The Trans-European Conventional Rail System

“TELEMATIC APPLICATIONS”
subsystem for Freight Services

CONFIGURATION MANAGEMENT

Concept
and
Generic Requirements
## History

<table>
<thead>
<tr>
<th>ED</th>
<th>Date</th>
<th>Author</th>
<th>Change Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>12.10.2004</td>
<td>R.Berger</td>
<td>First edition</td>
</tr>
</tbody>
</table>
Intended Audience

This document is one of the references to be used by designers and engineers responsible for the proper realisation and implementation of the TAF TSI requirements regarding the common components. It is also a reference for the operating entity of the common part during the life cycle of the system.

Evolution of this Document

Distribution:

Any new version of the document incorporated in any new version of annex E of the TAF TSI approved by the EC in accordance of the procedure set out in Article 21(2) of the Directive 2001/16/EC shall be distributed to:

- AEIF
- EC
- DGTREN
- European Railway Agency

AEIF SSG Telematic Applications
Brussels
DGTREN
Brussels
European Railway Agency
Valenciennes

The document shall be distributed by e-mail in MS-Word format or in PDF Format and published on the AEIF Web-site www.aeif.org.

Configuration Management:

Any change requests on the content of this document shall be reported to the TAF TSI responsible member at AEIF and at the European Railway Agency.
# Table of Contents

0. **Introduction** ..................................................................................................... 6  
  0.1. Purpose ............................................................................................................... 6  
  0.2. References .......................................................................................................... 7  
   0.2.1. Standards and Guidelines .............................................................................. 7  
   0.2.2. Other References ........................................................................................... 7  
  0.3. Definitions and Abbreviations ............................................................................ 8  

1. **Configuration Management (CM)** ................................................................. 11  
   1.1 CM purpose ......................................................................................................... 11  
   1.2 CM Areas ............................................................................................................. 11  
    1.2.1 Configuration Identification .......................................................................... 12  
    1.2.2 Configuration Change Control .................................................................... 14  
    1.2.3 Configuration Status Accounting ............................................................... 16  
    1.2.4 Configuration Audits and Reviews .............................................................. 16  
    1.2.5 Software Baseline Library Establishment and Maintenance ..................... 17  
    1.2.6 Product Build and Release .......................................................................... 19  

2 **Configuration Management Process** ............................................................. 20  
   2.1 CM Process Description ...................................................................................... 20  
   2.2 CM Version Description Document .................................................................. 23  
   2.3 CM Planning ....................................................................................................... 25  
    2.3.1 Purpose .......................................................................................................... 25  
    2.3.2 Procedure ...................................................................................................... 25  

3 **Change / Problem Management Process** ....................................................... 29  
   3.1 Definition and Scope ......................................................................................... 29  
   3.2 Objectives ......................................................................................................... 29  
   3.3 Organisational Entities ....................................................................................... 30  
   3.4 General Process .................................................................................................. 31  
   3.5 Life-Time Aspects ............................................................................................. 32  
    3.5.1 Change Factors ............................................................................................ 32  
    3.5.2 Handling of Changes .................................................................................. 33  
   3.6 Release Management Organisation ................................................................. 34  
    3.6.1 Change Review Team ................................................................................. 34  
    3.6.2 Change Control Board ............................................................................... 34  
    3.6.3 TAF TSI Project Teams .............................................................................. 36  
    3.6.4 Supplier ....................................................................................................... 37  
    3.6.5 Other related Projects ................................................................................ 37  
   3.7 Decision and Control Mechanism ..................................................................... 38  
    3.7.1 Decision and Escalation Authorities ......................................................... 38  
    3.7.2 Control Mechanism ................................................................................... 38  
   3.8 The Release Plan ............................................................................................... 39  
   3.10 Procedure for Change Implementation .......................................................... 40  

Flow Chart for the Change Process ........................................................................... 39  

Configuration Management 4 / 72  
Version 1.0 12.10.2004  
Orig. Language : EN
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.10.1 Roles and Responsibilities</td>
<td>40</td>
</tr>
<tr>
<td>3.10.2 Activities</td>
<td>40</td>
</tr>
<tr>
<td>3.10.3 Checklists</td>
<td>43</td>
</tr>
<tr>
<td>3.10.4 State Machine for fault processing</td>
<td>43</td>
</tr>
<tr>
<td>3.11 Example of CR/FR Template</td>
<td>44</td>
</tr>
<tr>
<td>4 Baseline Management Process</td>
<td>46</td>
</tr>
<tr>
<td>4.1 Preface</td>
<td>46</td>
</tr>
<tr>
<td>4.2 Scope</td>
<td>47</td>
</tr>
<tr>
<td>4.3 Attributes of a baseline</td>
<td>48</td>
</tr>
<tr>
<td>4.4 Process Description</td>
<td>49</td>
</tr>
<tr>
<td>5 Some special CM Procedures</td>
<td>52</td>
</tr>
<tr>
<td>5.1 Procedure for Configuration Identification</td>
<td>52</td>
</tr>
<tr>
<td>5.1.1 Background/Purpose</td>
<td>52</td>
</tr>
<tr>
<td>5.1.2 Roles and Responsibilities</td>
<td>52</td>
</tr>
<tr>
<td>5.1.3 Activities</td>
<td>52</td>
</tr>
<tr>
<td>5.1.4 Checklists</td>
<td>56</td>
</tr>
<tr>
<td>5.2 Procedure for Configuration Status Accounting</td>
<td>56</td>
</tr>
<tr>
<td>5.2.1 Background/Purpose</td>
<td>56</td>
</tr>
<tr>
<td>5.2.2 Roles and Responsibilities</td>
<td>56</td>
</tr>
<tr>
<td>5.2.3 Activities</td>
<td>56</td>
</tr>
<tr>
<td>5.2.4 Checklists</td>
<td>58</td>
</tr>
<tr>
<td>5.3 Procedure for Conducting Configuration Audits</td>
<td>58</td>
</tr>
<tr>
<td>5.3.1 Background/Purpose</td>
<td>58</td>
</tr>
<tr>
<td>5.3.2 Roles and responsibilities</td>
<td>59</td>
</tr>
<tr>
<td>5.3.3 Activities</td>
<td>59</td>
</tr>
<tr>
<td>5.3.4 Checklists</td>
<td>62</td>
</tr>
<tr>
<td>5.4 Procedure for Library Maintenance</td>
<td>62</td>
</tr>
<tr>
<td>5.4.1 Background/Purpose</td>
<td>62</td>
</tr>
<tr>
<td>5.4.2 Roles and Responsibilities</td>
<td>62</td>
</tr>
<tr>
<td>5.4.3 Activities</td>
<td>63</td>
</tr>
<tr>
<td>5.4.4 Checklists</td>
<td>65</td>
</tr>
<tr>
<td>6 Appendix A: Definition</td>
<td>66</td>
</tr>
</tbody>
</table>
0. Introduction

0.1. Purpose

This document is a detailed description of the Configuration Management (CM) Process for TAF TSI deployment projects. It contains information on procedures and requirements for the configuration management activities in these projects. The document refers to the following Configuration Management related subjects:

- Change Management (chapter 3)
- Problem / Release Management (chapter 4)
- Baseline Management (chapter 5)

This paper provides an introduction to Configuration Management (CM), its concepts (chapter 1), functionality and planning (chapter 2). It is not intended to provide a look at the possible complexities of implementation. CM controls the:

- Product including all deliverable items (e.g., code and documentation)
- Items needed to develop the product including compilers, tools, simulators, etc.
- Changes to both.

The control helps to maintain the integrity of the product and provide the basis for communication among all participants in the development and use of the product (i.e., everyone knows the approved changes, the status of components/items/versions/changes, etc.)

This control is exercised throughout the life of the product and may result in the maintenance of many different versions and variants of the product being maintained simultaneously. This is where the complexity arises (see picture below) in what is otherwise a conceptually simple topic.
The application area of this paper is mainly the development and maintenance of all common components of the TAF TSI, rather than the individual adaptation of the existing systems at each RU or IM. The use of standard CM systems is strongly recommended, the CM processes must be adapted accordingly.

