

Projektisuunnitelma

Koskelo

Helsinki 16.12.2004

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Tom Bertell
Johan Brunberg
Lauri Liuhto
Eeva Nevalainen
Harri Tuomikoski

Asiakas

Teemu Kerola

Johtoryhmä

Juha Taina
Turjo Tuohiniemi

Kotisivu

<http://www.cs.Helsinki.FI/group/koskelo/>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	12.9.2004	Ensimmäinen versio
1.0	17.9.2004	Lopullinen versio
1.1	10.11.2004	Lisätty työmääräarvio

Sisältö

1 Johdanto	1
1.1 Tavoite	1
1.2 Resurssit	1
2 Projektin organisointi	2
2.1 Projektin vastualueet	2
3 Työvaiheet	3
3.1 Aikataulu	4
4 Menetelmät ja työvälineet	5
4.1 Versionhallinta	5
4.2 Dokumentointi	5
4.2.1 Tuntilistat ja raportointi	5
4.3 Tiedotus	6
4.3.1 Kokousten materiaali	6
4.4 Suunnittelu	6
4.5 Ohjelmointi	6
4.6 Testaus	6
4.7 Seuranta	7
4.8 Tarkastukset	7
5 Ohjelmisto	8
5.1 Rakenne	8
5.2 Effective lines of code arvio – eLOC	8
5.3 Function Points – FP	9

	ii
5.4 Projektin kokoarvio COSMIC- ja Experience pisteillä	11
5.4.1 Experience –pisteet	11
6 Projektin riskit	11
6.1 Riskianalyysi	11
6.2 Riskien hallinta	14

1 Johdanto

1.1 Tavoite

Koskelo on Helsingin Yliopiston Tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojekti, jonka tavoitteena on tuottaa järjestelmä joka liittyy yhteen aiemmin tuotetut Titokone- ja eAssari-järjestelmät. Projektissa Koksia laajennetaan sellaiseksi, että sen syötteitä ja tulosteita voidaan käyttää eAssari-järjestelmässä.

Projektin tavoitteena on tuottaa muutama Tito-kurssille sopiva automaattisesti eAssarilla tarkastettavissa oleva harjoitustehtävä. Harjoitustehtävä voi olla sellainen, että siinä on dynaamisesti vaihtuvia osia (esim. taulukon T pituus on jokin väliltä $[5,15]$), jolloin yhdestä tehtävämäärittelystä saadaan joka kerta vähän erilainen tehtävä.

Koski-projektin kotisivu: <http://www.cs.Helsinki.FI/group/koski/>

Assari-projektin kotisivu: <http://www.cs.Helsinki.FI/group/assari/>

Projektin toinen tavoite on korjata osa Titokoneessa ilmenneistä bugeista.

1.2 Resurssit

Ryhmällä on käytössään Tietojenkäsittelytieteen laitoksen atk-tilat, ohjaajan antama kopiokortti sekä säilytyslokero huoneessa A307.

2 Projektin organisointi

2.1 Projektin vastualueet

Projektissa ovat mukana seuraavat henkilöt:

Nimi	Rooli	Puhelin
Tom Bertell	WWW- ja tukipalveluvastaava	040 - 511 9818
Johan Brunberg	Dokumentaatiovastaava	050 - 563 8945
Lauri Liuhto	Koodivastaava	040 - 700 9603
Eeva Nevalainen	Testausvastaava	044 - 222 1177
Harri Tuomikoski	Projektipäällikkö	044 - 350 8394

WWW- ja tukipalveluvastaava ylläpitää ryhmän www-sivuja ja ryhmähakemiston rakennetta sekä vasaa muusta ylläpito- ja tukitoiminnasta

Dokumentaatiovastaava kehittää dokumentaatio-ohjeiston muulle ryhmälle, vastaa dokumenttien ulkonäöstä ja viimeistelystä

Koodivastaava pitää huolta että tuotettava koodi on selkeää ja kommentoitua

Testausvastaava vastaa testaussuunnitelmasta ja valvoo sen noudattamista; vastaa myös testausraportista

Projektipäällikkö johtaa ryhmää, vastaa projektisuunnitelmasta, valvoo projektin etenemistä, on yhteydessä asiakkaaseen ja jakaa työt

Laadusta vastataan yhdessä, koska laatua on hankala yhden ihmisen arvioida ja useampi henkilö pystyy todennäköisesti takaamaan sen paremmin.

