

Loppuraportti

Kohahdus

Helsinki 14.12.2006

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 Keskiöja

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	6.12.2006	Ensimmäinen versio
0.2	14.12.2006	Valmis versio

Sisältö

1 Johdanto	1
2 Projektin kulku	1
2.1 Ensimmäinen sykli	1
2.1.1 Vaatimusmäärittely	1
2.1.2 Suunnittelu	2
2.1.3 Toteutus	2
2.1.4 Testaus	2
2.2 Toinen sykli	2
2.2.1 Vaatimusmäärittely	3
2.2.2 Suunnittelu	3
2.2.3 Toteutus	3
2.2.4 Testaus	4
3 Suhteet muihin osajärjestelmiin	4
3.1 Titokone	4
3.2 EAssari-kehys	5
4 Projektityöskentely	5
4.1 Ryhmän henki	5
4.2 Käytetyt työkalut	5
4.3 Ohjaaja	6
4.4 Asiakkaat	6
5 Kehittämisenäkymiä	6
6 Loppusanat	6

1 Johdanto

Toteutimme Ohjelmistotuotantoprojekti-kurssina konekielisen ohjelmoinnin harjoitusympäristön (TitoTrainer).

Järjestelmä sisältää kaksi puolta, opettajan ja opiskelijan käyttöliittymät.

Opettajan on mahdollista luoda konekielisiä tehtäviä ja määritellä niiden tarkastuksessa käytettävät kriteerit. Lisäksi opettaja voi katsoa kurssi- sekä oppilaskohtaisia tilastotietoja.

Opiskelija voi luoda järjestelmään tunnuksen ja ratkaista ohjelmointitehtäviä.

Järjestelmä on tarkoitettu ottaa käyttöön Tietojenkäsittelytieteen laitoksella Tietokoneen toiminta -kurssia luennoitaessa. Kurssin opiskelijat voivat käydä ratkaisemassa järjestelmässä tehtäviä ja luennoitsija saa tästä tilastotietoa.

2 Projektin kulku

Projekti käynnistettiin 4.9.2006. Tällöin pidetyssä aloituspalaverissa vastuualueet jaettiin melko sujuvasti. Projektipäällikön, vaatimusmäärittelyvastaavan ja dokumentointivastaavan paikat jaettiin halukkaille. Suunnitteluvastaavan, koodivastaavan ja testausvastaavan vastuualueita pyöriteltiin aikansa, kunnes yksi kerrallaan ehdokkaat kyllästyivät ja valitsivat jonkin vaihtoehdoista.

Ryhmämme työskenteli suurimmaksi osaksi ryhmätyömuodossa. Aikaa vietettiin paljon yhdessä ja pääasiallisena työskentelytilana oli ATK-luokka Exactumissa. Tässä ATK-luokassa nähtiin lähes joka päivä. Kaikki ryhmän jäsenet osallistuivat ainakin jollain tavalla projektin kaikkiin työvaiheisiin.

2.1 Ensimmäinen sykli

Projektipäällikkö laati alussa aikataulun, jossa pääosin pysyttiin koko projektin ajan. Projektipäällikön ehdotuksesta käyttöön otettiin iteratiivinen prosessimalli, joka osoittautui juuri tälle projektille sopivaksi. Projektin alussa ei olisi ollut riittävästi tietoa vesiputousmallin käyttöönottoon. Ensimmäisen iteraation aikana projekti oli tästä aikataulusta noin viikon myöhässä, mikä näkyi siinä, että testaukseen varattu 1,5 viikkoa kutistui puoleen viikkoon.

2.1.1 Vaatimusmäärittely

Alussa aikaa kului paljon EAssariin ja Titokoneeseen tutustumiseen. Vaatimusmäärittelyyn osallistuivat kaikki, tosin vaatimusvastaavalle kertyi reilusti muita enemmän työtunteja. Vaatimusdokumentista pidettiin epävirallinen katselmustilaisuus, joka paljasti dokumentista useita virheitä ja puutteita.

2.1.2 Suunnittelu

Suunnitteluvaihe pääsi alkamaan hieman myöhässä, ja alussa oli vähän vaikeuksia päästä alkuun, koska osajärjestelmistä ei ollut kattavaa yleiskuvaa. Ensimmäisen iteraation aikana tutustuimme lähinnä EAssariin ja etenkin sen tietokantarakenteeseen. EAssariin tutustuminen oli hidasta ja sen ymmärtäminen kesti kauan. Lisähaittaa aiheutti se että EAssari ei sisältänytäkään sitä mitä oli luvattu ja sen dokumentaatio oli puutteellista. Toisin sanoen EAssari-kehys oli keskeneräinen.

