Horus
IMSETY
Software Configuration Management Plan
Version 0.7  14th May 2007

Project team:  Jeroen Keiren  0569081
               Frank Koenders  0575629
               Thijs Nugteren  0574426
               Joeri de Ruiter  0578312
               Stijn Stiefelhagen  0579816
               Carst Tankink  0569954
               Pim Vullers  0575766
               Freek van Walderveen  0566348

Project manager:  Egbert Teeselink

Senior management:  L. Somers  TU/e (HG 7.83)
                    M. v.d. Brand  TU/e (HG 7.44)

Adviser:  R.J. Bril  TU/e (HG 5.09)

Customer:  E. v. Breukelen  ISIS

Computer Science, Eindhoven University of Technology, Eindhoven
Abstract

This is the Software Configuration Management Plan (SCMP) for the IMSETY project. This project is part of the Software Engineering Project (2IP40) and is one of the assignments at the Eindhoven University of Technology. The document complies with the SCMP from the Software Engineering Standard, as set by the European Space Agency [2]. This document contains the rules, guidelines and procedures for versioning, naming and storage of all documents produced during the project. It also describes how document changes are to be handled and describes the backup- and safety procedures.
# Contents

1 Introduction ............................................. 1
   1.1 Purpose ........................................... 1
   1.2 Scope ............................................. 1
   1.3 List of definitions .................................. 2
   1.4 List of references .................................. 2

2 Management ............................................... 3
   2.1 Organization ....................................... 3
   2.2 Responsibilities .................................... 3
   2.3 Interface management ................................. 3
   2.4 SCMP implementation ................................. 3
   2.5 Applicable procedures ............................... 4

3 Configuration identification ........................... 6
   3.1 Naming conventions ................................ 6
   3.2 Baselines .......................................... 6

4 Configuration control .................................... 7
   4.1 Library control .................................... 7
   4.2 Media control ...................................... 7
   4.3 Change control ..................................... 7

5 Status accounting ........................................ 8

6 Tools, techniques and methods ......................... 9
   6.1 Version control ..................................... 9
   6.2 File server ......................................... 9
   6.3 Wiki and ticket system ............................... 9

7 Supplier control ......................................... 10

8 Records collection and retention ....................... 11

A Software requirements phase ........................... 12

B Architectural design phase .............................. 13

C Detailed design phase .................................. 14

D Transfer phase .......................................... 15
Document status sheet

<table>
<thead>
<tr>
<th>Document title</th>
<th>Software Configuration Management Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document identifier</td>
<td>IMSETY/doc/SCMP</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Freek van Walderveen</td>
</tr>
<tr>
<td>Version</td>
<td>0.7</td>
</tr>
<tr>
<td>Document status</td>
<td>Internally approved</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author(s)</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1 (Revision 175)</td>
<td>25-02-2007</td>
<td>Freek van Walderveen</td>
<td>First version up for internal review.</td>
</tr>
<tr>
<td>0.2 (Revision 238)</td>
<td>01-03-2007</td>
<td>Freek van Walderveen</td>
<td>Fixed problems found during first review. Up for second internal review.</td>
</tr>
<tr>
<td>0.3 (Revision 270)</td>
<td>04-03-2007</td>
<td>Freek van Walderveen</td>
<td>Fixed problems found during second review.</td>
</tr>
<tr>
<td>0.4 (Revision 584)</td>
<td>31-03-2007</td>
<td>Freek van Walderveen</td>
<td>Fixed problems found during review with Sylwia.</td>
</tr>
<tr>
<td>0.5 (Revision 586)</td>
<td>31-03-2007</td>
<td>Freek van Walderveen</td>
<td>Wrote AD appendix.</td>
</tr>
<tr>
<td>0.6 (Revision 985)</td>
<td>01-05-2007</td>
<td>Freek van Walderveen</td>
<td>Wrote DD appendix.</td>
</tr>
<tr>
<td>0.7 (Revision 1197)</td>
<td>14-05-2007</td>
<td>Freek van Walderveen</td>
<td>Fixed problems found during review.</td>
</tr>
</tbody>
</table>
# Document change report

