Projektisuunnitelma

Dogma

Helsinki 19.9.2006
Ohjelmistotuotantoprojekti
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9 op, 6 ov)

Projektiryhmä

Bjorkgren Joakim
Kuronen Ville
Ristola Toni
Tani Antti
Vihavainen Arto

Ryhmän ohjaaja

Moilanen Ilari

Asiakas

Niinivaara Olli

Kotisivu

http://www.cs.helsinki.fi/u/group/dogma

Dokumentin vastuu

Arto Vihavainen

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	6.9. 2006 19:30	Alustava versio
0.2	10.9. 2006 17:00	Täydennetty dokumenttia ja lisätty pääkohdat
0.3	16.9. 2006 18:00	Täydennetty dokumenttia
0.5	19.9. 2006 09:20	Ensimmäinen versio
0.6	21.9. 2006 20:00	Muutoksia, Alustavan kokoarvion lisäys

Sisältö

1. Johdanto

1.1 Yleistä

Dogma on Helsingin yliopiston tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojektina toteutettava dokumenttiselain. Dokumenttiselaimen tavoitteena on helpottaa työasemalla olevien dokumenttien selaamista ja tarjota tietoa dokumenttien suhteista (lähdeviittaukset yms.) muihin dokumentteihin sekä niihin liittyviin resursseihin (dokumentin tekijät yms.). Selain tulee myös toimimaan pohjana mahdolliselle jatkokehitykselle.

Projekti alkoi 4.9.2006 ja päättyy 17.12. 2006.

1.2 Kuvaus toteutettavasta tuotteesta

Moderni tietotyöläinen saattaa selailla ja käsitellä vuosien mittaan jopa tuhansia dokumentteja. Sekä hakupalveluiden (Google, CiteSeer., Helka, jne.) että työaseman tiedostonhallinnan (Konqueror, Windows Explorer, jne.) tarjoamat palvelut dokumenttien tehokkaaseen selailuun ovat kuitenkin rajalliset. Lisäksi näkymät ja toiminnot ovat palvelukohtaisia, mikä huomattavasti lisää dokumenttienhallinnan tuskaisuutta.

Käytössämme on aikaisempien ohjelmistotuotantoprojektien ansiosta metadataa, joka kuvaa dokumentteja yhtenäisellä tavalla. Tavallisten attribuuttien (dokumentin nimi yms.) se sisältää erityisesti tietoa dokumenttien suhteista muihin dokumentteihin (lähdeviittaukset) ja muihin resursseihin (dokumentin tekijät yms.). Tähän metadataan perustuen pitäisi olla mahdollista toteuttaa dokumenttiselain, joka on ominaisuuksiltaan ylivoimainen muihin välineisiin verrattuna ja (periaatteessa) korvaa niiden käyttöliittymät.

Keskeiset dokumenttiselaimelta vaadittavat toiminnot ovat:

- Resurssien lisääminen ja niiden attribuuttien muuttaminen
- Resurssien (dokumentit, tekijät, jne.) haku attribuuttien perusteella
- Resurssien ja niiden attribuuttien selaaminen taulukossa
- Taulukon järjestäminen haluttujen attribuuttien mukaan
- Osajoukkojen poiminta uusiin taulukoihin ja osajoukkojen nimeäminen
- Resurssien välisten suhteiden visualisointi 2-ulotteisesti verkkona

Lisäksi dokumenttiselain voi tarjota seuraavia toimintoja:

- Dokumentin metadatan (l. attribuuttien) automaattinen luonti dokumentin sisältöä jäsentämällä
- Vapaasanahaku dokumenttien sisällön perusteella
- Käyttäjän toimintaa kuvaavat attribuutit (esim. kuinka usein dokumenttia on katsottu)
- Laskennalliset attribuutit ja monimutkaisemmat kyselyt ("Näytä dokumentit, joiden yksikin tekijä on tehnyt yli 10 dokumenttia")
- Metadatan haku ja julkaisu verkossa niin, että käyttäjän näkymä on työasemariippumaton

1.3 Toteutusympäristö

Ohjelmisto toteutetaan Java-kielellä tietojenkäsittelytieteen laitoksen laitteistoympäristössä.

