

Projektisuunnitelma

Karstula

Helsinki 3.5.2007

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (9 + 1 op)

Projektiryhmä

Juha-Pekka Juutilainen

Hannu Kukko

Antto Mäkinen

Antti Rajasärkkä

Ari Raunio

Mika Tantarimäki

Asiakas

Eija Mäntyharju

Johtoryhmä

Sanna Keskiöja

Kimmo Simola

Kotisivu

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

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.1	22.1.2007	Dokumentti luotu
0.9	31.1.2007	Dokumenttia päivitetty
1.0	2.5.2007	Dokumentti latex-muotoon

Sisältö

1 Johdanto	1
1.1 Toteutettava ohjelmisto	1
1.2 Aikataulu	1
2 Sanasto	1
3 Projektioorganisaatio	1
3.1 Vastuualueet	2
4 Riskianalyysi	2
4.1 Kuvaustapa	2
4.2 Riski: Aikataulu ei pidä	2
4.3 Riski: Projektin jäsenen pitkä poissaolo/keskeytys	3
4.4 Riski: Dokumentteja tuhoutuu TKTL:n palvelimelta	3
4.5 Riski: Työvälineitä ei hallita riittävän hyvin	4
4.6 Riski: Työvälineet tai valittu tekniikka eivät sovellu projektin tarpeisiin	4
4.7 Riski: Asiakas ei osaa määritellä ohjelmaa tarpeeksi tarkasti	4
4.8 Riski: Asiakas muuttaa toiveitaan vaatimusmäärittelyn jäädytyksen jälkeen	4
4.9 Riski: Asiakas haluaa toimintoja, joita ei ole mahdollista toteuttaa	5
4.10 Riski: Asiakkaan tekninen ympäristö ei olekaan kuvatus kaltainen	5
4.11 Riski: Ominaisuuksia jää suunnittelemaan	6
4.12 Riski: Rajapintojen suunnittelu puutteellista	6
4.13 Riski: Toteutus ei vastaa vaatimusmäärittelyä	6
4.14 Riski: Toteutettu koodi bugista	7
4.15 Riski: Toteutettu koodi vaikeaselkoista	7
4.16 Riski: Suuret ongelmat jonkin asian toteuttamisessa	7
4.17 Riski: Usea ihminen muuttaa samaa tiedostoa ristiriitaisin tavoin	8
4.18 Riski: Kokonaisuuden testaaminen jää puutteelliseksi	8
4.19 Riski: Järjestelmään jää testaamattomia osia	8
4.20 Riski: Ei osata testata kaikkea mahdollista, kuten esim. tietoturvaa	9
5 Projektin laitteisto- ja ohjelmistovaatimukset	9

6	Koko- ja työmääräarviot	9
6.1	Toimintopistearvio (function point)	9
6.2	Koodirivien määrä (lines of code)	12
6.3	Työmääräarvio	12
7	Työn ositus	13
8	Projektin aikataulu	13
9	Seuranta- ja raportointimenetelmät	15
9.1	Tietojärjestelmä	15
9.2	Ryhmän kotihakemisto	15
9.3	Dokumenttien käsittely	15
9.4	Ryhmän sisäinen kommunikointi	15

1 Johdanto

1.1 Toteutettava ohjelmisto

Projekti toteuttaa Karstulan evankeliselle kansanopistolle ohjelmiston lääkelaskujen harjoitteluun ja osaamisen testaamiseen. Testit tulevat tapahtumaan koulun tiloissa olevilla Windows XP –koneilla; harjoittelua voisi tehdä myös kotona tai missä tahansa muualla koulun tilojen ulkopuolella.

1.2 Aikataulu

Aloituspäivä 15.1.2007 ja valmistumisen takaraja 29.4.2007. Yhteensä 15 viikkoa, joista yksi viikko lomaa. Projekti viedään läpi noudattaen suoraviivaista vesiputousmallia. Tarkempi aikataulu on omana lukunaan: 8 Projektin aikataulu.