This paper gives only the generic requirements on Configuration Management and doesn’t describe detailed procedures as for the realisation of the TAF TSI requirements - even for the common part - various entities or parties at different time with different responsibilities may be involved having different scopes of Configuration Management. As an example: The CM process for the group writing the Strategic European Deployment Plan according chapter 7 of the TAF TSI starts with the TAF TSI, also Change Requests (regarding the TSI) must be handled. But Software or other OEM products are - at that time - not yet items of the CM process. Nevertheless, the CM process must be set-up in a way, that later enlargements are not only possible, but that these enlargements can also be traced back to items of a previous deployment phase of the TAF TSI.

0.2. References

0.2.1. Standards and Guidelines

<table>
<thead>
<tr>
<th>Id</th>
<th>Title</th>
<th>Doc ID, Edition</th>
<th>Date</th>
<th>Author / Publisher</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On the allocation of railway infrastructure capacity and the levying of charges for the use of railway infrastructure and safety certification</td>
<td>Directive 2001/14/EC</td>
<td>26/02/01</td>
<td>EC</td>
</tr>
<tr>
<td>2</td>
<td>On the interoperability of the Trans-European conventional rail system</td>
<td>Directive 2001/16/EC</td>
<td>19/03/01</td>
<td>EC</td>
</tr>
</tbody>
</table>

0.2.2. Other References

<table>
<thead>
<tr>
<th>Id</th>
<th>Title</th>
<th>Doc ID, Edition</th>
<th>Date</th>
<th>Author / Publisher</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conventional Rail TSI Telematic Applications for Freight</td>
<td>TAF_TSI_V20_document_22-06-2004.doc Version 2.0</td>
<td>22/06/04</td>
<td>AEIF</td>
</tr>
<tr>
<td>2</td>
<td>CR Telematic Applications for freight: Glossary (English)</td>
<td>AEIF_TAF_Glossary_V10_040622.doc Version 1.0</td>
<td>22/06/04</td>
<td>AEIF</td>
</tr>
</tbody>
</table>
0.3. Definitions and Abbreviations

For definitions see Appendix A of this document. The Abbreviations are listed in the following table.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Competence Area</td>
</tr>
<tr>
<td>CCB</td>
<td>Configuration Control Board / Change Control Board</td>
</tr>
<tr>
<td>CI</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>CMM</td>
<td>Capability Maturity Model</td>
</tr>
<tr>
<td>CMP</td>
<td>Configuration Management Plan</td>
</tr>
<tr>
<td>CR</td>
<td>Change Request</td>
</tr>
<tr>
<td>CSAR</td>
<td>Configuration Status Accounting Report</td>
</tr>
<tr>
<td>DDTS</td>
<td>Distributed Defect Tracking System</td>
</tr>
<tr>
<td>EDITION</td>
<td>Version of a document</td>
</tr>
<tr>
<td>FCA</td>
<td>Functional Configuration Audit</td>
</tr>
<tr>
<td>FR</td>
<td>Fault Report</td>
</tr>
<tr>
<td>ICS</td>
<td>Version of an item</td>
</tr>
<tr>
<td>OEM</td>
<td>Other Equipment Manufacturer</td>
</tr>
<tr>
<td>OGS</td>
<td>Ongoing Support Team</td>
</tr>
<tr>
<td>PCA</td>
<td>Physical Configuration Audit</td>
</tr>
<tr>
<td>PCP</td>
<td>Project Control Plan</td>
</tr>
<tr>
<td>PCS</td>
<td>Product Change Status</td>
</tr>
<tr>
<td>PIT</td>
<td>Project Implementation Team</td>
</tr>
<tr>
<td>PL</td>
<td>Project Leader</td>
</tr>
<tr>
<td>PMP</td>
<td>Project Management Plan</td>
</tr>
<tr>
<td>PTF</td>
<td>Program Temporary Fix - is a temporary change to the Operating System or Operating Sub-system Software, which subsequently is superseded by a version change.</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Meaning</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td>QA</td>
<td>Quality Assurance</td>
</tr>
<tr>
<td>RM</td>
<td>Release Management</td>
</tr>
<tr>
<td>RP</td>
<td>Release Plan</td>
</tr>
<tr>
<td>SEDP</td>
<td>Strategic European Deployment Plan</td>
</tr>
<tr>
<td>SoC</td>
<td>Statement of Compliance</td>
</tr>
<tr>
<td>SRS</td>
<td>System Requirement Specification</td>
</tr>
<tr>
<td>TAF TSI</td>
<td>Telematic Applications for Freight Services Technical Specification for Interoperability</td>
</tr>
<tr>
<td>TRB</td>
<td>Technical Review Board</td>
</tr>
<tr>
<td>VDD</td>
<td>Version Description Document</td>
</tr>
<tr>
<td>WP</td>
<td>Work Product</td>
</tr>
</tbody>
</table>
1. Configuration Management (CM)

1.1 CM purpose

The purpose of Configuration Management (CM) is to establish and preserve the integrity of all work products throughout the entire life cycle, within a project and across projects. It applies not only to all work products delivered to a customer respectively put into operation but also to intermediate work products. The Configuration Management process supports all other processes described in this paper.

To maintain control of a project, work products must be uniquely identified, their version and status must be maintained, changes to released work products must be authorised and controlled, dependencies between work products must be maintained (traceability), problems must be registered and traced to their solution. It is not sufficient that all these activities are correctly carried out, they must also be recorded to provide full visibility and to avoid ambiguity and rework.

CM tasks are executed by CM specialists and also by all practitioners in all processes. The following CM functions must be implemented in all processes:

- All work products are uniquely identified.
- All work products are maintained in CM Libraries.
- All work products are verified against CM rules (identification, library, presentation, format, traceability).
- Baselines are defined and established serving as basis for work in each phase of the life cycle.
- The CM roles are nominated for all work products.
- Change Management and Problem Management are established where applicable and an integrated change/problem management is used to control changes to work products.
- Control Boards are established to authorise changes to baselines and the generation of derived products from these baselines.
- Interfaces with external configuration management processes (customer, operations,..) are established.
- Periodic and event-driven baseline audits are performed.

Configuration Management includes Change Management, Document Management, Problem Management and Baseline Management.

1.2 CM Areas

CM provides visibility into the status of the evolving software product. Software developers, testers, project managers, Quality Assurance (QA) personnel, and the customer benefit from CM information. CM answers the following questions:

- What changes were made to the software?
- Who made the changes?
- When were the changes made?
- Why were the changes made?
- Were the changes approved?
- Where are the products?
- How are the products built?
Configuration Management is divided into the four functional areas: Identification, Change Control, Status Accounting, and Audit and Reviews and two major supporting activities: Library Establishment and Maintenance, and Product Build and Release (see Figure 1).

Figure 1 - Configuration Management Areas

A standard definition highlights the following operational aspects of CM:

- **Identification**: identifying components and their type, definition of baselines labelling of product elements (filenames, document titles, document identification, etc.), and making them accessible. The identification scheme should reflect the structure of the product.

- **Change Control**: having controls in place to ensure the integrity of the product by controlling changes to and release of approved baselines throughout the life of the product.

- **Status Accounting**: recording and reporting the status of components and change requests, and gathering and reporting vital statistics about components in the product.

- **Audit and review**: verifying the completeness and integrity of a product and maintaining consistency among the components by ensuring that the product is a well-defined collection of components.

- **Library Maintenance**: establishing and maintaining the CM library which serves as the repository for all work products and controls the promotion, physical release and acceptance (check-out/check-in) of the work products.

- **Product Release**: ensuring the audits, reviews, and tests have been completed, recommending the release to the Configuration Control Board (CCB) and building the product.

