

Verso-sovellusprojekti

Projektisuunnitelma

Tero Hänninen

Juho Nieminen

Marko Peltola

Heikki Salo

Versio 0.6.0

Julkinen

8.4.2010

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, Projektisuunnitelma

Sivumäärä: 27

Tiedosto: Projektisuunnitelma.tex

Tiivistelmä: Verso-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle lähdekoodin julkistamisjärjestelmän. Projektisuunnitelmassa kuvataan projektin läpivientiä muun muassa projektin aikataulutuksen osalta.

Avainsanat: aikataulu, tavoitteet, käytänteet, projektiorganisaatio, projektisuunnitelma, riskit, työnjako, Git, versiohallinta

Muutoshistoria

| Versio | Päivämäärä | Muutokset | Tekijät |
|--------|------------|--|---------|
| 0.1.0 | 22.2.2010 | Dokumentin pohja luotu. | HS |
| 0.1.1 | 24.2.2010 | Laadittu tavoitteita, taustaa, käytänteitä, riskejä ja prosessimallia kuvaavat luvut. | HS |
| 0.2.0 | 8.3.2010 | Laadittu projektiorganisaatiota kuvaava luku. Korjauksia Santasen kommenttien pohjalta. | HS |
| 0.2.1 | 9.3.2010 | Laadittu johdantoa ja taustoja kuvaavat luvut. Korjauksia Santasen kommenttien pohjalta. | HS |
| 0.3.0 | 18.3.2010 | Lisätty Trac-käytänteitä kuvaava luku. Täydennetty taustaa ja riskejä kuvaavia lukuja. Korjauksia Leena Peltomaan kommenttien perusteella. | HS |
| 0.4.0 | 24.3.2010 | Paavo Nieminen lisätty tilaajaksi. Korjauksia Santasen esitysten pohjalta. | HS |
| 0.5.0 | 6.4.2010 | Päivitetty taustaa kuvaavaa lukua. Korjauksia Santasen esitysten pohjalta. | HS |
| 0.5.1 | 7.4.2010 | Lisätty henkilökohtaiset oppimistavoitteet. Korjattu ilmaisuja. | HS |
| 0.6.0 | 8.4.2010 | Lisätty luvut lähdekoodin käytänteistä, integraatiotestauksesta sekä yhteenveto. Korjattu kirjoitusvirheitä. | HS |

Tietoa projektista

Verso-projekti toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle lähdekoodien julkaisujärjestelmän.

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 |

Tilaja:

- | | | |
|--------------------|---------------------------|-------------|
| • 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 |
| 3.1 | Taustalla olevat ongelmat ja tarpeet | 3 |
| 3.2 | Git | 3 |
| 3.3 | Gitorious | 3 |
| 4 | Tavoitteet | 5 |
| 4.1 | Kehitettävä sovellus | 5 |
| 4.2 | Tulokset | 5 |
| 4.3 | Oppimistavoitteet | 7 |
| 5 | Organisaatio ja resurssit | 8 |
| 5.1 | Projektioorganisaatio | 8 |
| 5.2 | Projektin tilat, laitteet ja ohjelmistot | 8 |
| 5.3 | Ohjelmointi- ja dokumentointityökalut | 9 |
| 5.4 | Luennot ja perehdytykset | 9 |
| 6 | Käytänteet | 11 |
| 6.1 | Tiedotus | 11 |
| 6.2 | Palaverit | 11 |
| 6.3 | Lähdekoodin käytänteet | 12 |
| 6.4 | Versiohallinnan käytänteet | 12 |
| 6.5 | Dokumentoinnin käytänteet | 13 |
| 6.6 | Julkistettujen dokumenttiedostojen versionumerointi | 13 |
| 6.7 | Esikäyttäjien hyödyntäminen | 13 |
| 7 | Trac-käytänteet | 14 |
| 7.1 | Luku- ja muokkausoukset | 14 |
| 7.2 | Traciin kirjattavat tehtävät | 14 |
| 7.3 | Kirjoitusasusta ja otsikoista | 14 |
| 7.4 | Sähköposti ja tiketit | 15 |
| 7.5 | Tikettityypit | 15 |
| 7.6 | Prioriteetit | 16 |