Ensimmäiseksi yritimme miettiä luokkakaavioita, mutta niiden avulla emme päässeet asiassa alkuun. Sitten vain päätettiin että tehdään jotain ja aloimme tekemään sekvenssikaavioita järjestelmän toiminnoista. Sekvenssikaaviot osoittautuivat hyödyllisiksi ja niiden avulla saatiin suhteellisen kattava kuva järjestelmän toiminnasta.

Myös suunnitteluun osallistuivat kaikki, mutta suunnitteluvastaavan panos oli kuitenkin ratkaiseva.

2.1.3 Toteutus

Toteutusvaihe pääsi alkamaan viikon myöhässä aiempien vaiheiden myöhästelyn vuoksi. Ensimmäisen toteutusvaiheen hidastavana elementtinä oli alkuun se, että osa ryhmästä joutui opettelemaan JSP:tä ja JSTL:ää. Samaten Eclipseen ja CVS:ään tutustuttiin kunnolla vasta toteutuksen alkaessa. Tutustumiseen käytetyn ajan vuoksi testausvaiheen aloitus venyi hieman aikataulustaan.

Tehtävien jaossa pyrimme huomioimaan ryhmäläisten vahvuudet. Osa ryhmäläisistä alkoi tekemään JSP-sivuja ja osa sai tehtäväkseen java-luokkia ja yksi alkoi rakentamaan JavaScript- ja käyttöliittymäkomponentteja. Alun hankaluuksien ja hitaahkon käynnistymisen jälkeen päästiin kuitenkin vauhtiin, ja ryhmä onnistui tuottamaan ohjelmakoodia tehokkaasti.

2.1.4 Testaus

Ensimmäisessä iteraatiossa testaus jäi aikataulun sanelemana aika vähiin. Toteutusvaihe venyi yli osuutensa aikataulusta, ja testausvaiheelle jäi ensimmäisessä iteraatiossa paljon vähemmän aikaa kuin oli alkuun suunniteltu. Tämä pyrittiin huomioimaan toisen iteraation testausvaiheen suunnittelussa.

Testaussuunnitelma saatiin tehtyä ja toteutusvaiheen yhteydessä suoritettiin kattavia yksikkötestejä.

2.2 Toinen sykli

Projektimme sai pitää ansaitun lomaviikon yliopiston taukoviikolla. Toisessa iteraatiossa työkalut ja menetelmät olivat jo tuttuja, ja työt käynnistyivät huomattavasti rivakammin.

Toisen iteraation alussa esiteltiin asiakkaalle projektin sen hetkistä tuotosta, joka jo sisällsi toimivan opettajan käyttöliittymän ilman vastattujen tehtävien tilastointia. Asiakkaan palautteen perusteella tehtiin joitain muutoksia ja otettiin muutama uusi vaatimus.

2.2.1 Vaatimusmäärittely

Toisen iteraation vaatimusmäärittely meni nopeasti, koska ensimmäisessä iteraatiossa oli jo määritelty suurin osa koko projektin vaatimuksista. Toisessa iteraatiossa tehtiin tarkka syötteet ja tulosteet dokumentti, joka tarkasti määritteli millaisia syötteitä järjestelmä hyväksyy.

2.2.2 Suunnittelu

Toisessa suunnitteluvaiheessa täydennettiin suunnitteludokumenttia ja pidettiin tästä virallinen katselmus. Virallinen katselmus oli aika vaisu tilaisuus, ja totesimme että vaatimusdokumentin epävirallinen katselmus oli tätä virallista parempi.

Toisen iteraation suunnitteluvaiheessa päätimme hylätä EAssarin (paitsi tietokantaa), koska se ei toteuttanut tarpeellisia toimintoja. Tätä ei pidetty mitenkään suurena asiana, mutta asiakkaan kanssa pidettiin kuitenkin kokous aiheesta. Kokouksessa päästiin yhteisymmärrykseen asiakkaan ja EAssari-asiantuntijan kanssa siitä, että projektiryhmän valitsema tie on oikea.

2.2.3 Toteutus

Toisen iteraation toteutuksessa olimme jo kokeneempia kuin ensimmäisessä iteraatiossa. Välineet olivat jo käyneet tutuiksi ja kaikilla oli suhteellisen hyvä kuva toteutettavasta järjestelmästä.

Toteutusvaiheessa ei enää kohdattu suurempia ongelmia. Osittain tähän varmasti vaikutti ryhmän aiempi päätös jättää EAssari-yhteensopivuus (paitsi tietokannan kanssa) projektin ulkopuolelle. Toteutusvaiheelle on vaikeaa määrittää selkeää lopetuspäivää, koska viilautta voisi jatkaa loputtomiin. Kaikki korkeimman prioriteetin vaatimukset saatiin kuitenkin toteutettua aikataulun mukaisesti. Toteutusta jatkettiin vielä limittäin testauksen kanssa ja lopulta saatiin lähes kaikki vaatimukset toteutettua.

