

Ohjelmistotuotantoprojekti

Muutos- ja korjauspyyntöjen priorisointityökalu

Ryhmä Muppett

PROJEKTISUUNNITELMA

Helsinki 25.5.2008

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi:

581260 Ohjelmistotuotantoprojekti, kesä 2008

Asiakas:

Oodi-konsortio/ Sampo Lehtinen

Ryhmä:

Arto Chydenius

Laura Haverinen

Merja Linden

Topi Musto

Laura Ojala

Toni Sormunen

Ohjaaja:

Marko Lehtimäki

Vastuhenkilö:

Kimmo Simola

Dokumentin versiohistoria

<u>Versio</u>	<u>Päiväys</u>	<u>Muutokset</u>	<u>Muuttaja</u>
0.0	25.05.2008	Pohja	LH
0.1	25.05.2008	Ensimmäinen versio	LO
0.2	31.05.2008	Korjattu versio	LO
0.3	03.06.2008	Korjattu pikkuasioita	LO
0.4	10.06.2008	Kokoarvio/FP	LO
0.5	15.06.2008	Korjattu aikataulu	LO

Sisältö

1 Johdanto	1
2 Sanasto	1
3 Projektioorganisaatio	2
3.1 Ryhmän jäsenet ja vastualueet.....	2
3.2 Ryhmän ulkopuoliset jäsenet.....	2
3.3 Vastuualueiden kuvaus.....	2
4 Työn kuvaus	4
4.1 Liittymät muihin järjestelmiin.....	4
5 Projektin laitteisto- ja ohjelmistovaatimukset	4
5.1 Ympäristö.....	4
5.2 Kieli.....	5
5.3 Kokoarvio.....	5
5.3.1 Toimintopisteanalyysi (FP function points).....	5
5.3.2 LOC	7
6 Riskianalyysi	8
6.1 Teknologia.....	9
6.2 Ryhmä.....	10
6.3 Asiakas.....	11
6.4 Aikataulu.....	12
7 Aikataulu	13
7.1 Projektin aikataulu.....	13
7.2 Ryhmän resurssit.....	14
7.3 Tärkeitä päivämääriä.....	14
8 Projektityöskentely	14
8.1 Kotisivut.....	15
8.2 Työajan seuranta.....	15
8.3 Tuotettavat dokumentit	15
9 Projektin vaiheet	16
9.1 Projektisuunnitelma	16

9.2	Vaatimusten keruu ja analyysi.....	16
9.3	Ohjelmiston suunnittelu.....	17
9.4	Toteutus ja yksikkötestaus.....	17
9.5	Järjestelmätestaus.....	17
9.6	Projektin päätös.....	17
Liitteet		18

1 Johdanto

Dokumentti on Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojektin Muppett -ryhmän projektisuunnitelma, jossa määritellään kurssilla toteutettava tuote, ryhmän työkäytännöt ja vastualueet, riskit, kokoarvio sekä aikataulu. Projekti kattaa kesän 2008 viikot 21-35.

2 Sanasto

Java

Oliopohjainen ohjelmointikieli, joka on valittu ryhmän pääasialliseksi ohjelmointikieleksi.

GPL

General Public Licence. Ohjelmistotuotantoprojekteissa käytettävä lisenssi.

LGPL

Lesser General Public Licence. Ohjelmistotuotantoprojekteissa käytettävä lisenssi, joka rajoittaa ohjelmiston uudelleenkäytettävyyttä.

PostgreSQL

Avoimen lähdekoodin tietokannan hallintajärjestelmä.

SQL

Structured Query Language. Standardoitu kyselykieli relaatiotietokannan käsittelyyn.

SVN

Subversion, versionhallintajärjestelmä.

Vesiputousmalli

Prosessimalli ohjelmistotuotantoprojektien toteuttamiseen.

3 Projektiorganisaatio

Kullekin ryhmäläiselle on jaettu vastuualue, jonka toteuttamisesta hän huolehtii. Omalla vastuualueellaan hänellä on ylin päätäntävalta, joskin päätökset pyritään tekemään demokraattisesti ryhmän jäsenten kesken. Ryhmän ilmapiiriin tulee kannustaa avointa keskustelua, jossa kaikki saavat esittää kysymyksiä, kommentoida ja antaa palautetta.