Mahdollisesti käytettävien kolmansien osapuolten komponenttien tulee olla jonkin vapaan lähdekoodin lisenssin alaisia.

1.4 Immateriaalioikeudet

Projekti toteutetaan Helsingin yliopiston tietojenkäsittelytieteen laitoksen yleisen lisenssisopimuksen alaisuudessa.

2. Projektiorganisaatio

2.1 Projektiorganisaatio

Dogma-ryhmä seuraa hajautettua ryhmämallia.Jokaisella ryhmän jäsenellä on vastuualue, josta hänellä on päätäntävalta. Muuten ryhmän toiminta on tasaarvoista. Kaikki saavat esittää kysymyksiä, kommentteja ja palautetta kenelle tahansa.

Ryhmän jäsenten vastuualueet ovat seuraavat, suluissa varavastuualueet:

Björkgren Joakim. Suunnitteluvastaava (Koodausvastaava).

Kuronen Ville. Testausvastaava, Vaatimusmäärittelyvastaava (Projektipäällikkö).

Ristola Toni. Koodausvastaava (Testausvastaava, Dokumenttivastaava).

Tani Antti. Dokumenttivastaava (Suunnitteluvastaava).

Vihavainen Arto. Projektipäällikkö (Vaatimusmäärittelyvastaava).

2.2 Vastuualueiden kuvaus

Projektipäällikkö. Projektipäällikkö vastaa projektisuunnitelmasta, projektin aikataulusta henkilöiden allokoinnista tehtäviin ja riskienhallinnasta.

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.

2.3 Projektiryhmän käytännöt

Projektiryhmä kokoontuu virallisiin kokouksiin kaksi kertaa viikossa. Pääsääntöisesti kokoukset järjestetään tiistaisin kello 8:30 salissa A218 ja torstaisin kello 16:15 salissa CK109. Kokousten aikoja voidaan muuttaa projektiryhmäläisten pyynnöstä. Pääsääntöisesti projektiryhmän kokouksissa puheenjohtajana toimii käsiteltävän aiheen vastaava henkilö. Sama henkilö laatii myös ennen palaveria kokouksen esityslistan ja välittää sen muille nähtäväksi. Asiakaspalaverit sovitaan aina tarvittaessa asiakkaan kanssa.

Sihteeri toimittaa tekemänsä kokouspöytäkirjan koko projektiryhmälle kokouksen päätöksen jälkeen. Sihteeri valitaan aakkosjärjestyksessä projektiryhmän osallistujista, kuitenkin niin ettei puheenjohtaja voi toimia sihteerinä.

Esityslistan ja pöytäkirjan suunnittelussa käytetään osoitteessa http://www.cs.helsinki.fi/group/ohtu/resurssit/kokoukset/ olevia dokumenttimalleja.

Kokouksien ja tapaamisien ulkopuolella yhteydenpito hoidetaan IRC-kanavan (IRCNET, #ohtu06-dogma), wikin ja sähköpostin välityksellä. Projektille on luotu yhteinen postituslista: ohtus06-dogma-list@cs.helsinki.fi.

2.4 Projektiryhmän resurssit

Projektiryhmän käytössä on tietojenkäsittelytieteen laitoksen atk-palvelut, joihin kuuluu ryhmähakemisto (home/group/dogma), sähköpostilista sekä SVN-versionhallintajärjestelmä. Lisäksi käytössä on myös WIKI-palvelin. Projektiryhmä käyttää tietojenkäsittelytieteen laitoksen saleja ja tietokoneita.

3. Riskianalyysi

Riski on tapahtuma, josta toivoisi ettei se koskaan toteutuisi. Riskien kartoituksella pyritään ennakoimaan projektia uhkaavia tapahtumia, jotta niiden aiheuttama vahinko voidaan minimoida tai jopa välttää kokonaan. Tässä osuudessa kartoitetut Dogma-projektin riskit on jaoteltu ohjelmisto-, henkilöstö- ja järjestelmäriskeihin.

