Software Configuration Management Plan

Project Manager:
Sebastiaan Candel

Quality Assurance Manager:
Ronald van Zon

Senior Management:
Mark van den Brand, MF 7.0.96
Lou Somers, MF 7.145

Advisor:
Erik Scheffers, MF 7.146

Customer:
Lex Lemmens, HG -1.27

Authors:
Peter van Heck (0649886)
Peter Koymans (0748876)
Kay Lukas (0758084)
Astrid Pieterse (0743073)
Robbert Raats (0675661)
Willem Sonke (0752229)
Roby Visser (0654428)

October 26, 2012 Eindhoven
Abstract

This document contains the Software Configuration Management Plan (SCMP) for the kroket application, which aids students in choosing electives for the new Bachelor College. This application is made as part of the Software Engineering Project at Eindhoven University of Technology. This document contains information on the standards to be used for writing the documentation required for this project, as well as information about the processing and storage of these documents.

The document complies with the SCMP of the Software Engineering Standards, as set by the European Space Agency (ESA) [1].
Contents

Document Status Sheet 4

Document Change Records 5

1 Introduction 6
   1.1 Purpose ................................................. 6
   1.2 Scope ................................................ 6
   1.3 List of definitions ...................................... 7
   1.4 List of references ...................................... 7

2 Management 8
   2.1 Organization .......................................... 8
   2.2 Responsibilities ....................................... 8
   2.3 Interface management .................................. 8
   2.4 SCMP implementation .................................. 8
   2.5 Applicable procedures ................................. 9

3 Configuration identification 10
   3.1 Naming conventions .................................... 10
   3.2 Baselines ............................................. 10

4 Configuration control 11
   4.1 Library control ....................................... 11
      4.1.1 Development library ................................ 11
      4.1.2 Master library .................................... 12
      4.1.3 Archive library ................................... 12
   4.2 Media control ......................................... 13
   4.3 Change control ....................................... 13
      4.3.1 Development library ............................. 13
      4.3.2 Master library .................................... 13
      4.3.3 Archive library ................................... 13

5 Status accounting 14
6 Tools, techniques and methods
   6.1 Server ......................................................... 15
   6.2 SVN ............................................................... 15
   6.3 Apache .......................................................... 15
   6.4 Python ........................................................... 15
   6.5 Trac ............................................................... 16
   6.6 LaTeX ............................................................. 16
   6.7 Log files .......................................................... 16
   6.8 Future changes .................................................. 16

7 Supplier control ...................................................... 17

8 Records collection and retention ................................... 18
Document Status Sheet

General

Document title Software Configuration Management Plan
Identification Documentatie.SCMP.0.1.0
Author Willem Sonke, Astrid Pieterse
Document status Final

Document history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0</td>
<td>10-9-2012</td>
<td>Willem Sonke</td>
<td>Initial version</td>
</tr>
<tr>
<td>0.0.1</td>
<td>12-9-2012</td>
<td>Astrid Pieterse, Willem Sonke</td>
<td>Adding all required chapters</td>
</tr>
<tr>
<td>0.1.0</td>
<td>24-10-2012</td>
<td><em>not applicable</em></td>
<td>First internal review</td>
</tr>
</tbody>
</table>
Document Change Records

General

Date 25-10-2012
Document title Software Configuration Management Plan
Identification Documentatie.SCMP.0.1.0

Document history (relative to 0.0.1)

<table>
<thead>
<tr>
<th>Sections</th>
<th>Reason of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Internally approved</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to provide rules and guidelines for the storage and of all documents that are created during this project. Additionally it will give naming and layout conventions for all documents and describe the identification of all major documents.

1.2 Scope

This document is about:

- Listing the specific documents which need to be written during the course of the project.
- Giving naming conventions for these documents.
- Providing a structured way to create, store and update the documents.
- Providing a standard layout for these documents.