| | | |
|-----------|---|-----------|
| 7.7 | Tarkistuspisteet ja versiot | 16 |
| 8 | Tehtävät, työmäärät ja vastualueet | 17 |
| 8.1 | Vastualueet dokumenttien osalta | 17 |
| 8.2 | Tehtävien työmäärät ja työnjako | 17 |
| 9 | Prosessimalli ja aikataulu | 19 |
| 9.1 | Prosessimalli | 19 |
| 9.2 | Sovelluksen ominaisuuksien hyväksyminen | 19 |
| 9.3 | Jatkuva integraatiotestaus | 19 |
| 9.4 | Aikataulu | 20 |
| 10 | Riskit ja niiden seuranta | 22 |
| 10.1 | Riskien todennäköisyydet ja haitat | 22 |
| 10.2 | Muutostarve kehityskoneissa | 23 |
| 10.3 | Tavoitteiden rajausta | 23 |
| 10.4 | Valitun alustan ongelmat | 23 |
| 10.5 | Poissaolot | 24 |
| 10.6 | Testipalvelimen ongelmat | 24 |
| 10.7 | Motivaation puute | 24 |
| 11 | Yhteenveto | 25 |
| 12 | Lähteet | 26 |

1 Johdanto

Verso-projekti on Jyväskylän yliopiston tietotekniikan laitoksella keväällä 2010 toteutettava sovellusprojekti. Projekti määrittelee, suunnittelee ja toteuttaa Jyväskylän yliopiston tietotekniikan laitokselle prototyypin lähdekoodin julkistamisjärjestelmästä.

Yliopistolla ei tällä hetkellä ole käytössä yhtenäistä järjestelmää lähdekoodien jakamista tai julkistamista varten. Jokainen lähdekoodia tuottava tutkimusryhmä joutuu kehittämään omat käytänteensä lähdekoodin kanssa toimimiseen.

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

Verso-projektin projektisuunnitelmassa kuvataan läpivientiä, tehtäviä, aikataulua, prosessimallia, käytänteitä, työnjakoa, resursseja ja riskejä. Dokumentti on tarkoitettu projektiryhmälle, tilaajan edustajille ja projektin ohjaajille. Projektissa toteutettavan prototyypin ominaisuusvaatimukset on määritelty erilliseen Trac-järjestelmään.

Luvussa 2 kuvataan projektiin liittyviä termejä. Luvussa 3 tarkastellaan projektin taustaa. Luku 4 esittelee projektille asetetut tavoitteet ja tulokset. Luvussa 5 esitellään projektiorganisaatio ja muita projektin käytössä olevia resursseja. Luku 6 määrittelee projektin käytänteet. Luku 7 määrittelee projektissa käytettävät Trac-käytänteet. Luku 8 kuvaa suunniteltuja tehtäviä ja työnjakoa. Luku 9 määrittelee projektin prosessimallin ja aikataulun. Riskejä on käsitelty luvussa 10.

2 Termit

Dokumentissa esiintyvät aihealueen termit ovat seuraavat:

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

3 Taustaa

Luvussa käsitellään Verso-projektin taustaa.

3.1 Taustalla olevat ongelmat ja tarpeet

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 [5].

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].

3.2 Git

Git-ohjelmisto on Linus Torvaldsin kehittämä versiohallintaohjelmisto. Git kehitettiin alun perin Linux-ytimen kehittämistä varten ja se sopii käytettäväksi kaikenkoisten tietovarastojen kanssa. Git-valittiin Verso-projektiin, koska se oli projektiorganisaatiolle tutuin hajautetun versiohallinnan järjestelmä.

3.3 Gitorious

Gitorious on sekä palvelu että avoimen lähdekoodin sovellus. Internet-palvelu <http://gitorious.org> tarjoaa tilaa ja työkaluja avoimen lähdekoodin projekteille, jotka käyttävät Git-versiohallintaa. Gitorious-palvelussa käytettävä Gitorious-sovellus käyttää AGPL-lisenssiä, joten sovelluksen lähdekoodi on vapaasti käytet-

tävissä ja muokattavissa omaan käyttöön, kunhan muokkaukset ovat vapaasti saatavissa. [6]

Wikipedian [7] mukaan Gitoriousia käyttää moni suuri avoimen lähdekoodin projekti, kuten Qt, Amarok, Maemo, Amarok ja jatkossa koko KDE. Sovelluksen lähdekoodi on saatavilla Gitorious-palvelussa. Gitoriousia kehittää norjalainen Shortcut AS.