3.1 Ryhmän jäsenet ja vastuualueet

Jäsen	Email	Rooli	Varalla
Laura Ojala	lwojala@cs.helsinki.fi	Projektipäällikkö	Laura Haverinen
Merja Linden	mzlinden@cs.helsinki.fi	Vaatimusmäärittelyvastaava	Toni Sormunen
Arto Chydenius	chydeniu@cs.helsinki.fi	Suunnitteluvastaava	Topi Musto
Laura Haverinen	lahaveri@cs.helsinki.fi	Dokumenttivastaava	Merja Linden
Topi Musto	musto@cs.helsinki.fi	Koodivastaava	Arto Chydenius
Toni Sormunen	toni.sormunen@cs.helsinki.fi	Testausvastaava	Laura Ojala

3.2 Ryhmän ulkopuoliset jäsenet

Marko Lehtimäki marko.lehtimaki@helsinki.fi Ohjaaja
 Sampo Lehtinen sampo.lehtinen@helsinki.fi Asiakas

3.3 Vastuualueiden kuvaus

- **Projektipäällikkö** vastaa projektisuunnitelmasta ja aikataulusta sekä toimii puheenjohtajana kokouksissa.
- **Vaatimusmäärittelyvastaava** toimii asiakkaan ja projektiryhmän välisenä yhdyshenkilönä ja vastaa yhtenäisestä vaatimusdokumentin rakenteesta.
- **Suunnitteluvastaava** vastaa yhtenäisistä suunnittelutason rajapinnoista ja suunnitteludokumentin yhdenmukaisesta rakenteesta.
- **Dokumenttivastaava** vastaa dokumenttien yhtenäisestä ulkoasusta ja projektin www-sivusta.
- **Testausvastaava** vastaa testauksen kattavuudesta ja

testaussuunnitelmasta.

- **Koodivastaava** vastaa koodin yhtenäisestä ulkoasusta ja rajapintojen yhtenäisyydestä.

4 Työn kuvaus

Asiakas haluaa web-järjestelmän, jolla eri yliopistot pystyvät sijoittamaan Oodi-järjestelmän muutos- ja korjauspyynnöt haluamaansa tärkeysjärjestykseen. Järjestelmä toimii alustana äänestyksille ja laskee reaaliajassa Oodi-konsortiolle, mitkä korjaus- ja muutospyyntöt ovat suosituimpia. Projektipäällikkö saa tulokset suoraan valmiina raporttina ulos järjestelmästä.

Asiakas kuuluu Oodi-konsortioon, joka kehittää yhdessä 13 suomalaisen yliopiston kanssa opintohallinnon tietojärjestelmää Oodia. Järjestelmän kehitystyötä tehdään ryhmissä, jotka tyypillisesti koostuvat Oodi-konsortion projektipäällikön lisäksi yliopistojen edustajista. Yliopistojen tarpeet eroavat hyvin useasti toisistaan ja tietty uusi toiminto on tärkeämpi osalle yliopistoja kuin toisille. Versioon valittavien muutos- ja kehitystoiveiden asettaminen keskinäiseen järjestykseen tapahtuu tällä hetkellä manuaalisesti ja kuormittaa projektipäällikköä.

4.1 Liittymät muihin järjestelmiin

Projektiryhmän tulee ottaa huomioon, että tuotettava ohjelmisto on mahdollista integroida johonkin toiseen järjestelmään. Tässä vaiheessa ohjelmisto kuitenkin toteutetaan toimimaan itsenäisesti.

5 Projektin laitteisto- ja ohjelmistovaatimukset

5.1 Ympäristö

Toteutettava ohjelmisto koostuu web-käyttöliittymästä ja sen

ohjaamasta tietokannasta. Ohjelman toteutus tehdään Javalla (versio 1.5 tai 1.6) ja tietokannaksi on valittu PostgreSQL.