This document will not describe the detailed contents of the individual documents. The Configuration Items (CIs) that will be written during the project are:

- Architectural Design Document (ADD)
- Detailed Design Document (DDD)
- Software Configuration Management Plan (SCMP)
- Software Project Management Plan (SPMP)
- Software Quality Assurance Plan (SQAP)
- Software Requirements Document (SRD)
- Software Transfer Document (STD)
Kroket – SOFTWARE CONFIGURATION MANAGEMENT PLAN

- Software User Manual (SUM)
- Software Verification and Validation Plan (SVVP)
- User Requirements Document (URD)
- Code
- Minutes
- Test plans for Unit Test (UT), System Test (ST), Integration Test (IT) and Acceptance Test (AT)
- Product Backlog

1.3 List of definitions

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>Architectural Design Document</td>
</tr>
<tr>
<td>AT</td>
<td>Acceptance Test</td>
</tr>
<tr>
<td>ATP</td>
<td>Acceptance Test Plan</td>
</tr>
<tr>
<td>CI</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Manager</td>
</tr>
<tr>
<td>DDD</td>
<td>Detailed Design Document</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>PM</td>
<td>Project Manager</td>
</tr>
<tr>
<td>QM</td>
<td>Quality Manager</td>
</tr>
<tr>
<td>SCMP</td>
<td>Software Configuration Management Plan</td>
</tr>
<tr>
<td>SEP</td>
<td>Software Engineering Project</td>
</tr>
<tr>
<td>SL</td>
<td>Software Librarian</td>
</tr>
<tr>
<td>SPMP</td>
<td>Software Project Management Plan</td>
</tr>
<tr>
<td>SQAP</td>
<td>Software Quality Assurance Plan</td>
</tr>
<tr>
<td>SRD</td>
<td>Software Requirements Document</td>
</tr>
<tr>
<td>STD</td>
<td>Software Transfer Document</td>
</tr>
<tr>
<td>SUM</td>
<td>Software User Manual</td>
</tr>
<tr>
<td>SVVP</td>
<td>Software Verification and Validation Plan</td>
</tr>
<tr>
<td>SVVR</td>
<td>Software Verification and Validation Report</td>
</tr>
<tr>
<td>URD</td>
<td>User Requirements Document</td>
</tr>
</tbody>
</table>

1.4 List of references

Chapter 2

Management

2.1 Organization

The persons that are directly involved in the Configuration Management are the configuration manager and the vice configuration manager. Their names are stated in the SPMP.

2.2 Responsibilities

The configuration manager should provide a working environment for configuration management all the time. Any problems should be reported as soon as possible.

The CM is also responsible for putting versions of documents in the master and archive libraries, as mentioned in chapter 4. Creating and updating the document templates is also their responsibility.

The (primary) configuration manager is the first responsible for configuration management, however he may delegate tasks to the vice configuration manager. Also, whenever the configuration manager is not available or unreachable, the vice configuration manager should take over his tasks. Furthermore, all group members are responsible for their own documents (this includes updating the document status sheet; see also chapter 5).

2.3 Interface management

In case of failure of any hardware supplied by the university the CM or vice CM will contact ‘Bureau Computer Faciliteiten’ (BCF), as they are assumed to have expert knowledge of the hardware configuration of the supplied hardware.

2.4 SCMP implementation

Contrary to the ESA Software Engineering Standard [1], there will not be a separate SCMP document for each phase of the project. Instead this document will be updated with appendices for every phase of the project. For more information concerning planning of the phases, refer to the SPMP.
2.5 Applicable procedures

All the documents are subject to the standards described in the ESA standard and must also adhere to the requirements as described in the SQAP and the SVVP.

Every document has to be made with \LaTeX and use the kroket.cls document class (this can be found in the development library). Furthermore, instead of using one large .tex file, a document should be split up in smaller files if necessary (with \input).

Every document must have its title set using a \texttt{\title} command. \texttt{\maketitle} should be used to create the title page, which will happen automatically. Before the table of contents (generated by \texttt{\tableofcontents}), an abstract must be placed.

The CM will advise other group members when they experience problems with \LaTeX.
Chapter 3

Configuration identification

3.1 Naming conventions

All documents that will be handed to external parties will have a unique version number according to the following scheme. Also internally every document will have a version number, but that does not need be unique.