Riskien todennäköisyydelle käytetään seuraavaa luokistusta:

Erittäin todennäköinen, riski toteutuu todennäköisyydellä $p > 0.85$

Todennäköinen, riski toteutuu todennäköisyydellä $0.6 < p < 0.85$

Mahdollinen, riski toteutuu todennäköisyydellä $0.35 < p < 0.6$

Vähäinen, riski toteutuu todennäköisyydellä $0.15 < p < 0.35$

Epätodennäköinen, riski toteutuu todennäköisyydellä $p < 0.15$

Riskien vakavuudelle käytetään seuraavaa luokitusta:

Katastrofaalinen, riskin toteutuminen lopettaa projektin.

Vakava, riskin toteutuminen vahingoittaa projektia ja voi estää sen jatkumisen.

Siedettävä, riskin toteutuminen haittaa projektia ja saattaa estää eräiden haluttujen ominaisuuksien toteuttamisen sekä haitata projektin pysymistä aikataulussa

Vähäpätöinen, riskin toteutuminen haittaa projektia ja aiheuttaa lisätyötä, muttei estä projektia valmistumasta aikataulussa toivotuin ominaisuuksin

3.1 Ohjelmistoriskit

3.1.1 Ohjelmistoriskit / Yleiset

Riski:	**Lisenssiriski**
Kuvaus:	Kolmannen osapuolen ohjelmiston lisenssi on epäyhteensopiva käytettävän DocMan lisenssin kanssa
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Vakava / Katastrofaalinen
Toimenpiteet:	Etsitään tarvittaessa saatavilla olevia samankaltaisia komponentteja joiden lisenssit ovat yhteensopivia.

Riski:	**Komponenttien bugit**
Kuvaus:	Komponenteissa olevat bugit hankaloittavat integrointia. Tarkistamalla avoimien projektien bugilistaa tarvittaessa voidaan tämä riski minimoida.
Todennäköisyys:	Mahdollinen
Vakavuus:	Bugista riippuen Vähäpätöinen / Siedettävä
Toimenpiteet:	Yritetään kiertää bugien aiheuttamat haitat. Bugilistoissa usein neuvoja tähän, jos tunnettu bugi. Tuntemattoman bugin ilmoittaminen komponentin valmistajille. Valitaan komponenteista Stable-versiot.

Riski:	**Komponenttien epätäydellisyys**
Kuvaus:	Komponentin kaikki ominaisuudet eivät vastaa haluttuja ja sitä on paikattava. Toinen variaatio tästä on se että komponentin rajapinta on rajallinen ja vaikeuttaa integraatiota. Komponenttien ohjelmointirajapinnan tarkistus helpottanee näiden huomioimista suunnittelu- ja vaatimusmäärittelyvaiheessa.
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Siedettävä / Vakava
Toimenpiteet:	Selvitettävä kiertämismahdollisuudet, pahimmassa tapauksessa supistettava vaatimuksia.

3.1.2 Ohjelmistoriskit / Käyttöliittymä

Riski:	**Käyttöliittymän ja GUESSin välinen rajapinta ei toimi halutulla tavalla.**
Todennäköisyys:	Vähäinen
Vakavuus:	Siedettävä
Toimenpiteet:	Ennaltaehkäisevät toimenpiteet ehkäisevät tätä riskiä, guessiin tarkka tutustuminen.

Riski:	**Käyttöliittymän dynaaminen haku on hidas tai hidastaa muuten liikaa muuta käyttöliittymää.**
Todennäköisyys:	Vähäinen
Vakavuus:	Vähäpätöinen
Toimenpiteet:	Tehokkaiden algoritmien etsiminen ja tarvittaessa tietokannalle rakennettava yksinkertainen välimuisti.

3.1.3 Ohjelmistoriskit GUESS

Riski:	**Jython tuottaa ongelmia johtuen vähäisestä ohjelmointikokemuksesta Pythonin kanssa.**
Todennäköisyys:	Todennäköinen
Vakavuus:	Siedettävä / Vähäpätöinen
Toimenpiteet:	Jython-komponenttien tekijöiden tulisi ainakin käydä läpi Python-tutoriaali

Riski:	**GUESSin verkon interaktiivisuuden tuki ei ole riittävä.**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Vakava
Toimenpiteet:	Katsottava voidaanko interaktiivisuuden tuen voi kiertää käyttöliittymän kautta.

