

Verso-sovellusprojekti

Projektiraportti

Tero Hänninen

Juho Nieminen

Marko Peltola

Heikki Salo

Versio 0.1.0

Julkinen

5.5.2010

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2010		
Tilaja	__.__.2010		
Ohjaaja	__.__.2010		

Tietoa dokumentista

Tekijät:

• Tero Hänninen (TH)	tejohann@jyu.fi	0400-240468
• Juho Nieminen (JN)	juho.nieminen@jyu.fi	050-3831825
• Marko Peltola (MP)	marko.peltola@jyu.fi	041-4498622
• Heikki Salo (HS)	heikki.ao.salo@iki.fi	050-3397894

Dokumentin nimi: Verso-projekti, Projektiraportti

Sivumäärä: 30

Tiedosto: _doc.tex

Tiivistelmä: Verso-projekti toteutti Jyväskylän yliopiston tietotekniikan laitokselle lähdekoodin julkistamisjärjestelmän. Dokumentti kuvaa projektin taustaa, resursseja ja läpivientiä. Dokumentissa tarkastellaan myös suunniteltujen tavoitteiden, käytänteiden ja aikataulujen toteutumista.

Avainsanat: Sovellusprojekti, projektiraportti, tavoitteiden toteutuminen, työnjako, resurssit, aikataulu, projektiorganisaatio, versiohallinta, projektin läpivienti, ketterä ohjelmistokehitys

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	21.4.2010	Ensimmäinen luonnos dokumentista. Kirjoitettu taustaa ja tavoitteiden toteutumista kuvaavat luvut.	HS
0.0.2	23.4.2010	Kuvattu oppimistavoitteiden ja tulosten toteutumisista.	HS
0.0.3	29.4.2010	Korjattu termi.	HS
0.0.4	4.5.2010	Kuvattu organisaatio ja resurssit sekä tehtävien jakautumista.	HS
0.1.0	5.5.2010	Korjattu lähdeviitteet.	HS

Tietoa projektista

Verso-projekti toteutti Jyväskylän yliopiston tietotekniikan laitokselle lähdekoodien julkistamisjärjestelmän.

Tekijät:

- | | | |
|----------------------|-----------------------|-------------|
| • Tero Hänninen (TH) | tejohann@jyu.fi | 0400-240468 |
| • Juho Nieminen (JN) | juho.nieminen@jyu.fi | 050-3831825 |
| • Marko Peltola (MP) | marko.peltola@jyu.fi | 041-4498622 |
| • Heikki Salo (HS) | heikki.ao.salo@iki.fi | 050-3397894 |

Tilaaaja:

- | | | |
|--------------------|---------------------------|-------------|
| • Paavo Nieminen | paavo.j.nieminen@jyu.fi | 040-5768507 |
| • Tapani Tarvainen | tt@it.jyu.fi | 050-3130446 |
| • Ville Tirronen | ville.e.t.tirronen@jyu.fi | 014-2604987 |
| • Tero Tuovinen | Tero.Tuovinen@jyu.fi | 050-4413685 |

Ohjaajat:

- | | | |
|--------------------------|---------------------|-------------|
| • Jukka-Pekka Santanen | santanen@mit.jyu.fi | 014-2602756 |
| • Antti-Juhani Kaijanaho | antkaij@jyu.fi | 014-2602766 |

Yhteystiedot:

- | | |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| • Sähköpostilistat | verso@korppi.jyu.fi
yousource-users.group@korppi.jyu.fi |
| • Sähköpostiarkistot | https://korppi.jyu.fi/kotka/servlet/list-archive/verso/
https://korppi.jyu.fi/kotka/servlet/list-archive/yousource-users.group/ |
| • Työhuone | AgC 222.2, puh. 014-2604963. |

Sisältö

1	Johdanto	1
2	Termit	2
3	Taustaa	3
4	Tavoitteiden toteutuminen	4
4.1	Tavoitteet sovelluksen osalta	4
4.2	Jatkokehitys	5
4.3	Oppimistavoitteet	5
5	Tulosten toteutuminen	7
5.1	YouSource-sovellukseen liittyvät tulokset	7
5.2	Oheisdokumentit	8
5.3	Tulosten koostaminen	8
6	Organisaatio ja resurssit	9
6.1	Projektioorganisaatio	9
6.2	Projektin tilat, laitteet ja ohjelmistot	9
6.3	Ohjelmointi- ja dokumentointityökalut	10
6.4	Luennot ja perehdytykset	10
7	Tehtävien jakautuminen	12
7.1	Vastuualueet dokumentoinnin osalta	12
7.2	Vastuualueet ohjelmoinnin osalta	12
7.3	Työtunnit ja tehtävänjako	12
8	Käytänteet	13
8.1	Tiedotus	13
8.2	Palaverit	13
8.3	Lähdekoodin käytänteet	14
8.4	Versiohallinnan käytänteet	14
8.5	Dokumentoinnin käytänteet	15
8.6	Julkistettujen dokumenttitiedostojen versionumerointi	15
8.7	Esikäyttäjien hyödyntäminen	15

9	Trac-käytännöt	16
9.1	Luku- ja muokkausoikeudet	16
9.2	Traciin kirjattavat tehtävät	16
9.3	Kirjoitusasusta ja otsikoista	16
9.4	Sähköposti ja tiketit	17
9.5	Tikettityypit	17
9.6	Prioriteetit	18
9.7	Tarkistuspisteet ja versiot	18
10	Tehtävät, työmäärät ja vastualueet	19
10.1	Vastualueet dokumenttien osalta	19
10.2	Tehtävien työmäärät ja työnjako	19
11	Prosessimalli ja aikataulu	21
11.1	Prosessimalli	21
11.2	Sovelluksen ominaisuuksien hyväksyminen	21
11.3	Jatkuva integraatiotestaus	21
11.4	Aikataulu	22
12	Riskit ja niiden seuranta	24
12.1	Riskien todennäköisyydet ja haitat	24
12.2	Muutostarve kehityskoneissa	24
12.3	Tavoitteiden rajaus	25
12.4	Valitun alustan ongelmat	25
12.5	Poissaolot	25
12.6	Testipalvelimen ongelmat	25
12.7	Motivaation puute	26
13	Kokemukset ja oppiminen	27
14	Yhteenveto	28
15	Lähteet	29

1 Johdanto

Verso-projekti oli Jyväskylän yliopiston tietotekniikan laitoksella keväällä 2010 toteutettu sovellusprojekti. Projekti määritteli, suunnitteli ja toteutti Jyväskylän yliopiston tietotekniikan laitokselle prototyypin lähdekoodien julkistamisjärjestelmää.

Yliopistolla ei ollut käytössä yhtenäistä järjestelmää lähdekoodien jakamista tai julkistamista varten. Jokainen lähdekoodia tuottava tutkimusryhmä joutui kehittämään omat käytänteensä lähdekoodin kanssa toimimiseen.

Lähdekoodien julkistamista varten Verso-projekti kehitti WWW-sovelluksen, jolla käyttäjät voivat luoda tietovarastoja ja julkistaa lähdekoodeja. Tietovarastoja voi ylläpitää käyttäen hajautetun versiohallinnan työkaluja.

