

eMo
eAssari Moodle-ympäristössä
Projektisuunnitelma

Helsinki 15.12.06
Ohjelmistotuotantoprojekti
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteenlaitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9+1op/6ov)

Projektiryhmä

Järviniitty Juho
Karkulahti Ossi
Katainen Riku
Tverin Teemu
Vainio Ville-Pekka

Asiakas

Laine Harri

Vastuhenkilö

Taina Juha

Ohjaaja

Halko Antti

Kotisivu

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

Versiohistoria

<i>Versio</i>	<i>Päiväys</i>	<i>Tehdyt muutokset</i>
0.1	12.9.	Johdanto, riskejä
0.2	13.9.	Organisaatio, riskejä
0.4	14.9.	Aikataulutus
0.6	15.9.	Sanasto, koko- ja kustannusarviot
0.7	16.9.	Seuranta- ja raportointimenetelmät
0.8	17.9.	Laitteisto- ja ohjelmistovaatimukset
1.0	18.9.	Viimeistely
1.1	18.9.	Kokoarvio, viimeistely
1.1.1	22.9.	FP-analyysin korjaus
1.2	25.9.	Dokumentin ulkoasu

<i>Versio</i>	<i>Päiväys</i>	<i>Tehdyt muutokset</i>
1.3	9.10.	Gantt-kaavio
1.4	31.10.	Riskejä
1.5	27.11.	Aikatauluja päivitetty
1.5.1	14.12.	Viimeistely

Sisällys

1 Johdanto.....	1
1.1 Tavoite.....	1
2 Sanasto.....	1
3 Projektiorganisaatio.....	2
3.1 Projektiorganisaatio:.....	2
3.1.1 Ryhmän vastualueet.....	2
3.1.2 Projektin sidosryhmät.....	3
3.1.3 Vastuualueiden tehtäväkuvaukset.....	3
4 Riskianalyysi.....	4
4.1 Projektikohtaiset riskit.....	4
4.2 Tuotekohtaiset riskit.....	6
4.3 Organisaatiokohtaiset riskit.....	8
5 Projektin laitteisto- ja ohjelmistovaatimukset	11
6 Koko- ja kustannusarviot.....	12
6.1 Toimintopistemalli:	12
6.1.1 Syötteet.....	12
6.1.2 Tulosteet.....	12
6.1.3 Interaktio.....	13
6.1.4 Monimutkaisuus.....	13
6.1.5 Toimintopisteet.....	14
6.1.6 Koodirivit.....	14
6.2 Koodirivimalli:	14
7 Työn ositus.....	14
7.1 Projektin päivämääriä.....	14
7.2 Projektin vaiheet.....	15
7.2.1 Aiheeseen tutustuminen.....	15
7.2.2 Projektisuunnitelman tekeminen.....	15
7.2.3 Vaatimusmäärittely.....	15
7.2.4 Suunnittelu.....	15
7.2.5 Toteutus.....	15
7.2.6 Testaus.....	16
7.2.7 Käyttöönotto ja ylläpito.....	16
8 Projektin aikataulu.....	16
8.1 Aikataulukko.....	16
8.2 Gantt-kaavio.....	17
9 Seuranta- ja raportointimenetelmät	17
9.1 Kokoukset.....	17
9.2 Seuranta.....	18
9.3 Dokumentit.....	18

1 Johdanto

Tässä dokumentissa kuvataan ohjelmistotuotantoprojekti-kurssilla suoritettavaa projektia, jonka tarkoituksena on siirtää eAssari-ohjelmisto Moodle-ympäristöön. Lyhykäisyydessään eAssari on tietokantapohjainen ympäristö, jossa suoritetaan ja tarkastetaan koe- ja harjoitustehtäviä ohjelmallisesti. Moodle on ilmainen oppimisolusta, jota voi käyttää erilaiset käyttäjäryhmät erilaisiin käyttötarkoituksiin. Vapaa lähdekoodi ja lukemattomat pluginit mahdollistavat erittäin dynaamisen työympäristön sen käyttäjille ja kehittäjille.

