

Projektisuunnitelma

Oppimistavoitteiden hallintajärjestelmä harri

Helsinki 12.10.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9 op)

Projektiryhmä

Petri Kinnunen

Lasse Leino

Anne Pääkkö

Minna Ulmala

Asiakas

Harri Laine

Johtoryhmä

Kimmo Simola, vastuhenkilö

Aleksi Yrttiaho, ohjaaja

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
1.1	11.10.2007	Suunnitteluvaihetta, aikataulua ja kustannusarvioita tarkennettu
1.0	1.10.2007	Projektisuunnitelma valmis
0.4	30.9.2007	Koko- ja kustannusarviot, riskianalyysin tarkennus
0.3	29.9.2007	Huomattavia tarkennuksia työn ositukseen ja aikatauluun
0.2	24.9.2007	Muutettu L ^A T _E X-muotoon ja tarkennuksia työn ositukseen
0.1	17.9.2007	Ensimmäinen versio

Sisältö

1 Johdanto	1
2 Sanasto	1
3 Projektioorganisaatio	1
3.1 Henkilöt	1
3.2 Vastuualueet	2
3.3 Vastuualueiden kuvaukset	2
4 Riskianalyysi	3
4.1 Asteikot	3
4.2 Projektikohtaiset riskit	3
4.3 Tuotekohtaiset riskit	6
5 Laitteisto- ja ohjelmistoympäristön vaatimukset	6
6 Koko- ja kustannusarviot	6
6.1 Toimintopisteet	7
6.2 Funktiopistearvo	8
6.3 Koodirivien lukumäärä arvio	8
7 Dokumentit	9
8 Työn ositus	9
8.1 Vaatimusmäärittely	9
8.1.1 Vaatimusten esiinkaivelu	10
8.1.2 Vaatimusten luokittelu ja priorisointi	10
8.1.3 Vaatimusten validointi	11
8.2 Suunnittelu	11
8.2.1 Arkkitehtuurisuunnittelu	11
8.2.2 Rajapintasuunnittelu	11
8.2.3 Komponenttisuunnittelu	11
8.2.4 Olios suunnittelu	12
8.2.5 Testaussuunnitelma	12

	ii
8.3 Toteutus- ja yksikkötestaus	12
8.4 Integrointi ja järjestelmätestaus	12
8.5 Projektin viimeistely ja luovutus	12
9 Aikataulu	13
9.1 Projektin työkulku	13
10 Seuranta- ja raportointimenetelmät	14
10.1 Palaverit	14
10.2 Seurantakatsaus	14
10.3 Projektisuunnitelman päivitys	14

1 Johdanto

Helsingin yliopiston Tietojenkäsittelytieteen laitoksen kursseilla ollaan ottamassa käyttöön kirjattuja oppimistavoitteita. Kurssilla voi olla eritasoisia oppimistavoitteita, joita voidaan käyttää myös muiden kurssien tai niiden osien esitietovaatimuksina. Oppimistavoitteita voidaan määrittellä hierarkisesti ja niihin voidaan liittää kysymyksiä tai täsmäntäviä esimerkkejä asioista, joita oppimistavoitteeseen sisältyy. Tavoitteita pystytään määrittelemään joko kurssitasolla tai erikseen kurssin osalle.

Kehitettävä järjestelmä on tarkoitettu apuvälineeksi oppimistavoitteiden määrittelyyn ja ylläpitoon. Sen avulla on tarkoitus valvoa oppimistavoitteiden ja esitietovaatimusten yhdenmukaisuutta sekä oppimistavoitteiden ja kurssisisällön vastaavuutta. Järjestelmän on tarkoitus tarjota myös helppo näkymä oppimistavoitteisiin ja esitietovaatimuksiin, sekä kurssikytkentöihin.

Järjestelmä tarjoaa välineistöä kurssien luennoitsijoille sekä myös opiskelijoille. Luennoitsijat voivat hyödyntää muiden kurssien oppimistavoitteita esitietovaatimuksia määrittelyssään. Opiskelijat voivat tutkia oppimistavoitteita ja sitä kautta arvioida esitietojensa riittävyyttä. Kurssille voidaan generoida oppimistavoitepohjaisia palautelomakkeita.

Järjestelmän prototyypin teko aloitettiin 3.9.2007 ja se valmistuu 14.12.2007. Projektiin kuluu 14 viikkoa (1. periodi 7 viikkoa, yksi väliviikko, 2. periodi 7 viikkoa).