Projektin kehitysversiota pidetään TKTL:n palvelimilla. Kaikki tiedostot säilytetään SVN-versionhallinnassa. Dokumentit luodaan OpenOfficella. Ohjelmointiympäristönä toimii Eclipse.

5.2 Kieli

Kokousten ja dokumenttien kielenä on suomi. Koodin muuttujat, funktiot sekä kommentit kirjoitetaan englanniksi.

5.3 Kokoarvio

Toteutettavaa ohjelmistoa arvioidaan toimintopistemethodella sekä LOC-methodella. Ensimmäinen arvio koosta tehdään alustavan vaatimusmäärittelyn valmistumisen jälkeen.

Yleisen käsitteen mukaan jokainen projektiryhmän jäsen pystyy tuottamaan kurssilla noin 500-700 riviä uutta koodia, jolloin koodirivimäärä voisi liikkua välillä 3000-4300. Tämä eroaa kuitenkin ratkaisevasti toimintopisteanalyysin antamasta koodirivimäärästä.

5.3.1 Toimintopisteanalyysi (FP function points)

Toimintopisteanalyysiä varten on arvioitava järjestelmän syötteiden, tulosteiden, ulkoisten liittymien, tiedostojen, kyselyjen ja tietokannan taulujen määrää. Syötteet ovat dataa, joka viedään järjestelmään, tulosteet taas järjestelmässä näytettävää dataa. Ulkoiset liittymät kuvaavat yhteyksiä muihin järjestelmiin ja tiedostot viittaavat järjestelmän tietoliikenteeseen, dokumentteihin ja conf-tiedostoihin. Kyselyt ovat tilanteita, jossa käyttäjä antaa syötteen ja jolle heti generoituu jonkinlainen vastaus järjestelmästä. K* ilmaisee toimintopistekertoimen, joka määräytyy ryhmän ja vaikeusluokituksen mukaan.

	Helppo	k*	Normaali	k*	Vaikea	k*	Pisteet
Syötteet	2	3	3	4	3	6	6+12+18=36

Tulosteet	5	4	3	5	1	7	20+15+7=42
Kyselyt	3	3	2	4		6	9+8=17
Tiedostot	10	7	8	10		15	70+80=130
Liittymät	1	5	1	7		10	5+7=12
Σ							237

Alla määritellään kompleksisuustekijät, joiden avulla saadaan kompleksisuukerroin. Kysymyksiä arvioidaan asteikolla 0-5p, jossa

0p = Ei koskaan

1p = Harvoin

2p = Toisinaan

3p = Keskimääräisesti

4p = Merkittävästi

5p = Oleellisesti.

1. Onko järjestelmä vikasietoinen? Tarvitaanko luotettavaa tietojen varmistus- ja palautusmenettelyä? (Harvoin, 1)
2. Tarvitaanko tietoliikenneominaisuuksia? (Oleellisesti, 5)
3. Onko hajautettua prosessinhallintaa? (Ei, 0)
4. Onko suorituskyky kriittinen elementti? (Keskimääräisesti, 3)
5. Käytetäänkö järjestelmää olemassa olevassa raskaassa käytössä olevassa koneympäristössä?
(Ei, 0)
6. Tarvitaanko interaktiivista tietojen syöttöä? (Keskimääräisesti, 3)
7. Täytyykö interaktiivinen tietojen syöttö synkronoida usealle näytölle tai operaatiolle?
(Toisinaan, 2)
8. Päivitetäänkö tiedostoja interaktiivisesti? (Harvoin, 1)
9. Ovatko syötteet, tulosteet, tiedostot tai kyselyt monimutkaisia? (Toisinaan, 2)
10. Onko ohjelman toiminta monimutkaista? (Merkittävästi, 4)
11. Onko koodi tarkoitettu uudelleenkäytettäväksi? (Merkittävästi, 4)
12. Ovatko ohjelmiston muunnokset ja asennus mukana suunnitelmassa?
(Merkittävästi, 4)
13. Onko ohjelmisto suunniteltu toimivaksi useina asennuksina eri organisaatioissa?
(Ei, 0)
14. Onko sovellus suunniteltava käyttäjäystävälliseksi? (Oleellisesti, 5)

