

Projektisuunnitelma

Kohahdus

Helsinki 8.12.2006

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhmä

Taro Morimoto, Projektipäällikkö
Tuomas Palmanto, Vaatimusmäärittelyvastaava
Mikko Kinnunen, Suunnitteluvastaava
Markus Kivilä, Koodivastaava
Jari Inkinen, Testausvastaava
Paula Kuosmanen, Dokumenttivastaava

Asiakas

Teemu Kerola

Johtoryhmä

Sanna Keskkioja

Kotisivu

<http://www.cs.helsinki.fi/group/kohahdus>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	18.9.2006	Ensimmäinen versio
1.0	2.10.2006	Ensimmäinen valmis versio
1.1	16.10.2006	Muutettua riskejä ja lisätty LOC
1.2	8.12.2006	Päivitetty ottaen huomioon myös toisen syklin muutokset

Sisältö

1	Johdanto	1
2	Projektiorganisaatio	1
3	Riskianalyysi	2
3.1	Projektin riskit	3
3.2	Tuotteen riskit	4
3.3	Riskien hallinta	4
4	Laitteisto- ja ohjelmistoympäristön vaatimukset	5
5	Koko- ja kustannusarviot	5
5.1	Toimintopisteanalyysi	6
6	Työn ositus	7
6.1	Projektin ositus kahteen sykliin	9
7	Aikataulu	9
8	Seuranta- ja raportointimenetelmät	9

1 Johdanto

Tämä on ohjelmistotuotantoprojektiryhmän Kohahdus projektisunnitelma. Projektin tarkoituksena on tuottaa konekielisen ohjelmoinnin harjoitusympäristö eAssari-ympäristöön.

Projektissa luodaan järjestelmä automaattisesti tarkastettavien TTK-91-konekielen harjoitustehtävien luomiseen ja ratkomiseen. Lisäksi toteutetaan käyttäjien hallinta ja tehtävien tilastointi. Tehtävien ratkaisuun ohjelmisto käyttää olemassa olevaa Titokone-simulaattoria. Projektin painopisteenä on saada aikaan valmis tuote, jossa painopiste on toimivassa opiskelijan ja opettajan käyttöliittymässä.

Edellinen ohjelmistotuotantoprojektiryhmä Koskelo teki pari vuotta sitten lähes samoilta vaatimuksilla tuotetun ohjelmiston. Asiakas ei kuitenkaan ollut tyytyväinen lopullisen tuotteen kokonaisuuteen. Tästä johtuen keskityimme hyvän käyttöliittymän toteuttamiseen, jotta asiakkaan ja käyttäjien olisi helppo käyttää tuotetta. Kohahdus projektissa ei uudelleenkäytetä edellisen projektiryhmän ohjelmistoa. Koskelo-projektiryhmän dokumenttien avulla pyrimme muodostamaan realistisen käsityksen projektissa olevista haasteista.

Toteutettava ohjelmisto sisältää

- webbiselainpohjaisen käyttöliittymä (tehtävän luonti/vastaus, käyttäjien hallinta ja tilastointi)
- tehtävien tarkastukseen käytettävä komponentti
- integroiminen Titokone-simulaattoriin
- eAssari-kehiksen takana olevan tietokannan käyttäminen

2 Projektioorganisaatio

Jokaisella ryhmän jäsenellä on vastuualue, josta hänellä on ylin päätäntävalta. Muuten ryhmän toiminta on tasa-arvoista. Kaikki saavat esittää kysymyksiä, kommentteja ja palautetta kenelle tahansa. Jos projektin eteneminen vaarantuu päätösten puutteesta tai riitailanteesta, projektipäälliköllä on oikeus tehdä lopullinen ratkaisu.

Ryhmän jäsenten vastuualueet ovat seuraavat:

Taro Morimoto. Projektipäällikkö, Vara-dokumenttivastaava.

Tuomas Palmanto. Vaatimusmäärittelyvastaava, Vara-testausvastaava.

Mikko Kinnunen. Suunnitteluvastaava, Vara-projektipäällikkö.

Jari Inkinen. Testausvastaava, vara-koodivastaava.

Markus Kivilä. Koodivastaava, vara-suunnitteluvastaava.

Paula Kuosmanen. Dokumenttivastaava, vara-vaatimusmäärittelyvastaava.