2 Sanasto

Projektitiimi, tiimi, jäsenet, ryhmä, ryhmän jäsenet

Niillä tarkoitetaan projektioorganisaation varsinaisia toteuttavia osapuolia.

Ohjelmisto, järjestelmä

Tarkoitetaan toteutettavaa prototyyppiä.

3 Projektioorganisaatio

Seuraavaksi esitellään projektioorganisaatioon kuuluvat henkilöt, vastualueet sekä niiden kuvaukset.

3.1 Henkilöt

Projektioorganisaatioon kuuluu neljä varsinaista projektitiimin jäsentä, asiakas sekä projektin ohjaaja ja Ohjelmistotuotantoprojektin vastuuhenkilö.

Nimi	Rooli	Sähköposti	Puhelin
Lasse Leino	Tiimi	ltleino@cs.helsinki.fi	(044) 2939 612
Minna Ulmala	Tiimi	ulmala@cs.helsinki.fi	(050) 095 2323
Anne Pääkkö	Tiimi	apaakko@cs.helsinki.fi	(050) 407 1746
Petri Kinnunen	Tiimi	pskinnun@cs.helsinki.fi	(040) 731 1883
Harri Laine	Asiakas	laine@cs.helsinki.fi	(09) 191 51149
Aleksi Yrttiaho	Ohjaaja	ayrttiah@cs.helsinki.fi	-
Kimmo Simola	Vastuuhenkilö	kksimola@cs.helsinki.fi	-

3.2 Vastuualueet

Vastuu projektitiimin jäsenten välillä jakautuu seuraavasti.

Vastuualue	Vastuuhenkilö
Projektipäällikkö	Lasse Leino
Varaprojektipäällikkö	Anne Pääkkö
Vaatusmäärittelyvastaava	Minna Ulmala
Suunnittelu- ja testausvastaava	Anne Pääkkö
Dokumentti- ja koodivastaava	Petri Kinnunen
Kotisivu- ja SVN-vastaava	Petri Kinnunen

3.3 Vastuualueiden kuvaukset

Projektipäällikkö

Vastuussa projektin johtamisesta. Tekee projektisuunnitelman ja loppuraportin, sekä pitää huolta siitä, että projekti etenee projektisuunnitelman mukaan ja pysyy aikataulussa.

Vaatusmäärittelyvastaava

Toimii yhteyshenkilönä asiakkaan ja ryhmän välillä, sekä vastaa vaatimusdokumentin rakenteesta ja eheydestä. Vastaa vaatimusdokumentin hyväksymiskelpoisuudesta.

Suunnittelu- ja testausvastaava

Vastaa yhtenäisistä suunnittelutason rajapinnoista ja suunnitteludokumentin rakenteesta, sekä testauksen kattavuudesta ja testaussuunnitelmasta. Toimii suunnitteludokumentin tarkastustilaisuuden puheenjohtajana.

Dokumentti- ja koodivastaava

Vastaa dokumenttien ja koodin yhtenäisestä ulkoasusta, sekä ohjelman rajapintojen yhtenäisyydestä.

Kotisivu- ja SVN-vastaava

Vastaa SVN:n ylläpidosta, hakemistorakenteesta ja opettamisesta ryhmäläisille sekä kotisivujen päivityksestä kurssin aikana kertyvän materiaalin osalta.

Puheenjohtaja

Tekee vastuullaan olevaan palaveriin esityslistan ja toimittaa sen kokousta edeltävänä päivänä ryhmän jäsenille. Yleensä palaverissa puheenjohtajana toimii projektipäällikkö. Poikkeuksena palaverit, joissa toimitaan jonkun muun ryhmän jäsenen vastuualueella

(dokumenttien tarkastukset).

Sihteeri

Sihteeri vaihtuu aina joka palaverin jälkeen. Kirjoittaa kokouksista pöytäkirjan, jonka välittää sähköpostilla ryhmän jäsenille.

4 Riskianalyysi

4.1 Asteikot

Riskianalyysin todennäköisyyksien hahmottamisessa:

- erittäin pieni
- pieni
- keskiveroinen
- suuri
- erittäin suuri

Vastaavasti vakavuusasteen hahmottamisessa:

- mitätön
- pieni
- keskiveroinen
- suuri
- fataali

4.2 Projektikohtaiset riskit

Riski: Projektin työ- ja kustannusarviot eivät pidä paikkaansa

Todennäköisyys: erittäin suuri

Vakavuusaste: mitätön

Vastatoimet: Seurataan projektin etenemistä sekä ryhmäläisten työntekoa ja tarkennetaan projektisuunnitelmaa sen ollessa tarpeellista.