Yhteensä: 34

Lopulliset toimintopisteet lasketaan kaavalla:

$$FP = N \times [0.65 + 0.01 \times \sum(F_i)]$$

- FP = Function Points
- N = Peruspisteet
- $\sum(F_i)$ = Kompleksisuuskerroin

Täten

$$FP = 237 \times [0.65 + 0.01 \times 34] = 234.63$$

Nyt voidaan laskea koodirivien määrä, kun tiedetään, että Javan toimintopisteiden suhde ohjelmariveihin (LOC/FP) on 31.

$$LOC = 234.63 \times 31 = 7273.53$$

5.3.2 LOC

LOC-menetelmällä arvioidaan ohjelmiston koodirivimäärää. Alla järjestelmän osat osin eritelty käyttötapausten mukaan. Apuna koodiriviarviossa käytetään edeltäviä ohjelmistotuotantoprojekteja, minkä lisäksi eri osat pyritään arvioimaan hieman yläkanttiin kattamaan unohduksia ja erheitä. Projektin alkuvaiheessa on tapana aliarvioida järjestelmän koko ja tehdä virheitä osituksessa.

- Käyttöliittymät - 500
 - Käyttäjät
 - Ylläpito
- Tietokannan hallinta - 300
- Järjestelmään kirjautuminen - 50
- Järjestelmästä poistuminen - 10
- Äänestämisen - 500
 - Järjestysäänestys
 - Luokitteluäänestys
- Salasanan resetointi - 20

- Yliopistotietojen lisääminen - 200
 - Yliopistotietojen lisääminen
 - Yliopistotietojen muokkaaminen
 - Yliopistotietojen poistaminen
- Käyttäjätiedot - 200
 - Käyttäjätunnuksen lisääminen
 - Käyttäjätietojen muokkaaminen
 - Käyttäjätunnuksen poistaminen
- Muutos- korjauspyynnöt - 300
 - Muutospyynnön lisääminen
 - Muutospyynnön tuominen
 - Muutospyynnön muokkaaminen
 - Muutospyynnön poistaminen
- Äänestykset - 700
 - Äänestyksen luominen
 - Äänestyksen muokkaaminen
 - Äänestyksen sulkeminen
 - Äänestyksen avaaminen
 - Äänestyksen poistaminen
 - Äänestystietojen vieminen järjestelmästä
 - Äänestyksen tulosten katsominen
- Käyttäjien selaaminen - 100
- Äänestystietojen katsominen - 100

Yhteensä 2980 riviä koodia.

6 Riskianalyysi

Riskien todennäköisyyttä ja vakavuutta arvioidaan asteikolla 1-5, jossa 5 on todennäköisin ja vakavin, ja 1 epätodennäköisin ja harmittomin.

	Todennäköisyys	Vakavuus
5	Tapahtuu luultavasti projektin aikana	Pysäyttää projektin etenemisen viikoiksi
4	Tapahtuu mahdollisesti projektin aikana	Hidastaa projektia viikon
3	Saattaa tapahtua projektin aikana	Hidastaa projektia päiviä
2	Mahdollisesti ei tapahdu	Hidastaa projektia päivän

1	Luultavasti ei tapahdu	Aiheuttaa hieman lisätyötä
---	------------------------	----------------------------

6.1 Teknologia

Suunnittelun puutteellisuus tulee esiin toteutusvaiheessa

Todennäköisyys: 2

Vakavuus: 5

Prioriteetti: 10

Ennakointi: Vaiheet jonkin verran limittäin

Jatkosuunnitelmia: Paluu suunnitteluvaiheeseen korjaamaan ongelma.

Aikataulua joutuu mahdollisesti tiukentamaan.

Tiedostojen katoaminen

Todennäköisyys: 2

Vakavuus: 4

Prioriteetti: 8

Ennakointi: Laitoksen palvelimella, viikottainen backup. Versionhallinta.

