

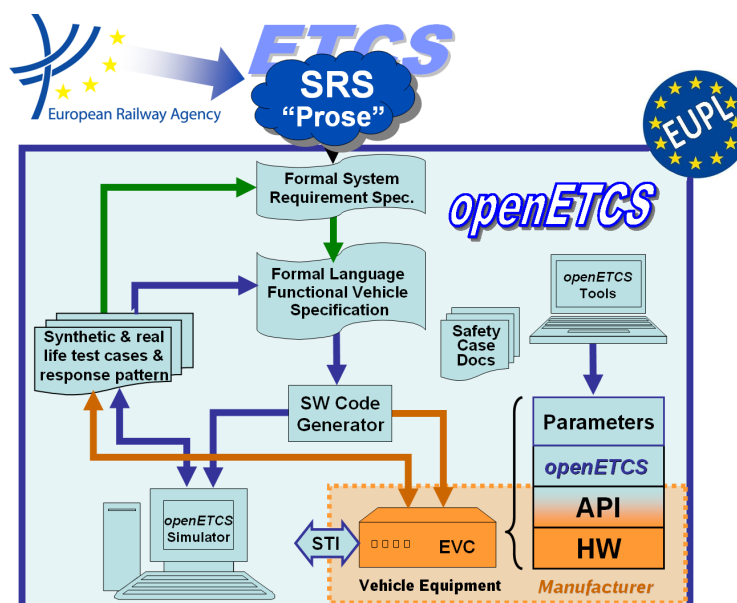
Work-Package 1: "Management"

Software Configuration Management Plan

DRAFT

Jürgen Weiss and Peer Jacobsen

October 29, 2015



Funded by:


 Federal Ministry
 of Education
 and Research

 Région de
 Bruxelles-
 Capitale

 GOBIERNO
 DE ESPAÑA
 MINISTERIO
 DE INDUSTRIA, ENERGÍA
 Y TURISMO

This page is intentionally left blank

Work-Package 1: ``Management''

**OETCS/WP1/D1.3
October 29, 2015**

Software Configuration Management Plan

DRAFT

Jürgen Weiss

ALSTOM Transport Deutschland GmbH
Friedrichstrasse 149
10117 Berlin, Germany

Peer Jacobsen

ALSTOM Transport Deutschland GmbH
Friedrichstrasse 149
10117 Berlin, Germany

Description of work

Prepared for openETCS@ITEA2 Project

Abstract:

Disclaimer: This work is licensed under the "openETCS Open License Terms" (oOLT) dual Licensing: European Union Public Licence (EUPL v.1.1+) AND Creative Commons Attribution-ShareAlike 3.0 -- (cc by-sa 3.0)

THE WORK IS PROVIDED UNDER openETCS OPEN LICENSE TERMS (oOLT) WHICH IS A DUAL LICENSE AGREEMENT INCLUDING THE TERMS OF THE EUROPEAN UNION PUBLIC LICENSE (VERSION 1.1 OR ANY LATER VERSION) AND THE TERMS OF THE CREATIVE COMMONS PUBLIC LICENSE ("CCPL"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS OLT LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

<http://creativecommons.org/licenses/by-sa/3.0/>
<http://joinup.ec.europa.eu/software/page/eupl/licence-eupl>

Contents

Figures and Tables	iv
Document History	v
0.1 Introduction	vi
0.1.1 Purpose	vi
0.1.2 Scope	vi
0.2 scope update	vii
0.2.1 References, Guidelines and Standards	vii
0.2.2 Key Terms, Abbreviations and Acronyms	viii
0.3 SCM Management	x
0.3.1 Organization	x
0.3.2 Responsibilities	x
0.3.3 Applicable Policies, Directives and Procedures	xi
0.4 SCM Activities	xii
0.4.1 Configuration Item Identification	xii
0.4.2 Baseline Identification	xvii
0.4.3 Configuration Control (Change Management)	xx
0.4.4 Configuration Status Accounting	xx
0.4.5 Configuration Evaluation and Review (Audits)	xxi
0.4.6 Interface Control	xxii
0.4.7 Subcontractor/Vendor Control	xxii
0.4.8 Release Management and Delivery	xxiii
0.5 SCM Schedules	xxiv
0.6 SCM Resources	xxv
0.7 SCMP Maintenance	xxvi
Bibliography	xxviii

Figures and Tables

Figures

Figure 1. Overview of SCM activities.....	xii
Figure 2. Signalling system CI example	xiii
Figure 3. CI identification and selection process	xiv
Figure 4. Baseline CI identification and selection process	xix

Tables

Table 1. Document history.....	v
Table 2. References.....	vii
Table 3. Guidelines.....	viii
Table 4. Key terms.....	viii
Table 5. Abbreviations and acronyms	ix
Table 6. Responsibilities.....	x
Table 7. CI checklist	xv
Table 8. CI attributes	xvi
Table 9. Defined baselines.....	xvii
Table 10. SCM resources	xxv

Document History

Table 1. Document history

Version	Date	Chapters modified	Reason	Name
0.0.0	09/08/2013	All New	Frame and Basics definition	Jürgen Weiss (ALSTOM)
0.0.1	13/02/2014	All	Changed structure and content to comply with IEEE Std 828-2005	Peer Jacobsen (ALSTOM)
0.0.2	23/10/2015	All	Remarks and request for review	F. Cochetti (ALSTOM)

0.1 Introduction

This chapter introduces the concept of Software Configuration Management (SCM), describes the purpose and scope of this document and defines references, guidelines and standards as well as key terms, abbreviations and acronyms.