2 Sanasto

Läakelasku Lääkkeiden annostukseen/sekoittamiseen liittyvä matemaattinen laskutehtävä

3 Projektioorganisaatio

Nimi	Rooli (vararooli)	Sähköposti Puhelin
Juutilainen Pekka	Juhas- Vaatimusmäärittelyvas- taava (Projektipäällikkö)	jjjuutil@cs.helsinki.fi 050 341 6876
Kukko Hannu	Dokumenttivastaava (Testausvastaava)	hmkukko@cs.helsinki.fi 040 578 8476
Mäkinen Antto	Suunnitteluvastaava (Koodivastaava)	aaomakin@cs.helsinki.fi 044 012 1284
Rajasärkkä Antti	Projektipäällikkö (Dokumenttivastaava)	rajasark@cs.helsinki.fi 050 514 9318
Raunio Ari	Koodivastaava (Vaatimusmäärittelyvastaava)	ajraunio@cs.helsinki.fi 044 060 2790
Tantarimäki Mika	Testausvastaava (Suunnitteluvastaava)	mtantari@cs.helsinki.fi 044 297 9150

Postituslista: ohruk07-karstula-list@cs.helsinki.fi

Kotisivu: <http://www.cs.helsinki.fi/group/karstula/>

Ryhmä kokoontuu kahdesti viikossa työpalaveriin: maanataisin klo 14-16 huoneessa A318 ja torstaisin klo 10-12 huoneessa A218. Poikkeuksista tiloihin ilmoitetaan erikseen. Pääasiallinen yhteydenpitotapa kokouksien lisäksi on sähköposti.

3.1 Vastuualueet

Projektipäällikkö vastaa projektisuunnitelman ja aikataulun laadinnasta ja niiden ylläpidosta koko projektin ajan.

Vaativuusmäärittelyvastaava toimii asiakkaan ja projektiryhmän välisenä yhdenhenkilönä ja vastaa yhtenäisestä vaatimuskäsittelyn rakenteesta.

Dokumenttivastaava vastaa dokumenttien yhtenäisestä ulkoasusta ja projektin www-sivusta.

Suunnitteluvastaava vastaa yhtenäisestä suunnittelutason rajapinnoista ja suunnitteludokumentin yhdenmukaisesta rakenteesta

Koodivastaava vastaa koodin yhtenäisestä ulkoasusta ja rajapintojen yhtenäisyydestä.

Testausvastaava vastaa testauksen kattavuudesta ja testaussuunnitelmasta

4 Riskianalyysi

4.1 Kuvaustapa

Riski Lyhyt kuvaus riskistä

Todennäköisyys Suuri – mahdollinen – pieni

Vakavuus Vähäpätöinen – siedettävä – vakava – tuhoisa

Minimointi Riskin todennäköisyyden ja vakavuuden pienentämiseksi tehtävät toimenpiteet

Tunnistaminen Miten riskin toteutuminen tunnistetaan

Toteutuessa Toimenpiteet riskin toteutuessa

4.2 Riski: Aikataulu ei pidä

Todennäköisyys Suuri

Vakavuus Siedettävä

Minimointi Seurataan töiden edistymistä suhteessa suunnitteluun tarkasti vähintään viikoittain. Kaikki osallistuvat suuriin työvaiheisiin kuten suunnitteluun ja toteutukseen.

Pyritään tekemään vaatimusmäärittely ja suunnittelu niin kattavasti, että ainakaan suuria muutostöitä ei tulisi enää toteutusvaiheessa. Omissa tehtävissä vaikeuksiin törmätessä pyydetään mieluummin apua kuin jäädään pitkäksi aikaa paikoilleen. Varaudutaan aikataulua suunniteltaessa myös pieniin viivästyksiin.

Tunnistaminen Tehtävät eivät valmistu ajoissa, eikä seuraavia tehtäviä saada käynnistettyä suunnitellussa aikataulussa.