Projektiryhmän ohjaajana toimii Hannu Räisänen.

3 Työvaiheet

Projekti toteutetaan vesiputousmallin mukaisesti. Vesiputousmalli on lineaarinen, ja jo toteutettuun vaiheeseen ei palata. Vaiheisiin annettu työaika on ehdoton maksimi, ja ne pyritään saamaan valmiiksi nopeammin. Protektille on varattu aikaa 15 viikkoa ja palautus on viimeistään 17.12.2004, mutta projektisuunnitelma on laadittu 14:lle viikolle, mikä jättää ryhmälle hieman pelivaraa.

Vaiheet järjestyksessä:

Vaatimusanalyysi

Kesto: 3 viikkoa

Osuus: 21%

Määritellään järjestelmän palvelut, rajoitukset ja tavoitteet. Kootut asiat analysoidaan että ne ovat realistisia, ristiriidattomia ja jäljitettävissä sekä asiakkaan vaatimusten mukaisia. Ristiriitatilanteissa pyritään sopimaan kompromisseista asiakkaan kanssa. Vaatimusten perusteelta tuotetaan määrittelydokumentti joka toimii sopimuksena ryhmän ja asiakkaan välillä sekä lähtökohdana projektin suunnitteluvaiheelle.

Suunnittelu

Kesto: 5 viikkoa

Osuus: 35%

Tarkennetaan järjestelmän kuvaavaa mallia riittävän tarkalle abstraktiotasolle.

Toteutus

Kesto: 4 viikkoa

Osuus: 28%

Toteutetaan suunnitelma osa kerrallaan ja yksikkötestataan kukin osa.

Integrointi ja testaus

Kesto: 2 viikkoa

Osuus: 14%

Toimivista osista kootaan osajärjestelmiä, jotka testataan. Toimivista osajärjestelmistä kootaan edelleen kokonainen järjestelmä jonka toimivuus verifioidaan ja validoidaan järjestelmä vaatimusdokumentin mukaiseksi.

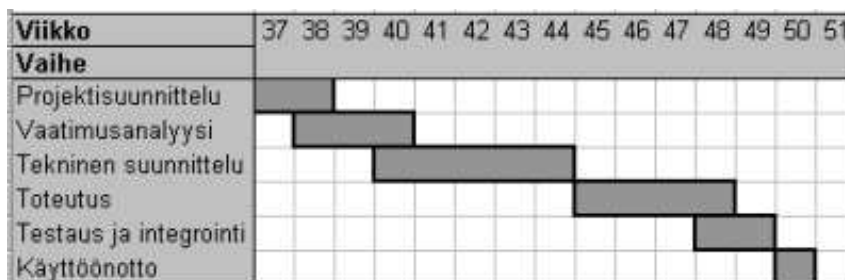
Viimeistely ja käyttöönotto

Kesto: 1 viikko

Osuus: 7%

Järjestelmä otetaan käyttöön. Järjestelmä asennetaan asiakkaalle toimintavalmiksi jotta arvosteluun ei kulu turhaa aikaa.

3.1 Aikataulu



Kuva 1: Aikataulu

Tärkeät päivämäärät:

Päivä jolloin valmis	Dokumentti	Vastuuhenkilö
17.9. perjantai	Projektisuunnitelma valmis	Harri
1.10. perjantai	Vaatimusdokumentti valmis	Harri
29.10. perjantai	Suunnitteludokumentti valmis	Lauri
29.10. perjantai	Testaussuunnitelma valmis	Eeva
10.12. perjantai	Toteutus- ja ylläpitodokumentti valmis	Lauri
10.12. perjantai	Käyttöohje valmis	Tom
10.12. perjantai	Testausraportti valmis	Eeva
10.12. perjantai	Loppuraportti valmis	Harri

29.11.–3.12. Demoviikko

4 Menetelmät ja työvälineet

4.1 Versionhallinta

Versionhallintaan käytetään CVS-ohjelmaa.