The first version of a document will have the version 0.0. Then for every externally reviewed version, this version will be increased by one, e.g. 0.1, 0.2, etc. When the document is approved by the customer, the version number is changed to 1.0.

If it is desirable, a third version digit can be added to the version number (e.g. 0.0.1). This number can be updated for distinguishing different versions that have not been reviewed externally. Since these versions are not externally approved, changes in the Change Records of a document refer to the previous externally approved version.

3.2 Baselines

Baselines are documents that have been externally reviewed and approved. They will be stored in the Master library (which will be discussed in chapter 4). According to the ESA standard new versions of the management documents need to be created for every stage in the project. Because of the small scale of this project the same management documents are used during the course of the project. Information specific for a stage in the project will be added to these documents in the form of appendices.
Chapter 4

Configuration control

4.1 Library control

This section describes the central storage facility for all CIs. All CIs are stored in the development library, and approved versions are moved to the master library or the archive library.

4.1.1 Development library

The development library contains CIs that are under construction, and CIs that are not official project documents. SVN will be used for the file management here. In the SVN development repository, only the latest version of files will be explicitly kept (the older versions will be automatically retained by SVN).

In the development SVN repository, the following directory structure will be used.

- **Code** – this folder contains all code produced, divided in the following subfolders:
  - **Authenticate** – code for authentication with the TU/e
  - **biztalk** – code for the communication with the TU/e administration (OWIS)
  - **CherryPy** – code for the CherryPy framework (see chapter 6)
  - **Django** – code for Django (see chapter 6)
  - **Web** – static files for the Apache webserver (accessible on http://kroket.win.tue.nl/Web)

- **Documentatie** – this folder contains all documents produced; there is a subfolder for every document, containing (only) the \LaTeX\ source code (these will be compiled automatically and appear on http://kroket.win.tue.nl/Documentation; error logs are accessible on http://kroket.win.tue.nl/Log)

- **Snippets** – this folder is a way to easily share files (for example tutorials or demos) with all other group members; for example, useful tutorials can be put here for the others to see
4.1.2 Master library

The second library, the master library, contains the current, externally approved version of all CIs (if there is already an externally approved version). The master library is a folder on the server outside of the SVN repository. This folder is made read-only accessible on the URL http://kroket.win.tue.nl/Master.

No document may be stored in this library without knowledge of the CM, and storage may only take place after the document has been reviewed and approved externally. Everybody is free to make copies of documents in this library by downloading them from the given URL, but these copies may not be put back into this library under any condition. No CIs will be deleted from this directory, they can only be replaced by a newer externally approved version. In that case the older version will be moved to the archive library.

In the master library a similar directory structure as in the development library will be used:

- **Code** – the code
- **Documentatie** – the documents, with subfolders for every document:
  - **URD** – the User Requirements Document
    
4.1.3 Archive library

The archive library contains older externally approved versions of all CIs. The archive library is stored in the same way as the master library and is made read-only accessible on the URL http://kroket.win.tue.nl/Archive.

The only way files can be placed in the archive library is by moving them from the Master library, as described above.

The archive library will have the same directory structure as the master library, but every (externally approved) version of CIs is put in a separate subfolder:

- **Code** – the code:
  - **0.1** – contains the code version 0.1
  - **0.2** – contains the code version 0.2
  
  etc.
- **Documentatie** – the documents, with subfolders for every document:
  - **URD** – the User Requirements Document:
    
    etc.
4.2 Media control

All documents and code will be stored on the virtual server from BCF. BCF backups all servers every night. For information about how to access the files, see chapter 6.

4.3 Change control

4.3.1 Development library

Because of the small development team, every group member is allowed to make changes to this library without explicit further permission. Of course, the team will informally discuss this often to avoid conflicts.

4.3.2 Master library

Once a CI is externally approved the CM can put it in the master library. If authors want to make changes to a document inside the master library, then that author has to take up contact with the QM. The QM will call up a review meeting where all the changes are approved or rejected. More information regarding the change procedure can be found in the SVVP. When changes to a CI are approved, the CM will copy the reviewed version of the CI to the master library. As the CM is the only one allowed to create and edit new documents in the master library, there is no need for change control.