Projekti tehdään Helsingin Yliopistossa Tietojenkäsittelytieteen laitokselle syksyllä 2006. Ohjelmistotuotantoprojekti-kurssin vastuuhenkilöt ovat koonneet työryhmän kurssille osallistuneista henkilöistä.

1.1 Tavoite

Kokonaistavoitteenamme on viedä ryhmämme kanssa läpi aito ohjelmistotuotantoprojekti kaikkine vaiheineen ja lopulta tuottaa valmis ohjelmisto. Projekti suoritetaan ”aidossa” työympäristössä, jolloin pääsemme pureutumaan työmaailmaan liittyviin toimintatapoihin ja niiden mukana tuomiin ongelmiin ja ratkaisuihin.

Projektin alussa tavoitteena on ottaa selvää Moodlen toimintaperiaatteista ja mahdollisuuksista siirtää eAssari tähän kyseiseen ympäristöön. Tämän jälkeen pyrimme toteuttamaan näissä puitteissa ohjelmistolle määritetyt vaatimukset kurssin kestoajan puitteissa.

2 Sanasto

Moodle

Moodle on ilmainen, avoimen lähdekoodin sähköinen oppimisolusta, joka soveltuu erilaisille käyttäjäryhmille monenlaisiin käyttötarkoituksiin. Moodle tarjoaa työvälineitä mm. vuorovaikutukseen, sisällöntuottamiseen ja materiaalin jakamiseen. Samankaltaisia oppimisolustoja ovat mm. WebCt, BlackBoard ja FirstClass.

eAssari

eAssari on tietokantapohjainen ympäristö ohjelmallisesti tarkastettavien harjoitus- ja koetehtävien suorittamiseen. Monikielisyys ja helppo laajennettavuus ovat olleet päätavoitteita järjestelmää suunniteltaessa. Ohjelmiston avulla opiskelijalle voidaan selainkäyttöliittymässä esittää tietokannassa määriteltyjä tehtäviä. Opiskelija antaa vastauksen tehtävään käyttöliittymän kautta. vastaus tarkistetaan normaalisti heti ja opiskelijalle annetaan palautetta tehdyn tarkistuksen perusteella.

PHP

PHP (PHP Hypertext Preprocessor) on ohjelmointikieli, jota käytetään erityisesti palvelinympäristöissä luotaessa dynaamisia web-sivustoja. PHP-koodi tulkitaan ohjelman suorituksen aikana kääntämisen sijaan.

3 Projektiorganisaatio

3.1 Projektiorganisaatio:

Jokaisella eMo-ryhmän jäsenellä on oma vastuualueensa, josta hän on vastuussa. Nämä roolit jaettiin vapaaehtoisuuteen perustuen ensimmäisessä projektikokouksessa. Kuitenkin jokainen ryhmän jäsen osallistuu kaikkiin työvaiheisiin. Varavastuualueita ryhmäläisille ei ole määritelty erikseen.

3.1.1 Ryhmän vastualueet

Juho Järvinen: projektipäällikkö

Riku Katainen: vaatimusmäärittelyvastaava

Ville-Pekka Vainio: suunnitteluvastaava

Ossi Karkulahti: dokumenttivastaava

Teemu Tverin: testausvastaava

3.1.2 Projektin sidosryhmät

Asiakas: Harri Laine

Ohjaaja: Antti Halko

Vastuuhenkilö: Juha Taina

3.1.3 Vastuualueiden tehtäväkuvaukset

Projektipäällikkö: Projektipäällikkö vastaa projektisuunnitelmasta, projektin aikataulusta, henkilöiden allokoinnista tehtäviin sekä riskienhallinnasta. Projektipäällikkö toimii puheenjohtajana kokouksissa, ellei kokouksen aihe kuulu jonkun toisen ryhmäläisen vastuualueeseen.

Vaatusmäärittelyvastaava: Vaatusmäärittelyvastaava toimii asiakkaan ja projektiryhmän yhdyshenkilönä, vastaa vaatusmäärittelyn osavaiheiden onnistumisesta, määrää vaatusdokumentin rakenteen, toimii puheenjohtajana vaatusmäärittelyyn 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 sekä siitä, että suunnitelma ja järjestelmäarkkitehtuuri ovat yhdenmukaiset.

