



**Software
Configuration
Management Plan
of Paperless GMP
Software**

PROTOCOL NO :

EFFECTIVE DATE :

PAGE NO.:

**Software
Configuration
Management Plan
Of Paperless GMP
Software**

TITLE :	
AUTHORING GROUP :	
DATE :	
SUPERSEDE PROTOCOL NO.:	



**Software
Configuration
Management Plan
of Paperless GMP
Software**

PROTOCOL NO :

EFFECTIVE DATE :

PAGE NO.:

TABLE OF CONTENTS

Sr. No.	Contents	Page No.
1	Identification	
1.1	Document overview	
1.2	Abbreviations and Glossary	
1.2.1	Abbreviations	
1.2.2	Glossary	
1.3	References	
1.3.1	Project References	
1.3.2	Standard and regulatory References	
1.4	Conventions	
2	Organization	
2.1	Configuration management principles	
2.2	Configuration management in a development cycle	
2.3	Configuration management of release	
2.4	Configuration management of prototypes	
2.5	Tasks in development and maintenance	
2.6	Archiving versions	
2.7	Link with bugs and features	
3	Configuration identification	
3.1	Identification rules of configuration items	
3.2	Identification rules of SOUPs	
3.3	Identification rules of installers	
3.4	Identification rules of archives	
3.5	Identification rules of documents	



Software Configuration Management Plan of Paperless GMP Software

PROTOCOL NO :

EFFECTIVE DATE :

PAGE NO.:

1 Identification

This document amplifies the “Configuration management” of the Project Management Plan.

If you instantiate this document, leave empty the §4 in the Project Management Plan and add a reference to this doc.

1.1 Document overview

This document contains the software configuration management plan of software Paperless GMP.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

Add here abbreviations

1.2.2 Glossary

Add here words definitions

1.3 References

1.3.1 Project References

#	Document Identifier	Document Title
[R1]	ID	Add your documents references. One line per document

1.3.2 Standard and regulatory References

#	Document Identifier	Document Title
[STD1]	ID	Add your standards references. One line per document

1.4 Conventions

Typographical convention.

Any other convention.



Software Configuration Management Plan of Paperless GMP Software

PROTOCOL NO :

EFFECTIVE DATE :

PAGE NO.:

2 Organization

Describe the organization in which CM resides.

Eg: The SCM support department, shared by all projects of the company, manages software configuration.

OR

The software configuration is managed by members of the project, with specific tools. Responsibilities are shared between

- The software configuration manager (SCM),
- The project manager,
- The technical manager.

2.1 Configuration management principles

The SCM is done with <your tool: GIT/SVN other>, a SCM tool that has a command line interface and integrates with <your bug tracking tool Redmine/trac/mantis/other> task management tool and the <your IDE Eclipse/other> Integrated Development Environment.

Describe how you manage different versions with branches, forks or other mean offered by your SCM tool.

Example:

A main development branch, called the Master, receives by default all software developments made by the software team. When a new major version is planned (for instance V1.0 or V2.0), a branch is created from the master. This branch is isolated to be tested, fixed, and finally delivered.

Use figures! A small diagram is better than a long explanation

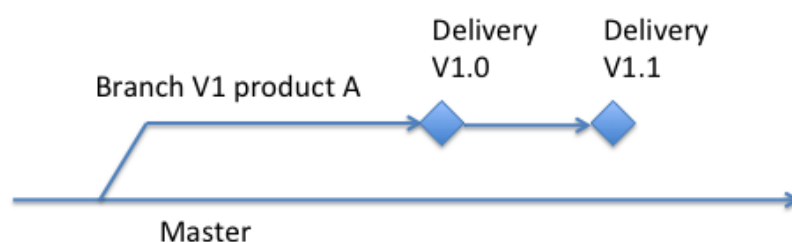


Figure 1 Master and branches in the SCM tool

Describe also how modifications in a branch (eg bugs fixes) can be diff-merged in another branch.

2.2 Configuration management in a development cycle

The changes made by developers during a development cycle are managed by the following method.

Describe how you manage the development cycle, checkout-checkin of each developer, if there is an integration branch. How the branch is merged in the current version at the end of the cycle.