Toteutetun järjestelmän vaatimuksia on ylläpidetty Trac-järjestelmässä [11]. Sovelluksen rakenne ja toiminta kuvataan dokumentissa [14]. Projektisuunnitelma [13] määritteli projektin käytänteet, resurssit ja aikataulun. Tässä dokumentissa arvioidaan projektisuunnitelman toteutumista.

Luvussa 2 kuvataan projektiin liittyviä termejä. Luvussa 3 tarkastellaan projektin taustaa. Luku 4 tarkastelee projektin tavoitteiden toteutumista. Luvussa 5 tarkastellaan projektin tuloksia. Luvussa 6 esitellään projektiorganisaatio ja muita projektin käytössä olevia resursseja. Luvussa 7 tarkastellaan tehtävien ja työmäärän jakautumista projektin jäsenten kesken. Luvussa 11 kuvataan Verso-projektissa käytetty prosessimalli. Luku 8 käsittelee projektin toteutuneita käytänteitä ja luku 12 riskien toteutumista. Luvussa 13 ryhmän jäsenet kuvaavat kokemuksiaan projektista.

2 Termit

Dokumentissa esiintyvät aihealueen termit ovat seuraavat:

Git	on hajautetun versiohallinnan ohjelmisto.
Gitorious	on WWW-sovellus Git-tietovarastojen selaamiseen ja hallintaan.
Hajautettu versiohallinta	tarkoittaa ilman keskustietovarastoa tehtävää versiohallintaa.
Julkistaminen	tarkoittaa materiaalien asettamista julkisesti saataville.
Lokaali tietovarasto	on paikallinen, vain käyttäjän tietokoneella sijaitseva tietovarasto.
Metatiedostot	ovat varsinaista sisältöä kuvailevaa tiedostoihin tallennettua lisätietoa.
Palveluprosessi	tietyin palvelun tarjoava, taustalla suoritettava ohjelma (engl. <i>daemon</i>).
Projekti	on YouSource-järjestelmässä käyttäjän luoma sisältökokonaisuus, jolla voi olla monia tietovarastoja.
Tietovarasto	on versioitu lähdekoodin tallennuspaikka (engl. <i>repository</i>).

3 Taustaa

Jyväskylän yliopiston alaisuudessa toimii monenlaisia tutkimusryhmiä, joissa tuotetaan lähdekoodia. Yliopistolla ei ole yhtenäisiä toimintatapoja tai järjestelmiä lähdekoodin säilyttämiseen, mistä seuraa monenlaisia ongelmia.

Hajanaisesti tallennettujen lähdekoodien takia tieto lähdekoodista ei leviä välttämättä edes kehityksen aikana eri ryhmien välillä, mikä johtaa mahdollisesti päällekkäisen työn tekemiseen. Työntekijän lähtiessä talosta hänen yksin tuottamansa lähdekoodi käytännössä menetetään, kuten tilaajan alustavassa aihekuvauksessa [6] kerrotaan.

Osa tutkimusryhmistä ei käytä ollenkaan versiohallintaa, ja osa vain paikallisesti. Ilman yhteisesti saatavissa olevaa lähdekoodin tallennuspaikkaa itse lähdekoodin jakaminen eri tutkijoiden kesken vaatii jatkuvaa soveltamista, ja kätevin ratkaisu voi lopulta olla lähdekoodin version siirtäminen muistitikulla tai sähköpostilla [3].

Kevään 2010 Verso-sovellusprojektin tavoitteena oli kehittää kehityskelpoinen prototyyppi lähdekoodien julkistamisjärjestelmästä, jolla lähdekoodeja voisi julkistaa.

4 Tavoitteiden toteutuminen

Verso-projekti tutki projektin alussa erilaisia alustavaihtoehtoja [12] ja valitsi lähtökohdaksi sovelluksen kehittämiseksi olemassa Gitorious-sovelluksen. Projektin alussa välttämättömiksi määritellyt vaatimukset ja lähes kaikki tärkeät toteutettiin. Myös monia pienemmälle prioriteetille määriteltyjä vaatimuksia toteutettiin.

4.1 Tavoitteet sovelluksen osalta

Tilajana toimivan Jyväskylän yliopiston tietotekniikan laitoksen tavoitteena oli kehittää lähdekoodien julkistamisjärjestelmä, jonka tulisi mahdollistaa koodin julkistaminen maailmalle, mutta toisaalta myös levittää tietoisuutta kehitetyistä ohjelmista oman yliopiston sisällä. Kehitettävän järjestelmän tavoitteena oli myös mahdollistaa yksityisen tilan luominen, jotta järjestelmää voi käyttää kehittämiseen ilman tarvetta julkaista kaikkea. Projektin ja sovelluksen tavoitteena oli myös yleisesti kannustaa aloittamaan versiohallinnan käyttöä.

Tärkeimmät tilaajan tavoitteet kohdistuivat käytön helppoon aloittamiseen ja yksityisen tilan mahdollistamiseen kehittämistä varten. Projektin kuluessa järjestelmälle syntyi paljon pienemmälle prioriteetille määriteltyjä vaatimuksia, kuten esimerkiksi tiedoston muokkaaminen WWW-selaimella sekä lähdekooditiedostosta keskustelemisen mahdollistaminen.

Suurin osa tärkeimmistä vaatimuksista oli selvillä projektin alusta asti. Esimerkiksi alustaksi valittu Gitorious kuitenkin aiheutti projektin edetessä pakolliseksi määritellyjä muutostarpeita. Yksi tällainen muutos oli tietovarastojen selaaminen, koska Gitorious mahdollisti selaamisen vain projekteille, jonka alaisuudessa tietovarastot Gitoriousissa ovat.

Käytön aloittamisen helpottamiseksi muun muassa tietovaraston luontia Gitorious-sovelluksessa helpotettiin muokkaamalla olemassaolevia toimintoja. Lisäksi sovellus kytkettiin 19.4.2010 käyttämään kirjautumisessa Jyväskylän yliopiston Korppi-opintotietojärjestelmän käyttäjätunnuksia, mikä poisti kokonaan käytön aloituksessa vaaditun erillisen rekisteröitymisen.

Käyttöliittymän osalta Gitoriousiin ei tehty merkittäviä muutoksia, vaan vain vaatimukseen liittyvät muutokset ja lisäykset sekä korjauksia projektin aikana ilmennei-

siin parannusehdotuksiin sovelluksen käytettävyydessä. Sovellukselle logon persoonallisen ja persoonallisen ulkoasun kehittivät graafiset suunnittelijat Auri Kaihlavirta ja Kati Valpe, jotka määrittivät sivustolla käytettävät värit sekä valmistivat sovellukselle logokuvan.

Toteutettu järjestelmä kattaa tärkeimmät siihen kohdistuneet vaatimukset. Projektin tuloksena on kehityskelpoinen järjestelmä, jota voi jo sellaisenaan käyttää lähdekoodien julkistamiseen ja työkaluna ohjelmoinnin aikana. Sovelluksen toteuttaminen kuvataan tarkemmin sovellusraportissa [14].

4.2 Jatkokehitys

Projektissa tuotettava YouSource-sovellus oli luonteeltaan prototyyppi, jonka kehittämiseksi kerättiin palautetta myös projektiorganisaation ulkopuolisilta esikäyttäjiltä. Projektin kuluessa oli selvää, ettei kaikkia ideoituja toiminnallisuuksia tulla toteuttamaan Verso-projektin aikana.