Riski: Projekti ei pysy aikataulussa

Todennäköisyys: suuri

Vakavuusaste: suuri

Vastatoimet: Pyritään raportoimaan mahdollisista viivästyksistä niin pian kuin mahdollista, jotta asiaan voidaan reagoida nopeasti.

Riski: Projekti ei valmistu määräaikaan mennessä

Todennäköisyys: keskiveroinen

Vakavuusaste: suuri

Vastatoimet: Pyritään saamaan toteutusvaiheessa järjestelmän runko valmiiksi niin, että se toteuttaa kaikki prioriteetiltaan tärkeimmät toiminnot. Näin säilytetään mahdollisuus, että projektia voi jatkaa jokin muu ryhmä myöhemmin.

Riski: Tiimin jäsen sairastuu

Todennäköisyys: keskiveroinen

Vakavuusaste: keskiveroinen

Vastatoimet: Asiasta ilmoitetaan välittömästi, jotta työ voidaan jakaa tasaisesti muiden ryhmän jäsenten kesken ja työtehtävät tulee tehtyä.

Riski: Tiimin jäsen keskeyttää projektin

Todennäköisyys: erittäin pieni

Vakavuusaste: fataali

Vastatoimet: Ei oteta projektia liian vakavasti vaan ohjelmistoprosessin oppimisen kannalta ja pidetään yllä rentoa, mutta kuitenkin vastuullista ilmapiiriä.

Riski: Kommunikointi jäsenten välillä ei ole riittävää

Todennäköisyys: keskiveroinen

Vakavuusaste: suuri

Vastatoimet: Etsitään sähköpostin lisäksi kaikille sopiva kommunikointikanava, jolla voidaan harjoittaa intensiivisempää kommunikointia. Vaihtoehtoisesti järjestetään useampia tapaamisia palaverien ulkopuolella, joissa tehdään yhdessä työtä.

Riski: Muut kiireet (työ, harrastus) syövät projektin jäsenten aikaa

Todennäköisyys: suuri

Vakavuusaste: keskiveroinen

Vastatoimet: Ilmoitetaan ryhmälle kiireellisestä aikataulusta, jotta työtä voidaan tarvittaessa jakaa muiden ryhmäläisten kesken.

Riski: Sovitut työt eivät syystä tai toisesta tule tehdyksi

Todennäköisyys: keskiveroinen

Vakavuusaste: suuri

Vastatoimet: Pyritään projektin alusta alkaen siihen, että jokainen tiimissä tekee asioita oma-aloitteisesti. Tarvittaessa pyydetään muilta apua, jotta työtehtävä voidaan viedä päätökseen.

Riski: Sovitut työt eivät tule tehdyksi, koska alue on tiimin jäsenelle vieras

Todennäköisyys: suuri

Vakavuusaste: keskiveroinen

Vastatoimet: Pyritään pitämään rentoa ilmapiiriä ryhmässä, jotta kommunikointi on avointa ja tiimin jäsen voi rehellisesti ilmoittaa tarvitsevänsä apua.

Riski: Ryhmäläiset menettävät kiinnostuksensa projektiin

Todennäköisyys: pieni

Vakavuusaste: fataali

Vastatoimet: Ei oteta projektia liian vakavasti vaan oppimisen kannalta ja pidetään yllä rentoa, mutta kuitenkin vastuullista ilmapiiriä.

Riski: Töiden jakamisessa on epäselvyyttä

Todennäköisyys: keskiveroinen

Vakavuusaste: pieni

Vastatoimet: Seurataan projektisuunnitelman työn ositusta, jotta töiden jakaminen ryhmäläisten kesken helpottuu.

Riski: Yksittäisen vaiheen valmistumisen myöhästyminen

Todennäköisyys: keskiveroinen

Vakavuusaste: erittäin suuri

Vastatoimet: Jaetaan työtä tasaisesti ryhmän kesken ja pyritään ennakoimaan mahdolliset viivästykset. Muodostetaan projektin aikataulu niin, että vaiheita voidaan tarvittaessa venyttää ilman koko projektin viivästymistä.