<table>
<thead>
<tr>
<th><strong>Document title</strong></th>
<th>Software Configuration Management Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Document identifier</strong></td>
<td>IMSETY/doc/SCMP</td>
</tr>
<tr>
<td><strong>Date of changes</strong></td>
<td>01-03-2007</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Section number</strong></th>
<th><strong>Reason for change</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Wrote appendix.</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Purpose

Good configuration management is essential for efficient development and maintenance. This document is written in order to coordinate the configuration management and help the project members to benefit from configuration management instead of experiencing it as a necessary evil. The project group is therefore the primary audience. The (senior) project management might however also be interested in an overview of how the configuration management will be performed.

1.2 Scope

Configuration management is concerned with managing configuration items. For this project we distinguish the following configuration items:

- All project documents (including management plans and minutes, excluding agendas)
- All product documents (see also the SPMP [3])
- The product itself
  - Source code
  - Documentation
- All other documents and files related to the project of which either a history needs to be kept or which multiple project members will be working on

The librarians should facilitate configuration management by providing and maintaining the necessary services (chapter 6) and writing this document, including procedures, guidelines and document templates.
CHAPTER 1. INTRODUCTION

1.3 List of definitions

Baseline
A baseline is a document or a product that has been formally reviewed and agreed upon, and is a basis for further development. A baseline is an assembly of configuration items. Formal change control procedures are required to modify a baseline.

Configuration item
Any entity (single file or logical group of files) of which a history should be recorded and saved in a repository.

Commit
v. The action of making a new revision of one or more configuration items.
n. See revision.

ESA European Space Agency

Repository
Central location in which configuration items and their history are stored.

Revision
A particular version of a configuration item in the version control system. This is a natural number that should only be used internally. A higher revision number corresponds to a newer version of an item. Documents that are to be reviewed should have a version number instead (see section 3.1).

SCMP Software Configuration Management Plan (this document).

SPMP Software Project Management Plan.

SQAP Software Quality Assurance Plan.

SVVP Software Verification and Validation Plan.

Tag
v. Making a tag.
n. A version of a configuration item as stored in the version control system.

Version
String consisting of two natural numbers with a point in between. This is the version number that will be used for documents that are to be reviewed, accepted or approved, internally or externally. Intermediate changes only get a new revision number, but no new version number.

1.4 List of references

Chapter 2

Management

2.1 Organization

The librarians as mentioned in this and other documents are also configuration managers. We use “librarian” as a synonym for “configuration manager”. The SPMP designates Freek van Walderveen as librarian and Joeri de Ruiter as vice librarian.

2.2 Responsibilities

The librarian (or, in case of problems with either the main repository or the librarian himself, the vice librarian) should provide a neatly working environment for configuration management all the time. Any problems should be reported as soon as possible.

The librarians are also responsible for tagging new versions of documents and putting them in the binder as mentioned in chapter 8. Creating and updating document templates is also the responsibility of the librarians.

The (primary) librarian is the first responsible for configuration management, however he may delegate tasks to the vice librarian. Also, whenever the librarian is not available or unreachable, the vice librarian should take over his tasks (such as tagging documents when necessary).

Furthermore, all group members are responsible for their own documents (this includes updating the document status sheet; see also section 2.5), except for tagging versions. They should inform the librarian when such a version is ready to be tagged.

2.3 Interface management

There is no strictly external hardware involved in the project. All services are hosted by the librarians who are thus responsible for them.

GENSO interfacing is a special exception; if any hardware or software interfaces are provided by (for example) external GENSO development teams, Pim Vullers and Stijn Stiefelhagen are those who should be contacted in case of problems or questions.

2.4 SCMP implementation

Contrary to the ESA Software Engineering Standard [2] there will not be a separate SCMP document for each phase of the project. Instead, there is an appendix attached for each phase of
the project. For more information concerning planning of these phases, please refer to the SPMP [3].

2.5 Applicable procedures

Quality and verification related procedures can be found in the SQAP and SVVP [4, 5]. A number of lower level policies are listed below, they are complementary to common sense policies (such as writing useful commit messages).