Riski:	**GUESS aiheuttaa ongelmia suorituskyvyn kanssa käsiteltäessä suuria datamääriä.**
Todennäköisyys:	Todennäköinen
Vakavuus:	Siedettävä / Vakava
Toimenpiteet:	Pienennettävä GUESS-näkymää.

3.2 Henkilöstöriskit

Riski:	**Joku sairastuu pitkäaikaisesti**
Todennäköisyys:	Vähäinen
Vakavuus:	Vakava
Oireet:	Ilmoittaa asiasta, ei ole tavoiteltavissa
Toimenpiteet	Jos henkilö ei kykene hoitamaan tehtäviään kotoa käsin, on ryhmän jaettava kyseisen henkilön tehtävät keskinään.

Riski:	**Joku sairastuu lyhytaikaisesti**
Todennäköisyys:	Todennäköinen
Vakavuus:	Vähäpätöinen
Oireet:	Ilmoittaa asiasta
Toimenpiteet	Jos henkilö ei kykene hoitamaan tehtäviään kotoa käsin, jaetaan kyseisen henkilön tehtävät väliaikaisesti ryhmän kesken

Riski:	**Joku jättää kurssin kesken**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Vakava
Oireet:	Ilmoittaa asiasta, henkilö ei ole tavoiteltavissa
Toimenpiteet	Henkilön tehtävät jaettava ryhmän kesken, harkittava projektin vaatimusten karsimista

Riski:	**Ryhmän välinen kommunikointi ei toimi**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Siedettävä
Oireet:	Jäsenet eivät tiedä tehtäviään, kokoukset takkuilevat
Toimenpiteet:	Otetaan asia esille ja parannetaan kommunikoinnin tilaa esimerkiksi puhelimitse, irkissä, kokouksissa ja sähköpostitse

Riski:	**Kommunikointi asiakkaan kanssa ei toimi**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Vakava
Oireet:	Ryhmä ei tiedä mitä pitäisi tehdä, asiakas ei vastaa sähköposteihin
Toimenpiteet:	Ilmoitetaan asiasta ohjaajalle

Riski:	**Jonkun ryhmän jäsenen taidot eivät riitä**
Todennäköisyys:	Mahdollinen
Vakavuus:	Siedettävä
Oireet:	Aikatauluista myöhästyminen, henkilön ilmoittaminen asiasta
Toimenpiteet:	Koitetaan erilaisia työskentelytapoja, pari/ryhmätyötä yliopiston tiloissa, jaetaan vastuuta alueista enemmän muille. Tärkeintä on oppiminen, ei ongelmakohdan ohittaminen

Riski:	**Ryhmän taidot eivät riitä**
Todennäköisyys:	Vähäinen
Vakavuus:	Vakava
Oireet:	Aikataulusta myöhästyminen, työn takkuilu
Toimenpiteet:	Yritetään ratkoa ongelmakohdat yhdessä, karsitaan projektin vaatimuksia

3.3 Järjestelmä- ja teknologiariskit

Riski:	**Työkoneen rikkoutuminen**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Siedettävä
Oireet:	Tietokone menee rikki eikä toimi, tiedostojen katoaminen
Toimenpiteet	Tietokoneen omistajan on tultava laitokselle työskentelemään tai lainattava tietokonetta joltain toiselta.