Järjestelmän asentamisessa lopulliseen ympäristöönsä koettiin pieniä ongelmia siinä, että järjestelmän kehitys- ja tuotantoympäristö eivät olleet samanlaiset. Kehitysympäristössä oli Javan versio 1.5, kun taas tuotantoympäristössä on käytössä Java 1.4.2. Tätä olisi ollut syytä selvittää etukäteen, mutta toisaalta asiakaskaan ei tiennyt kuin vasta ihan lopussa minne lopullinen järjestelmä asennettaisiin. Näin lopuksi oli pakko muuttaa ohjelmakoodi vielä Java 1.4.2 yhteensopivaksi.

2.2.4 Testaus

Toisessa iteraatiossa oli varattu testauksellekin kunnolla aikaa ja se päästiin aloittamaan aikataulun mukaisesti. Kaikki ryhmän jäsenet osallistuivat aktiivisesti testaukseen ja testauksen yhteydessä käytetty bugilista osoittautui hyödylliseksi työvälineeksi.

Testauksen kokonaiskattavuus saatiin lopulta aika korkeaksi, ehkä noin 85

Jäsenet	Kirjattu	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	Yhteensä
Jari Inkinen	14.12.06	6	19	11	14	14	25	15		12	13,5	11	18	16	21	16,5	212
Mikko																	
Kinnunen	14.12.06	5,5	14,5	25	24,5	22	29,5	15		11,5	13,5	13	14,5	15	20,5	15,5	239,5
Markus Kivilä	14.12.06	11	13	14	20,5	16	25	18,5		11,5	11,5	12,5	13,5	12,5	19,5	15,5	214,5
Paula																	
Kuosmanen	14.12.06	9	16,5	20,5	5,5	6,5	8,5	7,5	1	10,5	8	28	41	33	19	12,5	227
Taro Morimoto	14.12.06	8,5	15	15,5	12	14,5	8	16,5		13	13	12,5	16,5	22	18,5	15	200,5
Tuomas																	
Palmanto	14.12.06	14	20	28	17	18	19,5	12	4	11	6,5	6,5	13	13,5	16	12,5	211,5
Yhteensä	1	54	98	114	93,5	91	115,5	84,5	5	69,5	66	83,5	116,5	112	114,5	80,5	1298

Kuva 1: Projektin kirjatut työtunnit

3 Suhteet muihin osajärjestelmiin

Titotrainerilla oli alun pitäen kaksi ulkoista osajärjestelmää Titokone ja EAssari. Tarkoitus oli että EAssarista olisi tullut alusta, jonka päälle Titotrainer voidaan rakentaa. Titokonetta puolestaan käytetään konekielisten ohjelmien kääntämiseen ja tarkastamiseen. EAssarin käytöstä kuitenkin myöhemmin kuvatulla tavalla osittain luovuttiin.

3.1 Titokone

Titokone oli hyvin toteutettu ja se oli helppokäyttöinen. Titokoneen integroimisessa ei kohdattu lähes ollenkaan ongelmia. Koskelo-ryhmä oli jo tehnyt Titokoneeseen erinäisiä

muutoksia, mutta pieniä korjauksia jouduttiin silti tekemään. Käytännössä muutoksia oli noin parikymmentä riviä. Kokonaisuudessaan olimme hyvin tyytyväisiä Titokoneeseen.

3.2 EAssari-kehys

Tehtävänannossamme sanottiin seuraavasti: "Projekti rakennetaan valmiina olevaan eAssari-kehykseen, joka sopii generisiin automaattisesti tarkistettavien tehtävien määrittelyyn ja toteutukseen." Aluksi näyttikin tältä, ja luulimme että EAssari mahdollistaa nämä kuvattut toiminnot. Projektiryhmän käsitys valmiista kehyksestä kuitenkin poikkeaa huomattavasti tästä. Koska projektin tavoitteena oli luoda järjestelmä, joka voitaisiin ottaa käyttöön nopeasti projektin päättymisen jälkeen, niin totesimme parhaaksi ratkaisuksi jättää kehys täysin projektin ulkopuolelle. Käytimme kuitenkin EAssarin tietokantaa sellaisenaan, jotta Titotrainer olisi tulevaisuudessa mahdollista siirtää jatkokehitettyyn EAssari-kehykseen.

EAssarista puuttuu mm. käyttäjän rekisteröityminen, käyttäjän tietojen muokkaaminen, käyttäjän kirjautuminen, kurssien ja moduulien hallinnointi, tehtävien selaaminen, suoritustilastojen katsominen ja monikielisyyden hallinta. Kaikki listassa mainitut toiminnot kuuluvat mielestämme jokaiseen tämän tyyppiseen valmiiksi kuvattuun kehykseen. Koska toimintoja ei ollut, jouduimme toteuttamaan ne itse. Toteutuksessa ei kuitenkaan otettu huomioon EAssarin teoriassa mahdollistamaa geneerisyyttä, vaan toteutetut ominaisuudet ovat TitoTrainer-kohtaisia.