**Rule of thumb** In principle, all source documents should be kept in the repository, but no derived entities.

There are however a few exceptions to this rule, most notably tags, which should always be filed as PDF versions to guarantee an identical document is available later on.

Another exception is formed by large binary files. These are preferably kept on the FTP server, due to performance issues. This restriction will most likely not interfere with needs of version control, because large binary objects are mostly derived works, of which the source form will be kept in the repository, or copies/backups of external libraries.

**Subversion lock** When there are potentially two team members working on the same part of a file, the first person editing this file is allowed to use a subversion lock to prevent merging problems later. If a file is locked for no apparent reason, breaking a lock is allowed. However, both making and breaking locks should be done sensibly: direct communication between team members is always preferred above the usage of locks.

**Repository subtrees** Inside the trunk tree (see section 3.1), three subtrees are present: doc, src and scratch. The names speak for themselves.

Everyone is allowed and encouraged to put for example notes and other unofficial documents in the scratch directory. Official documents should go in the doc subtree, all in separate directories.

Minutes of all meetings end up in the doc/minutes directory.

This directory structure (especially doc and src) should be used in the tags and branches trees as well.

**Starting a new document** When starting on a new (official) document, the template (located at trunk/doc/template/template.tex) should be copied. This ensures all documents use the same style. To allow subversion to automatically update the revision number inside the document, the subversion property svn:keywords should be set to Revision. This is done automatically when the template is “svn copy”-d to the new document (another approach is to automatically add this property to all new .tex files [1]).

**Quality policies** When committing anything to the trunk tree (except for the scratch directory), files that should be compiled or parsed by some other program (such as \LaTeX, Python or C++ sources or configuration files) should be verified not to contain syntax errors due to which compiling or parsing will fail. All project members are responsible for their own commits. This ensures that the complete system and documentation is buildable at all times.

**Bibliography** (also known as List of references) All documents that are (cross-)referenced from other documents should be added to the trunk/doc/bibliography.bib file. They will be included in the lists of references automatically when necessary (but do not forget to run Bib\LaTeX).
Chapter 2. Management

Tagging Tags are created by the librarians when they are informed of the availability of a new version of a document. The document author(s) are responsible for this notification and updating the document status sheet and document change report (see chapter 5). The librarians are responsible for updating the version information inside the document. For tagging the following actions are taken:

- The document author(s) update the document status sheet and document change report.
- The document author(s) notify the librarians.
- A librarian updates the version information inside the document (general document version and version information in the last document status sheet entry).
- The librarian commits the document.
- The librarian generates a PDF from the document sources, excluding the revision number in the general version.
- The librarian puts the PDF in the appropriate directory inside the tags tree in the version control system.
- The librarian prints the PDF and puts it in the binder.
Chapter 3

Configuration identification

3.1 Naming conventions

Repository overview The repository is split up into the three trees trunk, tags and branches. All development is done in either the trunk or branches trees. In general, only the trunk tree is used. If large (experimental or otherwise deviating) changes need to be made that will break other parts of the system, a branch of trunk tree could be created, in which these changes can first be worked out, before they are merged into the trunk. The tags tree should not be touched by any team member except the librarian, because it contains the official (external) versions of documents and programs. For every version as described below, a copy should be made in the tags tree by one of the librarians.

Document versioning Documents that are still work-in-progress have a revision number. This number is used internally instead of the version number to provide fine-grained identification. When a document is released, a PDF of the document is stored in the tags tree, tagged with a new version number. If, for example, the SCMP is released, a PDF of trunk/doc/scmp/scmp.tex should be stored as tags/doc/scmp/scmp-version.pdf, and the version information inside the document should be updated to reflect the new version. The revision number should be removed from the version information in released documents.

As stated in the list of definitions (section 1.3), a version number always consists of two natural numbers, separated by a point. The first version to be reviewed internally should have version number “0.1”. For internal reviews, the minor number (after the point) is incremented, for external reviews, the major number (before the point) is incremented. Drafts that are not yet ready for review should use version number “0.0”.