Projektiryhmä vertaili erilaisia versiohallinnan tukijärjestelmiä käytettäväksi sovelluksen alustana [11]. Alustoista valittiin Gitorious, koska se tarjosi vertailussa eniten tavoiteltuja ominaisuuksia ja sen kehitys nähtiin aktiiviseksi.

Vaihtoehtona oli myös olla käyttämättä valmista sovellusta alustana ja kehittää WWW-sovellus käyttäen pohjalla WWW-ohjelmistokehystä. Projektiryhmä tutustui alustavasti Django, TurboGears ja Ruby on Rails -ohjelmistokehyksiin. WWW-sovelluskehukset olivat projektiryhmälle pääosin vieraita ja ryhmä katsoi, että Ruby on Railsiin pohjautuvan Gitorious-järjestelmän opettelussa ja kehittämisessä ei ole suurempi vaiva kuin täysin kehityksen aloittamisessa täysin uudella ohjelmistokehysellä.

4 Tavoitteet

Luvussa kuvataan projektin tavoitteita sekä oppimisen että kehitettävän sovelluksen kannalta.

4.1 Kehitettävä sovellus

Projektiryhmä tuottaa prototyypin lähdekoodien julkistamisjärjestelmästä.

Projektissa toteutettavan prototyypin ominaisuusvaatimukset on määritelty erilliseen Trac-järjestelmään [10]

4.2 Tulokset

Sovelluksen ohella projektiryhmä toteuttaa seuraavat tulokset:

| | |
|----------------------------|---|
| Asennusohje | sisältää sovelluksen asennusohjeet. |
| Esittelymateriaali | sisältävät väli- ja loppuesittelyn materiaalit. |
| Lähdekoodi | sisältää tuotetun lähdekoodin kommentteineen. |
| Projektiraportti | kuvaava projektin läpiviennin ja asetettujen tavoitteiden saavuttamista. |
| Projektisuunnitelma | kuvaava projektin tavoitteita, tehtäviä, aikataulua, yleisiä käytäntöjä ja riskien hallintaa. |
| Sovellusraportti | kuvaava toteutetun sovelluksen osat ja toiminnot sekä jatkokehitysideat. |
| Vaatimusmäärittely | kuvaava sovelluksen käyttäjille tarjoamia toimintoja. |

Edellä mainittujen dokumenttien lisäksi ryhmä laatii seuraavat tulokset:

| | |
|---------------------------|---|
| Ajankäyttöraportti | sisältää ryhmän jäsenten kirjaamat työtunnit. |
|---------------------------|---|

| | |
|------------------------------|---|
| Itsearviointit | sisältävät ryhmän jäsenten arvioinnit omasta panoksesta, onnistumisesta ja oppimisesta. |
| Palaverien dokumentit | sisältävät palaverien esityslistat ja pöytäkirjat. |
| Sähköpostiarkistot | sisältävät kaikki projektin sähköpostilistalla käydyt keskustelut. |

4.3 Oppimistavoitteet

Ryhmän jäsenet saavat projektin kuluessa kokemusta projektin suunnittelusta ja läpiviennistä sekä tarvittavien dokumenttien laatimisesta. Jäsenet oppivat myös koustekniikkaa toimimalla projektipalavereissa puheenjohtajana ja sihteerinä.

Projektiryhmä oppii käyttämään projektissa suuressa osassa olevaa Git-versiohallintaa. Ryhmän jäsenet saavat kokemusta versiohallintaan liittyvistä teknisistä järjestelyistä, mutta myös Gitin käyttöön liittyvistä käytänteistä avoimen lähdekoodin ohjelmistoprojektissa.

Projektin ohjelmoinnin aikana jäsenet käyttävät lisäksi uusia uusia ohjelmointikieliä. Ohjelmoinnissa käytetty Ruby-kieli ja Ruby on Rails -sovelluskehys ovat koko projektiryhmälle ennalta vieraita, mutta projektiryhmällä on ennen projektin aloittamista ollut kiinnostusta perehtyä siihen.