Toteutuessa Suuret ongelmatilanteet pyritään ratkaisemaan joukolla. Jos viivästyks koskee vain muutamaan henkilöä, pyritään jakamaan heidän seuraavia töitä muille.

4.3 Riski: Projektin jäsenen pitkä poissaolo/keskeytys

Todennäköisyys Pieni

Vakavuus Vakava

Minimointi Jokainen pitää oman vastuualueensa lisäksi itsensä ajan tasalla varatehtävänsä tilanteesta. Lisäksi jokainen tallentaa vapaavalintaisin väliajoin työtiedostonsa (myös ei-valmiit) projektin kotikansioon. Jos jollekulle tulee selväksi, että projekti on keskeytettävä, siitä on ilmoitettava mahdollisimman nopeasti.

Tunnistaminen Jäsen ilmoittaa keskeyttävänsä tai häneen ei yrityksistä huolimatta saada yhteyttä viikkoon.

Toteutuessa Tiedotetaan asiasta koko projektiryhmälle, jonka jälkeen sovitaan, miten poistuvan henkilön työt jaetaan.

4.4 Riski: Dokumentteja tuhoutuu TKTL:n palvelimelta

Todennäköisyys Pieni

Vakavuus Tuhoisa

Minimointi Jos mahdollista, säilytetään jotain varmuuskopiota myös TKTL:n ulkopuolella. Varataan muutama päivä pelivaraa ennen määräaikoja siltä varalta, että tarvitaan palautusta laitoksen varmistusnauhoilta.

Tunnistaminen Tiedostoja on kadonnut.

Toteutuessa Otetaan yhteyttä laitoksen ATK-ylläpitoon ja pyydetään heitä palauttamaan kadonneet tiedostot varmistuksiltaan; tai palautetaan kadonnut tiedosto omalta varmistukselta, jos sellainen on olemassa.

4.5 Riski: Työvälineitä ei hallita riittävän hyvin

Todennäköisyys Mahdollinen

Vakavuus Vähäpätöinen

Minimointi Valitaan sellaisia työvälineitä, jotka kaikki tuntevat entuudestaan. Uudet välineet esitellään kaikille tai vähintään jaetaan käyttöohjeet.

Tunnistaminen Työvaihe ei tahdo edetä tai lopputulos ei ole sovituslainen.

Toteutuessa Tehdään ko. työ mahdollisimman hyvin, esimerkiksi dokumenttien tekstin voi tuottaa, vaikkei tuntisikaan Latex-käskyjä. Tarvittaessa pyydetään apua muilta projektin jäseniltä.

4.6 Riski: Työvälineet tai valittu tekniikka eivät sovellu projektin tarpeisiin

Todennäköisyys Mahdollinen

Vakavuus Vähäpätöinen

Minimointi Tutustutaan ennen valintaa työvälineeseen/tekniikkaan.

Tunnistaminen Jonkin halutun ominaisuuden toteuttaminen tuottaa vaikeuksia.

Toteutuessa Pyritään etsimään "kiertotie".

4.7 Riski: Asiakas ei osaa määritellä ohjelmaa tarpeeksi tarkasti

Todennäköisyys Suuri

Vakavuus Siedettävä

Minimointi Pidetään huolta siitä, että määrittely tulee tehtyä riittävän tarkasti. On projektin jäsenten tehtävä saada asiakkaalta riittävästi tietoa määrittelyn pohjaksi. Viime kädessä määrittelyn tarkastuksessa pitää löytää kaikki mahdolliset tulkinnanvaraiset kohdat.

Tunnistaminen Suunnittelu- tai toteutusvaiheessa huomataan, että jonkin kohdan voi ymmärtää monella eri tavalla, tai ettei toimintoa ole määriteltä ollenkaan.