Jatkosuunnitelmia: Pelastetaan mitä pystytään ja jatketaan siitä.

Valitut työkalut osoittautuvat huonoiksi

Todennäköisyys: 2

Vakavuus: 4

Prioriteetti: 8

Ennakointi: Valitaan aina ryhmästä joku tutustumaan tuntemattomaan työkaluun. Valitaan tuttuja työkaluja.

Jatkosuunnitelmia: Pyritään vaihtamaan työkaluun, joka ajaa saman asian, mutta on helppokäyttöisempi tai tuttu ryhmäläisille.

Laitteisto-ongelmat

Todennäköisyys: 1

Vakavuus: 4

Prioriteetti: 4

Ennakointi: Pidetään omat laitteet kunnossa. Luotetaan laitoksen laitteistojen toimintaa. Ei olla riippuvaisia yhdestä laitteesta. Erityisesti ongelmat tietokantapalvelin [db:llä](#) ovat vakavia. Kesäaika saattaa aiheuttaa sen, että

mahdollisten ongelmien korjaukset kestävät kauan.

Jatkosuunnitelmia: Tehdään tiiviimpää ryhmätyötä, kunnes laitteet ovat kunnossa. Eivät kuitenkaan kaikki yhtäaikaa mene

6.2 Ryhmä

Jäsen on yllättäen estynyt työskentelemään useampana päivänä peräkkäin

Todennäköisyys: 4

Vakavuus: 4

Prioriteetti: 16

Ennakointi: Kaikki ovat valmistautuneita paikkaamaan ja tekemään (osan) estyneen jäsenen työstä. Keskustelu tärkeää.

Jatkosuunnitelmia: Tehtävien mahdollinen uudelleenjakaminen, jos projekti on kriittisessä vaiheessa.

Ryhmän jäsen keskeyttää yllättäen

Todennäköisyys: 2

Vakavuus: 5

Prioriteetti: 10

Ennakointi: Avoin ilmapiiri, keskustellaan asioista ajoissa.

Jatkosuunnitelmia: Tehtävien uudelleenjakaminen. Keskustellaan asiakkaan kanssa vaatimusten vähentämisestä.

Puutteellinen tietotaito yksilötasolla

Todennäköisyys: 4

Vakavuus: 3

Prioriteetti: 12

Ennakointi: Ennakoidaan, mitä tietoja/taitoja tarvitaan ja opetellaan.

Jatkosuunnitelmia: Ylityötunteja opetellessa.

Puutteellinen tietotaito ryhmätasolla

Todennäköisyys: 3

Vakavuus: 3

Prioriteetti:9

Ennakointi: Ennakoidaan, mitä tietoja/taitoja tarvitaan ja opetellaan.

Jatkosuunnitelmia: Ylityötunteja opetellessa.

Ryhmän sisäisessä tiedonvälityksessä katkoksia

Todennäköisyys: 2

Vakavuus: 3

Prioriteetti: 8

Ennakointi: Kaikesta ilmoitetaan kaikille ircissä/mailitse/wiki:n välityksellä.

Viikottaiset tärkeät tiedot myös SVN:ssä.

Jatkosuunnitelmia: Ryhmä tapaa ja keskustelee ongelmasta.

Ryhmän ilmapiiri muuttuu huonoksi

Todennäköisyys: 3

Vakavuus: 3

Prioriteetti: 9

Ennakointi: Pidetään ilmapiiri rentona ja vastaanottavaisena. Yksittäiset riidat jäsenet hoitavat keskenään; vasta viimekädessä ryhmän myötävaikuttaessa. Seurataan kaikkien ryhmän jäsenten osallistumista projektiin, ja toimitaan heti, kun näyttää siltä, että joku ryhmäläinen erakoituu. Viime kädessä ilmoitetaan ohjaajalle tai kurssin vastuuhenkilölle. Palautteen antaminen projektin aikana on suotavaa, mutta se pitäisi osata antaa ohjeistavana eikä ainoastaan negatiivisella tavalla.