Riski: Asiakas ei osaa kuvata järjestelmää tarpeeksi tarkasti

Todennäköisyys: suuri

Vakavuusaste: suuri

Vastatoimet: Konsultoidaan myös asiantuntijoita järjestelmän vaatimusmäärittelyvaiheessa. Käytetään vaatimusmäärittelyyn paljon aikaa, jotta suunnitteluvaiheessa ei ole enää epäselvyyttä järjestelmän rakenteesta.

Riski: Kommunikointi asiakkaan kanssa ei ole riittävää

Todennäköisyys: pieni

Vakavuusaste: suuri

Vastatoimet: Haastattelujen lisäksi pidetään asiakkaaseen yhteyttä myös haastattelujen ulkopuolella.

Riski: Kommunikointi asiakkaan ei onnistu asiakkaan kiireiden takia

Todennäköisyys: pieni

Vakavuusaste: suuri

Vastatoimet: Pyritään löytämään yhteisiä tapaamisaikoja ja joustetaan tarvittaessa omasta viikkoaikataulusta.

Riski: Integrointi Opetuksensuunnitelujärjestelmään osoittautuu vaikeaksi

Todennäköisyys: keskiveroinen

Vakavuusaste: erittäin suuri

Vastatoimet: Konsultoidaan asiantuntijoita asiasta ja keskustellaan asiakkaan kanssa tarkasti millä tasolla integrointi tulisi tehdä ja mitä tietoja järjestelmästä haetaan.

4.3 Tuotekohtaiset riskit

Riski: Tuote ei ole sitä mitä asiakas halusi

Todennäköisyys: keskiveroinen

Vakavuusaste: suuri

Vastatoimet: Valmistellaan haastattelutilaisuudet huolella ja käytetään vaatimusmäärittelyvaiheeseen tarvittaessa enemmän aikaa. Kommunikoidaan asiakkaan kanssa ongelmatilanteissa myös haastattelujen ulkopuolella.

Riski: Käyttöliittymä osoittautuu vaikeakäyttöiseksi

Todennäköisyys: keskiveroinen

Vakavuusaste: suuri

Vastatoimet: Suunnitellaan useita rinnakkaisia käyttöliittymäratkaisuja, jotta niiden käytettävyyttä voidaan vertailla. Rakennetaan niiden pohjalta protyyppejä, joita voidaan testata käytännössä.

5 Laitteisto- ja ohjelmistoympäristön vaatimukset

Järjestelmästä kehitettävä protyyppi ohjelmoidaan Javalla ja se käyttää hyväkseen tietokantayhteyttä. Käyttöliittymä toteutetaan selainympäristöön.

Järjestelmän lähdekoodin ja dokumenttien jakeluun käytetään SVN-versionhallintaa tiimin ryhmähakemistoa hyväksikäyttäen. Kotisivujen kehitys on myös versionhallinnan sisällä, sillä erolla, että niiden kehitys hoidetaan ryhmähakemiston sisällä. Järjestelmän lähdekoodin tekemiseen käytetään Eclipseä SVN-pluginin kera. Kaikki dokumentit ladotaan lopulliseen muotoonsa \LaTeX :illa. Muistiot ja pöytäkirjat saavat olla ASCII-tekstinä.

6 Koko- ja kustannusarviot

Seuraavaksi esitellään arvio projektiryhmän tuotettavasta koodimäärästä. Arviointi tehdään toimintopisteiden (FP, Function Points) avulla, joiden perusteella lasketaan tehtävien koodirivien määrä (SLOC, Source lines of code).