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

Dokumenttivastaava: Dokumenttivastaava vastaa siitä, että dokumenttien ulkoasu on yhteneväinen, dokumentit ovat luettavassa kunnossa ja dokumenttien sisältö on kattava. Dokumenttivastaava pitää myöskin yllä projektin kotisivua. Dokumenttivastaava vastaa myös Koodivastaavan tehtävistä, eli koodin yhtenäisestä ulkoasusta ja rajapintojen yhtenäisyydestä, sillä ryhmässä ei ole nimettyä koodivastaavaa.

4 Riskianalyysi

Riskianalyysillä pyritään kartoittamaan tuotantoprojektiin liittyviä riskejä ja selvittämään ennaltaehkäisykeinoja ja vastatoimia kullekin riskille. Jokaisella riskillä on todennäköisyys (1-5) ja vakavuus (1-5).

Jaottelemme riskit projekti-, tuote- ja organisaatiokohtaisiin.

Riskien luokitteluun käytettävät asteikot ovat seuraavat:

Riskien todennäköisyys:

- 1 epätodennäköinen
- 2 mahdollinen
- 3 keskinkertainen
- 4 todennäköinen
- 5 varma

Riskien vakavuus:

- 1 vähäinen
- 2 lievä
- 3 kohtalainen
- 4 vakava
- 5 katastrofaalinen

4.1 Projektikohtaiset riskit

Projektikohtaisilla riskeillä tarkoitetaan aikatauluun ja resursseihin kohdistuvia riskejä.

Riski: Projekti myöhästyy

ID: projR-001

Todennäköisyys: 4

Vakavuus: 2

Ennaltaehkäisy:

- Pyritään arvioimaan ohjelmiston koko mahdollisimman hyvin, jotta aikataulusta saa täsmällisemmän.
- Aikataulun ositus ja lomitukset tulee suunnitella hyvin
- Noudatetaan käytössä olevaa mallia ja panostetaan edistymisen seurantaan.

Vastatoimet:

- Laaditaan uusi aikataulutus projektille
- Tehdään ylitöitä

Riski: Projektin dokumentit ja koodi häviävät

ID: projR-002

Todennäköisyys: 1

Vakavuus: 5

Ennaltaehkäisy:

- Pidetään varmuuskopioita eri koneilla
- Käytetään versionhallintajärjestelmää

Vastatoimet:

- Etsitään omilta koneilta ja välimuisteista uusimmat mahdolliset versiot ja jatketaan siitä

Riski: Laitteisto- ja ohjelmisto-ongelmat

ID: projR-003

Todennäköisyys: 3

Vakavuus: 3

Ennaltaehkäisy:

- Varmuuskopiointi
- Versionhallinta

Vastatoimet:

- Tiedostojen palauttaminen varmuuskopioista / versionhallinnasta

4.2 Tuotekohtaiset riskit

Tuotekohtaiset riskit käsittävät tuotteen laatuun kohdistuvat riskit

Riski: Tuote ei vastaa määrittelyä

ID: tuoteR-001

Todennäköisyys: 2

Vakavuus: 3

Ennaltaehkäisy:

- Vaatimusmäärittely tulee olla täsmällinen
- Suunnittelussa tulee ottaa huomioon kaikki vaatimusmäärittelyssä esille tulleet seikat

- Testauksen tulee kattaa kaikki määritellyt kohdat

Vastatoimet:

- Korjataan ohjelmistoon kohdat, jotka menevät vaatimusmäärittelyn kanssa ristiin ja testataan ohjelmisto uudelleen, kunnes vaatimukset täytetään

Riski: Kukaan ei ymmärrä toisten koodeista mitään, eikä erillään tehdyt moduulit osaa kommunikoida keskenään

ID: tuoteR-002

Todennäköisyys: 3

Vakavuus: 4

Ennaltaehkäisy:

- Sovitaan yhteinen ulkoasu
- Kiinnitetään erityistä huomiota kommentointiin
- Testaillaan ohjelman eri komponentteja riittävän usein

Vastatoimet:

- Lisätään kommentointia
- Pidetään tapaaminen, jossa selvitetään epäselvät kohdat toisille