### 1.2.1 Configuration Identification

Identification involves identifying the structure of the system, uniquely identifying individual components, and making them accessible in some form. The goal of identification is to have the ability to identify the components of a system throughout its life cycle and to aid traceability between work products. Identification answers the following:
- What is the configuration of my system?
- What version of the file is this?
- What are the components of the system?
- What components constitute the baseline for developing this system?

Identification Activities:

- Select, identify, classify the components/items to be placed under CM control
- Create an identification scheme that reflects the software hierarchy (e.g., represents the architecture of the product)
  - Accommodate new items without modifying identifiers of existing items
  - Identify and maintain relationships between components
  - Model the structure of the product via a system model that represents the inventory of components for that product
  - Define relationships and interfaces between the various work products
  - Identify a way of partitioning the product for limiting the effects of changes to it and for isolating safety- or security-critical or other special components
- Denote baselines for a product
- Define a standard labelling scheme for work product and module files
- Uniquely identifying the various revisions of the work products
  - Identify a group of components (and their versions) that make up a configuration
  - Specify interfaces among components, versions, and configurations.

Figure 2 presents a typical breakdown of software into its distinct parts and presents a numbering scheme uniquely identifying each component.

![Software Configuration Identification Hierarchy](image)

Figure 2- Software Configuration Identification Hierarchy

Although most organisations tend to focus on the management of source code, the user deliverables, related documentation and data should also be identified and placed under CM control. It is important to store and track all environment information and support tools used throughout the software life cycle to ensure that the software can be
reproduced. The following are examples of items typically put under some form of CM control:

- Plans
  - Test Plans/Procedures and Test Cases/Data
  - CM Plans/Procedures
  - Quality Assurance (QA) Plans
  - Software Development Plans
- Support Software (software required to build and maintain the software throughout its life cycle) such as:
  - System Build Files
  - Compilers
  - Operating System
  - Link/Loaders
  - Shell Scripts
  - Third-party Tools.
- Source Code
- Object Code
- Documentation
  - Requirements Specifications
  - Architectural Design
  - Interface Control
  - Detailed Design
  - User Manual
  - Maintenance Documentation
  - Test cases, scenarios, procedures
- Software Development Folders (paper or electronic)
- Data structures and their life cycle documentation
- Libraries
- Development and test environments
- Problem/Enhancement Reports/Requests
- Hardware Platform Information
- CM Reports

A Process description is given in chapter 5.1

1.2.2 Configuration Change Control

Configuration Change Control involves evaluating and controlling the changes to work products throughout the software life cycle. The goal of Change Control is to establish mechanisms that will help ensure the production and maintenance of quality software. A generic change process is displayed in Figure 3.

Change Control answers the following:

- What is controlled?
• How are changes to the products controlled?
• Who controls the changes?

Change Control Activities:

• Defining the change process
  − Defines how changes can be requested or problems can be reported
  − Controls how and when changes are made
  − Defines level of authority required for changes
• Establishing change control policies and procedures
• Maintaining baselines
• Processing changes
  − Facilities for doing change impact analysis that predict all the possible effects of making a change
  − Propagation of changes, in a controlled manner, across different, but related, versions of the product
  − Tracking bugs, and how, when, and by whom they are dealt with
• Developing change report forms
  − Support for change request forms and problem reports
• Controlling release of the product.

Figure 3 - Generic Change Process
1.2.3  **Configuration Status Accounting**

Configuration Status Accounting involves the recording and reporting of the change process. The goal of status accounting is to maintain a status record of all items in a baseline, thus providing the traceability of all changes to the baseline throughout the software life cycle. Status Accounting answers the following:

- *What changes have been made to the system?*
- *How many files were affected by this problem report?*

Status Accounting Activities:

- Determining the type and content of logs and reports required to provide sufficient communication among managers, developers, and users of the work products.
- Tracking the status of CM items
  - Examine and evaluate the status of a product or item
- Tracking the status of changes to the system
  - Gather statistics about the product and the process
- Generating status reports
  - About all aspects of the product and process
- Recording and reporting the activities of CM.

A Process description is given in chapter 5.2.

1.2.4  **Configuration Audits and Reviews**

A Configuration Audit verifies that the software product is built according to the requirements, standards, or contractual agreement. Test reports, configuration status reports, traceability matrices, and documentation are used to verify that the software meets the stated requirements, that the requirements are traceable throughout the life cycle products, and that only the approved requirements and/or changes have been implemented.

The goal of Configuration Audit is to verify that all software products have been produced, correctly identified and described, and that change requests have been resolved. Informal audits are conducted at key phases of the software life cycle.

There are two types of formal audits that are conducted before the software is delivered to the customer:

- Functional Configuration Audit (FCA) and
- Physical Configuration Audit (PCA).

FCA verifies that the software actually satisfies the specified software requirements. PCA determines whether or not the design and reference documents represent the software that was built. Figure 4 illustrates the relationship between the FCA and PCA. Configuration audits answer the following:

- *Does the system satisfy the requirements?*
- *Are all changes incorporated in this version?*
Does the documentation represent the system as built?

Configuration Audit Activities:

- Defining audit schedule and procedures
- Performing audits on the established baselines
- Keeping an audit trail of the product and its process
  - Ensures maintenance of a history of all changes
  - Ensures traceability between all related components in the product and their evolution
  - Maintains a log of all the details of work done
- Generating audit reports.

Figure 4 - Configuration Audits

A Process description is given in chapter 5.3.

1.2.5 Software Baseline Library Establishment and Maintenance

In support of the above activities, a software baseline library is established. The library is the heart of the CM system. It serves as the repository for the work products created during the software life cycle. Changes to baselines in the library, and the release of software products from the library, are systematically controlled via the change control, configuration auditing, and product build and release functions. Repositories or libraries are needed to store and capture CM information as well as the different kinds of components such as source and object code, executables, diagrams, documentation and baselines.

Library Establishment and Maintenance answers the following:

- What components are available?
- What versions and variants are there?
- Who internally has which versions and variants?
- Who has access to which versions?
• Which version should I be working from?
• Was the version being checked-in based on the correct checked-out version?

The Software Library:

• Provides for and controls the storage and retrieval of configuration items/units
• Provides for the storage and recovery of archive versions of configuration items/units
  – Record versions of components, their differences, and reasons for those differences
• Provides for the maintenance of the library structure
• Ensures correct creation of products from the software baseline library based on approved releases
• Supports multiple control levels of CM
  – Provides controlled access to components in the system to avoid any unwarranted changes or change conflicts.
  – Provides for the sharing and transfer of configuration items/units between control levels within the library
• Provides storage, update, and retrieval of CM records
• Supports production of CM reports

The three main types of software libraries which are maintained by CM are the:

• Development (or Dynamic) library
• Master (or Controlled) library
• Static (or Archive) library

The relationship between these libraries is shown in Figure 5. Software is coded and tested as a set of modules in the development library (which consists of both the shared and the individual developer libraries). After unit tests, modules are transferred (sometimes called promoted) to master libraries for integration testing and system testing. When changes to master library modules are necessary, the appropriate modules are transferred back to a development library from the master library only after the appropriate authorisation has been given. When a baseline is established, the master libraries for the baseline are copied into the static libraries for archival purposes. The copies in the static library are never changed (hence the designation static).
A Process description is given in chapter 5.4

1.2.6 Product Build and Release

An integral part of controlling the products is to control their construction and release. This is in part handled through the change control process and library maintenance with the check-in and check-out procedures. In addition to these activities, and more importantly for independent verification and validation, CM should build the test releases. Also, CM will build all external releases and construct the release support documentation (e.g., release notes). Typical steps involved in the build and release of a product are shown in Figure 6.