Toteutuessa Vaatimusmäärittelyvastaava päättää mitä tehdään, tarvittaessa neuvoteltuaan asiakkaan kanssa. Päätöstä tehtäessä on otettava huomioon sen mahdollisesti muualla järjestelmässä aiheuttamat muutokset.

4.8 Riski: Asiakas muuttaa toiveitaan vaatimusmäärittelyn jäädytyksen jälkeen

Todennäköisyys Pieni

Vakavuus Vakava

Minimointi Tehdään tiivistä yhteistyötä asiakkaan kanssa vaatimusmäärittelyä luottaessa ja painotetaan, että jäädytyksen jälkeen muutoksia ei voi tehdä.

Tunnistaminen Asiakas toivoo jotain, joka on ristiriidassa määrittelyn kanssa.

Toteutuessa Selvitetään, onko muutos mahdollista toteuttaa helposti (esim. päivämäärän lisääminen käyttöliittymän ylänurkkaan) vai aiheutuuko toiveesta suuria vaikeuksia. Jos näin on, selitetään asiakkaalle, miksi muutosta ei voida tehdä ja tarvittaessa informoidaan kurssin vastuuhenkilöä.

4.9 Riski: Asiakas haluaa toimintoja, joita ei ole mahdollista toteuttaa

Todennäköisyys Mahdollista

Vakavuus Siedettävä

Minimointi Vaatimusmäärittelyä laadittaessa arvioidaan toivomusten monimutkaisuus ja karsitaan selvästi mahdottomat toivomukset joukosta asiakkaan kanssa yhteistyössä.

Tunnistaminen Projektin kaikkia vaatimuksia ei näytetä saavan valmiiksi millään takarajan lähestyessä.

Toteutuessa Karsitaan ei-keskeisiä ominaisuuksia ohjelmistosta siten, että toimiva kokonaisuus saadaan tehtyä projektin lopputuloksena. Tästä tietysti informoidaan myös asiakasta.

4.10 Riski: Asiakkaan tekninen ympäristö ei olekaan kuvatus kaltaisen

Todennäköisyys Mahdollista

Vakavuus Vakava

Minimointi Pyritään selvittämään asiakkaan tekniseltä yhteyshenkilöltä mahdollisimman tarkasti ympäristöön liittyvät kysymykset. Tehdään myös ennen varsinaista toimitusta pieni testiasennus esimerkiksi Hello world –tyyppisellä ohjelmalla ja käyttöliittymän osalla. Tarvittaessa liitetään asennuspakettiin myös sopiva versio JRE:stä.

Tunnistaminen Testiasennus ei onnistu.

Toteutuessa Selvitetään, mistä ongelmat johtuvat, ja pyritään ratkaisemaan ne ensisijaisesti saattamalla ympäristö alkuperäisen kuvauksen kaltaiseksi (viimekädessä tämä on asiakkaan ongelma)

4.11 Riski: Ominaisuuksia jää suunnittelematta

Todennäköisyys Mahdollinen

Vakavuus Vakava

Minimointi Vaatimusmäärittelyn on oltava riittävän tarkka ennen jäädytystä. Suunnitellaan kriittiset kohdat erityisellä huolella, jotta mahdolliset puutteet koskevat vain järjestelmän reuna-alueita.

Katselmoidaan (epävirallisesti) suunnittelua sitä mukaa, kun osia valmistuu. Tarkastuksissa pyritään löytämään kaikki puutteet.

Suunnitellaan ohjelmisto toimimaan modulaarisesti, jolloin yhden moduulin ongelmat eivät kuitenkaan vaikuta koko ohjelmistoon.

Tunnistaminen Toteutusvaiheessa tulee ongelmia.

Toteutuessa Tehdään suunnitelma, jonka vaikutukset muualle järjestelmään analysoidaan. Pyritään välttämään suuria lisäyksiä, jotka vaikuttavat koko järjestelmän rakenteeseen.