Verso-projekti toteutti YouSource-järjestelmään kaikki pakollisiksi määritellyt ja lähes kaikki tärkeiksi määritellyt ominaisuudet. Suurin osa poisjääneistä ominaisuuksista karsittiin pieneksi määritellyn prioriteetin takia, mutta esimerkiksi Kerberos-autentikoinnin toteuttaminen nähtiin liian suuritöiseksi tehtäväksi projektin loppuun.

Toteutetut ja toteuttamattomat ominaisuudet sekä ominaisuuksiksi hyväksymättömät ideat on kirjattu Trac-järjestelmään [11].

4.3 Oppimistavoitteet

Sovellusprojekti on tietotekniikan syventävien opintojen kurssi, jossa neljästä opiskelijasta muodostettu ryhmä toteuttaa tilaajalle ohjelmiston tietotekniikan laitoksen ohjauksessa. Kurssissa tutustutaan projektiluontoiseen toimintatapaan, ja sen tärkeimpiin tavoitteisiin kuuluu antaa ryhmän jäsenille kokemusta ryhmätyöstä, projektin läpiviennistä sekä erilaisista käytänteistä kirjallisessa ja suullisessa viestinnässä. Sovellusprojektin ohessa ryhmän jäsenet suorittavat oheiskurssin, jossa he saavat yleistä koulutusta ohjelmistokehitykseen liittyvistä aiheista.

Oheiskurssin opetustapahtumissa Verso-ryhmä sai koulutusta projekti- ja ryhmätyöskentelystä, projektin johtamisesta ja hallinnasta, tekijänoikeuksista, sopimuksista ja käytettävyydestä. Oheiskurssiin kuului myös kaksi väliesittelyä, joissa projektiryhmä harjoitteli projektin esittämistä loppuesittelyä varten.

Verso-projektin jäsenet saivat Gitorious-sovellusta jatkokehittäessään kokemusta Ruby-ohjelmointikielestä, Ruby on Rails -ohjelmistokehyksestä sekä WWW-sovelluksen ylläpitämisestä projektin aikana. Tietokantana toimi MySQL 5.0 -tietokannan hallintajärjestelmä, ja projektin aikana ryhmäläiset oppivat muun muassa SQL-kieltä.

Projektiryhmä oppi projektin aikana käyttämään hajautettua versiohallintaa käyttämällä Git-versioohjelmistoa sekä sovelluksen kehittämiseen että projektin dokumenttien hallintaan. Ryhmä oppi myös avoimen lähdekoodin ohjelmistoprojekteihin liittyviä versiohallintakäytänteitä. Lisäksi oman sovelluksen käyttämiselle ja esikäyttöön tarjoamiselle projektin aikana edellytys oli, että ryhmä oppi kehittämään keskeneräisiä ominaisuuksia omissa haaroissaan versiohallinnassa, jotta päähaara on jatkuvasti otettavissa käyttöön.

Projektin vaatimuksia, kehitysideoita sekä löydettyjä virheitä ylläpidettiin projektin ajan Trac-projektinhallintasovelluksessa. Projektin aikana ryhmä oppi perustaidot yleisen projektinhallintatyökalun käytöstä.

Projektin aikana ryhmän jäsenet oppivat käyttämään Linux-työpöytäympäristöä sekä esikäyttöä varten olevaa Linux-virtuaalipalvelinympäristöä. Projektiryhmä oppi ylläpitämään virtuaalipalvelinta sovellukseen liittyvien palveluiden osalta itse. Ryhmä oppi myös toimimaan tietotekniikan laitoksen ATK-käytänteiden kanssa halutessaan muutoksia työpöytäkoneille esimerkiksi asennettavien sovellusten osalta.

5 Tulosten toteutuminen

Projektin aikana ryhmä toteutti Gitorious-sovellusta pohjana käyttäen prototyypin lähdekoodien julkistamisjärjestelmästä, joka on nimeltään YouSource. Ryhmän jäsenet laativat myös dokumentaation muun muassa asennuksesta ja käytettävyydestä tuloksista.

5.1 YouSource-sovellukseen liittyvät tulokset

Projektiryhmä toteutti seuraavat julkiset tulokset:

Sovellustietovarasto	sisältää YouSource-prototyypin lähdekoodin sekä Git-versiohallinnan.
Projektitietovarasto	sisältää projektiin liittyvät dokumentit ja tiedostot sekä niiden Git-versiohallinnan.

Projektiryhmä toteutti lisäksi seuraavat julkiset suunnitelmat ja raportit:

Asennusohje	sisältää sovelluksen asennusohjeet.
Ajankäyttöraportti	sisältää ryhmän jäsenten kirjaamat työtunnit.
Esittelymateriaali	sisältävät väli- ja loppuesittelyn materiaalit.
Esityslistat	sisältävät projektin palavereissa käsiteltävät asiat.
Käytettävyyshuomiot	sisältävät kahden käytettävyydestä tuloksen huomiot.
Projektiraportti	kuvaava projektin läpiviennin ja asetettujen tavoitteiden saavuttamista.
Projektsuunnitelma	kuvaava projektin tavoitteita, tehtäviä, aikataulua, yleisiä käytäntöjä ja riskien hallintaa.
Pöytäkirjat	sisältää projektipalavereissa käsitellyt, asiat, päätökset ja tehtävät.
Sovellusraportti	kuvaava toteutetun sovelluksen osat ja toiminnot sekä jatkokkehitysideat.

Edellä mainittujen dokumenttien lisäksi ryhmä laatii seuraavat tulokset:

Itsearviointit	sisältävät ryhmän jäsenten arviointit omasta panoksesta, onnistumisesta ja oppimisesta.
Palaverien dokumentit	sisältävät palaverien esityslistat ja pöytäkirjat.
Sähköpostiarkistot	sisältävät kaikki projektin sähköpostilistalla käydyt keskustelut.
Lähdekoodi	sisältää tuotetun lähdekoodin kommentteineen.

5.2 Oheisdokumentit

Projektille perustettiin kaksi sähköpostilistaa, joiden viestit arkistoitiin ja projektin lopuksi arkiston viestit liitettiin projektikansioon. Toinen sähköpostilista oli tarkoitettu koko projektiorganisaation väliseen viestintään ja toinen esikäyttäjien tiedottamista ja viestintää varten.

FIXME: tulevaisuudessa Projektin aikana ryhmä piti kaksi väliesittelyä ja loppuesittelyn projektistaan ja kehitetystä järjestelmästä. Näiden esitysten **esitysmateriaalit** ja **esityksien arviointit** liitettiin projektikansioon.

FIXME: tulevaisuudessa Projektin lopuksi kukin ryhmän jäsen kirjoitti henkilökohtaisen arvion ryhmän työstä sekä omasta panoksestaan projektissa. Ryhmän jäsenen **itsearvio** liitettiin projektikansioon ja CD:lle.

5.3 Tulosten koostaminen

FIXME: tulevaisuudessa Sovellukseen liittyvät tulokset koostettiin CD-levylle, joka toimitettiin tilaajalle.

6 Organisaatio ja resurssit

Luvussa esitellään projektiorganisaatioon kuuluvat henkilöt, ryhmän käytössä olevat tilat, laitteet ja ohjelmistot sekä jäsenille järjestettävät perehdytykset.