Riski: Vaatimusmäärittelyn epäonnistuminen

ID: tuoteR-003

Todennäköisyys: 2

Vakavuus: 4

Ennaltaehkäisy:

- Vaatimusmäärittelyn suorittamiseen käytettävä riittävästi aikaa
- Asiakkaan kanssa kommunikointi

Vastatoimet:

- Vaatimusmäärittelyn uudelleen toteuttaminen
- Täsmällisempi määrittely asiakkaan kanssa

Riski: Ohjelmiston yhteensovittaminen Moodleen osoittautuu todella hankalaksi

ID: tuoteR-004

Todennäköisyys: 4

Vakavuus: 4

Ennaltaehkäisy:

- Moodleen dokumentaation ja koodien tarkka tutkiskelu

Vastatoimet:

- Lisää tutkiskelua
- Kiertää yhteensovitusongelmat suunnittelemalla joitain komponentteja Moodlesta riippumattomiksi

4.3 Organisaatiokohtaiset riskit

Organisaatiokohtaiset riskit koskevat organisaation jäseniä ja asiakkaita

Riski: Ryhmän jäsen sairastuu merkittävästi työskentelyä haittaavasti

ID: orgaR-001

Todennäköisyys: 2

Vakavuus: 2

Ennaltaehkäisy:

- Laitetaan aikatauluun hieman pelivaraa, jotta pienet takaiskut eivät myöhästyä koko projektia
- Liian tiukka aikataulu stressaa ja uuvuttaa työntekijöitä
- Järjestetään mahdollisimman hyvät etätyömahdollisuudet, jotta kotoa käsin voi olla tehokkaasti mukana ryhmän toiminnassa

Vastatoimet:

- Siirretään sairastuneen työtehtäviä muille ryhmän jäsenille tasapuolisesti
- Selvitetään, missä asioissa sairastunut jäsen voi auttaa tilastaan huolimatta ja jaetaan tehtäviä uudelleen

Riski: Ryhmän sisäinen kommunikointi ei toimi tehokkaasti

ID: orgaR-002

Todennäköisyys: 4

Vakavuus: 5

Ennaltaehkäisy:

- Sovitaan yhteiset kommunikointivälineet ja opastetaan kaikkia niiden käytössä
- Pyritään järjestämään mahdollisimman paljon yhteisiä tapaamisia

Vastatoimet:

- Mietitään parempia tapoja kommunikointiin
- Sovitaan yhteisiä aikoja myös etätönteekoon, jolloin kaikki on samaan aikaan esim. Ryhmäkeskustelussa

Riski: Ryhmän jäsenen keskeyttää kurssin.

ID: orgaR-003

Todennäköisyys: 1

Vakavuus: 4

Ennaltaehkäisy

- Hyvän työskentelyilmapiirin ylläpitäminen
- Aikataulun ja työn jaotuksen älykäs suunnittelu

Vastatoimet

- Tehtävien uudelleenjako

Riski: Ryhmän jäsenen osaamistaso on liian alhainen annetun tehtävän suorittamiseen

ID: orgaR-004

Todennäköisyys: 3

Vakavuus: 4

Ennaltaehkäisy:

- Töiden jakaminen osaamisen mukaisesti
- Omaan tehtävään perehtyminen

Vastatoimet:

- Osaamaton henkilö ilmoittaa asiasta tarpeeksi aikaisin
- Tehtävän suorituksen uudelleenjakaminen
- Lisää perehtymistä

Riski: Ryhmäläisellä ei ole riittävästi aikaa projektin tekemiseen, esim. töiden takia

ID: orgaR-005

Todennäköisyys: 5

Vakavuus: 4

Ennaltaehkäisy:

- Ottaa selvää mahdollisista muista kiireistä ajoissa ja jakaa työtehtäviä ryhmän kesken
- Priorisoi projektin muiden asioiden edelle

Vastatoimet:

- Ylitöitä muita kiireitä ennen ja jälkeen
- Muut tekevät estyneen ryhmäläisen työt

5 Projektin laitteisto- ja ohjelmistovaatimukset

Tuotettava järjestelmä tulee toimimaan Tietojenkäsittelytieteen laitoksen tietokantakäyttöön tarkoitettulla palvelimella ja se tullaan toteuttamaan Moodlen versio 1.6:n pohjalta. Moodle vaatii MySQL-tietokannasta vähintään version 4.1.12. Moodle käyttää PHP-kieltä ja pienin vaadittava versio siitä on 4.3.0.