Vastuualueiden tehtäväkuvaukset ovat seuraavat:

Projektipäällikkö. Projektipäällikkö vastaa projektisuunnitelmasta, projektin aikataulusta, henkilöiden allokoinnista tehtäviin ja riskienhallinnasta. Lisäksi projektipäällikkö on viimekädessä vastuussa projektin onnistumisesta.

Vaatimusmäärittelyvastaava. Vaatimusmäärittelyvastaava toimii asiakkaiden ja projektiryhmän yhdyshenkilönä, vastaa vaatimusmäärittelyn osavaiheiden onnistumisesta, määrää vaatimusdokumentin rakenteen, toimii puheenjohtajana vaatimusmäärittelyyn liittyvissä kokouksissa ja vastaa siitä, että kaikki tuotteelta halutut vaatimukset saadaan kirjattua ylös.

Suunnitteluvastaava. Suunnitteluvastaava vastaa siitä, että komponenttien väliset rajapinnat ovat yhdenmukaiset, suunnittelu tehdään projektin kannalta riittävällä tarkkuudella, tietokannan määrittely täyttää sille asetetut tavoitteet ja suunnitelma ja järjestelmäarkkitehtuuri ovat yhdenmukaiset.

Koodivastaava. Koodivastaava vastaa siitä, että koodien ulkoasu on yhteneväinen, kaikki tekevät yksikkötestauksen, rajapinnat ovat yhtenevät myös kooditasolla ja koodi vastaa suunnittelua sekä arkkitehtuuri- että komponenttitasolla.

Testausvastaava. Testausvastaava vastaa siitä, että kaikki käyttötapaukset testataan, kaikki käyttäjän vaatimukset testataan, kaikki kirjatut poikkeustilanteet testataan ja asiakkaalle annetaan mahdollisuus hyväksymistestaukseen.

Dokumenttivastaava. Dokumenttivastaava vastaa siitä, että dokumenttien ulkoasu on yhteneväinen, dokumentit ovat luettavassa kunnossa ja dokumenttien sisältö on kattava. Dokumenttivastaava pitää yllä projektin kotisivua.

Jokaiselle vastuualueelle on määrätty varavastaava. Varavastaava perehtyy alueeseen, ja jos varsinainen vastaava jonkin riskin toteutumisen seurauksena ei pysty jatkamaan vastaavana, varavastaava ottaa tehtävän itselleen.

3 Riskianalyysi

Projektiryhmä käyttää riskin todennäköisyydelle seuraavaa luokitusta:

Lähes varma (+++++). Riski toteutuu todennäköisyydellä $p > 0,9$

Varma (++++). Riski toteutuu todennäköisyydellä $0,7 < p \leq 0,9$

Keskiverto (+++). Riski toteutuu todennäköisyydellä $0,5 < p \leq 0,7$

Alle keskiverto (++) . Riski toteutuu todennäköisyydellä $0,3 < p \leq 0,5$

Mahdollinen (++). Riski toteutuu todennäköisyydellä $0,1 < p \leq 0,3$

Epätodennäköinen (+). Riski toteutuu todennäköisyydellä $p < 0,1$

Projektiryhmä käyttää riskin vakavuudelle seuraavaa luokitusta:

Katastrofaalinen (***)**. Riskin toteutuminen lopettaa projektin.

Erittäin vakava (**)**. Riskin toteutuminen vahingoittaa projektia ja voi estää sen jatkumisen.

Vakava (**)**. Riskin toteutuminen haittaa projektia ja voi estää sen pysymisen aikataulussa.

Keskiuerto (*)**. Riskin toteutuminen haittaa projektia ja voi estää kaikkien haluttujen ominaisuuksien toteuttamisen.

Lievä ()**. Riskin toteutuminen aiheuttaa projektiin lisätyötä, mutta ei estä projektia valmistumasta aikataulussa toivotuin ominaisuuksin.

Erittäin lievä (*). Riskin toteutuminen aiheuttaa pientä epämukavuutta projektissa.

Projektiryhmä käyttää riskien priorisointiin seuraavaa asteikkoa:

Ykkösluokka (1). Riskin toteutumisen todennäköisyyttä pienennetään aktiivisesti koko projektin kestävin vastatoimin. Riskin toteutuessa on varasuunnitelma.

Kakkosluokka (2). Riskin toteutumisen todennäköisyyteen pyritään vaikuttamaan projektin alussa tehtävillä päätöksillä. Riskin toteutuessa on varasuunnitelma.