0.1.1 Purpose

This document is the Software Configuration Management Plan (SCMP) for the openETCS project. It defines the methods and tools to identify and control the software and its functional attributes throughout its development and use. It also documents how and when these methods and tools are to be used, states who is responsible, and specifies the required resources.

The overall purpose is to maintain software integrity and configuration traceability during the entire software life cycle (including development as well as maintenance). The intended audience of the SCMP are persons involved in or responsible for this process. Roles and responsibilities as well as respective activities are defined or referred to in the corresponding sections of this SCMP (see sections [0.3.2](#) and [0.4](#)).

0.1.2 Scope

This SCMP applies to the entire life cycle of critical software as well as to noncritical software and software already developed within the context of the openETCS project. It provides technical and supervising direction to ensure that SCM activities are successfully implemented for the openETCS project.

SCM includes the following activities:

- Identification and definition of configuration items and baselines.
- Review, approval, and control of changes.
- Tracking and reporting of changes.
- Audits and reviews of the software product in development.
- Control of interface documentation.
- Management of software release and delivery activities.

To provide all necessary information, the SCMP is structured into the following parts:

Note: [This is the structure given in IEEE 828-2005.](#) *noted*

- Introduction: Describes the purpose and scope of the SCMP. Also lists reference documents, guidelines and standards as well as key terms, abbreviations and acronyms.
- SCM Management (see section [0.3](#)): Describes authorities and responsibilities related to managing and implementing SCM activities (i.e. who).
- SCM Activities (see section [0.4](#)): Describes the SCM activities required for the openETCS project (i.e. what).

- SCM Schedules (see section 0.5): Describes the coordination of SCM activities with other activities within the openETCS project (i.e. when).
- SCM Resources (see section 0.6): Describes tools as well as physical and human resources required for SCM activities (i.e. how).
- SCMP Maintenance (see section 0.7): Describes how the SCMP will be updated while in effect.

0.2 scope update

In the current project phase the main Software delivery is called *SCADE Model*. This artifact is managed as an unitary and generic software model. The Software model is delivered according to a predefined release scheme based on a progressive iteration sequence derived from a selection of use cases sieved to provide a consistent functional scope. The generic model does not require a CI register being managed as an unique item. Standard synchronization of partial deliveries are managed according to a periodic updates of the delivery plan. For a detailed description of the roles and responsibilities reference see [1]. Currently the WP3 develops the generic model, which is conceived as platform independent, deferring the hardware and implementation specific issues to a later stage by WP5 and maybe by other teams adopting similar exploitation concepts in form of demonstrators or even for reuse on EVC platforms. To the purpose of a traceable neutralization and abstraction a vendor independent API concept was adopted and In later phase, where platform specific issue are expected it is recommended to implement the configuration management Possibly a SCMP and CI register could support team(s) working on the applications demonstrators. This items should be evaluated and managed in WP5 and maybe even as an indication for later use of the artifacts outside from current OpenEtcS project. If this plan is supposed to provide a better control of the different artifacts. The review from WP5 would provide the correct feedback. In the following the review items indicated as still open should be reviewed by WP5 teams with indication of preferred solution.

0.2.1 References, Guidelines and Standards

In general, the references, guidelines and standards listed in the Project Quality Assurance Plan (QA Plan) also apply to the SCMP (see QA Plan section 1.5). This section of the SCMP only lists references, guidelines and standards that specifically apply to the SCMP and are not listed in the QA Plan.

0.2.1.1 References

The following table lists reference documents that provide additional information relevant to the SCMP:

Table 2. References

Internal Code	Name	Version/ Edition/ Date	Repository	Responsible
[2]	Quality Assurance Plan	0.10.6	governance	QA Manager
[3]	Change/Problem Management Process	0.1.0	governance	QA Manager
[4]	Full Project Proposal	3.0.2	management	n/a
[5]	Software Release and Deployment Plan	n/a	n/a	n/a

Open: Figure out how to add break in table header. Fill gaps. Additional relevant docs? Closed:
It can see relevant literature

0.2.1.2 Guidelines

The following table lists guidelines that have been followed in the creation of the SCMP:

Table 3. Guidelines

Internal Code	Name	Version/ Edition/ Date	Repository	Responsible
[6]	IEEE Std 828 - Standard for Software Configuration Management Plans	IEEE STD 828-2005	n/a	n/a

0.2.1.3 Standards

There are no standards that apply specifically to the SCMP. Standards that apply to the entire openETCS project are listed in the QA Plan (see QA Plan section 1.5).

0.2.2 Key Terms, Abbreviations and Acronyms

A central glossary is part of the openETCS project (see QA Plan section 1.6.1). The following sections only list important key terms, abbreviations and acronyms that are used in this SCMP.

Open: Not sure what to refer to here. Do we have a central glossary yet? see [1] and refer to Jan Welte closed

0.2.2.1 Key Terms

Table 4. Key terms

Term	Meaning

0.2.2.2 Abbreviations and Acronyms

The following table lists abbreviations and acronyms used in this SCMP:

Open: Review when content of doc is fixed/final. open to review

Table 5. Abbreviations and acronyms

Abbreviation/Acronym	Meaning
CI	Configuration Item
CSA	Configuration Status Accounting
DMI	Driver Machine Interface
hline DRU	Data Recording Unit
hline ETCS	European Train Control System
hline EVC	European Vital Computer
hline FCA	Functional Configuration Audit
FPP	Full Project Proposal
n/a	Not applicable
PCA	Physical Configuration Audit
QA Plan	Project Quality Assurance Plan
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SRS	System Requirements Specification
V & V	Verification and Validation
WP	Work Package

0.3 SCM Management

This chapter describes the allocation and management of authority and responsibility regarding the SCM activities for the openETCS project.