Projektin vaatimuksia, kehitysideoita sekä löydettyjä virheitä ylläpidetään Trac-projektinhallinnassa. Projektin aikana projektiryhmä oppii perustaidot yleisen projektinhallintatyökalun käytöstä.

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

Projektiryhmän jäsenillä on lisäksi henkilökohtaisia oppimistavoitteita:

- Heikki haluaa oppia projektinhallintaa, projektin aikataulutusta sekä tilaajan kanssa toimimista.
- Juho haluaa oppia projektinhallintaa, ohjelmointia ja vastaanottamaan palautetta.
- Marko haluaa oppia ohjelmointia ja versiohallinnan käyttöä.
- Tero haluaa oppia terminaalinympäristön kanssa toimimista ja ohjelmointia.

5 Organisaatio ja resurssit

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

5.1 Projektorganisaatio

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

Tilajana olevan tietotekniikan laitoksen edustajina toimivat Paavo Nieminen, Tapani Tarvainen, Ville Tirronen ja Tero Tuovinen, joista Ville Tirrosella on ylin päätösvalta. Ryhmän vastaavana ohjaajana toimii Jukka-Pekka Santanen ja teknisenä ohjaajana toimii Antti-Juhani Kaijanaho.

Projektiryhmän työkoneiden ohjelmistojen ja ylläpidon hoitaa Jyväskylän yliopiston ATK-lähituki. Projektiryhmän käytössä olevan virtuaalipalvelimen ylläpidosta vastaa Harri Tuomi.

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

5.2 Projektin tilat, laitteet ja ohjelmistot

Projektiryhmän huone AgC222.2 sijaitsee Agoran C-siivessä toisessa kerroksessa sovellusprojektien tiloissa. Huoneen puhelinnumero on 014-2604963.

Ryhmän käytössä on neljä tietokonetta, joista kolmessa on käyttöjärjestelmänä Linux Fedora Core ja yhdessä Windows XP, jolla kirjataan ryhmän jäsenten työtunnit. Projektiryhmällä on lisäksi verkon kautta käyttöoikeus myös erilliseen testipalvelimeen.

5.3 Ohjelmointi- ja dokumentointityökalut

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

Työmäärien kirjaamiseen on käytössä Petri Heinosen kehittämä Excel-pohjainen ajan-
käytönseurantasovellus.

Projektioorganisaation käytössä on myös Trac-sovellus, jota käytetään sovelluksen vaatimusten ja projektin tehtävien hallintaan. Traciin liittyvät käytänteet on kuvattu luvussa 7.

5.4 Luennot ja perehdytykset

Ohjelmointikielenä on Ruby ja alustana Ruby on Rails, jotka eivät ole ryhmän jäsenille ennestään tuttuja. Ryhmä tutkii kirjallisuutta verkosta ja tilaa tarvittaessa Santasen kautta käyttöönsä käyttökelpoiset kirjat.

Projektissa vaaditaan hyvää tuntemusta Git-versiohallinnasta, joka ei ollut ryhmäläisille entuudestaan tuttu. Tekninen ohjaaja Antti-Juhani Kaijanaho toimii asiantuntijana versiohallintaan liittyvissä kysymyksissä ja järjestää ryhmälle ongelma-
lähtöistä perehdytystä.

Projektissa toteutettavan prototyypin ohjelmointiin tarvitaan tuntemusta Rubystä, Ruby on Railsista sekä sovelluksen pohjana toimivan Gitoriousin toiminnasta. Projektiryhmä on perehtynyt Rubyyn ja Ruby on Railsiin omatoimisesti tutkimal-
la Internetissä saatavilla olevia ilmaisia oppaita. Gitoriousin asentamiseen ja toimintaan projektiryhmä on tutustunut lähdekoodeihin perehtymällä sekä tarvittaessa pyytämällä Gitoriousin kehitysyhteisöltä sen IRC-kanavalla apua.

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.

6 Käytänteet

Luvussa kuvataan projektissa käytettävät käytänteet.

6.1 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.

6.2 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.