Kolmosluokka (3). Riskin toteutumisen todennäköisyyteen ei vaikuteta. Riskin toteutuessa on olemassa varasuunnitelma.

Nelosluokka (4). Riskiä ei huomioida.

3.1 Projektin riskit

Aikataulu. Projekti ei pysy aikataulussa.

Keskeytys. Projektiryhmän jäsen keskeyttää projektin.

Hidastelu. Projektiryhmän jäsen ei tee tehtäviään luvatussa ajassa, vaan muiden pitää paikata hänen tekemisiään.

Kemia. Projektiryhmän jäsenet eivät tule toimeen keskenään.

Vastuuhenkilö. Vastuuhenkilö vaihtuu kesken projektin.

Nimi	Todennäköisyys	Vaikutus	Prioriteetti
Aikataulu	++++	****	(1)
Keskeytys	++	***	(3)
Hidastelu	+++	***	(1)
Kemia	+	**	(4)
Vastuuhenkilö	+	*	(4)

3.2 Tuotteen riskit

Vaativuudet muuttuvat. Asiakas ei osaa ilmaista vaatimuksia selvästi.

Tekniset haasteet. Ryhmä ei osaa tehdä haluttua tuotetta.

Ulkoiset rajapinnat. Rajapinnat muihin järjestelmiin puuttuvat tai ovat vajaat.

Kokoarvio. Tuote osoittautuu odotettua suuremmaksi.

Nimi	Todennäköisyys	Vaikutus	Prioriteetti
Vaativuudet	++	**	(2)
Tekniset	+	***	(4)
Ulkoiset	++++	****	(2)
Kokoarvio	++	***	(1)

3.3 Riskien hallinta

Seuraavat riskit on päätetty ottaa huomioon projektissa:

Nimi	Todennäköisyys	Vaikutus	Prioriteetti
Aikataulu	+++	**	(1)
Hidastelu	++	**	(1)

Riskeille on päätetty tehdä seuraavat vastatoimet ja varasuunnitelmat.

Ulkoiset. Projektin alussa pyritään aktiivisesti saamaan kaikkien rajapintojen ja järjestelmien dokumentoidut ja dokumentoimattomat ominaisuudet selville. Lisäksi pyritään saamaan eAssari-kehiksen asennuspaketti mahdollisimman ajoissa. Riskin toteutuessa väljennetään projektin aikataulua vähentämällä alhaisen prioriteetin toiminnallisuuksia.

Vaativuudet. Projektin alkuvaiheessa pyritään olemaan asiakkaaseen mahdollisimman paljon yhteydessä, jotta saataisiin kaikki halutut vaatimukset selville. Lisäksi projekti jaetaan kahteen sykliin, jotta toisessa syklissä voidaan jäljellä olevia vaatimuksia sopeuttaa paremmin käytännössä saatuihin kokemuksiin projektiryhmän toteutusnopeudesta.

Aikataulu. Projekti pyritään aikatauluttamaan mahdollisimman väljästi siten, että ei yritetä toteuttaa enempää kuin on käytännössä mahdollista. Jos tämä riski toteutuu, niin muokataan aikataulua ja vähennetään alemmalla prioriteetilla toteutettavia toiminnallisuuksia.

Hidastelu. Kaikkien jäsenten osallistumista seurataan aktiivisesti. Jokainen pitää huolen siitä, että luvatut tehtävät tehdään. Jos tuntuu siltä, ettei pysty tekemään tehtävää luvatussa ajassa, niin ilmoittaa asiasta mahdollisimman nopeasti, jotta voidaan delegoida tehtävä toiselle ryhmän jäsenelle. Ennen tehtävän vastaanottamista jokainen miettii tarkasti, että ehtiikö tehdä ks. tehtävän luvatussa ajassa. Pyritään aikatauluttamaan tehtävät tarpeeksi väljästi.

Kokoarvio. Projektisuunnitelman ja vaatimusmäärittelyn jälkeen arvioidaan ohjelmiston koot. Arviointiin lasketaan noin viidenneksen joustovaraa ylöspäin. Jos tuotteen kokoarvio kasvaa vaatimusmäärittelyn jälkeen yli N koodirivin, asiakkaan kanssa keskustellaan alhaisen prioriteetin vaatimusten pudottamisesta.