Riski:	**Tehdyn työn katoaminen järjestelmän/palvelimen kaatuessa**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Siedettävä
Oireet:	Tietoja ei löydy tietokoneen kaatumisen jälkeen
Toimenpiteet	Menetetty työ tehtävä uudestaan, varmuuskopioiden käyttö, tallennetaan työ tarpeeksi usein

Riski:	**Asiakkaan laitteet eivät ole riittävän tehokkaita**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Siedettävä
Oireet:	Järjestelmä takkuilee asiakkaan laitteella, järjestelmä ei toimi asiakkaan laitteella
Toimenpiteet	Muokataan ohjelmistoa asiakkaan järjestelmään sopivaksi tai suositellaan asiakasta hankkimaan sopivammat laitteet

Riski:	**Toteuttamisessa tarvittavien työkalujen toimimattomuus työskentelypisteillä**
Todennäköisyys:	Epätodennäköinen
Vakavuus:	Siedettävä
Oireet:	Työkalu ei toimi työskentelypisteellä
Toimenpiteet:	Selvitetään miksei työkalu toimi työskentelypisteellä. Henkilön on muokattava työpistettään siten, että vastaavaa työkalua voi käyttää työasemalla. Henkilö saattaa joutua vaihtamaan työskentelypistettään

3.4 Muut riskit

Riski:	**Aikataulun venyminen**
Todennäköisyys:	Mahdollinen
Vakavuus:	Vakava
Oireet:	Aikaa jäljellä vähemmän kuin työt vaativat
Toimenpiteet:	Tingitään projektin ominaisuuksista / vaatimuksista

Riski:	**Aikataulun väärät mittasuhteet**
Todennäköisyys:	Vähäinen
Vakavuus:	Siedettävä
Oireet:	Alussa jää paljon ylimääräistä aikaa ja lopussa painaa kiire päälle.
Toimenpiteet:	Karsitaan vaatimuksia jäljellä olevista osista, panostetaan oleellisiin asioihin eikä käytetä liikaa aikaa oheistuotteiden hiomiseen.

Riski:	**Ohjelmiston koon aliarviointi**
Todennäköisyys:	Todennäköinen / Mahdollinen
Vakavuus:	Vakava / Siedettävä
Oireet:	Aikataulu venyy
Toimenpiteet	Tingitään jäljellä olevista projektin ominaisuuksista / vaatimuksista

Riski:	**Vaatimukset muuttuvat jälkikäteen**
Todennäköisyys:	Todennäköinen
Vakavuus:	Siedettävä
Oireet:	Asiakas haluaa poiketa ennalta sovituista vaatimuksista muuttaen niitä tai tuoden uusia vaatimuksia toteutettavalle ohjelmistolle
Toimenpiteet:	Suuria lisävaatimuksia ei enää hyväksytä, pienet vaatimukset voidaan lisätä määrittelydokumenttiin riippuen niiden vaikutuksesta aikatauluun

Riski:	**Vaatimusmäärittely laadittu väärin**
Todennäköisyys:	Vähäinen
Vakavuus:	Vakava
Oireet:	Toteutus ei tyydytä asiakasta
Toimenpiteet:	Määrittely tehdään uusiksi. Otetaan selvää, voiko jo tehdystä työstä käyttää osan hyväksi uudessa järjestelmässä

4. Laitteisto- ja ohjelmistoympäristön vaatimukset

Dogman tuottama ohjelmisto tullaan sijoittamaan yksittäisiin työasemiin. Projektiryhmä on saanut ohjeen, ettei sen tule miettiä järjestelmän toimintojen transitiivisuutta, vaan tämä voidaan toteuttaa myöhemmissä versioissa.

Järjestelmän on määrä toimia laitoksen tyypillisessä työasemassa, mutta myöhemmin järjestelmän olisi tarkoitus olla käytössä myös "normaaleissa" työpisteissä.

Järjestelmän oletusohjelmointikielenä on Java sen portattavuuden takia, mutta järjestelmän osakomponentteja voidaan ohjelmoida joillain muillakin kielillä.

5. Koko- ja kustannusarviot

Aikataulun puitteissa on sovittu että toteutusvaiheeseen voidaan käyttää noin 6 viikkoa. Ryhmän jäsenten on kurssin sääntöjen mukaan tarkoitus työskennellä viikossa noin 15-20 tuntia, joten yhteensä työtunteja viikossa saadaan 75 – 100. Näin ollen 6 viikon toteutusvaiheeseen voidaan allokoida noin 450 – 600 työtuntia, joiden rajoissa toteutus pitää saada tehtyä. Käytännössä kuitenkin työtunneista tullaan joustamaan tarpeen vaatiessa.