6.1 Toimintopisteet

Ulkoiset syötteen	Kerroin
Haku	4
Pääteeman lisäys, poisto ja muokkaus	3
Alateeman lisäys, poisto ja muokkaus	3
Oppimistavoitetekstin lisäys, poisto ja muokkaus	3
Oppimistavoitteen tason valinta	3
Esimerkin lisäys, poisto ja muokkaus	3
Tenttikysymyksen lisäys, poisto ja muokkaus	3
Esitietovaatimuksen lisäys, poisto ja muokkaus	3
Esitietovaatimuksen lisäys listasta	3
Palautekysymyksen lisäys, poisto ja muokkaus	3
Kurssin tilan muutos	3
Muokkaajan, vastuuhenkilön ja tarkastajan nimeäminen	3
Yhteensä 11 kpl	$\Sigma=34$
Ulkoiset tulosteet	Kerroin
Kurssin löytäminen listasta	4
Haku	5
Esitietovaatimusten valinta	4
Pääteeman lisäys, poisto ja muokkaus	4
Alateeman lisäys, poisto ja muokkaus	4
Oppimistavoitetekstin lisäys, poisto ja muokkaus	4
Oppimistavoitteen tason valinta	4
Esimerkin lisäys, poisto ja muokkaus	4
Tenttikysymyksen lisäys, poisto ja muokkaus	4
Esitietovaatimuksen lisäys, poisto ja muokkaus	4
Esitietovaatimuksen lisäys listasta	4
Ilmoitus muutoksesta	5
Palautekysymyksen lisäys, poisto ja muokkaus	4
Tavoitepohjaisen palautelomakkeen generointi	5
Kurssin tilan muutos	4
Muokkaajan, vastuuhenkilön ja tarkastajan nimeäminen	4
Yhteensä 16 kpl	$\Sigma=67$
Ulkoiset kyselyt	Kerroin
Kirjautuminen	3
Uloskirjautuminen	3
Haku	4
Esitietovaatimusten valinta	3
Tavoitepohjaisen palautelomakkeen generointi	4
Yhteensä 5 kpl	$\Sigma=17$

Sisäinen data (Tietokanta)	Kerroin
Kurssitaulu	7
Teemataulu	7
Oppimistavoitetaulu	7
Palautekysymystaulu	7
Tenttikysymystaulu	7
Esimerkkitaulu	7
Käyttäjätaulu	7
Yhteensä 7 kpl	$\Sigma=49$

Ulkoinen data (Opetuksensuunnitelujärjestelmä)	Kerroin
Kurssitiedot	7
Käyttäjätiedot	7
Yhteensä 2 kpl	$\Sigma=14$

Järjestelmän yleispiirteet	Kerroin
Tiedonvälitys	2
Hajautettu laskenta	0
Suorituskyky	2
Suoritusympäristön käyttöaste	2
Tiedonvälityksen aste	4
Reaaliaikainen tietojen syöttö	5
Loppukäyttäjätehokkuus	2
Reaaliaikainen päivitys	5
Prosessoinnin vaativuus	0
Käytettävyys	5
Konversion ja asennuksen helppous	2
Käyttäjystävällisyys	5
Useita kohdeorganisaatioita	1
Kehitettävyys	2
Yhteensä 14 kpl	$\Sigma=37$

6.2 Funktiopistearvo

Toimintopisteitä tulee yhteensä 55 kappaletta ja niiden kertoimien summaksi saadaan 181.

Funktiopistearvoksi tulee täten $181 * (0.65 + 0.01 * 37) = 181 * 1,02 = \underline{184,62}$.

6.3 Koodirivien lukumäärä arvio

Johdetaan LOC-arvio toimintopistemethoden avulla. Ohjelmiston toteutuskielenä käytetään seuraavaa yhdistelmää: Java + HTML + CSS + SQL. Javan FP/SLOC arvo on 31 ja sen osuus toteutuskielistä on selvästi suurin.

LOC saa täten arvokseen $31 * FP = 31 * 184,62 = \underline{5723,22}$.

Se tekee noin 1430 (eli $5723,22/4$) riviä koodia per jäsen.

Tilastollisesti Ohjelmistotuotantoprojekteissa kukin ryhmän jäsen kirjoittaa 300-350 riviä koodia per kuukausi, eli 1050-1225 riviä koko projektissa. Tämä tarkoittaisi, että neljän hengen ryhmä kirjoittaisi tilastollisesti noin 4200-4900 riviä koodia.

Perustoiminnallisuuden (ensimmäisen prioriteetin toiminnot) toteuttaminen onnistunee noin 5000 rivillä koodia, jonka projektitiimi saavuttaa tilastoihin nojaten 84-98% todennäköisyydellä. 500 rivin lisäys ennusteeseen vähentää toteutumistodennäköisyyttä noin 7-9%. Projektin koko toiminnallisuuden tekemiseen menisi luultavasti noin 6000 riviä koodia, joten todennäköisyys sille, että projektin saataisiin toimintojen osalta täysin valmiiksi on 70-81%.

7 Dokumentit

Projektin aikana laadittavat sisäiset dokumentit:

- Projektisuunnitelma
- Suunnitteludokumentti
- Testaussuunnitelma
- Yhteenvetodokumentti

Projektin aikana laadittavat ulkoiset dokumentit:

- Vaatimusdokumentti
- Käyttöohje
- Ylläpitodokumentti

8 Työn ositus

Projektissa noudatetaan vesiputousmallia ja se jakautuu seuraavaksi esiteltäviin vaiheisiin. Vaiheet jakautuvat osavaiheisiin, joissa määritellään tarkemmin miten kyseisen prosessi viedään läpi. Kun vaihe saadaan valmiiksi, sitä käytetään seuraavan vaiheen lähtökohtana, eikä edeltävään vaiheeseen enää palata.