0.3.1 Organization

The following table provides the organizational units that participate in or are responsible for SCM activities within the openETCS project:

Not sure if it makes sense to list the WPs. Review once activities etc. are defined. removed
Closed

The openETCS project structure is described in more detail in the QA Plan (see QA Plan chapter 2). The individual work packages are described in detail in the Full Project Proposal (see FPP chapter 5).

0.3.2 Responsibilities

The following table matches responsibilities for the individual SCM activities that have been defined for the openETCS project (see chapter 0.4) to organizational units or specific positions:

Open: Review once activities etc. are defined. Match to organizational units. the list roles is shown in the following template, they are not needed and not assigned in the OpenEtc project as it should provide a proof of concept open to review

Table 6. Responsibilities

SCM Activity	Responsible
Responsible for monitoring that the SCMP is implemented and followed. Select and approve CIs and baselines.	QA Manager (WP1 Task Leader/Project Leader QA)
System configuration owner and responsible for the SCMP. Select and approve CIs and baselines.	Configuration Manager
Initiate CI identification and selection process in work package. Select and approve CIs and baselines.	Work Package Leader
Responsible for monitoring content and implementation of the SCMP.	Assessor
Responsible for ensuring that software and derived specifications and requirements are under system configuration and change management control.	Requirement Manager
Responsible for ensuring that the design documents are under system configuration and change management control.	Designer
Responsible for maintaining the generated or modified data/codes/models under system configuration and change management control.	Implementer

The individual roles are described in more detail in the QA Plan (see QA Plan Appendices CAT1 to CAT 3).

0.3.3 Applicable Policies, Directives and Procedures

As the organizational unit responsible for Quality Assurance, WP1 is responsible for the overall SCM process. Within WP1, the role Quality Assurance Manager is responsible for implementation of the SCMP and corresponding SCM activities as well as identification, assessment and management of risks related to SCM activities.

In addition, the role Configuration Manager is responsible for the content of the SCMP, the definition of SCM activities and the content of the Configuration Item Register.

Independent monitoring as well as independent identification and assessment of risks related to SCM activities is the responsibility of the role Assessor.

Open: Org unit for Assessor? closed assessment is currently done in WP4

0.4 SCM Activities

This chapter describes the functions and tasks that are necessary to manage the configuration of the software system. It includes technical as well management-related SCM activities that are grouped as follows:

- Configuration item identification
- Baseline identification
- Configuration control
- Configuration status accounting
- Configuration evaluation and review
- Interface control
- Release management and delivery

The following figure gives an overview of how these SCM activities function together:

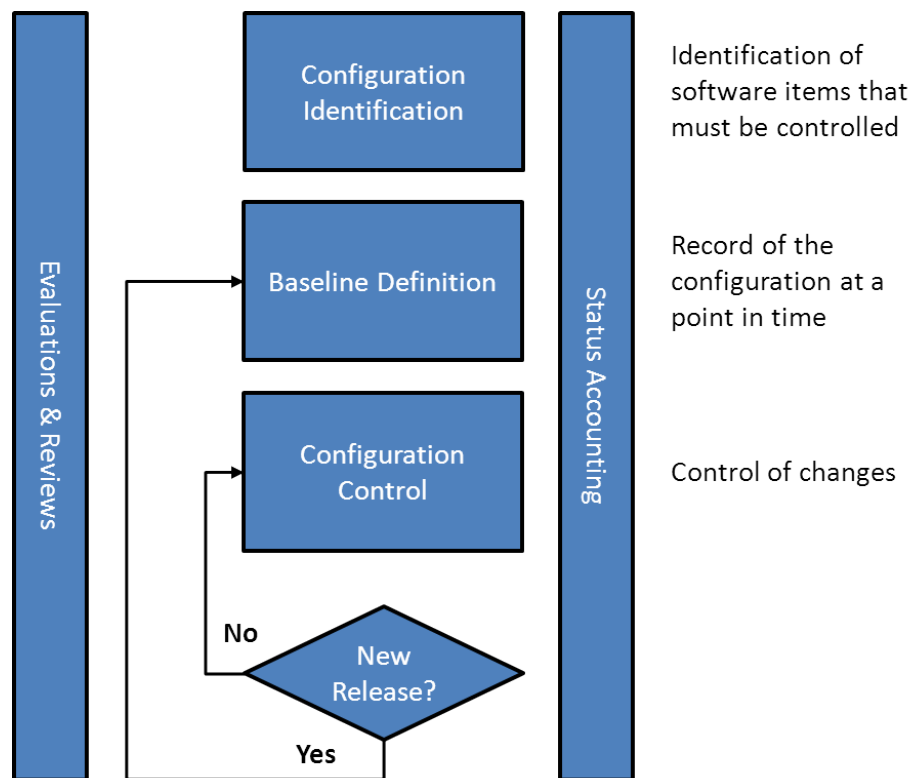


Figure 1. Overview of SCM activities

0.4.1 Configuration Item Identification

Configuration identification establishes and maintains the definitive basis for configuration control of a system and its configuration items (CIs). It defines how CIs are selected, grouped, classified and defined by physical and functional characteristics. The goal is to ensure that all relevant CIs are properly identified, documented and tracked throughout their life cycle.

A CI is an entity that is subject to configuration control. CIs differ in complexity, size and type and encompass both the output of the development process (e.g. code, databases, test plans) as well as elements of the support environment (e.g. compilers, programming tools, test beds).