Product Build and Release answers the following:

- *Which products can be built for release?*
- *What products have been released to whom?*
- *What versions and variants are scheduled to be built?*

Product Build and Release activities:

- Select compatible components to be made into a valid and consistent version of the product
- Support the construction of the product and its artefacts
- Take a snapshot or freeze the status of the product at any time
- Optimise systems construction by reducing the need to recompile components and saving space (this is partially a library function also)
- Regenerate any part of the product at any point in time
- Build all releases, both internal for test/validation and external for release to the customer
- Ensure that the customer gets the correct and complete product(s).
2 Configuration Management Process

2.1 CM Process Description

The CM process for a project begins at the beginning of the project, which is for the TAF TSI reakisation the start of the development of the “Strategic European Deployment Plan” (SEDP). The first document which arrives - in this case it is the TSI Telematic Applications for Freight - needs to be controlled in some fashion. Even if it is simply logging the receipt of the document (more formal control can be applied later). The project plans need to be drawn up and controlled, and the CM plan must be created and controlled (since these plans should evolve throughout the life of the project). This section will describe the CM process and referral to the two illustrations, Figure 7 – CM Process and Figure 8 – CM Process Interrelationships will aid understanding of the explanations.

The first area of CM activity for a project is configuration identification which is also a part of the initial project planning (Figure 10). The major task areas included within configuration identification are to:

- Develop the identification scheme including naming conventions
- Determine the baselines, their contents, and their acceptance criteria
- Establish the CM Library.

As work on the project progresses and work products begin to be completed, the other activities within CM come into play. When a work product is complete, if it has been identified for inclusion in a baseline, it is submitted to the CCB for baselining. The major task areas included within baselining a work product (or collection of work products) are to:
• Using the acceptance criteria defined in the planning phase, review of the work product and process used to produce it to ensure it is ready for baselining and has successfully completed all testing and/or verification processes
• Requesting and examining the status accounting information to ensure that all outstanding changes to the work product have waivers
• Obtain agreement by all CCB members that the work product is ready for baselining.

Once a work product has been baselined or for a non-baselined but CM-controlled work product placed under CM control, it can only be accessed for updating by using the check-out and check-in procedures. The process of producing and controlling the work products continues until ready for integration and testing (Note that overlap may occur when part of a system is in integration and testing while other parts are still being created). Integration and test are composed of the following processes:

• CM builds the executable
• Testing executes the planned testing using the test cases
• Problems which arise are entered into the change control process.

In addition to problems encountered during testing, those that arise after delivery and requested changes are also subject to the change control process. The major task areas within the change control process are:

• Log in the changes and categorise them (emergency, bug, change, level, etc.)
• Perform an impact analysis to estimate the technical, cost, and schedule impacts
• Obtain CCB approval
• Negotiate which release the change will be included in (not always necessary)
• Perform any necessary re-planning and schedule the release.

When the system has reached the level of required stability and is ready for delivery, it is subjected to the release product process. Please note that this process is for the initial release of a product or version/variant of a product. It is not for releasing/shipping of subsequent copies of the version/variant. The major task areas within the product release process are:

• CM builds the release using the instructions in the VDD
• Testing tests the product according to the test plan and procedures, which for all versions after the first should include regression testing
• Status Accounting is performed to ensure that problems, requested changes, and action items have been handled and the components are ready for release
• A Configuration Audit is performed to ensure that the products conforms to the requirements (FCA) and the design and documentation (PCA)
• CCB Approval is obtained to baseline and release the product
• The product is baselined and the delivery package is assembled
Figure 7- CM Process
2.2 CM Version Description Document

One of the more difficult tasks of CM is the control of multiple versions and variants of a product. Each version and variant will consist of a different mix of components/items from the base line library. For example, if a product has four components labelled A, B, C, D and each component has four versions labelled 1, 2, 3, 4. Assuming all components must be used to construct a release and that the interfaces between the components has not changed for the different versions, then 16 different releases of the product, labelled P, could have been released (e.g., P1 composed of A1,B1,C1,D1; P2 composed of A1,B1,C1,D2; P3…). There needs to be a way to control the combinations that are valid.

A means for controlling multiple product releases (versions and variants of the product) within a product line (i.e., where the versions/variants are strongly based on a foundation product) is to use the Version Description Document (VDD) as the primary control device. The change control process, also uses the VDD as an integral part of its process. The VDD is created early in each project during the Configuration Identification activities of the project’s CM planning effort. Each release of a product,
whether it is a version or a variant has its own VDD describing the contents of that release.

The creation of the VDD is composed of three tasks:

- Identify the components/items
- Identify the components/items to be included in each baseline
- Indicate the version of each component/item which will be used and/or created by the project

Throughout the life cycle of a project, the VDD will be updated, in particular for the development of a new product where new elements will be created during the architectural and detailed design phases. Additionally, as each baseline is approved and created during the life cycle, the VDD will be updated. The VDD will be used by all the main CM activities in order to control the products. Figure 9 displays the relationship between the VDD, the major CM activities, and the baseline library.

During the life cycle, the VDD may also be used for:

- Performing impact analyses
- Describing each build
  - Instructions for the build
  - Installation instructions
- Allocating changes to builds
- Tracking baselines and baselined elements
- Supporting configuration audits

Figure 9 - Relationship of the VDD and Baseline Library to CM Activities
2.3 CM Planning

2.3.1 Purpose

In order to perform any process in a controlled and disciplined manner when more than one person is involved, a plan is necessary, and Configuration Management, is not different. The previous sections discussed what configuration management is and the associated concepts and activities. In this section, some of the concepts and activities associated with CM planning will be discussed. This discussion is not intended to replace the CM planning procedures, but is an introduction to familiarise the reader with CM.

CM planning needs to start early in the project, as work products will be produced almost from day one (e.g., the feasibility study). Figure 10 presents an overview of CM planning and shows the relationship between project planning and CM planning.

2.3.2 Procedure

This procedure outlines the activities involved in the preparation of the Configuration Management Plan (CMP). The objective of an CMP is to document the plans that establish and maintain work products throughout the project's life cycle. The CMP identifies Configuration Management (CM) activities and procedures, the schedule of activities, resources required, equipment and CM tools, and unique training required for CM staff assigned to the project. When project CM requirements change, the CM Manager updates the CMP to reflect current project CM requirements. The CMP may be a standalone document included by reference in the Project Management Plan (PMP) or an integral part of the PMP.

2.3.2.1 Roles and Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>Provides inputs to the CMP. Ensures that co-ordination is obtained from all program participants directly affected by the CMP. Reviews the CMP. Approves the CMP. Ensures that the CMP procedures are being followed. Ensures that the software personnel comply with their CM responsibilities.</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Develops the CMP. Checks the CMP periodically to ensure that it's correct and up to date.</td>
</tr>
<tr>
<td>Manager/Staff</td>
<td></td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>Provides inputs to the CMP. Review the CMP. Audits CM against the CMP.</td>
</tr>
<tr>
<td>Senior Management</td>
<td>Reviews the CMP, as appropriate</td>
</tr>
</tbody>
</table>

2.3.2.2 Activities

2.3.2.2.1 Procedure Overview

<table>
<thead>
<tr>
<th>Entry Criteria</th>
<th>Requirement for CM is present.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>• Project schedule (from Project Planning)</td>
</tr>
<tr>
<td></td>
<td>• Allocated system requirements (from Requirements Management)</td>
</tr>
<tr>
<td></td>
<td>• Identified work products and defined baselines (from Project Planning)</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Analyse the requirements, review the project schedule and any established baselines</td>
</tr>
</tbody>
</table>
2. Using the standard procedures, define the processes for configuration identification, baselining, problem reporting, change control, version control, library requirements, status reporting, audits, CM tools, and metrics.
3. Estimate CM size, effort, cost, and schedule
4. Develop the draft CMP using the template
5. Conduct a review of the draft CMP, including all groups
6. Generate final CMP
7. Periodically throughout the project life cycle, review CMP for correctness

<table>
<thead>
<tr>
<th>Outputs</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Management Plan</td>
<td></td>
</tr>
<tr>
<td>CM schedule and milestones</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metrics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Time spent on CM planning and reviews</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exit Criteria</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Signed and baselined CMP</td>
<td></td>
</tr>
</tbody>
</table>

2.3.2.3 Detailed Procedural Steps

The following paragraphs provide guidance for Software Configuration Management Planning

**Analyse the Requirements, Review the Project Schedule and the Established Baselines**

CM requirements are normally compiled from several sources. Marketing requirements, operational requirements, contract requirements, budget constraints, the Capability Maturity Model (CMM), the system specification, and analysis of historical need, contribute to the set of CM requirements. The schedule of CM activities is related to events in the project schedule. Analysis of the project's established baselines (for maintenance projects) dictate the activities CM performs to maintain and update those baselines.