6 Koko- ja kustannusarviot

Projektin kokoa arvioidaan sekä käyttäen toimintopistemallia, että koodirivimallia.

Toimintopistemallin avulla arvioidaan käyttötapausten perusteella ohjelmiston toteutukseen tarvittavien koodirivien määrä. Koodirivimallissa arvioidaan ohjelmiston koodirivit suoraviivaisesti.

6.1 Toimintopistemalli:

Toimintopistemallissa ohjelmiston syönteille, tulosteille, käyttäjän tekemille kyselyille, tiedostoille ja ulkoisille rajapinnoille annetaan pistearvo, joka kuvaa niiden toteutuksen laajuutta. Elementin pistearvo määräytyy sen monimutkaisuuden mukaan.

Toimintopistemallin ensimmäinen kokoarvio voidaan tehdä kun vaatimusmäärittelyvaiheessa on selvitetty ohjelmiston käyttötapauksia. Seuraavat karkeat käyttötapaukset voidaan kuitenkin jo määritellä:

Pistearvot ovat seuraavat: helppo (3), keskivaikea (5), vaikea (7).

6.1.1 Syönteet:

<i>Kuvaus</i>	<i>Vaikeus</i>	<i>Pisteet</i>
Tehtävän lisääminen järjestelmään:	keskivaikea	7
Vastauksen syöttö	helppo	3
YHTEENSÄ		10

6.1.2 Tulosteet

<i>Kuvaus</i>	<i>Vaikeus</i>	<i>Pisteet</i>
Tehtävän näyttäminen ruudulla	keskivaikea	5
Vastauksesta palautteen antaminen	helppo	3

YHTEENSÄ **8**

6.1.3 Interaktio

Kuvaus	Vaikeus	Pisteet
Syötetyn tehtävän käsittely	vaikea	7
Vastauksen käsittely	vaikea	7

YHTEENSÄ **14**

6.1.4 Monimutkaisuus

Monimutkaisuuskerroin lasketaan seuraavien kysymysten perusteella. Jokaiseen kysymykseen vastataan numeroarvolla väliltä 0 - 5, joka kertoo kuinka tärkeää kyseisen kysymyksen käsittelemä asia on ohjelmistolle. Kerroin on vastausten summa.

1. Tarvitseeko järjestelmä luotettavaa varmuuskopiointia ja palautumista? 1
2. Tarvitaanko tiedon kommunikointia? 3
3. Tekeekö järjestelmä hajautettua prosessointia? 0
4. Onko suorituskyky kriittinen elementti? 1
5. Tuleeko järjestelmä toimimaan olemassa olevassa, raskaassa käytössä olevassa ympäristössä? 4
6. Tarvitseeko järjestelmä tiedon syöttöä suorituksen aikana? 1
7. Rakentuuko interaktiivinen syöte monen ruudun tai operaation yli? 1
8. Päivitetäänkö tiedostoja suorituksen aikana? 5
9. Ovatko syötteet, tulosteet, tiedostot tai kyselyt monimutkaisia? 4
10. Onko sisäinen toiminta monimutkaista? 3
11. Tuleeko koodin olla uudelleenkäytettävää? 4
- 12.
13. Ovatko ohjelmiston muutokset ja asennus osa suunnitelmaa? 3
14. Onko järjestelmä tarkoitettu asennettavaksi moneen eri ympäristöön? 0

15. Onko ohjelmisto suunniteltu olemaan käyttäjän muokattavaksi ja helppokäyttöiseksi? 4

Monimutkaisuuskerroin yhteensä: 34

6.1.5 Toimintopisteet

Toimintopistemallin laskukaavalla saadaan seuraava:

$$32*(0,65+0.01*34)=31,68$$

6.1.6 Koodirivit

PHP:n kerroin on 67: $31,68*67=2122,56$

6.2 Koodirivimalli:

Ohjelmisto sisältää alle 10000 riviä koodia.