5.1 Kokoarvio LOC – menetelmää käyttäen

LOC-menetelmällä arvioidaan koodirivien määrää. Ohjelmisto jaetaan karkeasti pienempiin komponentteihin, jotta saadaan suhteellisen tarkka arvio koodirivien määrästä.

Komponentti	**Arvioitu rivimäärä**
Käyttöliittymä	
Taulukomponentti	500
Layout ja eventlistenerit	500
Leikepöytätoiminnot	400
Muut toimenpiteet	600
Resurssivarasto	
Tiedostojen hallinta	400
Muut resurssivaraston toiminnot	600
Tietokanta	
Hakuolio	400
Rajapinta	150
Muut tietokannan toiminnot	350
GUESS – integraatio ja toiminnot	1200
Muut järjestelmän toiminnot	1000
Yhteensä :	6100 riviä.

5.2 Kokoarvio FP – menetelmää käyttäen

FP-menetelmässä tarkastellaan ohjelmiston toiminnallisuutta; syötteiden, tulosteiden, kyselyiden, tiedostojen ja näyttöjen lukumäärää. Toiminnot luokitellaan (yksinkertainen, tavallinen, vaikea), jonka perusteella määritetään lukumäärälle kertoimet. Tästä tuloksena saadaan ohjelman toimintopisteiden (Function Points) lukumäärä.

Syötteet:

Tiedostojen lisäys

Attribuuttien muokkaus

Attribuuttien lisäys (kaaret, tagit)

Tulosteet:

Resurssien näyttö resurssibrowserissa

Resurssien välisten suhteiden visualisointi 2 – ulotteisesti verkossa

Kyselyt:

Resurssien haku attribuuttien perusteella

Taulukon järjestäminen haluttujen attribuuttien mukaan

Resurssien siirto uuteen näkymään

Tiedoston avaus ohjelmassa

Tiedostot:

Kaikki syötetiedostot

Liittymät:

Tietokantaliittymä

GUESS – liittymä

Tyyppi	**kpl**	**Kerroin**	**Tulos**	**Selitys**
Syötteet	3	6	18	Vaikea
Tulosteet	2	6	12	Vaikea
Kyselyt	4	4	16	Tavallinen
Tiedostot	1	7	7	Yksinkertainen
Liittymät	2	10	20	Vaikea
Yhteensä			**73**	

Kompleksisuuskertoimet:

Onko järjestelmä vikasietoinen? Tarvitaanko varmistuksia ja palautusta? 3

Tarvitaanko tietoliikenneominaisuuksia? 0

Onko hajautettua prosessinhallintaa? 0

Onko suorituskyky kriittinen elementti? 2

Käytetäänkö järjestelmää raskaassa käytössä olevassa ympäristössä? 4

Tarvitaanko interaktiivista tietojen syöttöä? 5

Täytyykö interaktiivinen tietojen syöttö synkronoida usealle näytölle tai operaatiolle? 2

Päivitetäänkö tiedostoja interaktiivisesti? 3

Ovatko syötteet, tulosteen tiedostot tai kyselyt monimutkaisia? 5

Onko ohjelman toiminta monimutkaista? 5

Onko koodi tarkoitettu uudelleenkäytettäväksi? 3

Ovatko ohjelmiston muunnokset ja asennus suunnitelmissa mukana? 0

Onko ohjelmisto suunniteltu toimivaksi useina asennuksia eri organisaatioissa? 0

Onko sovellus suunniteltava käyttäjäystävälliseksi? 3

Yhteensä: 35

Function Points $FP = 73 * (0.65 + 0.01 * 35) = 73$

5.3 Yhteenveto

LOC – menetelmällä saatiin rivien tulokseksi 6100 riviä. Ryhmällä on työaikaa näiden rivien tuottamiseen noin 600 työtuntia, joten tällä arviolla ohjelman pitäisi kasvaa noin 10 riviä tunnissa.