Minutes should be named minutes_yyyyymmdd_type-n.tex (in the trunk/doc/minutes directory) where type is the type of meeting (for example customer, adviser) and n is a serial number of the meeting of this type on that day. For progress meetings, no type should be added and for first meetings of a particular type on each day, -0 may be left out.

3.2 Baselines

Baselines are always tagged and present in the tags tree, as well as printed and kept in the binder.

According to the ESA standard [2], new versions of the management documents need to be created for every stage of the project. Because of the small scale of this project it has been decided that the same management documents are used during the course of the project. Information specific for each stage in the project can be found in the appendices.
Chapter 4

Configuration control

4.1 Library control

In case of unreachability of the primary services, the backup facility (provided by the vice librarian) should be able to take over configuration management activities with minimal hassle due to the use of a replicated backup repository. This repository only guarantees read-access, so commits will have to wait until either the main repository is reachable again or some other measures are taken. If not done automatically, team members can switch between the main and backup repository using the following subversion command:

```
svn switch --relocate svn://svn.vanwal.nl/sep svn://blackdemon.org/sep
```

4.2 Media control

This section is not applicable to this project, as all project files are stored on hard drives.

4.3 Change control

Once a document has been approved internally, it is tagged as such. If authors want to make changes to such a document, they should contact the quality assurance manager. He will call for a review meeting in which the changes are approved or rejected. More information regarding the change procedure can be found in the SVVP [5]. When the changes are accepted, a new version of the document will be tagged. According to the SVVP, the addition of appendices to a document also requires an additional review meeting.
Chapter 5

Status accounting

All documents should carry a document status sheet, as provided by the template (see section 2.5). It should contain entries for all tagged versions of a document, including a summary of the changes made since the previous version of the document. A document change report should also be present in every document and should contain a list of sections in which changes are made and the reason for each change. Only changes that were made since the previous version have to be recorded here.

Authors are responsible for updating both the document status sheet and the document change report, except for version information, which will be updated by the librarian tagging the document.
Chapter 6

Tools, techniques and methods

All produced documents and tools must be freely available to any team member. Therefore, a central storage facility has been set up that holds all files and can be accessed by all members. Team members have received the necessary information to reach the provided services and are supposed to know how to work with the tools and techniques used. If not, any other group member will be able to provide assistance. The services provided are listed in the following sections.

6.1 Version control

The version control system used is subversion. This system facilitates all necessary low-level configuration management primitives such as storing and retrieving current and past revisions of configuration items, viewing differences between revisions of a particular configuration item and updating revision numbers after every commit (including the update of in-document revision strings).

A number of subversion hooks are implemented for convenience, safety and quality. The usage thereof is documented in a separate (internal) document [6].

The version control system can be reached via svn://svn.vanwal.nl/sep/.

6.2 File server

The file server (also referred to as FTP server) may be used to store any project related data that is too large or for which it is otherwise inappropriate to keep them in the version control system. It also contains a backup of the course website because of unavailability at times.

The file server can be reached via ftp://sep@vanwal.nl/.

6.3 Wiki and ticket system

A wiki has been set up to provide an easy platform for information exchange between group members. It has no official status, but any team member is allowed and encouraged to put any project related information there. The system will also be used as a knowledge base for GENSO research.

The system also incorporates a ticket system. Whether or not this system will be used will be decided upon when necessary.

The wiki can be reached via http://sep.vanwal.nl/.
Chapter 7

Supplier control

Please refer to SQAP [4] for the demands that are placed on tools supplied by external sources.
Chapter 8

Records collection and retention

A printed copy of all tagged documents and project related documents that carry the client’s signature should be kept in the binder in room HG 5.14. The librarians are responsible for putting new documents in the binder. All documents in the binder are retained for the period of the project.
Appendix A

Software requirements phase

No additional software configuration related procedures for this phase are identified.
Appendix B

Architectural design phase

No additional software configuration related procedures for this phase are identified.
Appendix C

Detailed design phase

For this phase, the src subtree is further subdivided into server, client, stub and common folders. The contents of these folders are managed by the teams working on the respective parts of the system. Automatic testing (building, running unit tests) will be implemented if it does not require too much effort.
Appendix D

Transfer phase

To be written.