4 Projektityöskentely

4.1 Ryhmän henki

Ryhmän henki ja ilmapiiri oli erinomainen. Ryhmäläiset tukivat toisiaan, ja apua sai aina kun sitä tarvitsi. Kemia ryhmän jäsenten välillä toimi moitteettomasti - kaikki tulivat toimeen kaikkien kanssa.

4.2 Käytetyt työkalut

Projektin alussa käytettiin ryhmälle pystytettyä Wikiä, jonka käytöstä kuitenkin luovuttiin aika pian. Wikiä ei tarvittu, koska ryhmä tosiaan teki lähes 90% tuotoksesta yhdessä laitoksen luokissa.

Ohjelmointiympäristönä käytettiin Eclipseä ja sen kautta käytettiin CVS-versionhallintaa. Eclipsen käyttö sujui kohtuullisen hyvin ryhmäläisiltä pienen totuttelun jälkeen.

Dokumentit tehtiin pääsääntöisesti L^AT_EX:lla ja muutama Open Officella. L^AT_EX:in käyttökokemusta löytyi ennen projektia vähänlaisesti, joten siihen tutustumiseen meni hieman aikaa. Osa ryhmäläisistä meinasi kypsyä L^AT_EX:in käyttämiseen projektin kuluessa, mutta ainakin kaikki saivat sen käyttökokemusta. Etenkin kuvien sijoittelun kanssa oli suuria

ongelmia.

4.3 Ohjaaja

Ohjaajamme Sanna Kesquioja oli erittäin pätevä ja asiansa hallitseva. Sannalta saimme rakentavaa tukea kokouksissa ja hän hoiti hyvin tarvitsemiemme tilojen ja resurssien hankkimisen.

Sannalla oli hyvä asenne projektiin, ja olisi ollut vaikeaa kuvitella Sannaa parempaa ohjaajaa tälle projektille.

4.4 Asiakkaat

Asiakkaallamme Teemu Kerolalla oli selkeä näkemys tarvittavasta tuotteesta, ja Teemu osasi selkeästi tuoda julki järjestelmän yksityiskohtaiset tarpeet. Osaltaan tähän varmasti vaikutti, että järjestelmä on ollut ohjelmistotuotantoprojektin aiheena myös aiemmin. Pyrimme hyödyntämään asiakkaan palautetta projektin eri vaiheissa. Iteratiivisen prosessimallin vuoksi kaikkia vaatimuksia ei ollut tarpeen jäädättää heti vaatimusmäärittelyn jälkeen, joten asiakkaalla oli mahdollisuus tarkentaa omia näkemyksiään ja tarpeitaan.

5 Kehittämisenäkymiä

Koska projekti luopui EAssari-yhteensopivuuden takaamisesta, niin järjestelmien integroiminen ei ole aivan yksinkertaista. Titotrainer ja nykyinen EAssari poikkeavat toiminnaltaan lähes täysin. Ainoa yhteinen elementti on EAssarin tietokanta, johon on tehty vain pieniä muutoksia (lisätty sequence ja kasvatetty yhtä kenttää). Titotrainer ei kuitenkaan käytä tietokantaa sen mahdollistamassa laajuudessa ja muutamia sen mahdollistamia ominaisuuksia on yksinkertaistettu.

6 Loppusanat

Omasta mielestämme onnistuimme projektissa kohtuullisen hyvin. Asiakas oli tuotteen tyytyväinen ja ainakin tätä raporttia kirjoittaessa näytti siltä, että järjestelmä tullaan todella ottamaan käyttöön.

Projektista opittiin melko paljon. Tutuiksi tulivat CVS niille joille se ei jo ennestään ollut tuttu, ja opittiin myös tuntemaan JSP, JSTL, Eclipse, Tomcat-palvelinympäristö.

Ensiarvoisen tärkeää oli että projektissa oli mukana henkilö (Taro), joka tunsu hyvin Tomcat-ympäristön, Eclipsen ja JSP:n. Näin muut ryhmäläiset saivat tehokkaan pikarehdytyksen ennestään tuntemattomiin tekniikoihin.

Projektiryhmämme oli ilmeisesti koottu sillä perusteella, että kaikki olivat suorittaneet Käyttöliittymät-kurssin. Käyttöliittymä olikin priorisoitu tärkeimmäksi yksittäiseksi osaksi järjestelmää. Käyttöliittymät-kurssin suorittamisella ei kuitenkaan ollut ratkaisevaa merkitystä projektin onnistumiselle, vaikkakin sen suorittamisesta oli selkeästi hyötyä projektissa.