Jatkosuunnitelmia: Ryhmä tapaa ja keskustelee ongelmasta.

6.3 Asiakas

Vaatimusten muuttuminen kesken projektia

Todennäköisyys: 2

Vakavuus: 4

Prioriteetti: 8

Ennakointi: Tehdään vaatimusmäärittely kunnolla ja hyväksytetään se asiakkaalla.

Jatkosuunnitelmia: Radikaali muutos on epätodennäköistä, joskin pieniä muutoksia mahdollisesti tulee. Ne käsitellään ja sovitaan asiakkaan kanssa,

mitkä muutokset on mahdollista toteuttaa. Alkuperäisten vaatimusten hyväksymisen jälkeen asiakkaalle voi sanoa, että ehdotus ei välttämättä mene läpi, mutta että sen toteutusta harkitaan.

Epäselvät vaatimukset

Todennäköisyys: 2

Vakavuus: 5

Prioriteetti: 10

Ennakointi: Mahdollisimman paljon keskustelua vaatimusmäärittelyvaiheessa, huolellisuus. Asiakkaan kanssa tarkastustilaisuus vaatimusdokumentille ennen dokumentin jäädytystä.

Jatkosuunnitelmia: Käydään vaatimuksia läpi asiakkaan kanssa tarvittavan kauan ja hyväksytetään vaatimukset asiakkaalla.

Ryhmän ulkopuolinen avainhenkilö ei saatavilla pitkään aikaan/keskeyttää

Todennäköisyys: 2

Vakavuus: 2

Prioriteetti: 4

Ennakointi:

Jatkosuunnitelmia: Oletamme, että tilalle tuodaan toinen asiakas, jonka kautta hoitaa asioita.

6.4 Aikataulu

Aikatauluista myöhästyminen

Todennäköisyys: 4

Vakavuus: 4

Prioriteetti: 16

Ennakointi: Aikatauluun tilaa virhearvioille. Projektille määrätään sopivat tarkistuspisteet, joita valvotaan. Priorisoidaan tekemiset.

Jatkosuunnitelmia: Kiristetään tahtia tai siirretään deadlinejä mahdollisuuksien mukaan. Aikataulutetaan projekti uudelleen, ja mahdollisesti siirretään resursseja sinne, missä niitä eniten tarvitaan. Viimekädessä karsitaan alhaisen prioriteettitason vaatimuksia.

Ohjelmiston koon aliarviointi

Todennäköisyys: 2-3

Vakavuus: 4

Prioriteetti: 10

Ennakointi: Huolellinen vaatimusmäärittely. Priorisoidaan vaatimukset ja aloitetaan toteutus ajoissa.

Jatkosuunnitelmia: Karsitaan vaatimuksia, toteutetaan ensi kädessä vain olennaisimmat.