6.3 Lähdekoodin käytänteet

Ohjelmoidessa noudatetaan ensimmäisen palaverin [3] mukaisesti Ruby-kielen [8] Ruby on Rails -ohjelmistokehyksen ohjelmointityyliä sekä Gitorious- ohjelmiston ohjelmointikäytänteitä [8]. 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 6.1: Lähdekoodiesimerkki.

6.4 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.

6.5 Dokumentoinnin käytänteet

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

6.6 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.

6.7 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].

7 Trac-käytänteet

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

7.1 Luku- ja muokkausoikeudet

Verso-projektin Tracin selaamiseen on järjestetty avoin pääsy, mutta muokkaamista varten käyttäjän pitää kirjautua sisään. Muokkausoikeus on kaikilla luvussa 5.1 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.

7.2 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.

7.3 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.

7.4 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.

7.5 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 7.1: Trac-tikettien tyypit.

Taulukossa 7.1 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.

7.6 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 7.2: 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 kehitetyksi tulemiseen Verso-projektissa, vaan niitä käytetään hyödyksi, kun iteraatioihin ollaan kiinnittimässä tehtäviä.

7.7 Tarkistuspisteet ja versiot

Tarkistuspiste (engl. *milestoneilla*) viitataan projektin kehitysprosessin vaiheisiin eli *iteraatioihin*. Iteraatioiden aikataulut on kuvattu luvussa 9.4.

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

8 Tehtävät, työmäärät ja vastualueet

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

8.1 Vastualueet dokumenttien osalta

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

Taulukko 8.1: Dokumenttien kielet ja vastualueet.

8.2 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 8.1: Jäsenten työmääräarvio

9 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.

9.1 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, palaverikeskutelujen ja muun yhteydenpidon perusteella projektiryhmä tekee tilaajalle ehdotuksen seuraavan iteraation sisällöstä ja sen tehtävät hyväksytään projektipalaverissa.

9.2 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.

9.3 Jatkuva integraatiotestaus

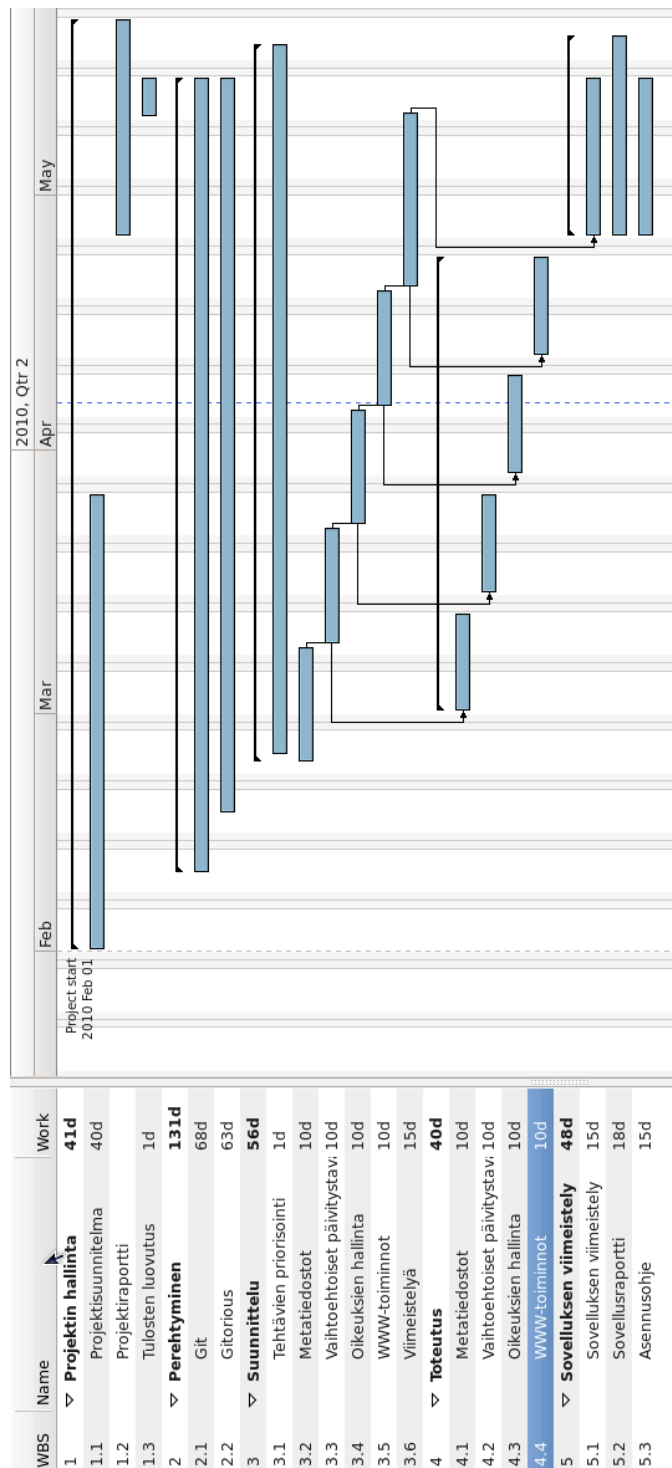
Sovellukseen tuotettavan lähdekoodin laatua testataan jatkuvasti. Kun uutta ominaisuutta kehitetään, sen lähdekoodi sijoitetaan omaan haaraansa (engl. *topic branch*) sovelluksen lähdekoodin tietovarastossa, kuten luvussa Versiohallinnan käytänteet 6.4 kuvataan. Kehityksen alla olevaa ohjelmakoodia testataan 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.