4.12 Riski: Rajapintojen suunnittelu puutteellista

Todennäköisyys Todennäköinen

Vakavuus Vakava

Minimointi Kaikkien on osallistuttava riittävästi suunnitteluun, jotta rajapinnat saadaan suunniteltua kunnolla. Tarkastuksissa kiinnitetään erityisesti huomiota rajapintojen kuvauksiin.

Tunnistaminen Toteutusvaiheessa törmätään monitulkintaisesti määritelyihin kohtiin tai integrointitestauksessa huomataan, että komponentit eivät toimi yhdessä.

Toteutuessa Törmätessä tulkintaongelmiin, ei tehdä oletuksia, vaan kysytään/sovitaan muutoksesta. Muutosta tehtäessä otetaan huomioon myös rajapintaa käyttävät komponentit. Viimekädessä tästä vastaa rajapintojen vastuuhenkilö.

4.13 Riski: Toteutus ei vastaa vaatimusmäärittelyä

Todennäköisyys pieni

Vakavuus vakava

Minimointi Tarkastetaan suunnitelmat määrittelyä vasten ja suoritetaan kattava ohjelmistotestaus. Määrittelyn yhteydessä tehdään käyttöliittymäproto. Erityisesti ohjelmiston keskeinen toiminta on määriteltävä ja suunniteltava huolellisesti, jotta ydintoiminnallisuus olisi joka tapauksessa asiakkaan tarpeita vastaava.

Tunnistaminen Ohjelmistoa testattaessa huomataan puutteita/ristiriitoja määritte-

lyyn verrattuna.

Toteutuessa Pyritään korjaamaan pienet puutteet ja virheet. Jos aikataulu ei salli muutoksia, sitten niitä ei tehdä.

4.14 Riski: Toteutettu koodi bugista

Todennäköisyys mahdollinen

Vakavuus siedettävä

Minimointi Testataan lopputulos huolellisesti: sekä koodaaja itse että joku muu. Katselmoidaan koodia ajoittain.

Tunnistaminen Testattaessa törmätään virheisiin.

Toteutuessa Jäljitetään virheet ja pyritään korjaamaan ne.

4.15 Riski: Toteutettu koodi vaikeaselkoista

Todennäköisyys mahdollinen

Vakavuus siedettävä

Minimointi Luodaan yhteiset koodauskäytännöt, joita kaikki noudattavat. Katselmoidaan koodia ajoittain. Kommentoidaan oma koodi nimenomaan sillä ajatuksella, että joku toinenkin saa koodista selvää.

Tunnistaminen Toisen kirjoittamasta koodista ei tahdo saada selvää.

Toteutuessa Ensisijaisesti koodin kirjoittanut henkilö korjaa koodinsa selvennämiseksi.

4.16 Riski: Suuret ongelmat jonkin asian toteuttamisessa

Todennäköisyys todennäköinen

Vakavuus siedettävä

Minimointi Varaudutaan aikataulussa pieniin viivästyksiin ja jaetaan työtehtävät tasaisesti, jotta tarvittaessa apua pystytään antamaan.

Tunnistaminen Tehtävät eivät valmistu ajoissa

Toteutuessa Ongelmiin törmätessä pyydetään apua muilta projektin jäseniltä nopeasti;

4.17 Riski: Uusea ihminen muuttaa samaa tiedostoa ristiriitaisin tavoin

Todennäköisyys pieni

Vakavuus siedettävä

Minimointi Käytetään versionhallintaa CVS-ohjelmaa ja sovitaan tiedostojen käytöstä muutenkin. Selkeät sopimukset siitä, minkälainen kunkin tiedoston on määrä olla.

Tunnistaminen Tehtyjä muutoksia katoaa. Halu tehdä muutoksia toisen vastuulla olevaan tiedostoon.

Toteutuessa Pyritään ratkaisemaan ristiriitaisuudet CVS:n avulla. Pahimmassa tapauksessa muutokset joudutaan tekemään uudestaan. Neuvotellaan ja sovietaan yhdessä, minkälainen tiedostosta piti tulla.