Software Configuration Management Plan of Paperless GMP Software

PROTOCOL NO :

EFFECTIVE DATE :

PAGE NO.:

2.3 Configuration management of releases

Explain how a release is managed in configuration. Is there a fork, a branch, a tag and so on.

2.4 Configuration management of prototypes

If you have prototypes (not ce marked, not fda cleared) that are released to selected users for clinical research or the like, explain how they are managed in configuration.

2.5 Tasks in development and maintenance


The tasks depend on the phase of the software development project or of maintenance. The SCM Manager does the following operations, in the software life-cycle.

Event	Operation
Launching the development of a new product	Creating the source folder structure in the master branch
Deciding to create a major version	Fork of a branch from the current state of the master branch
Releasing a major version	Tagging the current version in its branch. Archiving the tagged version
Releasing a minor version or a patch	Adding a new tag to the current version in the branch Archiving the tagged version
Closing an iteration cycle	Adding a new tag to the current version in the master branch
Other events...	

The software developers update the source code in the branch that corresponds to the state of the software and the type of modification.

List the locations of changes

Type of modification	Location of the modification
Creating a new functionality in the next major version (iteration cycles)	In the master branch
Creating a new functionality in the next minor version	In the branch of the major version.
Modifying an existing functionality in the new minor version	In the branch of the major version.
Fixing a bug in a released version	In the branch of the major version
Fixing a bug in a version in verification	In the branch of the major version in

	Software Configuration Management Plan of Paperless GMP Software	PROTOCOL NO :
		EFFECTIVE DATE :
		PAGE NO.:

phase (not yet released)	verification phase
--------------------------	--------------------

2.6 Archiving versions

Each version is archived on XXX (a server/network URL ...).

The versions are kept in the form of a tree structure, with:

- Source code, configuration files, database scripts,
- Generated binaries and installers,
- Technical documentation,
- ...

2.7 Link with bugs and features

Explain how is made the link between SCM tool and bug tracking tool and tasks.

This is important to explain how the link is made. It ensures the traceability of code changes with tasks/features/bug fixes. This is the main advantage of the integration of tools inside an IDE. At every iteration, it shall be possible to know what tasks were done and which parts of the source code has changed.

3 Configuration identification

3.1 Identification rules of configuration items

The identification of configuration item is:

- <configuration item name>_Vm.n.p

where:

- "Vm " is the major version of the configuration item,
- n is the minor version number,
- p is the incremental minor version number, if necessary.

The version number of the configuration item Vm.n.p starts at V1.0.0.

The number "m" of major version is incremented when substantial modifications are made to the device, for example:


- Updating of the intended use,
- Adding new modules / functionalities,

The number "n" of minor version is incremented when non-substantial modifications are made to the device, for example:

- Adding new functionalities to existing modules,
- Updating the GUI.

The number "p" of incremental minor version is incremented when minor modifications are made to the device, for example:

- Modifying existing functionalities,

	Software Configuration Management Plan of Paperless GMP Software		PROTOCOL NO :
			EFFECTIVE DATE :
			PAGE NO.:

- Minor update of the GUI.

Explain also how versions are identified during iterations.

3.2 Identification rules of SOUPs

SOUPs are identified by:

- The name of the manufacturer,
- The name of the library or software,
- The version if the library or software.

For open-source SOUP without manufacturer name, the name of the open-source project is used.

If a SOUP doesn't have its own identification, the identification rules in section 3.1 are applicable.

3.3 Identification rules of installers

Discard this section if there is no installer

Installers have the same version as the product they install. If an installer installs more than one product, the installer version is the concatenation of each product name and version.

3.4 Identification rules of archives

Explain how archives of §2.6 are identified.

3.5 Identification rules of documents

The identification of documents is described below:

XXX-<document number> Rev. <revision index> [Opt.]

Where:

- XXX is an acronym to identify the project
- " document number " is a incremental in the project,
- "Revision index" designates the approved iteration of the document. The revision index is 01 for the first revision, 02 for the second and so on.
- [Opt.] indicates an optional longer descriptive name.

The revision index is 01 for the first revision, 02 for the second and so on.

Explain also if there is a special rule to identify documents versions during iterations.