Projektin tulee kestämään 15 viikkoa.

7 Työn ositus

7.1 Projektin päivämääriä

Tärkeitä päivämääriä	
18.9.	Projektisuunnitelma valmis
5.10.	Vaatimusmäärittely valmis
8.11.	Suunnitteludokumentti valmis
9.11.	Testaussuunnitelma valmis
11.12.	Ohjelma valmis
15.12.	Projektin päättyminen

7.2 Projektin vaiheet

Ryhmä valitsi prosessimalliksi lineaarisen vesiputousmallin, jota mukailten projektin vaiheet ovat seuraavat.

7.2.1 Aiheeseen tutustuminen

Ryhmän tutustuminen aiheeseen tapahtuu omatoimisesti paneutumalla Internetistä löytyvään, sekä Harri Laineelta saatuun materiaaliin.

7.2.2 Projektisuunnitelman tekeminen

Projektisuunnitelma määrittää toteutettavan projektin kehykset. Se sisältää mm. aikataulun ja ryhmän sisäiset säännöt. Projektisuunnitelma laaditaan siten, että kukin projektiryhmän jäsen kirjoittaa vastuualuettaan koskevista aiheista. Hyväksyminen tapahtuu kokouksessa.

7.2.3 Vaatimusmäärittely

Vaatimusmäärittelyssä selvitetään asiakkaan asettamat vaatimukset ohjelmistolle. Nämä vaatimukset jaetaan toiminnallisiin ja ei-toiminnallisiin ja ne kirjataan vaatimusmäärittelydokumenttiin. Vaatimukset toimivat suunnittelun lähtökohtana.

7.2.4 Suunnittelu

Suunnittelussa pyritään kuvamaan toteutettava järjestelmä tarkasti ja ilmentämään sen arkkitehtuuri. Järjestelmän osa-alueet luokkarakenne, rajapinnat, toiminnallisuus ja käyttöliittymä suunnitellaan erikseen ja kootaan suunnitteludokumenttiin. Suunnittelun aikana saadaan selville myös testitapauksia, jotka kirjataan testaussuunnitelmaan.

7.2.5 Toteutus

Ohjelmointikieli on PHP. Toteutus pyrkii noudattamaan suunnitteludokumenttia, mutta luultavasti toteutukseen kuuluu myös suunnitelman päivittämistä. Toteutus etenee luokittain, ja toteutuksen kanssa rinnakkain tehdään myös integrointi- ja järjestelmätestausta.

7.2.6 Testaus

Testausta suoritetaan rinnakkain sekä suunnittelun, että toteutuksen kuluessa. Testaaminen jakautuu neljään osaan: yksikkötestaus, integrointitestaus, järjestelmätestaus sekä hyväksymistestaus. Näistä kolme ensimmäistä ovat projektiryhmän tekemiä testauksia. Hyväksymistestauksen suorittaa asiakas valmiille tuotteelle.

7.2.7 Käyttöönotto ja ylläpito

Ohjelmiston valmistuttua annetaan asiakkaalle mahdollisuus hyväksymistestaukseen sekä opastusta ohjelmiston käyttöön. Ohjelmiston ylläpito rajataan projektin ulkopuolelle.