4.18 Riski: Kokonaisuuden testaaminen jää puutteelliseksi

Todennäköisyys pieni

Vakavuus vakava

Minimointi Suunnitellaan komponentit sellaisiksi, että niitä pystytään testaamaan yhdessä ilman kaikkia komponentteja.

Tunnistaminen Komponenttien integrointitestausta ei onnistu ilman jotain tiettyä komponenttia.

Toteutuessa Käytetään apuna tyhmiä tynkäkomponentteja (dummyt).

4.19 Riski: Järjestelmään jää testaamattomia osia

Todennäköisyys todennäköistä

Vakavuus siedettävä

Minimointi Suunnitelman mukainen testaus kattaa keskeiset osat. Koodaaja kokeilee kaikkien rajapintojensa/käyttäjien syötteiden kelpoiset ja kelvottomat syötteet siten, että kaikki haarautumat tulisi testattua.

Tunnistaminen Asiakas törmää virheelliseen toimintoon joskus projektin päättämisen jälkeen.

Toteutuessa Asiakkaan on joko itse ratkaistava ongelma tai perustettava uusi ylläpitoprojekti

4.20 Riski: Ei osata testata kaikkea mahdollista, kuten esim. tietoturvaa

Todennäköisyys mahdollinen

Vakavuus siedettävä

Minimointi Käytetään suunnittelussa ja testauksessa hyväksi koko projektiryhmän yhteistä osaamista.

Tunnistaminen Asiakas törmää joskus projektin jälkeen tällaisiin ongelmiin.

Toteutuessa Asiakkaan on joko itse ratkaistava ongelma tai perustettava uusi projekti.

5 Projektin laitteisto- ja ohjelmistovaatimukset

1. Asiakkaan ympäristöä vastaava testikone (Windows XP)
2. Kehitysohjelmat: Eclipse tai vapaavalintainen Java-kehitin
3. Versionhallinta CVS
4. Dokumentointi – Latex (pohjat resurssisivulta)
5. Muuta: asennus, JRE jne?

6 Koko- ja työmääräarviot

6.1 Toimintopistearvio (function point)

	Helppoja	keskivaikeita	Monimutkaisia
Päävalikko			
Syötteitä	1	1	
Tulosteita	1		
Kyselyitä	1		
Tiedostoja		1	
Ulkoisia rajapintoja			

Harjoittelunäkymä			
Syötteitä	3		
Tulosteita	2	1	
Kyselyitä		1	
Tiedostoja		1	
Ulkoisia rajapintoja			
Testinäkymä			
Syötteitä		1	
Tulosteita	1	1	
Kyselyitä	2		
Tiedostoja	1		
Ulkoisia rajapintoja		1	
Ylläpitonäkymä			
Syötteitä	2	1	
Tulosteita	2	1	
Kyselyitä			
Tiedostoja (sisältyy aikaisempiin kohtiin)			
Ulkoisia rajapintoja			

Summa ja painoker- toimet	Helppo	Keskivaikea	monimutkai- nen	Sum- ma
Syötteitä	6 * 3	3 * 4	0 * 6	30
Tulosteita	6 * 4	3 * 5	0 * 7	39

Kyselyitä	3 * 7	1 * 10	0 * 15	31
Tiedostoja	1 * 7	2 * 10	0 * 15	27
Ulkoisia rajapintoja	0 * 5	1 * 7	0 * 10	7
Summa				134