4.2 Dokumentointi

Kaikki tehtävä dokumentaatio tuotetaan suomeksi PS ja PDF formaatteihin käyttäen apuna L^AT_EX ladontaohjelmaa. Dokumentaatiovastaava valvoo, että sovittu dokumentaatio toimitetaan ajoissa, ja tukivastaava huolehtii, että sovitut dokumentit linkitetään ryhmän sivuille.

4.2.1 Tuntilistat ja raportointi

Ryhmän jokaisen jäsenen vastuulla on pitää kirjaa omista työtunneistaan. Projektin eri vaiheiden kuvaukseen käytetään seuraavia lyhenteitä.

- PS – Projektisuunnitelma
- VA – Vaatimusanalyysi
- SU – Suunnittelu
- KO – Koodaus
- TE – Testaus
- TT – Tuki- ja ylläpitotoiminta
- MT – Materiaaliin tutustuminen

Työtunneista pidetään kirjaa puolentunnin tarkkuudella, ja merkinnässä pitää olla lyhyt kuvaus tehdystä työstä. Dokumentointivastaava tekee kahden viikon välein yhteenvedon kaikkien kirjanpidosta kokoukseen, jotta voidaan seurata ryhmän edistymistä (ks. 4.7 Seuranta). Raportit lähetetään myös johtoryhmälle.

Tuntikirjanpito on ryhmän hakemistossa:
`/group/home/koskelo/cvsroot/admin/`

Ohjeet kirjanpitoon:
`http://www.cs.Helsinki.FI/group/ohtu/resurssit/report/`

4.3 Tiedotus

Projektiryhmän pääasiallinen tiedonvaihto tapahtuu kahdesti viikossa ryhmän kokouksissa (TI 10–12, TO 14–16) Exactumin huoneessa A217. Kokouksissa on kiertävä sihteerin toimi ja sihteeri huolehtii pöytäkirjan puhtaaksi kirjoittamisesta ja tallettamisesta ryhmän hakemistoon (`/home/group/koskelo/`).

Ryhmän muu tiedonvaihto tapahtuu sähköpostilistan (`ohtus04-koskelo-list@cs.Helsinki.FI`) sekä IRC-kanavan `#koskelo` välityksellä. Listalle lähetettävien postien otsikkoon laitetaan etuliite `[Koskelo]`. Puhelimen välityksellä tapahtuvaa viestintää käytetään vain kiireellisissä asioissa. Ryhmän www-sivut löytyvät osoitteesta `http://www.cs.Helsinki.FI/group/koskelo/` jossa pidetään näkyvillä dokumentteja, pöytäkirjoja sekä muuta tarpeellista.

4.3.1 Kokousten materiaali

Kaikki kokouksissa käsiteltävä materiaali (koodi, dokumentit, esityslistat) on lähetettävä jokaiselle ryhmän jäsenelle viimeistään kahta päivää ennen kokousta, jotta kaikilla on aikaa tutustua materiaaliin. Mikäli materiaali myöhästyy, niin siitä on ilmoitettava mahdollisimman aikaisin projektipäällikölle tai seuraavan kokouksen puheenjohtajalle projektipäällikön poissaollessa.

4.4 Suunnittelu

Suunnittelussa käytetään UML kuvauskieltä.

4.5 Ohjelmointi

Käytettävä ohjelmointikieli on Java (versio 1.4.2), ja kehitysympäristönä toimii Linux. Koodi tuotetaan englannin kielellä, jotta yhtenäinen ulkonäkö aiempien järjestelmien koodaukseen säilyisi. Kommentointi tehdään kuitenkin suomeksi. Ohjelmoitaessa pyritään noudattamaan Sunin Java Coding Conventonia soveltuvien osien. Mikäli ryhmän jäsen tuottaa koodia ympäristössä joka ei ole Unixin mukainen, niin hänen on huolehdittava että rivien vaihdot ja muut mahdolliset merkit käännetään asianmukaisiksi ennen koodin lähetystä CVS-hakemistoon.