8.1 Vaatimusmäärittely

Vaatimusmäärittely on iteratiivinen prosessi seuraavien vaiheiden välillä. Työvaiheen loppuun saadaan tuloksena vaatimusmäärittelydokumentti.

Tässä kappaleessa dokumentilla viitataan aina vaatimusdokumenttiin.

8.1.1 Vaatimusten esiinkaivelu

Vaatimusten esiinkaivelu suoritetaan pääosin asiakas- ja asiantuntijahaastatteluilla.

Haastatteluja järjestetään kolme kappaletta:

Asiakashaastattelu - Pidetty 13.9.2007

Asiakashaastattelu tehdään asiakastapaamisessa, jonka tuloksena pyritään keräämään suurin osa järjestelmän vaatimuksista. Haastattelun pohjalta selvitetään:

- Järjestelmän yleinen rakenne
- Käytettävä sanasto
- Sidosryhmät

Haastattelun jälkeen siirrytään kerättyjen vaatimusten luokitteluun ja priorisointiin.

Asiantuntijahaastattelu - Pidetty 24.9.2007

Asiantuntijatapaamisen tarkoitus on antaa projektiryhmälle kokonaiskuva oppimistavoitteista ja niiden tekoprosessista. Haastattelun pohjalta tarkennetaan asiakastapaamisessa kerättyjä vaatimuksia.

Haastattelun jälkeen siirrytään tarkennettujen vaatimusten luokitteluun ja priorisointiin, sekä aloitetaan käyttöliittymäprototyypien suunnittelu.

Toinen asiakashaastattelu - Pidetty 4.10.2007

Haastattelussa käydään läpi dokumentin ensimmäinen versio yhdessä asiakkaan kanssa ja esitellään käyttöliittymäprototyypit asiakkaalle. Haastattelun tarkoituksena on varmistaa asiakkaalta, että vaatimukset ovat ymmärretty oikein.

Haastattelun jälkeen projektitiimi siirtyy validointivaiheeseen, sekä jatkaa käyttöliittymäprototyypien suunnittelua.

8.1.2 Vaatimusten luokittelu ja priorisointi

Selvitetyille sidosryhmille tehdään käyttötapauksia, joiden pohjalta laaditaan järjestelmävaatimuksia. Järjestelmävaatimukset luokitellaan toiminnallisiin ja ei-toiminnallisiin vaatimuksiin ja ne priorisoidaan siten, että kun ylimmän tason prioriteetit on toteutettu, ohjelman runko on toiminta kunnossa.

Käyttäjävaatimuksia ei tehdä, koska asiakas toimii samalla myös asiantuntijan roolissa. Täten asiakkaalla ole ongelmaa järjestelmävaatimuksien ymmärtämisessä.

Järjestelmävaatimusten tarkentuessa järjestelmäarkkitehtuuri, sekä järjestelmämalli dokumentoidaan selventävien kaavioiden kanssa dokumenttiin.

8.1.3 Vaatimusten validointi

Dokumenttiin tehdään asiakkaan vaatimat tarkennukset ja korjaukset, sekä aloitetaan dokumentin viimeistely.

Vaatimusmäärittelyvastaava varmistaa, että dokumentti on laadullisesti hyväksymiskelpoinen ja se hyväksytetään asiakkaalla.

8.2 Suunnittelu

Suunnittelu jakautuu seuraaviin osavaiheisiin. Rajapinta- ja komponenttisuunnittelua tehdään järjestelmän luonteen takia rinnakkain. Oliosuunnittelu aloitetaan toteutusvaiheen alkaessa. Testaussuunnitelmaa tehdään koko suunnitteluvaiheen aikana muiden vaiheiden rinnalla.

Työvaiheen lopuksi saadaan tuloksena suunnitteludokumentti ja testaussuunnitelma.

8.2.1 Arkkitehtuurisuunnittelu

Arkkitehtuurisuunnitelmassa tehdään järjestelmän arkkitehtuurikuvaus, joka selvittää:

- Osajärjestelmät
- Osajärjestelmien komponentit
- Niiden välisen kommunikaation

Arkkitehtuurimallin pohjana käytetään vaatimusmäärittelydokumentin järjestelmämallia.