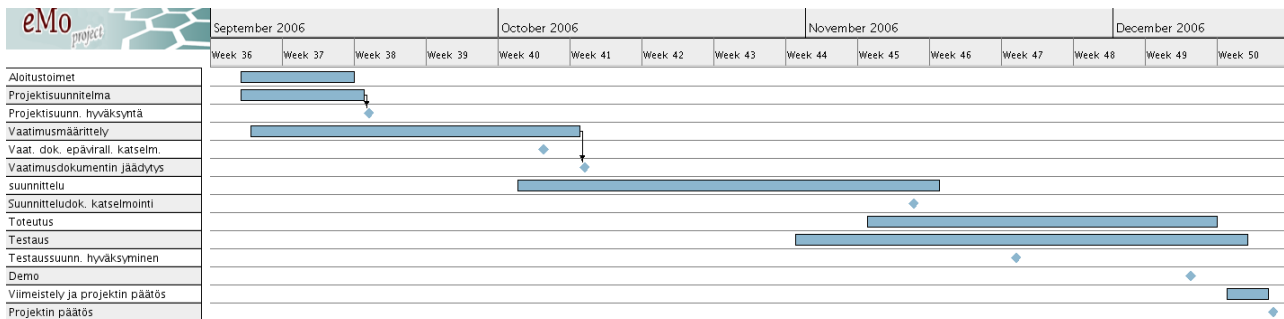
8 Projektin aikataulu

8.1 Aikataulukko

<i>Viikko</i>	<i>PVM</i>	<i>Vaihe</i>	<i>Kuvaus</i>
36	6.9.-10.9.	Aloitustoimet, projektisuunnitelma, vaatimusmäärittely	Tehtävään tutustuminen, nettisivut, ym. Projektisuunnitelman tekeminen
37	11.9.-17.9.	Aloitustoimet, projektisuunnitelma, vaatimusmäärittely	Versionhallinnan käyttöönotto, projektisuunnitelman tekeminen, vaatimuksien kartoittaminen
38	18.9.-24.9.	Vaatimusmäärittely, projektisuunnitelma	Projektisuunnitelma viimeistellään maanantaina. Vaatimusmäärittely (vaatimusten kaivelu ja analysointi), kelpoisuus selvitys
39	25.9.-1.10.	Vaatimusmäärittely	Vaatimusten tarkentaminen (käyttäjä- ja järjestelmävaatimukset, skenaariot)
40	2.10.-8.10.	Vaatimusmäärittely, suunnittelu	Vaatimusten validointi ja dokumentin viimeistely. Arkkitehtuurisuunnitelma
41	9.10.-15.10.	Suunnittelu	Arkkitehtuurisuunnitelma, abstrakti määrittely
42	16.10.-22.10.	Suunnittelu	Rajapintasuunnittelu, komponenttisuunnittelu
43	23.10.-29.10.	Loma	
44	30.10.-5.11.	Suunnittelu, testaus	Tietorakenteiden suunnittelu, testaussuunnitelma
45	6.11.-12.11.	Suunnittelu, toteutus, testaus	Algoritmien suunnittelu, ohjelmakomponentit, testaussuunnitelma
46	13.11.-19.11.	Toteutus, testaus	Ohjelmakomponentit, yksikkötestaus

47	20.11.-26.11.	Toteutus, testaus	Ohjelmakomponentit, yksikkötestaus
48	27.11.-3.12.	Toteutus, testaus	Osajärjestelmät, yhteistyötesti, käyttöohje
49	4.12.-10.12.	Toteutus, testaus, demo	Järjestelmä, integrointi- ja käyttöympäristötestit, ylläpidodokumentti, Demotilaisuus
50	11.12.-14.12.	Viimeistely, päätös	Asiakkaan demo, Viimeistely, dokumentit kuntoon, palautus.

8.2 Gantt-kaavio



Kuva 1: Projektin aikataulu Gantt-kaaviona

9 Seuranta- ja raportointimenetelmät

Seuranta tehdään etupäässä jokaisessa kokouksessa. Projektipäällikkö tarkkailee myös ryhmäläisten ajankäyttöä ohjelmistotuotantoprojektin tietojärjestelmästä.

9.1 Kokoukset

Kokouksia on pääsääntöisesti kahdesti viikossa: maanantaisin klo. 14-16 ja torstaisin klo 12-14. Tarpeen tullen kokouksia voidaan pitää muulloinkin. Puheenjohtajana toimii projektipäällikkö, ellei kokous liity erityisesti jonkun muun vastualueeseen. Sihteeri valitaan erikseen jokaiseen kokoukseen edellisessä kokouksessa.

Kokouksissa käsitellään esityslistalla olevia asioita. Pääsääntöisesti kokouksissa käsitellään ajankohtaista työvaihetta ja käydään läpi edellisen kokouksen jälkeen tehdyt tehtävät ja sovitaan seuraavaan kokoukseen mennessä tehtävät työt.