4.6 Testaus

Yksikkötestaukseen käytetään JUnit ohjelmaa.

4.7 Seuranta

Kahden viikon välein tehdään seuranta projektiryhmän edistymisestä, onko pysytty aikataulussa, onko ilmennyt uusia riskejä tai onko vanhoja toteutunut. Projekti-päällikkö tuottaa ennen kokousta listan tarkastettavista asioista.

4.8 Tarkastukset

Toteutusvaiheessa suoritetaan koodin tarkastus monimutkaisimpien osien osalta.

5 Ohjelmisto

5.1 Rakenne

Tuotettava ohjelmisto koostuu seuraavista osista:

Tehtävänteko

Ottaa vastaan parametrejä joiden perusteella luodaan tehtäviä. Vastauksen syntaktinen oikeellisuus tarkistetaan myöskin tehtävän parametrien perusteella.

Displayer

Tarjoaa käyttäjälle rajapinnan tehtävien tekoon ja lähettää ne edelleen tarkistajalle. Tulostaa käyttäjälle tehtävän tulokset.

Tarkistus

Monimutkaisin osa. Täytyy kyetä tarkistamaan tehtävät älykkäästi koko vastauksen osalta, eikä vain lopputulosta. Kykenee myöskin antamaan hyviä vihjeitä takaisin displayer-modulille, miksi annettu vastaus ei kelpaa.

5.2 Effective lines of code arvio – eLOC

Effective lines of code eli todellinen koodirivien määrä johon ei ole laskettu mukaan kommentointia ja tyhjiä rivejä.

Ohjelmiston kokoarvio LOC-menetelmällä on tehty tutkimalla aiempien ohjelmistotuotantoprojektien arvioituja määriä ja vertaamalla niitä loppuraporteissa ilmoitettuihin, sekä yrittämällä verrata aiempien projektien moduuleita suhteessa Koskelon moduuleihin.

Tuotettavan järjestelmän moduulien kokoarvio:

- Tehtävänteko (1000 LOC)
- Displayer (1000 LOC)
- Tarkistus (3000 LOC)

Yhteensä: 5000 riviä.

5.3 Function Points – FP

Funktiopisteitä laskettaessa ryhmä löysi seuraavia toimintoja:

Syötteitä

- Tehtävänanto
- Mallivastaus tehtävän tekijältä
- Opiskelijan ratkaisu
- Titokoneen antama palaute
- Titokoneen ratkaisu muuttuvaan tehtävään

Tulosteet

- Palaute opiskelijalle
- Ratkaisun oikeellisuus opiskelijalle
- Tehtävänanto opiskelijalle
- Opiskelijan ratkaisun anto Titokoneelle
- Ratkaisun kirjaaminen Assariin

Kyselyjä

- Käyttäjän tunnistus Assarissa
- Oikean ratkaisun parametrien kysely

Liittymiä

- Koski järjestelmä
- Assari järjestelmä

Tyyppi	kpl	Kerroin	Selitys
Syötteitä	5	3	Yksinkertainen
Tulosteita	5	4	Yksinkertainen
Kyselyjä	2	6	Vaikea
Tiedostoja	0	–	–
Liittymiä	2	7	Normaali

UFC (peruskokoarvio): 61

Kompleksisuuskertoimet:

- Onko järjestelmä vikasietoinen? Tarvitaanko varmistuksia ja palautusta? 3
- Tarvitaanko tietoliikenneominaisuuksia? 2
- Onko hajautettua prosessinhallintaa? 0
- Onko suorituskyky kriittinen elementti? 2
- Käytetäänkö järjestelmää raskaassa käytössä olevassa ympäristössä? 4
- Tarvitaanko interaktiivista tietojen syöttöä? 5
- Täytyykö interaktiivinen tietojen syöttö synkronoida usealle näytölle tai operaatiolle? 0
- Päivitetäänkö tiedostoja interaktiivisesti? 0
- Ovatko syötteet tulosten tiedostot tai kyselyt monimutkaisia? 5
- Onko ohjelman toiminta monimutkaista? 5
- Onko koodi tarkoitettu uudellenkäytettäväksi? 0
- Ovatko ohjelmiston muunnokset ja asennus suunnitelmissa mukana? 5
- Onko ohjelmisto suunniteltu toimivaksi useina asennuksina eri organisaatioissa? 0
- Onko sovellus suunniteltava käyttäjäystävälliseksi? 5