9.4 Aikataulu

Projektin tehtävien aikataulut on esitetty kuvan 9.1 Gantt-kaaviossa.



Kuva 9.1: Gantt-kaavio tehtävien aikataulutuksesta

10 Riskit ja niiden seuranta

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

10.1 Riskien todennäköisyydet ja haitat

Riskien todennäköisyydet ja niistä seuraavat haittavaikutukset on esitetty taulukossa 10.1. 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 10.1: Arvioidut riskit, niiden todennäköisyys ja haittavaikutus.

10.2 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 vastaantullut ongelma on haastava ja vaatii selvittämistä yhdessä lähituen kanssa, sen ratkaisuun kuluu kauan. Riskiä hallitaan ennakoimalla muutostarpeita ja pyytämällä tarvittavat muutokset etukäteen.

10.3 Tavoitteiden raja

Tähän juttua siitä, että vaikka periaatteessa Gitoriukselle 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.

10.4 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.

10.5 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.

10.6 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.

10.7 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.

11 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.

12 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 käyttämisestä" <URL: http://versotest.it.jyu.fi/verso/verso-project/blobs/master/project_docs/suunnitelma_esikayttajien_kayttamisesta.txt>.
- [3] Juho Nieminen, "1. palaverin pöytäkirja", saatavissa raakatekstimuodossa <URL: http://versotest.it.jyu.fi/verso/verso-project/blobs/master/project_docs/palaverit/poytakirja_1_palaveri.txt>.
- [4] Juho Nieminen, "2. palaverin pöytäkirja" <URL: http://versotest.it.jyu.fi/verso/verso-project/blobs/master/project_docs/palaverit/poytakirja_2_palaveri.txt>.
- [5] Ville Tirronen, "Alustavaa lukemistoa projektiin liittyen".
- [6] "About Gitorious", saatavissa HTML-muodossa <URL: <http://gitorious.org/about>>.
- [7] Wikipedia, "Gitorious", saatavissa XHTML-muodossa <URL: <http://en.wikipedia.org/wiki/Gitorious>>.
- [8] "Gitorious Coding Style Guide", saatavissa HTML-muodossa <URL: <http://gitorious.org/gitorious/pages/CodingStyleGuide>>.
- [9] "Ruby Style Guide", saatavissa raakatekstimuodossa <URL: <http://github.com/chneukirchen/styleguide/raw/master/RUBY-STYLE>>.
- [10] Verso-projekti, "Verso-projektin tehtävähallinta", saatavissa HTML-muodossa <URL: <https://trac.cc.jyu.fi/projects/verso/>>.
- [11] "Vertailu alustoista", saatavissa raakatekstimuodossa <URL: <http://versotest.it.jyu.fi/verso/verso-project>>

```
/blobs/master/project_docs  
/vertailu_alustoista.txt>
```