6.1 Projektiorganisaatio

Verso-projektiryhmään kuului neljä Jyväskylän yliopiston tietotekniikan opiskelijaa. Tero Hänninen on 3. vuoden ja Marko Peltola 7. vuoden tietotekniikan opiskelijaa. Juho Nieminen ja Heikki Salo ovat 4. vuoden tietotekniikan opiskelijoita.

Tilaaajana toimineen tietotekniikan laitoksen edustajina toimivat Paavo Nieminen, Tapani Tarvainen, Ville Tirronen ja Tero Tuovinen, joista Ville Tirrosella oli määräysvalta tilaajaa koskevista asioista. Ryhmän vastaavana ohjaajana toimi Jukka-Pekka Santanen ja teknisenä ohjaajana toimi Antti-Juhani Kaijanaho.

Projektiryhmän työkoneiden ohjelmistojen ja ylläpidon hoiti Jyväskylän yliopiston ATK-lähituki. Projektiryhmän käytössä olleen virtuaalipalvelimen ylläpidosta vastasi Jyväskylän yliopiston tietohallintokeskuksen sovelluspalvelut, pääasiassa Harri Tuomi.

Lisäksi sovellusprojektiurssin aikana projektiryhmää auttoi kirjoitusviestinnän opettaja Leena Peltomaa sekä puheviestinnän opettaja Minna Koljonen.

6.2 Projektin tilat, laitteet ja ohjelmistot

Projektiryhmän käytössä ollut huone AgC222.2 sijaitsi Agoran C-siivessä toisessa kerroksessa sovellusprojektien tiloissa. Huoneessa oli ryhmän käytössä neljä tietokonetta, joista kolmessa oli käyttöjärjestelmänä Linux Fedora Core 12 ja yhdessä Windows XP SP3 ryhmän jäsenten työtuntien kirjaamista varten.

Huoneen Linux-tietokoneisiin oli projektia varten asennettu MySQL 5.0 -tietokantaohjelmisto, Apache-WWW-palvelin, Ruby-ohjelmatulkki sekä Ruby on Rails -sovelluskirjasto. Projektiryhmällä on lisäksi verkon kautta käyttöoikeus myös erilliseen testipalvelimeen versotest, jonka käyttöjärjestelmänä oli Red Hat Enterprise Linux 5.0.

6.3 Ohjelmointi- ja dokumentointityökalut

Ryhmän käytössä oleviin Fedora Core -tietokoneisiin oli ATK-lähituen toimesta asennettu palvelinohjelmistoina Apache, Git, MySQL sekä Ruby ja Ruby on Rails -ohjelmakirjastot, jotka mahdollistivat Gitorious-sovelluksen kehittämisen. Windows XP -tietokoneeseen oli ATK-tuelta pyydetty TortoiseGit- ohjelmisto versiohallintaa ja Microsoft Office 2003 ajankäytön merkitsemistä varten.

Työmäärien kirjaamiseen käytettiin Petri Heinosen kehittämää Excel-pohjaista ajankäytönseurantasovellusta.

Projektiorganisaation käytössä oli myös Trac-sovellus, jota käytettiin sovelluksen vaatimusten ja projektin tehtävien hallintaan. Tracin käytössä noudatettuja käytäntöitä on tarkasteltu luvussa 9.

6.4 Luennot ja perehdytykset

Ohjelmointikielenä oli Ruby ja alustana Ruby on Rails, jotka eivät olleet ryhmän jäsenille ennestään tuttuja. Ryhmä tutki kirjallisuutta verkosta ja tilasi Santasen kautta käyttöönsä Agile Web Development with Rails (Ruby, Thomas, Heinemeier Hansson) -kirjan. Lisäksi Tarvainen lainasi projektin käyttöön Ruby on Rails Bible (Fisher) -kirjan.

Projektissa vaadittiin hyvää tuntemusta Git-versiohallinnasta, joka ei ollut ryhmäläisille entuudestaan tuttu. Tekninen ohjaaja Antti-Juhani Kaijanaho toimi asiantuntijana versiohallintaan liittyvissä kysymyksissä ja järjesti ryhmälle ongelmalähtöistä perehdytystä.

Projektissa toteutettun prototyypin ohjelmointiin tarvittiin tuntemusta Rubystä, Ruby on Railsista sekä sovelluksen pohjana toimivan Gitoriousin toiminnasta. Projektiryhmä perehtyi Rubyyn ja Ruby on Railsiin omatoimisesti tutkimalla Internetissä saatavilla olevia ilmaisia oppaita. Gitoriousin asentamiseen ja toimintaan projektiryhmä tutustui lähdekoodeihin perehtymällä sekä kysymällä apua Gitoriousin kehitysyhteisöltä sähköpostilla ja IRC-kanavalla.

Jukka-Pekka Santanen piti luennon projektin hallinnasta ja läpiviennistä. Luennolla käsiteltiin myös ryhmän jäsenten keskinäistä viestintää.

Oheiskurssilla käytäviin asioihin kuuluvat

- kokous- ja neuvottelukäytänteet
- esittely ja esiintyminen
- kirjoitusviestintä
- projektin johtaminen ja hallinta
- käytettävyyden luennot ja ryhmätyöt
- tekijänoikeus ja sopimukset
- versiohallinta sekä
- kaksi väliesittelyä.

7 Tehtävien jakautuminen

Luvussa tarkastellaan toteutunutta projektin vastuualueiden ja tehtävien jakautumista ja verrataan sitä suunniteltuun. Dokumentoinnin ja ohjelmoinnin vastuualueisiin ei tullut muutoksia projektin alussa suunniteltuun.

7.1 Vastuualueet dokumentoinnin osalta

Dokumenttien vastuualueita ei vaatimusmäärittelyä ja projektidokumentointia lukuunottamatta suunniteltu projektin alussa. Projektipäällikkönä Salo aloitti projektisuunnitelman kirjoittamisen ja Nieminen vaatimusmäärittelyn kirjoittamisen. Projektin neljännessä palaverissa [5] päätettiin, että erillisen dokumentin sijaan vaatimukset kirjataan Trac-järjestelmään. Tracissa vaatimuksista tehtyjä tikettejä on ylläpitänyt koko projektiorganisaatio.

Projektisuunnitelma jäädettiin 12.4.2010, koska suunnitelman tekemisen ei enää katsottu edistävän projektin läpiviemistä. Projektisuunnitelmassa oli valmiina Trac ja muiden käytänteiden kuvaus, projektin tavoitteet tuloksien ja oppimistavoitteiden osalta sekä kuvaus projektissa käytettävästä prosessimallista.

Projektisuunnitelmasta puuttui projektissa tuotettavan sovelluksen kuvaaminen, ja siinä oli vasta alustavasti kuvattu riskejä sekä tuntien jakautumista.

7.2 Vastuualueet ohjelmoinnin osalta

Kesken.

7.3 Työtunnit ja tehtävänjako

Kesken.

8 Käytänteet

Luvussa kuvataan projektissa käytettävät käytänteet.

8.1 Tiedotus

Projektiryhmän tiedotuksesta vastaa käsillä olevasta asiasta vastaava henkilö tai projektipäällikkö.

Projektiryhmän jäsenten, tilaajan ja ohjaajien välinen tiedotus hoidetaan sähköpostilla käyttämällä sähköpostilistaa `verso@korppi.jyu.fi`. Sähköpostilista on pääasiallinen yhteydenpitoväline, mutta kiireellisistä asioista voidaan myös soittaa.