8.2.2 Rajapintasuunnittelu

Rajapintasuunnittelussa suunnitellaan ja dokumentoidaan järjestelmän osien rajapinnat muihin osajärjestelmiin, käyttäen hyväksi arkkitehtuurisuunnitelmassa esitettyä arkkitehtuurimallia.

Rajapintakuvauksissa kuvataan rajapinnan palvelut, kunkin palvelun syöte- ja tulostetyypit sekä arvoalueet ja mitkä ehdot on kunkin palvelun kohdalla oltava voimassa ennen palvelun ja palvelukutsun jälkeen.

8.2.3 Komponenttisuunnittelu

Komponenttisuunnittelu tehdään osajärjestelmittäin ja se jakautuu:

- Tietokantasuunnitteluun
- Käyttöliittymäsuunnitteluun

- Integroinnin ja toiminnallisuuden suunnitteluun

Tietokantasuunnittelussa suunnitellaan tietokannan rakenne ja laaditaan siitä UML-kaavio. Käyttöliittymäsuunnittelussa käytetään pohjana vaatimusmäärittelyvaiheessa tehtyjä käyttöliittymäprototyyppejä, joita kehitetään eteenpäin.

Integroinnin ja toiminnallisuuden suunnittelussa suunnitellaan miten kommunikointi hoidetaan tietokannan, käyttöliittymän sekä muiden järjestelmän osien kanssa.

8.2.4 Oliosuunnittelu

Oliosuunnittelussa määritellään olioiden rajapinnat yksityiskohtaisesti ja dokumentoidaan ne.

8.2.5 Testaussuunnitelma

Testausdokumentissa kerrotaan miten ohjelmistoa testataan sen tekemisen aikana ja sen valmistuttua sekä mitä eri testausmuotoja käytetään.

8.3 Toteutus- ja yksikkötestaus

Suunnittelun valmistuttua aloitetaan varsinainen toteutusvaihe, jossa toteutetaan järjestelmän prototyyppi suunnitteludokumentin mukaisesti. Jokainen tiimin jäsen harjoittaa yksikkötestausta testaussuunnitelman mukaisesti.

Lopputuloksena saadaan ohjelmiston lähdekoodi.

8.4 Integrointi ja järjestelmätestaus

Testaaminen suoritetaan testausdokumentin mukaisesti ja sillä varmistetaan ohjelmiston laatu ja toimintojen toimivuus. Löydetyt virheet korjataan ja mahdolliset fataalit ongelmat dokumentoidaan ylläpitodokumenttiin. Kirjoitetaan ohjelmiston käyttöohje.

Lopputuloksena testauksen avulla varmistettu toimiva ohjelmisto, jonka puutteet on kirjattu ylös, sekä käyttöohje ja ylläpitodokumentti.

8.5 Projektin viimeistely ja luovutus

Projekti viimeistellään ja luovutetaan asiakkaalle sekä asennetaan asiakkaan testiympäristöön. Syntyneet dokumentit viimeistellään ja kirjoitetaan loppuraportti.