Kokouksen esityslista laitetaan esille projektin kotisivuille viimeistään kokousta edeltävänä päivänä ja pöytäkirja viimeistään kokousta seuraavana päivänä. Kokouksen sihteeri monistaa esityslistan jaettavaksi kokoukseen osallistujille. Kotisivujen päivittäminen on ensisijaisesti dokumenttivastaavan vastuulla, mutta myös muut ryhmän jäsenet voivat tehdä päivityksiä.

9.2 Seuranta

Projektiryhmän jäsenet pitävät kirjaa projektiin käyttämästään ajasta. Tunnit kirjataan ohjelmistotuotantoprojektin tietojärjestelmään joka viikko. Edellisen viikon tuntien tulee olla järjestelmässä viimeistään seuraavan viikon maanantaina. Työkalu on www-sovellus, joka käynnistyy osoitteesta http://db.cs.helsinki.fi/~tkt_ohtu/metrics/v0.

Oleellinen osa seuranta on myöskin kokouksissa tapahtuva projektin edistymisen seuranta.

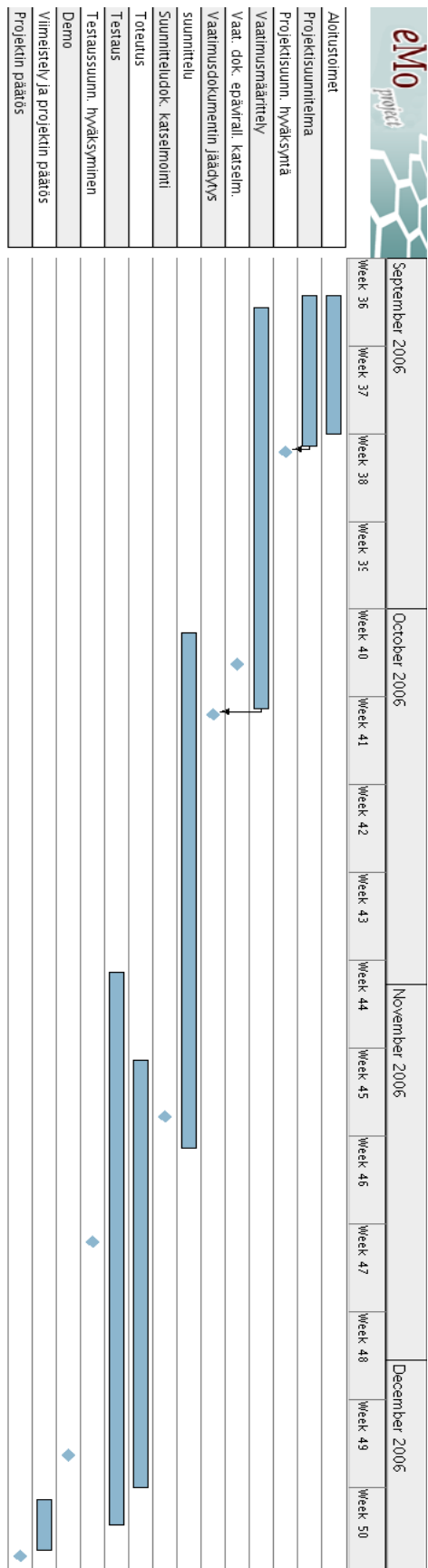
9.3 Dokumentit

Tärkein osa raportoinnista ovat projektissa syntyvät dokumentit. Dokumentit voidaan jakaa kahteen ryhmään seuraavasti sen mukaan kenelle ne on tarkoitettu.

Sisäiset dokumentit	Ulkoiset dokumentit
Projektisuunnitelma	Vaatimusmäärittelydokumentti
Suunnitteludokumentti	Käyttöohje
Testaussuunnitelma	Ylläpidodokumentti
Yhteenvedodokumentti	

Lisäksi projektin dokumentaatioon kuuluvat esityslistat(sisäinen), kokospöytäkirjat(sisäinen), sisäiset muistiot(sisäinen) ja tarkastusten tarkistuslistat(sisäinen). Toteutus dokumentoidaan kommentteina ohjelmakoodissa ja testitapaukset vastaavasti kommentteina testiskripteissä.

LIITE 1 Gantt-kaavio



Kuva 2: Projektin aikataulu gantt-kaaviona