Open: Not sure if the above is right. Maybe we only want to control CIs that are part of a baseline? open to review : should be decided at a later stage depending on real needs that could arise in follow ups.

In a complex system such as the openETCS software, CIs are structured hierarchically, meaning that complex high level or so-called aggregate CIs can be subdivided into several subordinate CIs that are easier to document and control. Because it is formed of several CIs, an aggregate CI can be very similar to a baseline. The key difference is that an aggregate CI is a logical and/or convenient grouping of CIs, whereas a baseline represents a deliverable (see also section 0.4.2).

The following figure provides an example of CIs in a signalling system:

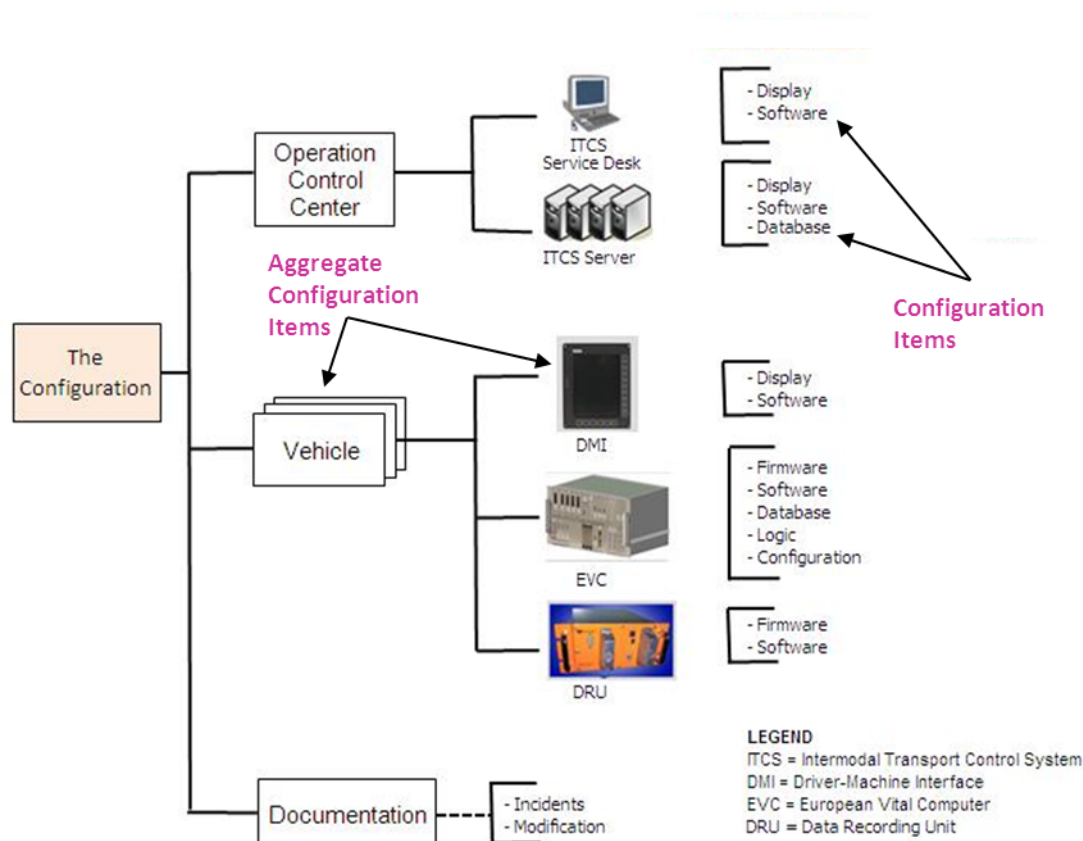


Figure 2. Signalling system CI example

The following sections describe how CIs are identified and managed in the openETCS project.

0.4.1.1 CI Identification and Selection Process

The identification and selection of CIs is an ongoing process that relies on the judgment and recommendations of people involved at all levels of the openETCS project. In most cases, the process will begin in the Scrum Teams of the individual work packages of the openETCS project. The following figure gives a general overview of the CI identification and selection process:

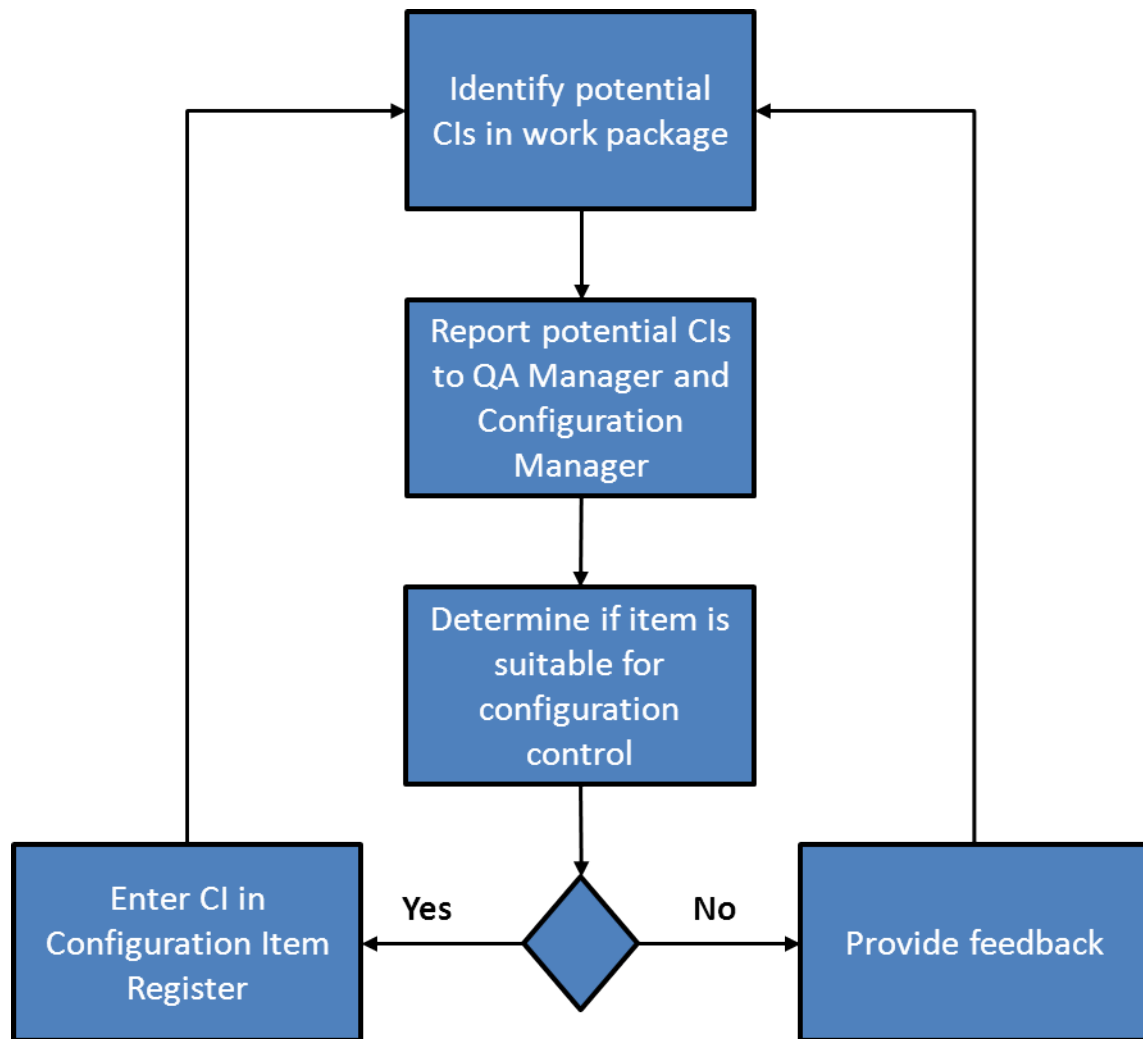


Figure 3. CI identification and selection process

Process description:

1. Work Package Leaders are responsible for starting and supervising the identification and selection of potential CIs in their WP. The identification and selection itself should involve people who are close to the subject matter (e.g. Scrum Teams). Criteria and a checklist for identifying potential CIs are provided in section 0.4.1.2.
2. Potential CIs are reported to the QA Manager and the Configuration Manager.
3. QA Manager and Configuration Manager provide feedback regarding the suitability of the potential CI for configuration control. They also coordinate the decision process with the involved persons in the WP that has proposed the CI.
4. If the CI is suitable for configuration control, the Configuration Manager enters it in the Configuration Item Register. Attributes that must be provided for each CI are given in section 0.4.1.3.

Open: Define CI Register. Where are we planning to enter and maintain CI information? Open to review should be decided in a later stage depending on real needs that could arise in follow ups.

0.4.1.2 CI Identification and Selection Criteria

The following general criteria can be used as a guideline when identifying and selecting CIs:

- Designating a component as a CI increases its visibility and control within the project. This is essential for system critical or high risk components.
- Major functional design components should be designated as (aggregate) CIs. These components form a good starting point for the initial selection.
- Complex (aggregate) CIs can be subdivided into individual CIs. This will likely happen during the design and development phases. Subordinate CIs should be functionally inter-related.
- High level (aggregate) CIs should be separable entities that implement at least one function.
- Operational software and support software should be differentiated as separate CIs.
- Selecting a large number of CIs can lead to delays and increase costs.
- Selecting very few CIs can mean that the individual CIs are unnecessarily complex and difficult to document and control.

The following table can be used as a checklist to determine if an item should be designated as a CI (if there are more “yes” answers than “no” answers, it should probably be a CI):

Table 7. CI checklist

Question	Yes	No
Is the item critical or high risk? Does failure of the item have a significant impact?		
Does the item implement critical functionality (e.g. collision avoidance)? Would control and verification of the functionality be increased by CI status?		
Does the item require changes in design or development?		
Is the item computer hardware or software?		
Does the item include untested technology?		
Does the item have an interface with another CI?		
Can the item be easily identified as an individual, controlled item?		
Is it necessary to track the configuration of the item over its lifecycle?		
Can or must the item be tested independently?		
Is the item released to the customer as a unit?		
Can the item be replaced as a unit?		
Does the item perform a meaningful stand-alone function?		

0.4.1.3 CI Attributes

To ensure that a CI can be properly identified and traced by configuration control, it must be entered in the Configuration Item Register.