Funktiopistemenetelmällä saatiin tulokseksi 73 funktiopistettä, joka on hieman alakanttiin sillä ohjelman todellinen toiminta tapahtuu “kuoren alla”. Ryhmän pitäisi saada yksi funktiopiste toteutettua 9 tunnissa, jotta aikataulussa pysyminen onnistuisi.

Koodirivejä funktiopisteiden ja empiirisellä tutkimuksella saadun kertoimen avulla 73*97 = 7081, joka on melko lähellä LOC – menetelmän antaman tuloksen kanssa.

(Yllä on käytetty kerrointa joka saatu www-osoitteessa http://www.qsm.com/FPGearing.html olleesta taulukosta, jossa on analysoitu eri ohjelmointikielten riippuvuuksia FP – menetelmällä saatuihin funktiopisteisiin.)

Toteutusvaiheessa yksikkötestaus ja toteutusvaiheen loppuvaiheessa myös integraatiotestaus ottavat aikataulusta osan, joten lopullinen aikamäärä itse ohjelmoinnille voi jäädä hieman arvioitua alemmaksi.

6. Työn ositus

Projekti ositetaan vesiputousmallin mukaisesti seuraaviin osioihin:

1. Aiheeseen tutustuminen ja käytännön järjestelyt. Tehtävänannon yhteydessä saatuun työn aiheeseen liittyvään materiaaliin tutustumista ja ryhmän toiminnan alkuorganisointi.

2. Projektisuunnitelma. Projektisuunnitelma kuvaa ryhmän työskentelytavat, aikataulut, työnjaon ja riskit. Projektipäällikkö kirjoittaa projektisuunnitelman alustavan version, jonka pohjalta ryhmä osallistuu suunnitelman työstämiseen. Projektisuunnitelmaa päivitetään projektin edetessä.

3. Vaatimusmäärittely. Vaatimusmäärittelyssä selvitetään asiakkaan järjestelmälle asettamat vaatimukset, joiden pohjalta kirjoitetaan vaatimusdokumentti. Vaatimusmäärittelyn yhteydessä tutkitaan mahdollisuutta käyttää valmiita komponentteja vaatimusten toteuttamiseen.

4. Suunnittelu. Toteutusvaiheen suunnittelu alkaa vaatimusmäärittelyn loppuvaiheessa, kun määrittely on oleellisilta osiltaan valmiina. Suunnittelun tuloksena syntyy suunnitteludokumentti. Alustava testaussuunnitelma laaditaan suunnitteluvaiheessa.

5. Testaussuunnitelman laatiminen. Alustava testaussuunnitelma laaditaan suunnitteluvaiheessa.

6 . Toteutus. Toteutusvaiheen tuotos on asiakkaan tilaama ohjelmisto. Toteutus toteutetaan kahdessa vaiheessa niin, että ensimmäisen vaiheen tuotos käy asiakkaalla kommentoitavana ja toisessa vaiheessa toteutetaan asiakkaan toivomat muutokset. Yksikkötestausta suoritetaan toteutuksen edetessä.

7. Testaus. Testauksen tavoitteena on saada aikaan toimiva ja stabiili ohjelmisto. Testaus suoritetaan testaussuunnitelman mukaan. Yksikkötestaus tapahtuu toteutuksen yhteydessä. Integrointitestausta suoritetaan osajärjestelmien valmistuessa ja järjestelmätestaus koko toteutuksen päätteeksi.

8. Muut tuotokset. Valmiille järjestelmälle laaditaan käyttöohje, ylläpitodokumentti ja projektista laaditaan loppuraportti.