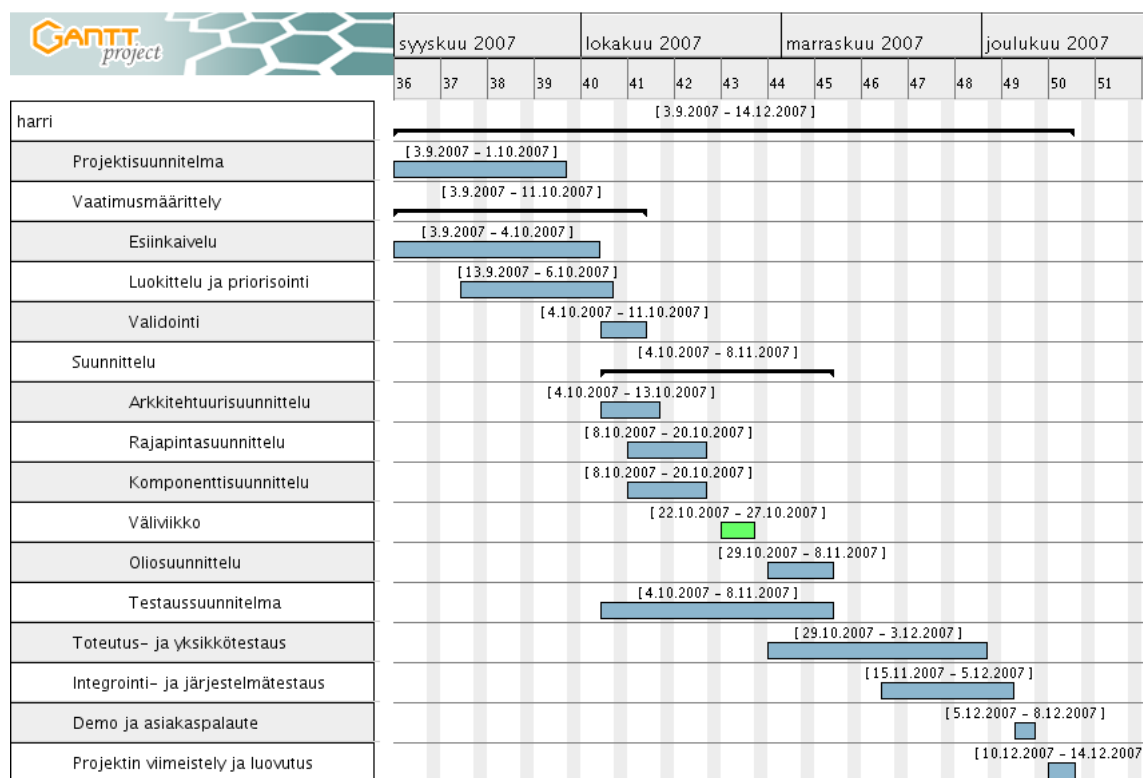
Lopputuloksena valmis ohjelmisto sekä loppuraportti.

9 Aikataulu

Projekti alkoi 3.9.2007 ja se kestää 14 viikkoa. Projekti valmistuu 14.12.2007. Ensimmäisen periodin väliviikko on viikko 43 eli päivien 22.-28.10 välillä. Väliviikko on lomaa ja se verottaa 7 päivää projektin suunnitteluvaiheesta.

9.1 Projektin työnkulku

Vaihe	Alkaa	Valmis	Kesto
Projektisuunnitelma	03.09.2007	01.10.2007	28 päivää
Vaativuusmäärittely	03.09.2007	11.10.2007	38 päivää
Suunnittelu	04.10.2007	08.11.2007	28 päivää
Toteutus- ja yksikkötestaus	29.10.2007	03.12.2007	35 päivää
Integrointi ja järjestelmättestaus	15.11.2007	05.12.2007	20 päivää
Demo ja asiakaspalaute	05.12.2007	10.12.2007	5 päivää
Projektin viimeistely ja luovutus	10.12.2007	14.12.2007	5 päivää



Kuva 1: Projektin aikataulu Gant-kaaviona

10 Seuranta- ja raportointimenetelmät

10.1 Palaverit

Ensimmäinen periodi (3.9.2007 - 21.10.2007)

Palaverit pidetään maanantaisin klo 14-16 ja torstaisin klo 8-10 Helsingin yliopiston Tietojenkäsittelytieteen laitoksessa Exactumissa salissa A219.

Toinen periodi (29.10.2007 - 14.12.2007)

Tarkennetaan toisen periodin alkuun mennessä.

10.2 Seurantakatsaus

Joka viikon ensimmäisessä palaverissa pidetään aluksi seurantakatsaus. Seuranta katsauksessa jokainen vuorollaan kertoo mitä on viime viikolla saanut aikaan, kuinka paljon siihen on käytetty työtunteja ja mitkä asiat vaativat vielä työtä. Seurantakatsauksen edellytyksenä on, että jokainen pitää työtunneistaan kirjaa ohjelmistotuotantoprojektin tietojärjestelmällä. Edellisen viikon työtuntien pitää olla kirjattuna järjestelmään viimeistään seurantakatsaukseen mennessä.

10.3 Projektisuunnitelman päivitys

Projektisuunnitelmaa päivitetään koko projektin ajan.

Seuraavia *työn osituksen* vaiheita tarkennetaan seuraavan aikataulun mukaisesti, jos se katsotaan tarpeelliseksi:

Tarkennettava vaihe	Päivä	Tila
Suunnittelu	11.10.2007	Tarkennettu
Toteutus- ja yksikkötestaus	22.10.2007	Ei vielä tarkennettu
Integrointi ja järjestelmätestaus	8.11.2007	Ei vielä tarkennettu