4 Laitteisto- ja ohjelmistoympäristön vaatimukset

Ohjelmisto tehdään TKTL:n Linux-ympäristössä Java 5.0:lla. Ohjelmistoa tullaan ajamaan Tomcat webbipalvelimella ja pohjautuu Servlet- ja JSP-teknologioihin. Tietokantana toimii Oracle 10.0, jota käytetään eAssari-kehiksen kautta. Työskentely-ympäristönä käytetään Eclipse-sovelluskehitystä. Versionhallintaan käytetään CVS-versionhallintajärjestelmää. Kaikki tarvittavat laitteistot ja ohjelmistot ovat käytettävissä TKTL:n koneilla Internet Explorer -ja Firefox -selaimella.

5 Koko- ja kustannusarviot

Kohadus-järjestelmän ydinohjelmisto voidaan jakaa seuraaviin osiin (vertaa Johdantoluvun listaan):

- Opettaja/oppilas kirjautuu sisään
- Opettaja/oppilas muokkaa omia tietojaan
- Opettaja selaa tehtäviä
- Opettaja luo uuden tehtävän tai muokkaa olemassa olevaa
- Opettaja selaa suoritettujen tehtävien tilastoja
- Opettaja selaa oppilaan tietoja
- Oppilas selaa tehtäviä

- Oppilas suorittaa tehtävän
- Lisäksi tehtävän suorituksessa käyttöliittymän takana ajetaan Titokone-simulaattoria ja verrataan sen tuottamia tulosteita tehtävän kriteereihin

5.1 Toimintopisteanalyysi

Toimintopisteanalyysi tehdään edellä listatuille järjestelmän ominaisuuksille.

Toiminto	Syötteet	Tulosteet	Kyselyt	Tiedostot	Ulkoiset liittymät
Sisäänkirjautuminen	2	0-1	1	1-4	0
Opettaja muokkaa tietojaan	2-3	1	1	1-2	0
Opettaja selaa tehtäviä	1	0	1	1-3	0
Opettaja luo/muokkaa tehtävää	10-20	1	2	1-4	0
Opettaja selaa tilastoja	1-2	2	2	3-5	0
Opettaja selaa oppilaan tietoja	1	1	1	1-3	0
Oppilas selaa tehtäviä	1	1	1	1-4	0
Oppilas suorittaa tehtävän	2	2	2	1-4	1

Edellisistä luvuista on laskettu minimi-, maksimi- ja odotettu määrä. Näistä on laskettu arvio kaavalla $a = (\text{minimi} + \text{maksimi} + \text{todennäköinen} \cdot 4) / 6$. Kaava on esitelty muun muassa alan oppikirjoissa yleisesti käytettynä arviomenetelmänä.

Edellisen analyysin perusteella saadaan seuraava taulukko:

Pistelähde	minimi	todennäköinen	maksimi	arvio	paino	pisteet
Syötteiden määrä	20	30	32	29	5	145
Tulosteiden määrä	7	8	8	8	5	40
Kyselyiden määrä	11	11	11	11	4	44
Tiedostojen määrä	10	20	29	20	10	200
Ulkoisten liittymien määrä	1	1	1	1	10	10
Yhteensä						439

Kysymysten painoarvot:

Backup and recovery	2
Data communications	2
Distributed processing	0
Performance critical	1
Existing operating environment	2
On-line data entry	4
Input transaction over multiple screens	3
Master files updated on-line	4
Information domain values complex	3
Internal processing complex	3
Code designed for reuse	3
Conversion/installation in design	1
Multiple installations	1
Application designed for change	3
0,01 * summa	0,32

Arvioitu toimintopisteiden määrä: $FP = \text{yht} * (0,65 + 0,32) = 439 * (0,65+0,32) = 426$

Normaalin ohjelmistotuotantoprojektin toimintopisteiden määrä on noin 150-300. Näin ollen arvioitu ohjelmisto on vajaa kaksi kertaa liian suuri. Projektin vaatimukset on priorisoitu ja ajanpuutteen sattuessa ominaisuuksia karsitaan pois prioriteettien mukaisesti. Myös projektin toisessa syklissä voidaan tilannetta vielä arvioida uudelleen.

Ohjelmiston koko tulisi kaavojen mukaan olemaan 13206 LOC.