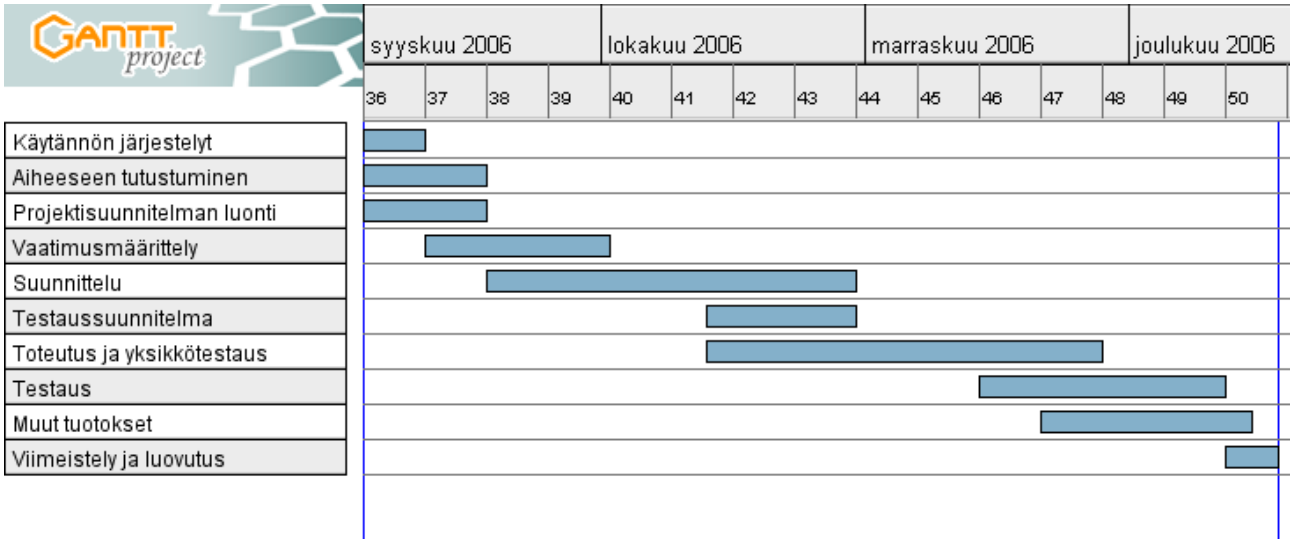
9. Projektin viimeistely ja luovutus.

7. Aikataulu

Projekti aloitettiin 4.9.2006 ja sen palautuspäivämäärä on 17.12.2006. Aikataulu pyritään saamaan sellaiseksi että projektin jakson ensimmäinen ja viimeinen päivämäärä on myös päivänä jolloin on kokous. Tällöin voidaan myös pitää palaveri jossa tarkastetaan tuotosta, tai sovitaan uuden jakson käytännöistä.

Käytännön järjestelyt	4.9.2006 – 11.9.2006
Aiheeseen tutustuminen	4.9.2006 – 19.9.2006
Projektisuunnitelman luonti	4.9.2006 – 19.9.2006
Vaatimusmäärittely	12.9.2006 – 3.10.2006
Suunnittelu	19.9.2006 – 31.10.2006
Testaussuunnitelma	13.10.2006 – 31.10.2006
Toteutus ja yksikkötestaus	13.10.2006 – 28.11.2006
Testaus	14.11.2006 – 12.12.2006
Muut tuotokset	21.11.2006 – 15.12.2006
Viimeistely ja luovutus	12.12.2006 – 17.12.2006

7.1 Gantt – kaavio aikataulusta



8. Mittaus- ja raportointitavat

Jokaisen ryhmäläisen työskentelyä mitataan ajallisesti ohjelmistotuotantoprojektien seurantaa varten kehitetyllä www-sovelluksella, joka löytyy www-osoitteesta http://db.cs.helsinki.fi/~tkt_ohtu/metrics/v0/index.php . Ryhmän jäsenet kirjaavat tuntiseurantansa kerran viikossa seurantajärjestelmään, siten että ennen edellisen viikon työt näkyvät ennen viikon ensimmäistä viikkopalaveria. Tällöin voidaan varmistua siitä, että palaverissa on aina käytettävissä viimeisin tieto kunkin henkilön ajankäytöstä. Sivuilta löytyvät myös ohjeet järjestelmän käyttöön. Jokaisen projektiryhmäläisen olisi suosituksen mukaan työskenneltävä noin 15-20 tuntia viikossa.

Jokaisessa kokouksessa käydään läpi tavoitteet ennen seuraavaa kokousta. Kyseiset asiat kirjataan kokouksen pöytäkirjaan, jota tarkastellaan taas seuraavalla viikolla.