Työn epätasainen jakaantuminen

Todennäköisyys: 3

Vakavuus: 3

Prioriteetti: 6

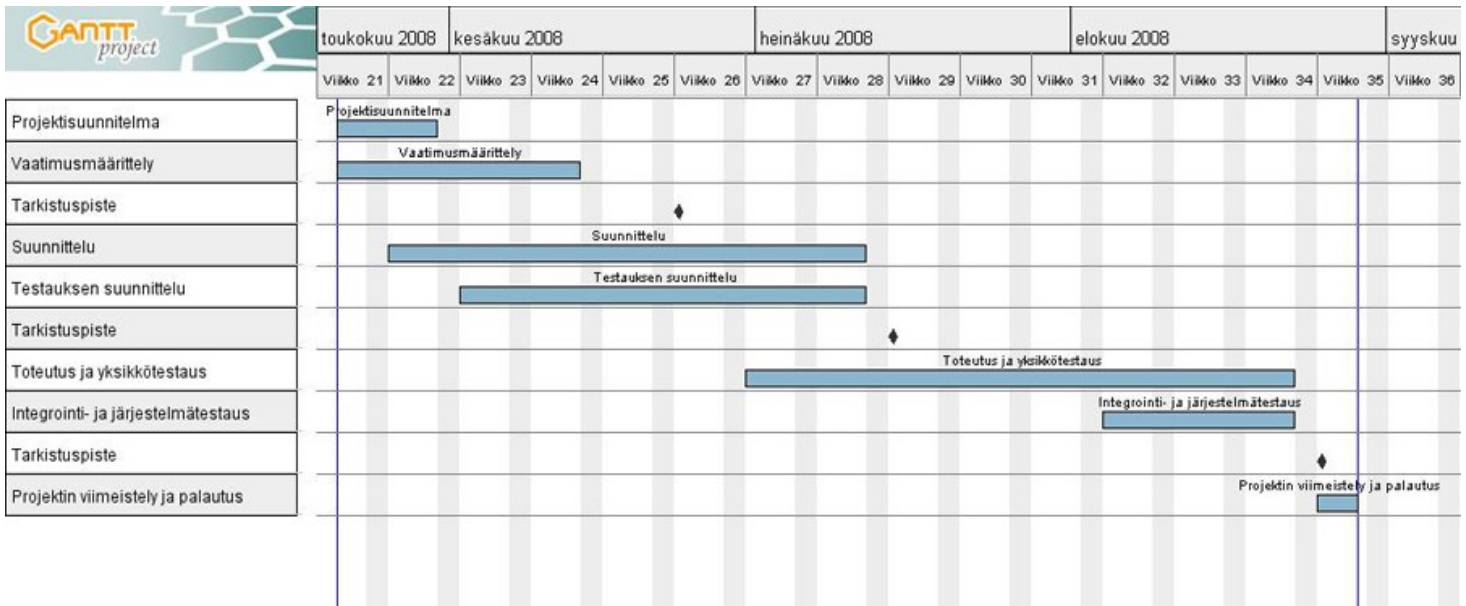
Ennakointi: Seuraus mahdollisesti ilmapiirin huononeminen, joten suomalaisuudesta huolimatta ihmisiä kannustetaan puhumaan ja kertomaan mielipiteensä asiasta. Viikkokokouksissa käydään työtilannetta läpi ja jaetaan tarvittaessa töitä uudelleen

Jatkosuunnitelmia: Ryhmä arvioi kunkin työtaakan ja säättää sitä. Ensi kädessä projektipäällikkö pyrkii pitämään työtaakat yhtenäisinä.

7 Aikataulu

7.1 Projektin aikataulu

Alla Gantt-kaavio projektin aikataulusta. Projektiryhmä pidättää oikeuden muutoksiin.



7.2 Ryhmän resurssit

Kurssi kestää noin 14 kalenteriviikkoa, ja sen laajuus on 9 opintopistettä. Yksi opintopiste vastaa noin 27 tunnin työpanosta, jolloin jokaisen ryhmän kuudesta jäsenestä tulisi yltyä noin 243 työtuntiin projektin aikana. Tällöin viikottainen tuntimäärä on noin 17.4 tuntia, joka on juuri ja juuri yhteensopiva suositellun viikottaisen työ määrän (17-20h/hlö) kanssa. Tuntimäärät ryhmän eri jäsenten välillä saattavat hieman vaihdella projektin eri vaiheissa.

7.3 Tärkeitä päivämääriä

Ensimmäinen käliproton demo 05.06.2008

Projekti päättyy 29.08.2008

8 Projektityöskentely

Ryhmä kokoontuu viikottain tiistaisin sekä torstaisin klo 15-17 salissa A319 tietojenkäsittelytieteen laitoksella, jolloin seurataan projektin edistymistä.

Kokouksissa on läsnä myös ohjaaja Lehtimäki. Kokouksista tuotetaan pöytäkirja, joka muiden ryhmän tuotosten ohella löytyy ryhmän kotisivulta. Virallisissa kokouksissa puheenjohtajana toimii LauraO ja sihteerin rooli vaihtuu kokouksittain aakkosjärjestyksessä. Epävirallisia tapaamisia järjestetään tarvittaessa, ja niistä tehdään tarpeen tullen muistio. Poissaolot ja kesälomat tulee ilmoittaa mahdollisimman hyvissä ajoin projektiryhmälle.