Toteutunut LOC (8.12.2006):

- Java 7000 LOC
- JSP/JS 3800 LOC

Yhteensä: 10800 LOC

6 Työn ositus

Projektin kesto on 15 viikkoa, josta yksi viikko on lomaa. Projektissa seurataan iteratiivista prosessimallia kahdella syklillä. Töiden ositus projektiryhmäläisten kesken pyritään järjestämään siten, että kaikki ryhmäläiset olisivat aktiivisesti mukana tekemässä jokaista projektin vaihetta.

1. sykli:

- Vaatimusmäärittely: 2-3 viikkoa
- Suunnittelu: 2 viikkoa

- Toteutus ja yksikkötestaus: 2 viikkoa
- Integrointi- ja järjestelmätestaus: 1,5 viikkoa
- Demo ja viimeistely: 0,5 viikkoa

2. sykli:

- Vaatimusmäärittely: 1,5 viikkoa
- Suunnittelu: 2 viikkoa
- Toteutus ja yksikkötestaus: 2,5 viikkoa
- Integrointi- ja järjestelmätestaus: 2 viikkoa
- Demo: 0,5 viikkoa
- Projektin viimeistely ja luovutus: 1 viikko

Projektin tarkistuspisteet 1.sykli:

- Ma 4.9. Projektin aloitus
- Ma 18.9. Projektisuunnitelma valmis
- Pe 22.9. Käyttöliittymädemo asiakkaalle
- Ma 25.9. Vaatimusmäärittelyn tarkastus
- To 28.9. Vaatimusmäärittelyn jäädytys
- To 5.10. Suunnitteludokumentin tarkastus ja jäädytys
- Ma 16.10. Ohjelmisto testausvalmiina
- To 19.10. Ohjelmisto valmis ja demottavana

Projektin tarkistuspisteet 2.sykli:

- Ma 30.11. Projektin aloitus
- Pe 3.11. Projektisuunnitelma valmis
- To 9.11. Vaatimusmäärittelyn tarkastus ja jäädytys
- To 16.11. Suunnitteludokumentin tarkastus ja jäädytys
- To 30.11. Ohjelmisto testausvalmiina
- To 7.11. Ohjelmisto valmis ja demottavana asiakkaalle
- To 14.12. Tuotteen luovutus

6.1 Projektin ositus kahteen sykliin

Projekti ositetaan kahteen sykliin. Sykliin välissä on lomaviikko viikolla 43. Kahteen sykliin osittamisen tarkoitus on, että ensimmäisessä syklissä pystytään toteuttamaan projektin suurimmalla prioriteetilla oleva toiminnallisuus. Toisen syklin toteutettavia vaatimuksia voidaan sopeuttaa ensimmäisessä syklissä saatuihin kokemuksiin projektiryhmän tehokkuudesta.

Ensimmäinen sykli:

- Toteutetaan toimiva käyttöliittymä
- Tehtävien tarkistukseen liittyvä toiminnallisuus toteutetaan vain tynkinä
- Integroidaan käyttöliittymä eAssari-kehikseen (tietokantaan)
-

Toinen sykli:

- Toteutetaan tehtävien tarkistus toiminnallisuus
- Mahdollisesti lisätään Titokone-simulaattoriin puuttuvia ominaisuuksia
- Hienosäädetään käyttöliittymää
- Toteutetaan ensimmäisestä syklistä jääneet korkeamman prioriteetin ominaisuudet
-

7 Aikataulu

Projektin ajoituskaavio on esitetty kuvassa 1. Projekti on jaettu kahteen sykliin. Toisen syklin ajoitus tarkennetaan, kun ollaan edetty ensimmäisen syklin loppupuolelle.

[width=15cm]gantkaavio20060918.eps

Kuva 1: Projektin ajoituskaavio

8 Seuranta- ja raportointimenetelmät

Projektiryhmä pitää seurantakokouksen joka maanantai projektin keston aikana. Seurantakokouksessa tarkistetaan, että projektissa on pysytty aikataulussa ja seurattu projektisuunnitelmaa. Seurantakokousten pöytäkirjat lähetetään palaverien jälkeen projektin sähköpostilistalle.

Projektiryhmä pitää yhden tarkastuksen molemmissa sykleissä. Torstaina 16.11. pidetään tarkastus suunnitteludokumentista. Muista suunnittelu- ja vaatimusdokumentista pidetään epämuodollinen tarkastustilaisuus.