4.3.3 Archive library

CIs in this library cannot be modified under any condition, only new files can be added. New versions may only be added after they have resided in the master library, and there is a new externally approved version that is being placed in the master library. Since documents in this library cannot be modified, there is no need for change control.
Chapter 5

Status accounting

CIs in the development library can be placed into the master library by the CM if they are externally approved. In case there already is a previous version of the CI in the master library, he will move that to the archive library in the corresponding folder.

Of course, all versions of a document contain a change record, in which the changes with respect to the previous version are recorded.

On the Trac system (see chapter 6), the CM will maintain a page on which the status of every document is recorded. The status records the dates on which the document has been put in the master library, and which versions are available in the archive library (with their inclusion dates in the master library). For example, such a status could look like the following.

<table>
<thead>
<tr>
<th>Version</th>
<th>Library</th>
<th>Date into master</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>current</td>
<td>Development</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>0.3</td>
<td>Master</td>
<td>2012-09-25</td>
<td>–</td>
</tr>
<tr>
<td>0.2</td>
<td>Archive</td>
<td>2012-09-18</td>
<td>–</td>
</tr>
<tr>
<td>0.1</td>
<td>Archive</td>
<td>2012-09-04</td>
<td>–</td>
</tr>
</tbody>
</table>
Chapter 6

Tools, techniques and methods

6.1 Server

For our file storage and testing requirements we have a virtual server from BCF (Bureau Computerfaciliteiten) available. The IP address of the server is 131.155.68.42; its hostname is kroket.win.tue.nl. The server is running Ubuntu 11.10.

The CM is responsible for maintaining this server and instructing the other group members how to use it. The following services will run on the server.

6.2 SVN

For the development library, a SVN repository will be available. The SVN server is configured such that after every commit, a post-commit hook is executed. This hook copies the contents of the Web folder of the repository to the Apache web server. In this way, the files on the web server can be easily changed by all group members, so that it is possible to test the website. Furthermore the CherryPy server will be restarted on every commit (see below).

The SVN checkout URL is svn://kroket.win.tue.nl/kroket-repo. To commit, a username and password is required. These will be created and given to the other group members by the CM.

6.3 Apache

For the web server, Apache will be used. Apache will be running on port 80, such that the following address can be used to access the web content: http://kroket.win.tue.nl/.

6.4 Python

We are using two frameworks: CherryPy and Django.

CherryPy A CherryPy server will run on port 8080. So, the web output can be observed on http://kroket.win.tue.nl:8080. The SVN server will restart CherryPy’s server after
CHAPTER 6. TOOLS, TECHNIQUES AND METHODS

every commit.

**Django**  Another Python framework, Django, will run on port 8081. It can thus be found on http://kroket.win.tue.nl:8081. Django’s admin interface can be found on http://kroket.win.tue.nl:8081/admin.

### 6.5 Trac

Trac is a bug tracking system. Furthermore it contains a wiki system with more information about the server setup. It can be accessed by using the URL http://kroket.win.tue.nl:8000.

### 6.6 L\TeX

L\TeX will be running on the server, and automatically compiling all L\TeX documents in the Documentatie folder of the development library. The documents will be copied to the Apache web server, such that http://kroket.win.tue.nl/Documentatie can be used to get all documents.

### 6.7 Log files

The server will automatically copy log files of several processes to the Apache server on http://kroket.win.tue.nl/Log. At the moment, these log files include the L\TeX’s output of the compilation of documents, and the CherryPy log.

### 6.8 Future changes

If in the future the need arises to install other services on the server, this will be the responsibility of the CM. Also the information on the Trac wiki system will be updated with the new services.
Chapter 7

Supplier control

The demands that are placed on external sources are described in the SQAP.
Chapter 8

Records collection and retention

In the development library, everyone is allowed to delete files (if there is agreement in the group, of course). However, SVN retains files also if they are deleted, so they can be recovered later.

In the master library, files can only be replaced by a newer version, but the old version will move to the archive library, where files cannot be removed. So files, that are once placed in the master library, will be retained for the entire project.