**Define the Processes**

Using the standard procedures, define the processes for configuration identification, problem reporting, change control, version control, library requirements, status reporting, audits, tools, and metrics. The standard procedures can be used without change, or tailored to the specific needs of the project. If they are used without change, they should be included by reference in the CMP. If the are tailored, the tailored version should be included with the CMP.

**Estimate Size, Effort, Cost, and Schedule**

The size, effort, cost, and schedule for CM are dependent upon the project’s size, effort, cost, and schedule and these should be estimated after the project’s estimates and based upon the project’s estimates (and are fed back into update the project’s estimates). CM estimates should be based upon projected CM activities and not simply calculated based upon a percentage of the project’s estimates.

**Using the Template, Develop Draft CMP**

CM generates the draft CMP using the template and the information gathered. As a part of this process, the organisation standard CM procedures will be tailored if necessary.
Conduct a Peer Review of Draft CMP

CM distributes the draft CMP to all project personnel, as well as Senior Management for review. Set a specific time for all affected groups (e.g., developers, marketing, customers) and individuals to meet and discuss comments, questions and concerns.

Generate Final CMP

Using the comments from the review, CM generates the final CMP. Distribute the final CMP to the Project Manager for approval. After project management approval, obtain the remaining approval signatures. These are Quality Assurance, Test Manager, and CM manager. After all signatures are obtained the CMP is submitted to the Configuration Control Board (CCB) to be placed in the project documentation baseline.

Periodically CM Reviews CMP

Throughout the project life cycle CM reviews the CMP for completeness and correctness. When differences are noted between the documented planned activities and the manner in which activities are actually being performed within CM, CM, or anyone else on the project team, generates a problem report or change request to have the CMP updated. The change is handled through the CCB.

2.3.2.4 Checklists

<table>
<thead>
<tr>
<th>No.</th>
<th>Activity</th>
<th>Yes</th>
<th>N/A</th>
<th>Signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CMP outline and format is tailored to the unique requirements of the project.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CM requirements are elicited from applicable documents.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CM risks have been identified and assessed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>The CM processes have been defined.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CMP has been reviewed and coordinated by all groups.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>QA review of the CMP has been completed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Senior management approval has been received for implementation of the CMP.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CMP has been distributed to all participating groups.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Lessons learned have been collected and recorded.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The above checklist contains some questions that can be used to ensure compliance with this procedure. This checklist is meant to be a starting point and should be modified to meet specific organisational and project needs.
1. Identify products to be delivered
2. Identify project life cycle
3. Define versions to be built
4. Identify internal work products
5. Identify configuration items (CIs)
6. Identify major milestones and baselines
7. Define baseline acceptance criteria

1. Define CM processes for:
   - Problem reporting
   - Change control
   - Version control
   - Status Reporting
2. Define CM software library requirements, responsibility, and use
3. Define CM processes for metrics and audits
4. Define CM roles and responsibilities
5. Define required CM tools
6. Estimate required size (number of products to be supported, size), effort, cost, and schedule

1. Develop Work Breakdown Structure (WBS) tasks
2. Develop project roles and responsibilities
3. Define project development processes, standards, etc.
4. Make estimates
5. Develop other project plan requirements

Note 1: Step 2.0 cannot begin until step 1.0 is complete
Note 2: Step 3.0 should not wait until step 2.0 is complete to begin
Note 3: This is not intended to show all inputs into the project plan, it is just a representation of CM’s role in planning

Figure 10 - CM Planning
3  Change / Problem Management Process

3.1 Definition and Scope

Change Management in the context of this paper covers the TAF TSI projects, which result from computer hardware and software changes. The following management systems may be distinguish:

**Change Management** is the authorisation of changes. It can be used to authorise change from non-existence to existence of a work product, from one version to another. Change Management can be performed via Formal Change Control (Fault Report, Change Request) or Comment Facilities.

**Problem Management** covers all types of technical problems ('defects') regardless of the functional area. However, the scope is limited to problems reported against registered work products. The goal of problem management is to optimise the process of reporting the problems and their solutions within a project and across projects. The problem management process must contribute to reduce the number of problems found.

In the further description Change Management and Problem Management are both addressed as Change Management, because from the point of view, related to generic requirements regarding Configuration Control, the handling for both is the same.

The Change Management (CM) Process covers the definition and management of the lifetime aspects of all systems and entities in the TAF TSI project. This covers aspects of

- Business and project plans
- Organisational necessities and process implications
- System environment and infrastructure.

Besides the TAF TSI project itself the release management process must take into account the implications of other projects (systems) which do interface to or do have a relationship with the TAF TSI project. This includes (but is not limited to):

- Customer care and billing
- Product development and marketing
- Service management and assurance
- Network management
- Network planning and construction
- IT operation and infrastructure.

3.2 Objectives

The objectives of an organised Change Management process are to:

- Enable efficient and reliable methods for executing changes to the software, hardware and related work procedures within the TAF TSI common environment
- Ensure that this is done according to standardised and proven methods and procedures
- Eliminate or at least minimise the impact on the production environment and therefore on the computer services to the clients
- Enable a continuous improvement process to take place by reporting, feedback and corrective measures
One of the major objectives is to limit the number of change projects; i.e. to bundle changes to be done together at one time or to include changes into scheduled maintenance. One of the reasons is to make the change process more efficient. The more important reason, however, is to minimise the potential negative impact upon the operating environment. Therefore it must be avoid making ad hoc changes unless they are needed as a result of a disruption of the operating environment. Then they will be subject to the procedures of Problem Management. If it is possible, changes will be bundled into a set schedule with set tasks and a set objective so that a proven procedure can be used which will minimise the manpower requirements and will have no or only minimum impact upon the production environment and service availability.

3.3 Organisational Entities

The following figure identifies the organisational entities which are involved in the change management process.

![Diagram of Change Management organisation]

Figure 11 - Change Management organisation

The main actors are the Change Control Board and the Change Review Team.

- The Change Review Team communicates and co-operates with the other project functions in order to collect and elaborate all changes requested on the TAF TSI project.
- The Change Control Board is the approval authority for the release / baseline definition (see chapter 4). Furthermore, it is the interface to the Change Control Boards of other projects and to other (upper) management functions in the entire organisation.

A detailed description of the involved entities is given in a subsequent section.
3.4 General Process

As mentioned above the centre point is the Change Review Team. This organisational unit is the owner of the change management process.

The overall change management process is independent of the project life-time. I.e., regardless any time aspects of any project phases, the procedure for defining, implementing and verifying of changes remains the same. This will be controlled by the Change Review Team to ensure consistency in working and in the process.

