tapahtuu toteutusvaiheen aikana ja tarkemmat järjestelyt sovitaan vaiheen alkaessa.

Ryhmän kokouksista kirjataan pöytäkirjat dokumentaationa projektin etenemisestä. Tapaamisissa voi ryhmän jäsenten lisäksi olla mukana kurssivastaava ja/tai asiakkaan edustaja.

## **8.2 Ryhmän ulkopuolinen projektin seuranta**

Projektipäällikkö raportoi viikoittain kurssivastaavalle työn etenemisestä. Kunkin syklin päättyessä pidetään post mortem –tilaisuus, johon myös kurssivastaava osallistuu. Projektin päättyessä järjestetään demotilaisuus, jossa projektin tuotokset esitellään. Tilaisuuteen osallistuvat kurssivastaavan lisäksi mahdollisesti rinnakkainen laudatur-projektiryhmä, Ohjelmistotuotantoprojektien vastuuhenkilöitä sekä muita aiheesta kiinnostunutta laitoksen henkilökuntaa

## **9 Muutoksenhallinta**

Projektin dokumentit ja tuotettavan ohjelmiston lähdekoodi säilytetään CVS-versionhallintajärjestelmässä, joka ylläpitää näiden versiotietoja. FTR-kokouksessa käsiteltävä dokumentti hyväksytään ja jäädytetään. Mikäli kyseistä dokumenttia joudutaan painavista syistä muuttamaan saman syklin sisällä jäädyttämisen jälkeen, kirjataan tehtävät muutokset erilliseen dokumenttiin nimeltä Muutokset jäädytettyihin dokumentteihin. Tämä ei kuitenkaan koske seuraavassa syklissä tehtäviä lisäyksiä.

## Liite 1. Aikataulu

