

# **Verso Project**

## **Requirement Specification**

**Tero Hänninen**  
**Juho Nieminen**  
**Marko Peltola**  
**Heikki Salo**

Version 0.1.1  
Public  
3.3.2010

**University of Jyväskylä**  
**Department of Mathematical Information Technology**  
**Jyväskylä**



**Authors:**

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

**Customers:**

Tapani Tarvainen	tt@it.jyu.fi	014-2602752
Ville Tirronen	ville.e.t.tirronen@jyu.fi	014-2604987
Tero Tuovinen	Tero.Tuovinen@jyu.fi	050-4413685

**Instructors:**

Jukka-Pekka Santanen	santanen@mit.jyu.fi	014-2602766
Antti-Juhani Kaijanaho	antti-juhani.kaijanaho@jyu.fi	014-2602756

**Contact information:**

Email lists	verso@korppi.jyu.fi and konkari_opetus@korppi.jyu.fi.
Email archives	<a href="https://korppi.jyu.fi/kotka/servlet/list-archive/verso/">https://korppi.jyu.fi/kotka/servlet/ list-archive/verso/</a> and <a href="https://korppi.jyu.fi/kotka/servlet/list-archive/verso_opetus/">https://korppi.jyu.fi/kotka/servlet/ list-archive/verso_opetus/</a>
Workroom	AgC 222.2, tel. 014-1234567.

Acceptor	Date	Signature	Clarification
Project manager	__.__.2010		
Customer	__.__.2010		
Instructor	__.__.2010		

## Version history

<b>Version</b>	<b>Date</b>	<b>Modications</b>	<b>Modifier</b>
0.1.0	17.2.2010	Chapters 1, 3, 4, 5 and ?? started.	JN
0.1.1	24.2.2010	LaTeX document created.	JN
0.1.2	25.2.2010	Document translated into English. Added requirements based on the record of summit 2 and continued chapter 1.	JN
0.1.3	2.3.2010	Continued the functional requirements chapter.	JN
0.1.4	3.3.2010	Added terms to chapter 2 and clarified items in chapter 6.	JN

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Terminology</b>	<b>2</b>
<b>3</b>	<b>Priorities</b>	<b>3</b>
<b>4</b>	<b>Requirement statuses</b>	<b>4</b>
<b>5</b>	<b>Description of current working methods</b>	<b>5</b>
<b>6</b>	<b>Functional requirements</b>	<b>6</b>
6.1	General . . . . .	6
6.2	Creating a repository . . . . .	6
6.3	WWW-interface . . . . .	7
6.4	Alternative interfaces . . . . .	7
6.5	Access control . . . . .	7
6.6	Lisencing . . . . .	7
6.7	Other requirements . . . . .	8
<b>7</b>	<b>Restrictions</b>	<b>9</b>
<b>8</b>	<b>Technical requirements</b>	<b>10</b>
<b>9</b>	<b>Summary</b>	<b>11</b>



# 1 Introduction

The goal of Verso project is to create a supporting software for version control. The software is a web application that will provide a publishing channel for source code and an easy to use interface for version control. The Verso website lets users to create projects and then share their work through automatically generated project home pages. On the website, it is possible to browse and search projects and examine the source code directly as well.

Currently, at the Department of Mathematical Information Technology in University of Jyväskylä, there is no common system for version control. This causes problems when a worker leaves the department because the source code he has done might leave too. The lack of a single version control system also prevents workers from knowing who is doing what, which may lead to producing overlapping work. Furthermore, the current disorganized situation presents licencing problems in which one is unaware who owns a piece of source code and how can it be used.

The idea is to get as many people to use proper version control as possible. Therefore, the system should be designed so that it can be used in many different ways. A WWW-interface is provided for the people unfamiliar with version control, a command line interface is provided for the more experienced and even features that will adapt to people's working methods will be implemented, such as reading and mirroring a zip-archive at supplied URL.

## 2 Terminology

<b>Commit</b>	contains file modification data and a log message from the user describing the changes.
<b>Database</b>	is a storage component used by the website which stores the user and WWW-interface data.
<b>Project</b>	is a user created project in the Verso system which has one or many repositories.
<b>Repository</b>	refers to a Git repository which belongs to a project.
<b>Sertification</b>	is a quarantee of quality which a priviledged user can give to source code.

### 3 Priorities

<b>Name</b>	<b>Symbol</b>	<b>Description</b>
Mandatory	+++	The requirement must be implemented in the software.
Important	++	The requirement is considered as very useful to the user.
Useful	+	The requirement is considered as useful but not essential to the user.
Left out	x	The requirement will not be implemented.
Undecided	-	The priority of the requirement is not decided yet.

## 4 Requirement statuses

<b>Name</b>	<b>Abbreviation</b>	<b>Description</b>
Third party	3.p.	The feature has been approved by the customer.
Approved	appr.	The feature has been approved by tutors.
Tested	test.	The feature has been tested.
Implemented	impl.	The feature has been made.
Under development	u.d.	The feature is being developed.
Pending	pend.	The feature is waiting for implementation.
Abandoned	aband.	The feature will not being implemented.

## 5 Description of current working methods

Many people already use version control at the Department of Mathematical Information Technology in University of Jyväskylä. However, they only use it locally for personal use or among their current project group so the source code is not shared inside the department.

A very common way to share source code between co-workers is to use a portable USB flash drive. It is considered easy and reliable.

## 6 Functional requirements

### 6.1 General

Item	Description	Pri.	State
1.1	The software interface is in English.	+++	u.d.
1.2	The software supports distributed source code management.	+++	impl.
1.3	The software has automated testing tools integrated.	+	pend.
1.4	The software offers project management tools.	+	pend.

### 6.2 Creating a repository

Item	Description	Pri.	State
2.1	A user can create repositories which are placed under a project.	+++	u.d.
2.2.	When a user who has no project creates a repository, a project will be created for him in the process.	+++	pend.
2.3	When a project is created the software will generate a home page for it.	+++	pend.
2.4	When a repository is created the software will generate a home page for it.	+++	pend.

### 6.3 WWW-interface

Item	Description	Pri.	State
3.1	The project home page has links to repository files and commits.	+++	pend.
3.2	A user can add and update repository files.	+	pend.
3.3	A user can edit text based files.	+	pend.
3.4	A user can comment source code.	+	pend.
3.5	A privileged user can give a certification for source code.	+	pend.
3.6	Existing and active projects are be well presented on the front page.	+	pend.
3.7	There is a link to the existing repositories on the front page.	++	pend.

### 6.4 Alternative interfaces

Item	Description	Pri.	State
4.1	The software is able to mirror a SVN repository on a set interval.	++	pend.
4.2	A user can add files to database via email.	+	pend.
4.3	The software is able to mirror a online ZIP archive on a set interval.	++	pend.

### 6.5 Access control

Item	Description	Pri.	State
5.1	A user can allow the access to his project for himself only, for a selected group or for everyone.	++	pend.

### 6.6 Lisencing

Item	Description	Pri.	State
6.1	A user can add and modify a licence of a project.	+++	u.d.

## 6.7 Other requirements

<b>Item</b>	<b>Description</b>	<b>Pri.</b>	<b>State</b>
7.1	Project metafiles are located in the database.	++	pend.
7.2	Repository metafiles are located in the repository.	++	pend.

## 7 Restrictions

- ...

## 8 Technical requirements

- ...

## 9 Summary

- ...