Open: Where do we want to store CI information? Github? seems reasonable to have a version controlled repository related to the Open ETCS project structure: open to review

The following table specifies which attributes should be available or are required for each CI:

Table 8. CI attributes

Attribute	Description
ID	Unique identifier
Name	Name of the CI
Description	Brief description of the CI
Owner	Person in charge of the CI
Class	- Category (e.g. hardware, software, ...) - Type (e.g. server, source code)
Manufacturer Information	- Manufacturer name - Serial number - License number/contract
Version	Version of the CI (must correspond to Modification History)
Modification History	- CI creation date - CI modifications (including date, description, person in charge)
Location	Describes where the CI can be accessed (e.g. medium and storage location, server and file path,...)
Status	- Present status of CI (e.g. active, out of operation,...) - Status history (including status changes, description, date)
CI Relationships	The CI's parent or subordinate CIs, if the CI is part of an aggregate CI
Interfaces	The interfaces the CI is connected to
Baseline	The baseline to which the CI belongs (must refer to a specific CI version)

Open: Review. Additional attributes? : open to review

0.4.1.4 CI Storage and Access

In addition to entering CI information in the Configuration Item Register, all CIs must be physically placed under configuration control. This means that procedures (including roles and responsibilities) have to be established for storage, retrieval and reproduction.

The following has to be specified (e.g. per CI class):

- Format
- Location
- Labelling
- Reception and inspection (including documentation)
- Access control
- Tracking of controlled copies

Open: Define. Will we simply keep everything in Github? open to review Github is the recognized repository and could be maintained

0.4.2 Baseline Identification

At its core, a baseline is a snapshot of the software configuration or a defined part of the software configuration at a significant point in the software life cycle. This snapshot can then be used for comparison and as a basis for tracking and verifying changes in the configuration. Baselines are essential to maintaining software integrity.

As a general rule, a baseline must represent a defined, tested/reviewed and approved deliverable (e.g. the software requirements documentation or a stable release version). A baseline is formed by one or more CIs and, once defined and approved, is subject to configuration control. CIs that are part of a baseline are called baselined CIs (Note: CIs are sometimes also called baselined as soon as they are entered into an Configuration Item Register and become subject to configuration control).

0.4.2.1 Defined Baselines

The baselines for the openETCS project represent the milestones from requirements definition to release versions of the product, which is also reflected in the way the project is structured into work packages. The following table gives an overview of the defined baselines:

Table 9. Defined baselines

Baseline	Contained CIs
SRS	- Requirements documentation - Approved SRS tools
Model/Code	- Code - Models - Approved model/code tools
Safety	- Safety documentation - Approved safety tools
V&V	- Test cases - Use cases - Test results - Test environment - Approved V&V tools

Table 9 -- continued from previous page

Baseline	Contained CIs
Product	- All of the above
	- Release documentation
	- Release tools
Archive	- All of the above

0.4.2.2 Baseline CI Identification and Selection Process

The process of identifying and selecting which CIs (including the CI version) form a baseline must be carried out and approved at a sufficiently high level of the openETCS project, i.e. by the Work Package Leaders together with QA Manager and Configuration Manager.

The following figure gives a general overview of the process to identify and select which CIs form a baseline:

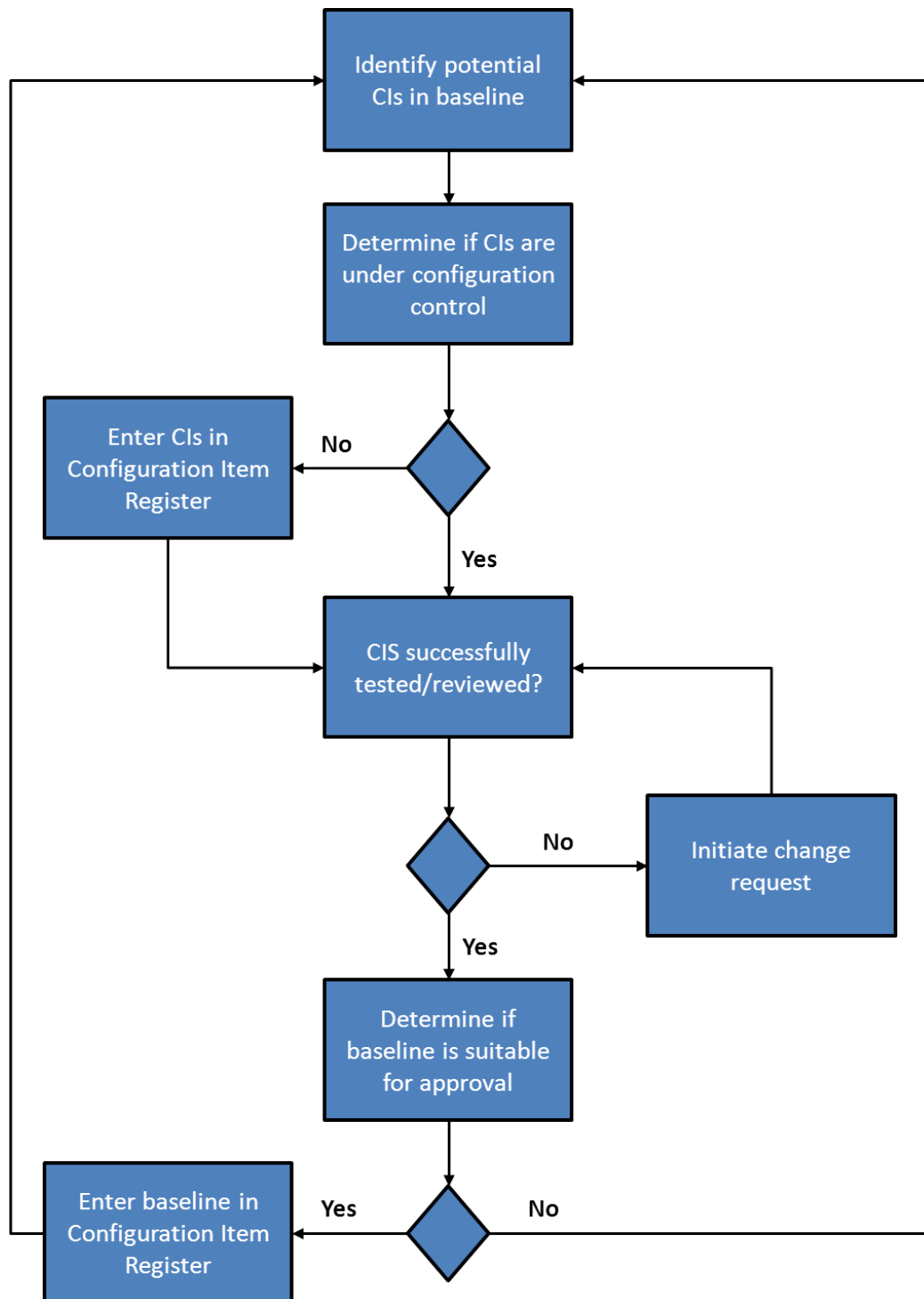


Figure 4. Baseline CI identification and selection process

Process description:

1. The Work Package Leaders together with the QA Manager and the Configuration Manager are responsible for identifying the CIs that belong to a specific baseline.
2. If a baseline includes a CI that has not been entered into the Configuration Item Register (i.e. placed under configuration control), the CI has to go through the CI identification and selection process (see section 0.4.1.1).

3. Before the baseline can be approved, its CIs have to be tested and reviewed individually and as part of the baseline as a whole. If a CI is unsuitable, a change request can be initiated (see section 0.4.3).
4. The Work Package Leaders together with the QA Manager and the Configuration Manager decide if the baseline can be approved. If the baseline is approved, the Configuration Item Register is updated accordingly.

Open: Are we tracking baselines through the CI Register? No closed

0.4.3 Configuration Control (Change Management)

Configuration control ensures that CIs are only changed after evaluation and approval. For the openETCS project, a corresponding change management process has been defined separately (see document Change/Problem Management Process in section 0.2.1.1).

0.4.4 Configuration Status Accounting

Configuration Status Accounting (CSA) is the process of tracking CIs through the steps in their evolution. Regular reports on the status and history of the CIs in the openETCS project are created to deliver the data required to measure project progress. The following data is tracked for each CI:

Open: The QA Plan lists a Software Assessment Plan and Software Assessment Report. Are the requirements below included there, meaning we can simply refer to it here? Yes Closed

- Approved versions and revisions
- Status of change requests
- Implementation status of approved changes
- Evaluations (Audits)

Note: IEEE 828-2005 names the above as the minimum. Noted

If necessary, this data makes it possible to rebuild previous baselines. It is also used to track change requests and document the results of evaluations and reviews.

Open: The following has to be documented in the SCMP:

- What data elements and SCM metrics are tracked and reported for baselines and changes? SCADE Model, SCDE generated code Integration code Test environment model test results
- What types of status accounting reports are generated (and their frequency)? sprint reviews on regular basis
- How is information collected, stored, processed, reported, and protected from loss?
- How is access to the status data is controlled?

open to further review

0.4.5 Configuration Evaluation and Review (Audits)

CI must be evaluated and reviewed (audited) to verify conformance to requirements defined, for example, in the specification documentation of the openETCS project or applicable standards. Audits of CIs must determine that a CI fulfills its required physical and functional characteristics.

Open: The QA Plan lists a Software Verification and Validation Plan (as well as Test, Validation and Verification Reports) and a Software Assessment Plan (and Software Assessment Report). Are those documents going to include the requirements below, meaning we can simply refer to them here? Yes Closed

0.4.5.1 Physical Configuration Audit

A Physical Configuration Audit (PCA) must be carried out prior to a software release or software update (i.e. before a new baseline). It includes all CIs that are part of the corresponding baseline. The PCA is performed to determine that all CIs (including documentation) are present in the correct version and match the data from the corresponding configuration status reports.

An unsuccessful audit (non-conformance) means that the software release or software update cannot take place.

Open: Roles and responsibilities? Process and schedule? Approval criteria and actions that occur upon approval? see [1] closed

0.4.5.2 Functional Configuration Audit

A Functional Configuration Audit (FCA) is carried out for each CI (including documentation) to determine that it has been tested or inspected and fulfills the functions defined in the corresponding openETCS specifications.

An unsuccessful audit (non-conformance) means that the CI cannot be released (e.g. to be part of a baseline).

Open: Roles and responsibilities? Process and schedule? Approval criteria and actions that occur upon approval? see [1] closed

0.4.5.3 Audit Reports

Every audit must be documented in a corresponding audit report with at least the following information:

- Auditor
- Date
- Audit reason/objective
- CIs audited
- Result for each CI
- Overall result

- Recipient of audit report

Open: Others? see assesment report on github (internal-assesment repository)open to further review

0.4.5.4 Non-Conformance Follow-Up

If a CI fails an audit, corresponding corrective actions have to be planned and implemented before a follow-up audit can be scheduled and carried out.

Open: Roles and responsibilities? Process? see [1]. for CI to be decided in later stage. closed

0.4.6 Interface Control

CIs may interface with other items that are outside the scope of the SCMP. The interfacing items may remain outside of configuration control, but the interfaces have to be identified and controlled.

Open: The QA Plan lists a Software Interface Specification. Does that document include the requirements below, meaning we can refer to it here? open to review

The following information has to be defined for each interface:

- Interface nature/type
- Affected CIs
- Process to control interface code, documentation and data
- Process to approve and release the interface

Open: Roles and responsibilities? Process? Do we track/control interfaces as part of individual CIs or as separate items? Is a separate Interface Management Plan planned? No Closed

0.4.7 Subcontractor/Vendor Control

Software items developed outside the openETCS project must be placed under configuration management. This includes software developed by subcontractors or software acquired in finished form from another source. It may not be possible to treat CIs that are not developed internally the exact same way as CIs developed as part of the openETCS project.