Yhteensä: 36

Function Points $FP = 61 * (0.65 + 0.01 * 36) = 61.61$

5.4 Projektin kokoarvio COSMIC- ja Experience pisteillä

Experience on kokemustietokantaan perustuva ohjelmistoprojektin koon arviointimenetelmä. Tietokantaan on kerätty tietoa Suomessa viimeisen viidentoista vuoden aikana ja projekteja on kertynyt yli 500. Experience -pisteiden laskeminen perustuu lomakepohjaiseen näyttöjen, algoritmien, syötteiden, tulosteiden, olioiden ja liittymäraja-pintojen määrän arviointiin. Ryhmän täyttämä lomake on dokumentin liitteenä.

Pisteiden laskemisen lisäksi ryhmä täytti monivalintakyselyn, josta laskettiin projektin vaativuuskerroin. Vaatimuskertoimeksi saatiin 0,87. Täytetty monivalintakysely on dokumentin liitteenä.

5.4.1 Experience –pisteet

Tuotettavassa ohjelmistossa on ryhmän arvion mukaan noin 260 Experience – pistettä. Näistä saadaan työmääräarvio seuraavasti:

$$260 * 4 \text{ tuntia} * 0,87 = 904,8 \text{ tuntia.}$$

6 Projektin riskit

6.1 Riskianalyysi

Tässä osassa luetellaan arvioidaan projektia uhkaavia riskejä. Riskejä seurataan ja etsitään säännöllisesti (ks. 4.7 Seuranta).

Analyysin asteikossa on todennäköisyydelle arvot vähäinen (<10%), mahdollinen (<25%), todennäköinen (>50%), erittäin todennäköinen (>75%) ja lähes varma (>90%).

Riskin vakavuutta kuvaavat arvot tuhoisa, vakava, siedettävä ja vähäpätöinen.

Asiakkaan vaatimusten muuttuminen

Todennäköisyys: Vähäinen

Vakavuus: Siedettävä

Minimointi: Kun vaatimusdokumentti on hyväksytty asiakkaalla se jäädytetään eikä ryhmä sitoudu mihinkään muutoksiin.

Riskin toteutuessa: Jos muutoksia vaatimusdokumenttiin tehdään, tarkastellaan kokonaisuutta ja suunnitellaan toteutuksen aikataulu tarvittaessa uudestaan.

Ryhmän jäsenen odottamaton poissaolo

Todennäköisyys: Todennäköinen

Vakavuus: Siedettävä

Minimointi: Huolellinen dokumentointi henkilön tekemästä työstä ja vastuusta, jotta hänen toimet on helppo ja nopea jakaa muulle ryhmälle.

Riskin toteutuessa: Jaetaan henkilön tehtävät muille viimeistään seuraavassa kokouksessa.

Ryhmän jäsenen pitkäaikainen poissaolo tai keskeytys

Todennäköisyys: Vähäinen

Vakavuus: Vakava

Minimointi: Huolellinen dokumentointi henkilön tekemästä työstä ja vastuusta, jotta hänen toimet on helppo ja nopea jakaa muulle ryhmälle.

Riskin toteutuessa: Ks. aikataulun pettäminen.

Yhteensopivuusongelmat aiempien järjestelmien kanssa

Todennäköisyys: Mahdollinen

Vakavuus: Vakava

Minimointi: Tutustutaan aikaisessa vaiheessa aiempien projektien materiaaliin.

Riskin toteutuessa: Selvitetään mitä voidaan asialle tehdä ja kauanko aikaa mahdollinen suunnitelman muutos vie. Ks. aikataulun pettäminen.

Yliopiston laitteiston pitkäaikainen käyttökatkos tai hajoaminen

Todennäköisyys: Vähäinen

Vakavuus: Vakava

Minimointi: Selvitetään vaihtoehtoisten työtilojen käyttö.