Projektiryhmän jäsenten välinen tiedotus hoidetaan sähköpostilla käyttämällä ryhmän jäsenten omaa sähköpostilistaa ja IRC-kanavaa.

8.2 Palaverit

Projektipalavereja pidetään 1–2 viikon välein. Projektipalavereihin osallistuvat ryhmän jäsenet, ohjaajat ja tilaajan edustajat. Ensimmäisen palaverin [3] mukaisesti palaveri on päätösvaltainen, kun projektiryhmästä ja tilaajista on paikalla vähintään yksi edustaja sekä ohjaajista paikalla on vastaava ohjaaja Santanen. Tilaaja voi nimittää tarvittaessa myös ulkopuolisen edustajan. Palaveri on laillinen, kun palaverikutsu ja esityslista on lähetetty vähintään vuorokautta ennen palaveria.

Palaverin alussa projektiryhmä ehdottaa keskuudestaan puheenjohtajaa ja sihteeriä. Projektiryhmä kierrättää puheenjohtajan ja sihteerin rooleja siten, että kukin toimisi projektin aikana vähintään kaksi kertaa puheenjohtajana ja sihteerinä. Palaverissa käydään läpi kokouskutsussa ollut esityslista, johon kokoukseen osallistujat voivat esittää palaverissa tarvittaessa lisäyksiä esityslistaa hyväksyttäessä.

Palaverissa edellisen palaverin puheenjohtaja käy läpi etukäteen alustavasti hyväksytyn edellisen palaverin pöytäkirjan. Palaverissa edellisen palaverin pöytäkirja voidaan hyväksyä tai se voidaan hyväksyä muutoksin. Palaverin jälkeen sihteeri laatii palaverista pöytäkirjan, josta hän pyytää kommentit ensin ryhmäläisiltä, minkä jälkeen sihteeri lähettää pöytäkirjan kirjoitusviestinnän opettajalle ja projektiorganisaatiolle.

8.3 Lähdekoodin käytänteet

Ohjelmoidessa noudatetaan ensimmäisen palaverin [3] mukaisesti Ruby-kielen [9] Ruby on Rails -ohjelmistokehyksen ohjelmointityyliä sekä Gitorious- ohjelmiston ohjelmointikäytänteitä [9]. Ohjelmakomponentit pyritään pitämään mahdollisimman yhteensopivana Gitorious-pääprojektin kanssa. Tavoitteena on tuottaa mahdollisimman monesta Gitorious-ohjelmistoon projektia varten tehdystä muokkauksesta muokauspyyntö (engl. *merge request*) Gitorious-pääprojektiin.

```
class Repository < ActiveRecord::Base
  include ActiveMessaging::MessageSender
  ...

  def can_view?(a_user)
    if REPO_VIEWABLE_EVERYONE
      return true
    else
      return viewer?(a_user)
    end
  end
end
...
end
```

Kuva 8.1: Lähdekoodiesimerkki.

8.4 Versiohallinnan käytänteet

Sovellukseen liittyvä lähdekoodi tallennetaan Git-versiohallintaan. Sovellusta kehitetään omaan tietovarastoonsa ja projektin dokumentteja, kuten muistioita, palaverien pöytäkirjoja, väliesittelymateriaaleja omaan tietovarastoonsa. Väliaikaishaaroja (engl. *topic branch*) käytetään sekä projektitietovarastossa että etenkin sovelluksen tietovarastossa. Väliaikaishaara sisältää esimerkiksi tiettyyn ominaisuuteen tai asiakokonaisuuteen (kuten tiettyyn pöytäkirjaan) liittyvät päivitykset.

Sovelluksen osalta haarojen käyttäminen on tärkeää, koska sovelluksen päähaaraa on yhtä aikaa julkaisuhaara esikäyttöä varten. Epävakaa ohjelmakoodi on pidettävä omissa haaroissaan sen koekäyttöön valmistumiseen asti. Kun sovellukseen liittyvän ominaisuus tulee valmiiksi, sen haara yhdistetään päähaaraan (engl. *master branch*) ja ominaisuuden historian sisältävä haara jätetään versiohallintaan. Ominaisuuksien tarjoaminen Gitorious-sovellukselle on helpompaa, kun projektissa tuotettavien ominaisuuksien haarat ovat lähdekoodeja tarjotessa tallessa.

8.5 Dokumentoinnin käytänteet

Pääsääntönä on, että sovellukseen liittyvät dokumentit kirjoitetaan englanniksi ja projektiin liittyvät dokumentit suomeksi. Englanninkielisissä dokumenteissa käytetään amerikanenglantia.

8.6 Julkistettujen dokumenttiedostojen versionumerointi

Dokumenteissa käytetään kolmitasoista versionumerointia ensimmäisen palaverin [3] mukaisesti. Ensimmäinen versio on 0.1. Ryhmän sisäiseen käyttöön tarkoitettuja versionumerot kasvatetaan 0.0.1:llä, ohjaajalle ja tilaajalle versionumerot kasvatetaan 0.1:llä. Dokumentin ensimmäinen hyväksytty versio on 1.0.0.

Sovelluksessa käytetään käyttöön innostavaa versionumerointia, joten sovelluksen ensimmäinen projektiryhmän ulkopuolisille testaajille julkinen versio on 1.0.

8.7 Esikäyttäjien hyödyntäminen

Projektin aikana sovellusta päästetään käyttämään kymmenkunta esikäyttäjää, joilta pyydetään palautetta sovelluksen ominaisuuksista ja käytettävyydestä. Esikäyttäjien käyttämisestä on laadittu erillinen dokumentti "Suunnitelma esikäyttäjien hyödyntämisestä" [2].

9 Trac-käytänteet

Projektissa käytetään vaatimusmäärittelyn hallintaan Trac-ohjelmistoa. Koska erillistä vikojenseurantaohjelmistoa (engl. *bugtracker*) ei käytetä, projektiryhmän ja tilaajan välisen tehtävähallinnan lisäksi Traciin kirjataan myös käyttäjien kautta tulevat vikailmoitukset ja toiveet.

9.1 Luku- ja muokkausoikeudet

Verso-projektin Tracin selaamiseen on järjestetty avoin pääsy, mutta muokkaamista varten käyttäjän pitää kirjautua sisään. Muokkausoikeus on kaikilla luvussa 6.1 mainituilla projektiorganisaation jäsenillä.

Avoin pääsy antaa mahdollisuuden selata tehtäviä myös esikäyttäjille ja muille projektista kiinnostuneille. Projektiorganisaation ulkopuolisen tahon kanssa yhteyttä pitäessä voidaan siis viitata suoraan Tracin tiketteihin.

9.2 Traciin kirjattavat tehtävät

Tracia ei käytetä koko Verso-projektin tehtävien hallintaan, vaan ainoastaan siinä tuotettavan YouSource-sovelluksen tehtävien osalta. Pääsääntönä on, että mikäli tehtävä tai idea liittyy sovellukseen, sen voi kirjata Traciin.

9.3 Kirjoitusasusta ja otsikoista

Kuten muissakin sovellukseen liittyvissä dokumenteissa, Tracissa käytetään kirjauskielenä englantia.

Tiketin tiivistelmäksi (engl. *summary*) tiketin lisääjä kirjaa mahdollisimman yksiselitteisen kuvauksen, koska tiivistelmä toimii tiketin otsikkona. Selkeän otsikon avulla tiketti löytyy tikettilistoista ja sähköpostikansioista helpommin.