The change management process can be split into 6 general phases:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Who</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Submission (Request Change)</td>
<td>Authorised entities from inside or outside the project organisation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Submission of an FR, CR, or new Req.:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Technical requests will be mainly initiated out of the project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tactical and organisational requests are expected mainly from high-level management and/or other relevant projects.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• It is expected, that the submitter already makes a first analysis and internal evaluation before submission. (E.g. change requests initiated by a developer team are expected to be co-ordinated and approved among all developer teams of a project.)</td>
</tr>
<tr>
<td>2</td>
<td>Evaluation</td>
<td>Change Review Team</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Evaluation of each change request concerning its technical, commercial and organisational impact on the project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Proposal for each request concerning its acceptance / non-acceptance.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Summing and structuring of the change requests in a release plan.</td>
</tr>
<tr>
<td>3</td>
<td>Decision</td>
<td>Change Control Board</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Decision on the proposed release plan.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Initiation of the relevant process / organisational changes.</td>
</tr>
<tr>
<td>4</td>
<td>Implementation</td>
<td>Project management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Implementation of the requested and approved changes.</td>
</tr>
<tr>
<td>5</td>
<td>Verification</td>
<td>Acceptance test team</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Verification of the (successful) implementation of the changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• In case on non-success or non-implementation this will result in a loop back to step 2 involving the Change Review Team. The according change may either be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Processed again through the steps 3 – 5, or be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Killed if there are indicators that a success can again not be guaranteed.</td>
</tr>
</tbody>
</table>
### 3.5 Life-Time Aspects

#### 3.5.1 Change Factors

Items which can impact the life-time of the project are defined as:

<table>
<thead>
<tr>
<th>Change Factor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fault Report</strong></td>
<td>Identification of a malfunction or the lack of a specified feature. Important for the evaluation of a FR is its impact on the system stability, on the system function and on the operational process. FRs are mainly initiated by the acceptance test team(s) and by the operational staff.</td>
</tr>
<tr>
<td><strong>Change Request</strong></td>
<td>This requests the modification of a specified and agreed feature. Like an FR, a CR must be carefully evaluated concerning its impact on the system stability, on the system function and on the operational process. CRs can be initiated by authorised entities inside and outside of this project.</td>
</tr>
<tr>
<td><strong>New Requirement</strong></td>
<td>This is a request for a new function not yet specified or available in the system. Any new requirement must be carefully evaluated concerning its impact on the system stability, on the system function and on the operational process. A new requirement can be initiated by authorised entities inside and outside of this project.</td>
</tr>
<tr>
<td><strong>New OEM Product Release</strong></td>
<td>Even if no change / failure is identified, the release of an OEM (core) product may change due to its independent release plan. Contractual issues (maintenance agreements) may force an according update of the installed system solution.</td>
</tr>
</tbody>
</table>

All of them could be applied to

- Hardware
- Software
- Documentation
- Services (e.g. training)
- Procedures and processes
- Organisational structure
- Roles and Responsibilities.
3.5.2 Handling of Changes

The main issue in the change management process is: when will be a change implemented. The time factor for the implementation is mainly driven by

- The importance (= necessity) of the change
- The introduced impact on the system and/or organisation.

I.e., a change may not be necessarily implemented immediately, but at some point in time during the project life-time. This is the main reason for an effective and careful change management, as changes have to be tracked over time, over various project releases. And, even more complex, a change request itself is a item of change.

The following figure 12 illustrates this time context.

![Figure 12 - Handling of changes over the project life-time](image)

The individual stages are defined as:

- **Release in operation**: Some changes/modifications have to be implemented immediately, mainly those due to failure reports. The resulting patches/bug-fixes will be introduced into the system when available (and verified).

- **Maintenance release**: A maintenance release is defined as a system modification concentrating on problem resolution. I.e., a defined set of bug-fixes / patches is introduced to the system.
  
  A maintenance release could also be driven by an upgrade (= new release) of an OEM product, if this is required for the harmonisation / synchronisation with the roadmap of an OEM product.
  
  Maintenance releases are characterised as having no impacting on the running operational processes and/or organisation.

- **New release**: Any modifications introducing new functionality or system changes must lead
to a new release. Such changes could be driven by change requests and/or new requirements which enhance the entire system.

A new release may not only provide new technical functions but also changes to processes and organisation.

Release in planning Dependent on the necessity/implications (e.g. commercial, organisational) of a change, its implementation may be shifted to the future. In order to plan for it and to give the project a time structure such system changes / modifications can be grouped into a release which is indicated as "in planning".

3.6 Release Management Organisation

3.6.1 Change Review Team

3.6.1.1 Scope and Responsibility

The Change Review Team is the central organisational entity in the change management process. Its main task is to:

- Be the owner of the change management process and the Change Management Specification.
- Collect and analyse all tracked FR, CR, and new requirements on the TAF TSI project. Evaluate their impact on:
  - Hardware
  - Software
  - Documentation
  - Services (e.g. training)
  - Procedures and processes
  - Organisational structure
  - Roles and responsibilities.
- Elaborate the cost factor. Not only the effort to implement the change but also internal costs (e.g. for reorganisation, training, etc.).
- Consolidate the release plans of all (basic) products used in the TAF TSI system. This includes also a mapping of all available features to the Statements of Compliance (SoCs).
- Consolidate the release plans of the TAF TSI systems, which are related to the TAF TSI project in respect to their impact on the TAF TSI project.
- Implement a phase-out plan for those TAF TSI systems which will be replaced by the TAF TSI. These plans must consider functional and geographical issues as well as dependencies to other project activities (e.g. cleansing, training, installation).
- Specify the release plan for the TAF TSI project taking the a.m. items into account. This includes a proposal for the acceptance / non-acceptance of the submitted change requests, especially for all technical aspects. For organisational / tactical issues a first evaluation is expected by the Change Control Board.
- Evaluate and/or propose requirements for the OEM products used in the TAF TSI project.
- Evaluate and/or propose requirements for systems outside the TAF TSI project.
- Track together with project management and the acceptance test team the implementation and verification of the approved changes. In case of non-success or non-implementation this will result in a loop back into the evaluation process.
3.6.1.2 Team Organisation and Work Approach

The Change Review Team shall meet on a periodic basis to evaluate all open CR/FRs.

A status report reflecting this evaluation shall be prepared and handed over to Change Control Board at least one week prior to the next meeting of the Change Control Board. For each of the requested changes the evaluation status shall cover:

- Impact on the current project status
- Proposal for resolution (accepted/rejected)
- Proposal concerning the implementation (release)
- Identification of relevant constraints / countermeasures.

The status report has to be worked out in close co-operation with project management and the affected project teams.

The Change Review Team evaluates each single change request concerning its technical impacts. The result shall be reflected in an according release plan. The approval of the release plan, plus the decision on organisational / tactical changes relies with the Change Control Board.

3.6.1.3 Input and Output

**Input**

- Any CR/FR submitted by an qualified authority out of the project organisation, the operational teams, or from management.
The TAF TSI project plan.
Any CR submitted by other related projects.
Release plans of the application products used in TAF TSI system.
Release plans of SW platform products (e.g. Operation System, database).
Release plans of HW components.
Release plans of those systems, which are related to TAF TSI.
Any business plans with impact on the existing organisation and processes.

Output
- Status reports
- Release plan
The release plan shall be updated every 6 months or in case of major impacts. In the last case this will have to be discussed with the Change Control Board.
- Requirements specifications for other systems / products.
- Requirements specifications shall be provided together with the release plan or on request.

3.6.2 Change Control Board

In the context of the release management process the Change Control Board co-ordinates the life-time aspects of the TAF TSI project:

- It co-ordinates all change requests initiated via the Change Review Team and decides further actions.
- Verification and approval of the release plan.
- Co-ordination with other related projects. Especially those which are related to TAF TSI to identify mutual requirements and constraints.
- Implementation of necessary organisational and/or process changes due to the release plan.

Furthermore, the Change Control Board takes the final management decisions on products, procedures, organisational impact and financial aspects. This is done in close co-operation with upper management.

- Definition of the project strategic.
- Definition of the project phases concerning major functionality and time. This includes relevant contracts and purchase orders.
- Negotiation and decision with the supplier (and indirectly with the OEM product suppliers) concerning change requests on the relevant OEM products.
- Resolution of escalated inter-project issues.

3.6.3 TAF TSI Project Teams

3.6.3.1 TAF TSI Project Management

- To plan, implement and monitor all project activities.
- To set-up the project plan and to track all project tasks.
- To analyse any submitted CR/FR and to evaluate together with the Change Review Team their impact on the project.
- To support the Change Review Team in specifying the release plan.
3.6.3.2 Acceptance Test Team

In general, this team is responsible for the specification, planning and implementation of the acceptance test process and related activities. This team is a central function within the overall project organisation. The acceptance test team is organised and managed by project management.