Riskin toteutuessa: Jaetaan tehtävät tarvittaessa uudestaan jos ryhmähakemisto pois käytöstä.

Projektimateriaalinen tuhoutuminen

Todennäköisyys: Vähäinen

Vakavuus: Tuhoisa

Minimointi: Jokainen projektin jäsen pitää tuottamastaan materiaalista henkilökohtaiset kopiot ryhmäkansion lisäksi.

Riskin toteutuessa: Otetaan yhteyttä Tkkl:n ylläpitoon.

Tarvittavan tietotaidon puuttuminen

Todennäköisyys: Mahdollinen

Vakavuus: Siedettävä

Minimointi: Tarkat suunnitelmat ennen toteutusta.

Riskin toteutuessa: Selvitetään onko mahdollista hankkia tietotaitoa kohtuullisessa ajassa ilman että projekti kaatuu.

Aikataulun pettäminen

Todennäköisyys: Todennäköinen

Vakavuus: Siedettävä/vakava

Minimointi: Huolellinen työn edistymisen seuranta, ja töiden vaativuuden arviointi tehtäviä jaettaessa.

Riskin toteutuessa: Arvioidaan aikataulua uudelleen seuraavassa kokouksessa ja mikä vaikutus osan/vaiheen viivästymisellä on. Äärimmäisessä tapauksessa neuvotellaan asiakkaan kanssa järjestelmän alemman prioriteetin toimintojen supistamisesta.

Kommunikaatio-ongelmat

Todennäköisyys: Mahdollinen

Vakavuus: Siedettävä

Minimointi: Ryhmän jäsenet kertovat edistymisestään toisilleen, vaikka olisi kyse pienemmästäkin asiasta. Projektiin liittymätönkin kommunikaatio parantaa tiimihenkeä ja pienentää kynnystä kertoa asioista jotka saattavat tuntua itselle merkityksettömältä, mutta toiselle tärkeältä.

Riskin toteutuessa: Selvitetään asioita ryhmän kesken ja äärimmäisessä tapauksessa ohjaajien ja vastuuhenkilöiden kanssa.

Titokoneen ulkoinen rajapinta epätarkoituksenmukainen

Todennäköisyys: Mahdollinen

Vakavuus: Siedettävä/vakava

Minimointi: Titokoneeseen tutustuminen ajoissa ja hyvin.

Riskin toteutuessa: Keskustellaan asiasta seuraavassa kokouksessa ryhmällä ja kenties myös asiakkaan kanssa. Ks. aikataulun pettäminen.

Titokone ei tue rinnakkaisuutta

Todennäköisyys: Pieni

Vakavuus: Vakava

Minimointi: Otetaan selvää toiminnasta Assarin jäseniltä.

Riskin toteutuessa: Keskustellaan asiasta seuraavassa kokouksessa ryhmällä, sekä asiakkaan kanssa. Harkitaan järjestelmän toiminnan toteutusta FIFO-periaattella asiakkaiden palvelussa.

eAssarin tietorakenteet eivät yhteensopivia

eAssarin tietorakenteet eivät yhteensopivia

Todennäköisyys: Mahdollinen

Vakavuus: Vähäinen

Minimointi: Assarin dokumentaatio ei selvennä tarpeeksi hyvin kuinka luotu tehtävä talletetaan ja missä muodossa. Eli onko DTO yhteensopiva uusien tehtävityyppien kanssa. DTO:a muokattava tässä tapauksessa, mikäli ei.

6.2 Riskien hallinta

Tässä osassa kuvataan kappaleessa 6.1 lueteltuja riskejä jotka ovat toteutuneet, sekä kerrotaan vastatoimista.

Titokoneen ulkoinen rajapinta epätarkoituksenmukainen

Titokoneen rajapinta todettu puutteelliseksi, ja ryhmä on päättänyt muokata sitä. Muokkausten jälkeen Titokoneen rajapinta tarjoaa statistiikkaa Titokoneen suorituksesta. Suoritustietoja ovat mm. käytettyjen konekäskyjen määrä. Katso tarkemmat kuvaukset vaatimusdokumentista.