Tiketin kuvausta (engl. *summary*) käytetään selvittämään tarkemmin tiketissä käsiteltävä asia. Kuvaukseen voi kirjata esimerkiksi tikettiin liittyviä kehitysratkaisuja tai sovellukseen lisättäviä tekstejä.

Tiketin luominen on samalla pienimuotoinen keskustelunavaus tiketissä kuvattuun tehtävään liittyen. Tiketin kommentteissa (engl. *comment*) voi kysyä tarkennuksia tehtävään liittyen, minkä perusteella sen lisääjä voi esimerkiksi muokata tiketin kuvausta tarkemmaksi.

Tiketin otsikko ei ole muuttumaton. Mikäli Tracista löytyy tiketti, jonka kuvaus ei projektiorganisaation jäsenen mielestä ole tarpeeksi selkeä, hän voi vapaasti muokata siitä selkeämmän. Muokatessa asiasisältö ei kuitenkaan saa muuttua, ellei siitä ole erikseen sovittu.

9.4 Sähköposti ja tiketit

Trac-tiketteihin merkitään cc-kenttään eli muutosilmoitusten vastaanottajaksi projektiorganisaation sähköpostilista. Lisäksi, jos ilmoitus tulee projektiorganisaation ulkopuolelta (esimerkiksi esikäyttäjiltä) jakelulistaan lisätään ilmoittajan sähköpostiosoite.

9.5 Tikettityypit

Tikettityyppi	Omistaja	Kuvaus
Feature	Tilaaja	Sovellukseen hyväksytty ominaisuus.
Chore	Tilaaja	Projektiryhmällä teetetty muu tehtävä.
Wish	Kuka tahansa	Kehitysidea tai muu toive.
Bug	Kuka tahansa	Sovelluksesta löytynyt vika.
task	Projektiryhmä	Pieni kehitystehtävä.

Taulukko 9.1: Trac-tikettien tyypit.

Taulukossa 9.1 on kuvattu erilaiset Tracissa käytettävät tikettityypit. Taulukon termillä omistaja tarkoitetaan sitä tahoa, jonka kommunikointivälineeksi tikettityyppi on tarkoitettu.

Verso-projektin Traciin on järjestetty lukuoikeus kaikille, mutta tietojen muokkaamiseen ovat oikeutettuja vain projektiorganisaatioon kuuluvat käyttäjät. Kehitysideoita ja bugeja tulee kuitenkin mahdollisesti myös projektiorganisaation ulkopuolelta. Omistajalla "Kuka tahansa" tarkoitetaan kenen tahansa lähettämää huomiota,

jonka vastaanottaja projektiorganisaatiossa tekee siitä tiketin.

9.6 Prioriteetit

Prioriteetti	Kuvaus
Mandatory	Pakolliseksi nähtävä, toteutetaan ensimmäisten joukossa
Important	Tärkeä, ei pakollinen, toteutetaan ajan salliessa
Useful	Hyödyllinen, ei käytön kannalta tärkeä, toteutetaan ajan salliessa
Left out	Projektin osalta tarpeettomaksi nähty
–	Toistaiseksi priorisoimaton ominaisuus.

Taulukko 9.2: Trac-tikettien prioriteetit selityksineen.

Prioriteetit (engl. *priority*) ovat tarkoitettu käytettäväksi Feature-tyyppisten tiketien kanssa, muiden tikettityyppien kanssa niitä voi käyttää suuntaa antavana lisätietona. Prioriteetti ei suoraan liity ominaisuuden kehityksi tulemiseen Verso-projektissa, vaan niitä käytetään hyödyksi, kun iteraatioihin ollaan kiinnittimässä tehtäviä.

9.7 Tarkistuspisteet ja versiot

Tarkistuspiste (engl. *milestoneilla*) viitataan projektin kehitysprosessin vaiheisiin eli *iteraatioihin*. Iteraatioiden aikataulut on kuvattu luvussa 11.4.

Versioilla viitataan sovelluksen julkistettuihin kehitysversioihin. Esimerkiksi versio 1.0 on ensimmäinen julkinen esikäyttöversio.

10 Tehtävät, työmäärät ja vastualueet

Luvussa kuvataan projektin tehtävät ja niiden jakautuminen ryhmän kesken. Luvussa arvioidaan myös tehtävien työtuntimäärät.

10.1 Vastualueet dokumenttien osalta

Tulos	Kieli	Vastuuhenkilö	Tarkastettavaksi	Hyväksytty
Projektisuunnitelma	suomi	HS		
Vaatimusmäärittely	englanti	JN		
Projektiraportti	suomi			
Sovellusraportti	englanti			

Taulukko 10.1: Dokumenttien kielet ja vastualueet.

10.2 Tehtävien työmäärät ja työnjako

Vaiheet ja tehtävät		Tekijä				
Vaihe	Tehtävä	Juho	Marko	Tero	Heikki	Yhteensä
Esikäyttäjät	käytettävyysestaus	2:00	2:00	2:00		6:00
	haastattelu	1:30	1:30	1:30		4:30
	esikäyttötiedotus				8:00	8:00
	esikäyttöviestintä	6:00	6:00	6:00	14:00	32:00
	suunnittelu				10:00	10:00
Esikäyttäjät Total		9:30	9:30	9:30	32:00	60:30
Esitutkimus	tutustuminen	30:00	50:00	50:00	10:00	140:00
Esitutkimus Total		30:00	50:00	50:00	10:00	140:00
Oheiskurssi	esittelyt	15:00	15:00	15:00	15:00	60:00
	katselmoinnit	5:00	5:00	5:00	5:00	20:00
	koulutus	30:00	30:00	30:00	30:00	120:00
Oheiskurssi Total		50:00	50:00	50:00	50:00	200:00
Palaverit	raportointi	20:00	20:00	20:00	20:00	80:00
	seuranta ja hallinta	15:00	15:00	15:00	15:00	60:00
Palaverit Total		35:00	35:00	35:00	35:00	140:00
Projektin hallinta	muut tehtävät				5:00	5:00
	suunnittelu	25:00			100:00	125:00
	raportointi				60:00	60:00
	tiedotus	2:00	2:00	2:00	30:00	36:00
Projektin hallinta Total		27:00	2:00	2:00	195:00	226:00
Suunnittelu	suunnittelu	20:00	20:00	20:00	30:00	90:00
Suunnittelu Total		20:00	20:00	20:00	30:00	90:00
Toteutus	käyttöliittymä	40:00	40:00	40:00		120:00
	metatiedostot	20:00	20:00	70:00		110:00
	oikeudet ja näkyvyydet	50:00	80:00	30:00		160:00
	päivitystavat	30:00	30:00	40:00		100:00
	tukitehtävät	10:00	20:00	10:00	50:00	90:00
Toteutus Total		150:00	190:00	190:00	50:00	580:00
Viimeistely	sovellusraportti	50:00	5:00	5:00	5:00	65:00
	lähdekoodin siistiminen	10:00	10:00	10:00		30:00
	merge requestit	10:00	20:00	20:00		50:00
Viimeistely Total		70:00	35:00	35:00	5:00	145:00
Yhteensä		391:30	391:30	391:30	407:00	1581:30

Kuva 10.1: Jäsenten työmääräarvio

11 Prosessimalli ja aikataulu

Projekti toteutetaan kevätlukukaudella 2010. Sovelluksen on määrä olla valmis esiteltäväksi julkisesti toukokouun puolella välissä. Projektin uusien ominaisuuksien ohjelmoinnin tulee olla valmis neljännen iteraation päättyessä 23.4.2010.