Kompleksisuuskerroimet

Asteikko 0=ei tarvita . . . 5=ehdottomasti tarvittava toiminto

1. Tarvitseeko järjestelmä luotettavaa varmuuskopiointia ja palautusta? **1**
2. Tarvitaanko tietoliikenneyhteyksiä? **0**
3. Onko järjestelmässä hajautettua laskentaa? **0**
4. Onko suorituskyky kriittinen tekijä? **0**
5. Toimiiko järjestelmä olemassa olevassa raskaasti kuormitetussa ympäristössä? **0**
6. Vaatiiko järjestelmä suorituksen aikaista tiedon syöttöä? **5**
7. Vaatiiko tiedon syöttäminen syöttöprosessin jakamista usealle näytölle tai ikkunalle? **0**
8. Päivitetäänkö tiedostoja suorituksen aikana? **3**
9. Ovatko syötteet, tulosteet, tiedostot tai kyselyt monimutkaisia? **1**
10. Onko sisäinen prosessointi monimutkaista? **1**
11. Suunnitellaanko koodi käytettäväksi uudelleen? **1**
12. Sisältyykö suunnitelmaan konversio ja asennus? **2**
13. Onko järjestelmä suunniteltu asennettavaksi useita kertoja eri organisaatioissa? **2**
14. Suunnitellaanko järjestelmä helposti muunneltavaksi ja käyttäjäystävälliseksi? **4**

Kompleksisuuskerrointen summa: $\sum F_i = 20$

Toimintopisteet: $134 * (0,65 + 0,01 * 20) = 113,9$

6.2 Koodirivien määrä (lines of code)

Arviot Javan koodirivien määrästä toimintopistettä kohti vaihtelevat suuresti keskiarvon ollessa noin 60 riviä/toimintopiste. Näin ollen toimintopistearviosta voidaan johtaa rivien määrä seuraavasti:

Rivien määrä = toimintopisteet * 60 riviä/toimintopiste = 113,9 * 60 riviä = **6834 riviä**.

6.3 Työmääräarvio

COCOMO II –mallin mukaan työmäärä henkilötyökuukausina arvioidaan seuraavasti:

Työmäärä = $2,94 * (\text{koko tuhansina riveinä})^B * M$

Vakio B (arvot 0=erittäin hyvä ... 5=erittäin vähäinen):

Ennakkotuntemus: 3

Kehitystyön joustavuus: 1

Riskien hallinta: 0

Tiimin yhteistyö: 2

Prosessin kehittyneisyys: 4

1. Summa: 10 => B:n arvoksi saadaan $1,01 + 10/100 = 1,1$

Vakio M

Tarkoitus olisi arvioida vielä projektin monimutkaisuutta ja riskejä seitsemän eri muuttujan avulla, jotka voivat lisätä tai laskea työmäärää:

Tuotteen vaadittu luotettavuus ja monimutkaisuus: 0,75

Uudelleenkäytön vaatimukset: 0,75

Ohjelmistoalustan monimutkaisuus: 0,75

Henkilöstön pätevyys: 1,3

Aikataulu: 1

Tukipalvelut: 1

1. $M = 0,75 * 0,75 * 0,75 * 1,3 * 1 * 1 = 0,55$

Työmäärä

Työmääräarvio on täten: $2,94 * 6,834^{1,1} * 0,55 = \mathbf{13}$ henkilötyökuukautta.

Projektissa on käytettävissä kuusi henkilöä 20 tuntia/viikko neljän kuukauden ajan = $6 * 4 * 0,5 = 12$ henkilötyökuukautta.

7 Työn ositus

Työmäärä pyritään pitämään tasaisena läpi projektin. Tämä vaatii sitä, että suuriin (ainakin suunnittelu, toteutus) työvaiheisiin osallistuvat kaikki, eikä vain ko. osa-alueen vastuhenkilö.