In the context of the release management process the responsibilities are defined as:

- To specify the acceptance test process, the test plans and the test activities for a defined release according to the release plan.
- To provide the Change Review Team with requested information on the acceptance process (e.g. test specification).
- To report to the Change Review Team the acceptance test results. Especially the results of the mandatory changes.

3.6.3.3 Other TAF TSI Project Teams (Authorised Authorities)

- To submit a CR/FR.
- To consolidate requests from other project teams before submitting a change request.
- To verify the implementation of approved changes (mainly the acceptance test team).

3.6.4 Supplier

If the overall system solution of the common part of the TAF TSI will be supplied by a main supplier, the term "supplier" is used for this commercially responsible entity. In addition, there are the suppliers (= OEM suppliers) of the OEM products. However, from a contractual point of view they are not directly dealing with the TAF TSI deployment team. This entity as the main supplier will manage the communication / information flow between the TAF TSI deployment team and the OEM suppliers.

In the context of the change management process the main supplier is responsible for:

- Provisioning of all roadmaps of the OEM products to the Change Control Board and the Change Review Team respectively. These roadmaps have to be requested from the according OEM suppliers.
- Provisioning of the roadmap of the (integrated) system solution to the Change Control Board and the Change Review Team respectively.
- To negotiate the technical and commercial aspects concerning the OEM core products with the relevant OEM supplier (e.g. due to a change request).

3.6.5 Other related Projects

- To submit a CR/FR for the TAF TSI project.
- To receive a CR/FR related to their project from the TAF TSI project.
3.7 Decision and Control Mechanism

3.7.1 Decision and Escalation Authorities

The following table outlines the various levels within the change management organisation, including the according authorities and responsibilities. Issues which cannot be resolved within a certain level have to be escalated to the next level for resolution.

<table>
<thead>
<tr>
<th>Organisational Entity</th>
<th>Authority / Responsibility</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Change Control Board</td>
<td>• Decide on strategic issues&lt;br&gt;• Define the project phases (incl. commercial aspects)&lt;br&gt;• Negotiate with supplier and other projects&lt;br&gt;• Resolve escalated problems&lt;br&gt;• Approve the release plan&lt;br&gt;• Resolve escalated problems&lt;br&gt;• Accept/reject CR / FR from other projects</td>
<td>• Organisation and technical issues&lt;br&gt;• Technical issues if submitted from other projects</td>
</tr>
<tr>
<td>3 Change Review Team</td>
<td>• Accept/reject CR / FR&lt;br&gt;• Evaluate raised CR / FR and map them into the release plan&lt;br&gt;• Resolve escalated problems&lt;br&gt;• Define and evaluate requirements on OEM products and other projects</td>
<td>• Technical issues&lt;br&gt;• Proposals on organisational issues</td>
</tr>
<tr>
<td>2 Project Management</td>
<td>• Submit a CR / FR&lt;br&gt;• Resolve escalated problems&lt;br&gt;• Evaluate FR raised during operation&lt;br&gt;• Define maintenance releases and propose them to the Change Review Team</td>
<td>• Technical issues</td>
</tr>
<tr>
<td>1 Qualified Authorities out of the TAF TSI Project or other Groups</td>
<td>• Submit a CR / FR</td>
<td>• Technical issues</td>
</tr>
</tbody>
</table>

3.7.2 Control Mechanism

The following control mechanisms have to be applied by the Change Review Team.

Status Tracking The status of each CR / FR / Req. shall be tracked continuously through the tracking database system. This will also identify who is responsible for a certain item at a given point in time.

Reviews Through reviews the implementation of approved changes shall be monitored. This applies especially for documents, services, organisational issues, etc. in cooperation with project management.

Verification System related changes have to be verified according to the defined acceptance process and the acceptance test specifications. This will be carried out by the acceptance test team under supervision of the Change Review Team. Verification of organisational and tactical changes will be carried together with high-level management.
3.8 The Release Plan

The release plan will be the basis for the following documents:

- Project Plan
- Migration Plan
- Acceptance Process Specification
- Training Plan.

3.9 Flow Chart for the Change Process

![Flow Chart for the Change Process](image-url)

Figure 14: Flow Chart for the Change Process
3.10 Procedure for Change Implementation

This procedure delineates the steps to review, verify, document, and incorporate changes to a project. Changes include elaboration, rewording, additions, deletions, and bug fixes. All changes must be reviewed by the affected parties. Changes which pose potential risks to the cost, schedule, or technical achievement of the development project must be clearly communicated to project management.

3.10.1 Roles and Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>Ensure that all required information is provided as needed</td>
</tr>
<tr>
<td></td>
<td>Participate in acceptance presentation(s) to the customer</td>
</tr>
<tr>
<td></td>
<td>Re-plan project if needed</td>
</tr>
<tr>
<td></td>
<td>Inform Senior Management of significant problems</td>
</tr>
<tr>
<td>Software Engineer</td>
<td>Analyse software changes</td>
</tr>
<tr>
<td></td>
<td>Ensure change is a software change</td>
</tr>
<tr>
<td></td>
<td>Identify and negotiate any needed information/support (users, team members, test, etc.)</td>
</tr>
<tr>
<td></td>
<td>Notify Project Manager if r-plan may be needed</td>
</tr>
<tr>
<td></td>
<td>Present results to team for acceptance</td>
</tr>
<tr>
<td></td>
<td>Present to the customer for acceptance (as needed)</td>
</tr>
<tr>
<td>Software Test Manager</td>
<td>Support Software Engineer in analysis</td>
</tr>
<tr>
<td></td>
<td>Review/understand software changes</td>
</tr>
<tr>
<td></td>
<td>Participate in acceptance presentation(s)</td>
</tr>
<tr>
<td>CM Staff</td>
<td>Provide baselined inputs</td>
</tr>
<tr>
<td></td>
<td>Assist in determining work product or version impacts</td>
</tr>
<tr>
<td></td>
<td>Baseline outputs and put notes in project history</td>
</tr>
<tr>
<td>QA Staff</td>
<td>Periodically participate in/audit change control activities</td>
</tr>
<tr>
<td></td>
<td>Assist in determining QA impacts</td>
</tr>
<tr>
<td></td>
<td>Participate in user acceptance (if it occurs)</td>
</tr>
</tbody>
</table>

3.10.2 Activities

3.10.2.1 Procedure Overview

| Entry Criteria | • An experienced, trained individual(s) is available for analysing the change(s) and assessing the technical impacts |
|               | • A team of trained personnel is designated to support this activity           |
|               | • Sufficient resources are allocated                                           |
|               | • Change Requests (CR)/Problem Reports (PR) contain complete data             |
| Inputs        | • CR/PR                                                                        |
|               | • Current project plans and status                                             |
|               | • Current technical baseline                                                   |
| Steps         | 1. Log the change into the change control system                              |
|               | 2. Identify type of change                                                     |
|               | 3. For an emergency bug fix begin work                                        |
|               | 4. Perform impact analysis                                                     |
|               | 5. Change Control Board (CCB) accepts/rejects change and recommends release for incorporation |
|               | 6. Negotiate with customer and other affected groups                          |
|               | 7. Schedule work/adjust plan                                                   |
|               | 8. Track change effort                                                        |
|               | 9. Close the change in the change control system                              |
3.10.2.2 Detailed Procedural Steps

Log the Change Into the Change Control System

Every change, whether it is from a Change Request or a Problem Report shall be logged into the change control system.