11.1 Prosessimalli

Projekti viedään läpi käyttäen ketterää ohjelmistokehitystä. Ohjelmistokehitys jaetaan kahden viikon mittaisiin iteraatioihin, joihin kiinnitetään käytettävissä oleviin resursseihin nähden järkevä määrä tehtäviä. Kiinnittämättömien Trac-tikettien, palaverikeskustelujen ja muun yhteydenpidon perusteella projektiryhmä tekee tilaajalle ehdotuksen seuraavan iteraation sisällöstä ja sen tehtävät hyväksytään projektipalaverissa.

11.2 Sovelluksen ominaisuuksien hyväksyminen

Kun ominaisuus eli vaatimusmäärittelyssä Trac-tikettityyppi Feature valmistuu, se hyväksytetään tilaajalla. Hyväksyttäminen tehdään tavallisesti siten, että projektiryhmä järjestää ominaisuuden nähtäville ja ilmoittaa siitä projektin sähköpostilistalla. Jos tilaaja ei hyväksy ominaisuutta, projektiryhmä korjaa ominaisuuteen liittyvät puutteet.

Sovelluksen osalta projekti on hyväksytty viimeiseen 5. iteraatioon kiinnitettyjen Feature, Chore ja Bug -tyyppisten tehtävien tultua hyväksytyksi. Viimeiseen iteraatioon ei oteta Feature-tyyppisiä tehtäviä, ellei niiden kehitystä ole jo aloitettu.

11.3 Jatkuva integraatiotestaus

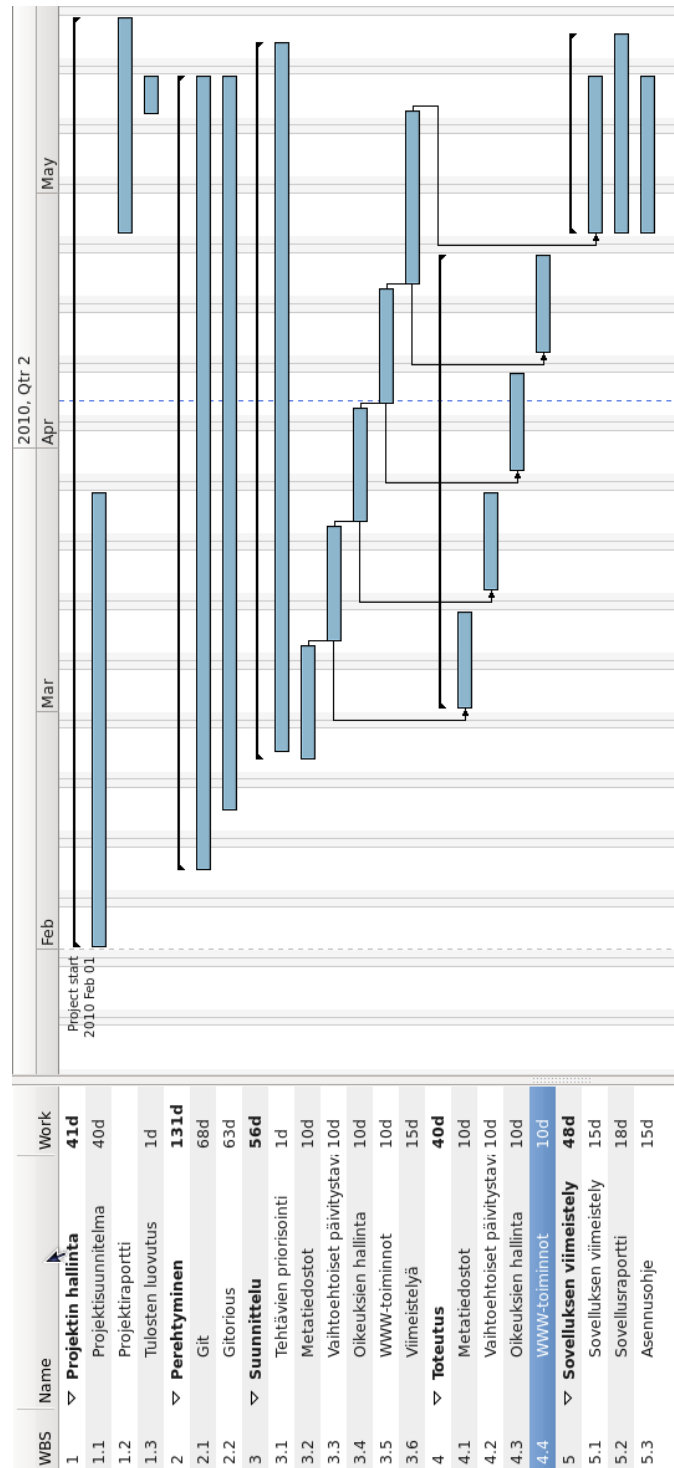
Sovellukseen tuotettavan lähdekoodin laatua testataan jatkuvasti. Kun uutta ominaisuutta kehitetään, sen lähdekoodi sijoitetaan omaan haaraan (engl. *topic branch*) sovelluksen lähdekoodin tietovarastossa, kuten luvussa Versiohallinnan käytänteet 8.4 kuvataan. Kehityksen alla olevaa ohjelmakoodia testataan pelkästään kehitystyö-

asemilla sen kehityksen ajan. Ominaisuuden ohjelmoinnin valmistuttua ja sen koodin vakauduttua se yhdistetään sovelluksen versiohallinnan päähaaraan ja esitellään tilaajalle virtuaalipalvelimella. Epävakaata ohjelmakoodia voidaan tarvittaessa esitellä tilaajalle myös kehitystyöasemilta.

Kun ominaisuus yhdistetään päähaaraan, se tulee projektiorganisaation ja esikäyttäjien käytössä olevaan testisovellukseen. Uusin vakaa ohjelmakoodi on siis projektin ajan jatkuvassa integraatiotestauksessa.

11.4 Aikataulu

Projektin tehtävien aikataulut on esitetty kuvan 11.1 Gantt-kaaviossa.



Kuva 11.1: Gantt-kaavio tehtävien aikataulutuksesta

12 Riskit ja niiden seuranta

Luvussa kuvataan projektissa tiedostetut riskit sekä kuvataan toimia niiden ennakoinniseksi ja ehkäisemiseksi.

12.1 Riskien todennäköisyydet ja haitat

Riskien todennäköisyydet ja niistä seuraavat haittavaikutukset on esitetty taulukossa 12.1. Todennäköisyyttä ja haittavaikutusta arvioidaan asteikolla pieni, keskinkertainen ja suuri.

Riski	Todennäköisyys	Haittavaikutus
Muutostarve kehityskoneissa	suuri	keskinkertainen
Tavoitteiden raja	keskinkertainen	suuri
Valitun alustan ongelmat	keskinkertainen	suuri
Poissaolot	keskinkertainen	pieni
Testipalvelimen ongelmat	keskinkertainen	keskinkertainen
Motivaation puute	pieni	keskinkertainen

Taulukko 12.1: Arvioidut riskit, niiden todennäköisyys ja haittavaikutus.