Muu yhteydenpito käy sähköpostin, IRC-kanavan (#muppet ircnetissä) ja wiki:n välityksellä. Häätötilassa otetaan yhteyttä puhelimitse. Ryhmän wiki on osoitteessa <http://wiki.helsinki.fi/display/muppet/>. Wikin hyöty ryhmälle on vielä testattavana, ja yhteydenpitomenetelmät rajataan tulevaisuudessa komboksi, joka toimii parhaiten.

8.1 Kotisivut

Ryhmän kotisivut löytyvät osoitteesta

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

Kotisivuilta löytyy kokousten esityslistat ja pöytäkirjat, dokumentit ja muuta projektiin liittyvää materiaalia.

8.2 Työajan seuranta

Kukin jäsen merkitsee tehdyt tunnit Ohjelmistotuotantoprojektin tietojärjestelmään, joka sijaitsee osoitteessa

http://db.cs.helsinki.fi/~tkt_ohtu/metrics/v0/index.php. Edellisen viikon

työtunnit pitää merkitä ennen viikon ensimmäistä palaveria.

8.3 Tuotettavat dokumentit

Dokumenttien teosta vastaa kunkin alueen vastuhenkilö, kuitenkin niin, että kaikki ryhmän jäsenet osallistuvat dokumenttien tekoon. Kunkin vaiheen merkittynä päättymispäivänä järjestetään katselmointikokouksia, joissa vaiheen aikana tuotettu dokumentti käydään läpi. Dokumentin tulee olla valmiina muutama päivä ennen tilaisuutta, jotta ryhmän jäsenet ehtivät

tutustua siihen. Mahdolliset virheet ja puutteet kirjataan ylös ja vastuuhenkilö hoitaa sovitut korjaukset dokumenttiin. Valmiit dokumentit viedään ryhmän kotisivulle ja muokattavat versiot pidetään SVN:ssä kaikkien saatavilla.

Ryhmä tuottaa projektin aikana seuraavat dokumentit

Sisäiset dokumentit

- Projektisuunnitelma
- Suunnitteludokumentti
- Testausdokumentti
- Yhteenvedodokumentti

Ulkoiset dokumentit

- Vaatimusdokumentti
- Käyttöohje
- Ylläpitodokumentti

9 Projektin vaiheet

Projektiryhmä käyttää projektin toteuttamiseen vesiputousmallia kuitenkin niin, että eri vaiheilla on päällekkäisyyttä. Iteraatio on mahdollinen. Projekti koostuu suuntaa-antavasti seuraavista vaiheista.

9.1 Projektisuunnitelma

Projektin aluksi luodaan alustava projektisuunnitelma, jota päivitetään projektin eri vaiheissa.

9.2 Vaatimusten keruu ja analyysi

Ensimmäisten viikkojen aikana suoritetaan pikainen prioriteettiin perustuva vaatimusten keruu, mikä mahdollistaa karkean käyttöliittymäproton luomisen. Tämä proto esitellään asiakkaalle projektin alkuvaiheessa, jonka jälkeen tai jo käyttöliittymän teon aikana palataan analysoimaan vaatimuksia.

- kartoitus ja analyysi
- sidosryhmät
- määrittely
- validointi

9.3 Ohjelmiston suunnittelu

- Arkkitehtuurisuunnittelu
- Rajapintasuunnittelu
- Oliopohjainen suunnittelu
- Lopullinen käyttöliittymäsuunnittelu

9.4 Toteutus ja yksikkötestaus

Tuotetaan ohjelmisto suunnitteludokumentin mukaan. Asiakas painotti yksikkötestauksen tärkeyttä.

9.5 Järjestelmätestaus

Testataan sovittujen vaatimuksien toimivuus ohjelmistossa.

9.6 Projektin päätös

Kirjoitetaan yhteenvetoraportti, käyttöohje sekä ylläpitodokumentti. Asennetaan ohjelmisto asiakkaan laitteistoon pyydettyäessä. Projekti päättyy 29.08.2008.

Liitteet