8 Projektin aikataulu

lihavoidut kurssin vaatimuksia

Kursivoidut vaativat varmistuksia

Vko	N	Pvm	Tuotos
3	1	Ma 15.1.	Projekti alkaa Vaatusmäärittely alkaa
4	2	Ma 22.1.	1. käyttöliittymädemo Arkkitehtuurisuunnittelu alkaa
5	3	Ma 29.1. To 1.2.	2. käyttöliittymädemo (asiakas) Projektisuunnitelma ”valmis” (sisäinen) Vaatusmäärittelyn raakaversio ohjaajalle ja testaajalle
6	4	Ma 5.2. To 8.2. Pe 9.2.	Vaatusmäärittelyn palaute ja korjaus Yksityiskohtainen suunnittelu alkaa - käyttöliittymä – komponenttien sisäiset rakenteet – testaussuunnittelu Vaatusdokumentti (ulkoinen) asiakkaalle Arkkitehtuurisuunnitelman katselmointi Arkkitehtuurisuunnitelma ohjaajalle
7	5	Ma 12.2. To 15.2.	Arkkitehtuurisuunnitelman palaute ja korjaus Vaatusdokumentin tarkastus ja jäädytys Arkkitehtuurisuunnitelman tarkastus
8	6	Ma 19.2. To 22.2. Pe 23.2.	Testauskoulutus klo 14-16 Suunnitteludokumentti ohjaajalle

9	7	Ma 26.2. To 1.3.	Suunnitteludokumentin palaute ja korjaus Testaussuunnitelma ohjaajalle Suunnitteludokumentin tarkastus Testaussuunnitelman palaute ja korjaus
10	8		[loma 5.3.-11.3.] Toteutus alkaa
11	9	Ma 12.3. To 15.3.	Testaussuunnitelman tarkastus
12	10		
13	11		
14	12	Ma 2.4. To 5.4.	Integrointitestaus alkaa Käyttöohje ohjaajalle
15	13	To 12.4.	[Pääsiäinen 5.-11.4.] Käyttöohjeen palaute ja korjaus Ylläpitodokumentti ohjaajalle Toteutus valmis: versio 0.9 – asiakasdemo (korjataan vain mahd. virheet)
16	14	Ma 16.4. To 19.4.	Käyttöohjeen tarkastus Ylläpitodokumentin palaute Yhteenvedodokumentti ohjaajalle Käyttöohjeen (ulkoisen) jäädytys Ylläpitodokumentin tarkastus Yhteenvedodokumentin palaute Lopullinen versio 1.0
17	15	Ma 23.4. Ke 25.4. To 26.4.	Ylläpitodokumentin (ulkoisen) jäädytys Yhteenvedodokumentin tarkastus Demopäivä Yhteenvedodokumentin (sisäinen) jäädytys
18	16	Ma 30.4. Su 6.5.	Viimeinen palaveri 10-12 Projekti päättyy

9 Seuranta- ja raportointimenetelmät

9.1 Tietojärjestelmä

Kaikki kirjaavat työtuntinsa ennen maanantain palaveria Ohjelmistotuotantoprojektien tietojärjestelmään: http://db.cs.helsinki.fi/~tkt_ohtu/metrics/v0

Projektipäällikkö vastaa siitä, että järjestelmään kirjataan pakolliset metriikkasarjat kunkin projektin osavaiheen jälkeen.

9.2 Ryhmän kotihakemisto

Laitoksen järjestelmissä polku: /group/home/karstula.

Jokaisella ryhmän jäsenellä on oma kansionsa ryhmän kotihakemiston alla. Sinne tallennetaan keskeneräisiä töitä vapaassa tahdissa.

9.3 Dokumenttien käsittely

Dokumentit kirjoitetaan valmiiden pohjien mukaisiksi. Pohjat on tallennettu projektinryhmän kansioon /karstula/dokumentit. Dokumentit julkaistaan PDF-muodossa kotisivulla.

Kokousten esityslistat ja pöytäkirjat kukin dokumentin kirjoittaja lähettää ryhmän jäsenille sähköpostiin. Dokumentointivastaava huolehtii dokumentit public_html-kansioon ja laittaa linkin kotisivulle.

9.4 Ryhmän sisäinen kommunikointi

1. Palaverit kaksi kertaa viikossa. Esityslista viimeistään edellisenä päivänä.
2. Sähköpostit luetaan arkipäivisin.
3. Ylättävissä tilanteissa ilmoitetaan sähköpostilla, saman päivän aikana puhelimitse.