12.2 Muutostarve kehityskoneissa

Projektiryhmän käytössä oleviin kehityskoneisiin on asennettu ryhmän toistaiseksi tarvitsemat kehitystyökalut ja sovelluskirjastot. Projektin edetessä kehitetään kuitenkin toisistaan hyvin poikkeavia kokonaisuuksia, joihin liittyy mahdollisesti tarve uusille asennuksille tai kehitysympäristön asetusten muutoksille. Projektiryhmä ei itse pysty toteuttamaan tarvittavia muutoksia, joten projektiryhmä on riippuvainen yliopiston lähituen palvelusta. Riskinä on, että jos vastaan tullut ongelma on haastava ja vaatii selvittämistä yhdessä lähituen kanssa, sen ratkaisuun kuluu kauan. Riskiä hallitaan ennakoimalla muutostarpeita ja pyytämällä tarvittavat muutokset etukäteen.

12.3 Tavoitteiden raja

Tähän juttua siitä, että vaikka periaatteessa Gitoriousille näyttäisi mahdolliselta toteuttaa kaavaillut ominaisuudet, sen käytön estyminen tekee projektista todella suuritöisen. Lisäksi, vaikka pohjaksi saataisiin toimiva alusta, ohjelmointitekniikat ovat uusia, ja vaatimusten toteuttamiseen saattaa liittyä ennalta tuntemattomia rajoitteita.

12.4 Valitun alustan ongelmat

Alustaksi valittu Gitorious valittiin, koska sen käyttöönoton estämiseen ei ollut tiedossa syytä ja se vaikutti lupaavimmaksi tavoitteiden saavuttamisen kannalta. On mahdollista, että ilmaantuu syy, joka estää sen käytön toistaiseksi. Esimerkiksi Gitorious-projektissa vielä toteutumattoman tietoturvariskin ilmeneminen, joka estää sen käytön yhdessä yliopiston autentikointimenetelmien kanssa. Tämä riski koskee kaikkea toimimista ulkopuolisten ohjelmaosien kanssa ja tätä voi hallita vain seuraamalla Gitorious-kehittäjälistalta ilmoitettuja vikoja ja niiden ratkaisuja.

12.5 Poissaolot

Ryhmän jäsenille voi tulla suunniteltuja poissaoloja, kuten matkat, ja yllättäviä, kuten sairastuminen. Jos ryhmän jäsen jää yllättäen pitkäksi aikaa pois, muu ryhmä joutuu ratkaisemaan tehtävät ilman poisjääneen tietämystä. Tämän riski on pieni, koska suuri osa ohjelmoinnista tehdään pariohjelmointina ja tietämys leviää sitä kautta hyvin.

12.6 Testipalvelimen ongelmat

Testipalvelimen ongelmilla tarkoitetaan odottamattomia tilanteita, joissa esikäytössä oleva sovellus toimii tuntemattomasta syystä väärin.

Projektissa tuotettava sovellus sisältää paljon erilaisia komponentteja, joiden yhteistoiminnasta sovellus on riippuvainen. Jos sovelluksen jokin osa vikaantuu ja alkaa

toimia väärin, tilanteen selvittäminen on mahdollisesti haastavaa ja aikaavievää, mikä myöhästyttää kehitystyötä.

Riskin hallitsemiseksi projektiryhmä harjoittelee erillisellä testipalvelimella erilaisien vikatilanteiden ratkaisua. Mahdollisia vikatilanteita pyritään myös ennakoimaan ja niissä toimimiseen määritellään selkeät käytänteet.

12.7 Motivaation puute

Tähän juttua siitä, että jos motivaatio laskee eikä intoa sovelluksen tekemiseen ole, alkaa sovelluksen hommat kasaantua kohti toukokuuta, mistä ei seuraa mitään hyvää.

13 Kokemukset ja oppiminen

Tähän kokemuksia ja oppimista.

14 Yhteenveto

Verso-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle lähdekoodien julkistamisjärjestelmän prototyypin. Prototyyppi kehitetään Ruby-kielellä Ruby on Railsia käyttävän Gitorious-sovelluksen päälle.

Projektin aikana ryhmä saa kokemusta työskentelystä ohjelmistoprojektissa, useista eri työkaluista sekä projektin hallinnasta.

Projekti alkoi 3.2.2010 ja se päättyy 22.5.2010 mennessä. Projektin projektin läpiviennessä noudatetaan iteratiivista prosessimallia ja käytetään 5 iteraatiota.

15 Lähteet

- [1] Petri Heinonen, Ajankäytönseurantasovellus, saatavissa Excel-muodossa
<URL: <http://appro.mit.jyu.fi/tools/ajankaytto/ajankaytonseuranta.xls>>, Jyväskylän yliopisto, informaatioteknologian tiedekunta.
- [2] Heikki Salo, "Suunnitelma esikäyttäjien hyödyntämisestä"
<URL: http://sovellusprojektit.it.jyu.fi/verso/src/docs/project/suunnitelma_esikayttajien_hyodyntamisesta.txt>.
- [3] Juho Nieminen, "1. palaverin pöytäkirja", saatavissa raakatekstimuodossa
<URL: http://sovellusprojektit.it.jyu.fi/verso/src/docs/project/palaverit/poytakirja_1_palaveri.txt>, 17.2.2010.
- [4] Juho Nieminen, "2. palaverin pöytäkirja"
<URL: http://sovellusprojektit.it.jyu.fi/verso/src/docs/project/palaverit/poytakirja_2_palaveri.txt>, 25.2.2010.
- [5] Marko Peltola, "4. palaverin pöytäkirja"
<URL: http://sovellusprojektit.it.jyu.fi/verso/src/docs/project/palaverit/poytakirja_4_palaveri.txt>, 12.3.2010.
- [6] Ville Tirronen, "Alustavaa lukemistoa projektiin liittyen".
- [7] "About Gitorious", saatavissa HTML-muodossa
<URL: <http://gitorious.org/about>>.
- [8] Wikipedia, "Gitorious", saatavissa XHTML-muodossa
<URL: <http://en.wikipedia.org/wiki/Gitorious>>.
- [9] "Gitorious Coding Style Guide", saatavissa HTML-muodossa
<URL: <http://gitorious.org/gitorious/pages/CodingStyleGuide>>.
- [10] "Ruby Style Guide", saatavissa raakatekstimuodossa
<URL: <http://github.com/chneukirchen/styleguide/raw/master/RUBY-STYLE>>.

-
- [11] Verso-projekti, "Verso-projektin tehtävähallinta", saatavissa HTML-muodossa
<URL: <https://trac.cc.jyu.fi/projects/verso/>>.
- [12] "Vertailu alustoista", saatavissa raakatekstimuodossa
<URL: http://sovellusprojektit.it.jyu.fi/verso/src/docs/project/vertailu_alustoista.txt>.
- [13] "Projektisuunnitelma", saatavissa PDF-muodossa
<URL: http://sovellusprojektit.it.jyu.fi/verso/src/docs/project/projektisuunnitelma/Verso_projektisuunnitelma_0.6.0.pdf>.
- [14] "Sovellusraportti", saatavissa PDF-muodossa
<URL: http://sovellusprojektit.it.jyu.fi/verso/src/docs/application/application_report/application_report_0.0.5.pdf>.