If necessary, the following must be determined for subcontractor/vendor CIs:

- SCM requirements the subcontractor has to fulfill
- Compliance monitoring of the subcontractor
- Evaluations and reviews of subcontractor CIs
- Process to test, verify, accept and integrate subcontractor CIs into openETCS project
- Handling of proprietary CIs (e.g. regarding copyright and royalties)

- Handling of changes (e.g. participation of subcontractor/vendor)

Open: Roles and responsibilities? Process? to be managed at a later stage, open to further review

0.4.8 Release Management and Delivery

The build, release and delivery of software products and documentation by the openETCS project must be formally controlled. For the openETCS project, a corresponding Software Release and Deployment Plan has been defined separately (see corresponding reference in section 0.2.1.1).

0.5 SCM Schedules

Because there may be dependencies between the individual SCM activities or relationships between SCM activities and other project activities (e.g. milestones), some SCM activities have to be carried out in sequence or according to a schedule.

Open: Do we have dependencies that require sequences/schedules? Are there milestones/events that require SCM activities? Is there some kind of timeline or something we could refer to?
No closed

0.6 SCM Resources

Carrying out the SCM activities defined in this SCMP requires resources. These resources may themselves be CIs that have to be placed under configuration control.

The following table lists the resources specified for each SCM activity:

Table 10. SCM resources

SCM Activity	Tools/Techniques/Equipment	Personnel	Training
CI Identification			
Baseline Identification			
Configuration Control			
Configuration Status Accounting			
Configuration Evaluation and Review			
Interface Control			
Subcontractor/Vendor Control			
Release Management and Delivery			

Open: Define resources? Can we also refer to corresponding other documents like QA Plan or Change Process? to be demanded to a later stage in case of follow ups: open to further review

0.7 SCMP Maintenance

SCM planning must continue throughout the life cycle of the openETCS project. This means the SCMP itself will be treated as a CI to ensure that the following information remains up-to-date:

- SCM responsibilities
- SCMP updates (including change management/approval)
- SCMP distribution

Open: Why do we have an additional "References" section that basically repeats what is already listed in the "References, Guidelines and Standards" section? Is intended as an extensive reference including documents papers etc not only guidelines and references : Closed

Bibliography

- [1] openETCS. *openETCS Architecture and Design Specification*, 2014. <https://github.com/openETCS/modeling/tree/master/openETCS%20ArchitectureAndDesign/D3.5.3>.
- [2] Izaskun de la Torre. Quality assurance plan.
- [3] Izaskun de la Torre. Change/problem management process.
- [4] openETCS Project Team. openETCS full project proposal, 2013.
- [5] n/a. Software release and deployment plan.
- [6] IEEE 828 standard for software configuration management plan, August 2005.
- [7] CENELEC. EN50128 railway applications - communication, signalling and processing systems - software for railway control and protection systems, June 2011.
- [8] CENELEC. EN50126 railway applications - the specification and demonstration of reliability, availability, maintainability and safety (rams), March 2000.
- [9] CENELEC. EN50129 railway applications - communication, signalling and processing systems - safety related electronic systems for signalling, December 2003.
- [10] CENELEC. EN61508 functional safety of electrical/electronic/programmable electronic safety-related systems, December 2010.
- [11] European Commission. 2008/57/EC - interoperability directive, June 2008.
- [12] European Commission. 2011/18/EU - interoperability directive, March 2011. change of annexes.
- [13] European Commission. 20012/88/EU - TSI CCS commission decision of 25 january 2012 on the technical specification for interoperability relating to the control-command and signalling subsystems of the trans-european rail system, March 2011.
- [14] Roman Pichler. *Agile Productmanagement with Scrum*. Addison-Wesley, 2010.
- [15] Ken Schwaber and Mike Beedle. *Agile Software Development with Scrum*. Prentice Hall, 2002.
- [16] Jean-Francois Monin. *Understanding Formal Methods*. Springer, 2008.
- [17] ERTMS. UNISIG subset 026, system requirements specification - baseline 3, January 2010.
- [18] ERTMS. UNISIG subset 036, fffis for eurobalise, January 2010.
- [19] openETCS. *openETCS SCADE model*, 2014. <https://github.com/openETCS/modeling/tree/master/model/Scade/System>.
- [20] ERTMS. UNISIG subset 076.
- [21] ERTMS. UNISIG subset 023, glossary of terms and abbreviations.
- [22] Izaskun de la Torre. Change/problem management process.

- [23] International Organization for Standardization.
- [24] Ainhoa Gracia. Review process.
- [25] Ainhoa Gracia. Revision process.
- [26] Bernd Hekele. Grieving handling process.
- [27] Bernd Hekele. Project co-operation agreement.
- [28] Bernd Hekele. Contribution guidelines.
- [29] Bernd Hekele. Openetcs ip policy.
- [30] Jonas Helming. Committer election guideline.
- [31] Jonas Helming. Committer approvement process.
- [32] Bernd Hekele. openetcs internal assessment.
- [33] Jonas Helming. openetcs development process.
- [34] Eclipse Community. Eclipse development process.
- [35] Hardi Hungar Marc Behrens. Openetcs validation & verification plan.
- [36] Izaskun de la Torre. Qa plan backlog.
- [37] Stefan Rieger. openETCS publishing guideline.