Identify Type of Change

The five main types of changes are:

1. Emergency bug fix
2. Non-emergency bug fix
3. Missing feature
4. Change of a feature
5. New feature

In this first step the most important objective is to accurately identify emergency bug fixes, since work must begin immediately and in parallel to the other steps of the change control process. This means that emergency bug fixes bypass the normal controls on the work products and work has begun without estimation of the technical and economic impact. Thus you should identify criteria to limit the number of emergency bug fixes or a project can quickly go out of control. An emergency bug can be defined as one which

- Substantially affects the development progress (e.g., the bug is in a critical module and all testing is halted until it is fixed) or the operation of the system
- Causes the program to become inoperative at the customer site
- May cause substantial harm, either tangibly or intangibly

For an Emergency Bug Fix Begin Work

Work must begin immediately on any change categorised as an emergency bug fix or alternatively, an emergency meeting of the CCB may be called to authorise the work. This does not mean that the impact analysis does not need to be performed or any of the other steps of the change control process do not need to be performed. It means that they are done in parallel with the emergency bug fix. In fact an emergency bug fix requires even closer monitoring, control, and co-ordination since it is being done outside the normal process. For example, an emergency baseline needs to be created and the changes to work products may need to be incorporated and/or back-fit into existing versions, or
the version under current development. Note, if Patches are used to make emergency fixes, the solution must be submitted back into the system to ensure that they are an approved solution and then must be incorporated into the work products as approved.

**Perform Impact Analysis**

The impact analysis will look at and analyse the change from four different perspectives:

- The scope impact - examine the CR/PR to determine that it is indeed a change/problem to/with portion of the system, or if the CR/PR should be processed by another organisation
- The technical impact - what work products (and versions/variants of them) are affected and by how much (what is the level of effort to make the change(s) to the work product(s)), and are new/different skills needed or new/different tools
- The schedule impact - examines the impact on the schedule of the project and determines the optimum release for the implementation of the change
- The cost or economic impact - an estimation of the cost of performing the change immediately, performing the change in the optimal release, and not performing the change.

An integral part of the impact analysis and one which should be performed early in the impact analysis is the cluster analysis of changes. The purpose of the cluster analysis is to cluster or group together changes which affect the same work product(s). This promotes efficiency through minimising the number of times a particular work product needs to be changed, reviewed, and tested.

At the end of the impact analysis, the things which should be known for a change are:

- What is affected
- How long should it take
- When is it best scheduled
- How much will it cost.

This will provide the CCB with the information it need to make its decisions and the project manager the information they need to know to negotiate the change with the customer and other affected groups.

**CCB Accepts/Rejects Change and Recommends Release for Incorporation**

Based upon the submitted documents, recommendations of the impact analysis team and their deliberations, the CCB will accept, reject, or defer the change until additional information is collected. If the change is accepted, the release must also be recommended for the change and the CCB authorisation for access to the controlled work products must be given. The submitter of the change, as well as other affected groups will be notified of the status of their change. One of the primary considerations of the CCB is that all impacts have been considered (and are reasonable) and that all affected parties have had input to the process.

**Negotiate with Customer and Other Affected Groups**

The next step in the process is to negotiate the change with the customer. The negotiations will primarily deal with the release and the cost, although, in the case of a rejected change, the rejection itself may be a subject of negotiation. One of the reasons for including the customer on the CCB is to attempt to limit the negotiations after the CCB, since most
issues should already have been resolved during the CCB. At the end of this step, it should be finalised which release the change will be incorporated into, and how much the change will cost.

**Schedule Work/Adjust Plan**

Based upon the accepted changes, the plan should be adjusted and the work scheduled and assigned.

**Track Change Effort**

The effort expended on the change shall be tracked.

**Close the Change in the Change Control System**

The final step is to close the change on the change control system when it has been incorporated into a release.

3.10.3 **Checklists**

<table>
<thead>
<tr>
<th>No.</th>
<th>Activity</th>
<th>Yes</th>
<th>N/A</th>
<th>Signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Has the change been logged?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>If it is an emergency bug fix have the special procedures been followed?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Has a scope, technical, cost, and schedule impact analysis been done?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Has a cluster analysis been done?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Has the change been entered into the CCB agenda?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Has the release been selected and negotiated with the customer?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Has the plan been updated and schedule adjusted?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Has the effort been tracked?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Has the information been logged into the Status Accounting data base as it has been updated?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Has the change been closed after it has been implemented in a release?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The above checklist contains some questions that can be used to ensure compliance with this procedure. This checklist is meant to be a starting point and should be modified to meet specific organisational and project needs.

3.10.4 **State Machine for fault processing**

The following figures 15 shows a state machine for the processing of defects. The according states are described below.
3.11 Example of CR/FR Template

Input fields of a submission template

| S | Submitted | Internal state, only required by the system. |
| N | New       | A new report has been submitted.         |
| A | Assigned  | A person has been assigned to the record to investigate the problem. |
| Y | Ass         | The reported problem has been analysed. |
| D | Duplicated | The report is related to a problem which has been submitted earlier. A pointer is generated to the initial report (the second report should be closed). |
| U | Suspended | The reported problem has been analysed and a solution being proposed. However, the implementation is suspended. |
| X | Accepted   | The proposed solution is accepted (technical, effort, time, etc.). The report is assigned to another person to implement the proposed solution. |
| E | External   | The reported problem must be fixed externally. It is therefore passed to the external entity for further handling. |
| F | Forwarded  | The reported problem is not resolvable in the assigned project, but by another project. Therefore it has been forwarded to this other project. |
| K | Killed     | No further actions.                        |
| R | Resolved   | The reported problem has been fixed and a solution implemented and pre-tested. |
| V | Verified   | System integration / validation team has tested and approved the solution. |
| I | Installed  | The approved resolution has been installed at customer site, either in form of a patch or as part of an official maintenance/upgrade release. |
| P | Approved   | The customer has approved the solution and closes his report internally. |
| C | Closed     | The report is closed in DDTS          |

Figure 15 - Example of a state machine for processing defects
<table>
<thead>
<tr>
<th>Template Segment</th>
<th>Input Parameter</th>
<th>Mand./Opt.</th>
<th>Possible Selection Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defect information</td>
<td>Project release</td>
<td>M</td>
<td>• Project releases</td>
</tr>
<tr>
<td></td>
<td>Defect type</td>
<td>M</td>
<td>• Fault report (default)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Change request</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Question</td>
</tr>
<tr>
<td></td>
<td>Categories of costs</td>
<td>O</td>
<td>• Warranty</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Maintenance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Consulting</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Other</td>
</tr>
<tr>
<td></td>
<td>Module</td>
<td>M</td>
<td>• Project specific modules</td>
</tr>
<tr>
<td></td>
<td>Need fix by</td>
<td>O</td>
<td>• Date</td>
</tr>
<tr>
<td></td>
<td>Breakdown</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Problem description</td>
<td>Headline</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Problem</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test description</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Build label for CM</td>
<td>O</td>
<td>• for internal use only</td>
</tr>
<tr>
<td>Defect information</td>
<td>Problem severity</td>
<td>M</td>
<td>• Critical error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Major error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Minor error</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Not specification</td>
</tr>
<tr>
<td></td>
<td>Detected in phase</td>
<td>M</td>
<td>• Analyse / design</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Coding</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Module test</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• System integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Acceptance test</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• During operation</td>
</tr>
<tr>
<td></td>
<td>Platform</td>
<td>M</td>
<td>• Selection list of project specific items</td>
</tr>
<tr>
<td></td>
<td>Where discovered</td>
<td>M</td>
<td>• Customer site</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Development</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Product management</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Others</td>
</tr>
<tr>
<td></td>
<td>Customer responsible</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customer report ID</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Submitter information</td>
<td>Submitter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Phone number</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Organisation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4 Baseline Management Process

4.1 Preface

The baseline is a key concept of CM. A baseline is defined as

"a specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change procedures".

Thus it is an approved snapshot of the product at a given point in its evolution which is used as a reference point against which to control changes. When an item is baselined, it becomes frozen - the item can only be changed by creating a new version. The number and type of baselines depend on which life cycle model the project is implementing and other considerations such as the volatility of the requirements and the understanding of the product to be built. In general, the less potential chance for change and/or error and thus the need for returning to a previous state in the development, the fewer baselines you need. Picture 16 shows the typical baselines which may be established during the life cycle and when they may be established.

A standard V-model life cycle would use the developmental (life cycle dependent) baselines depicted in Picture 1. Other life cycle models, such as the spiral, incremental development, and rapid prototyping, require more flexibility in the establishment of baselines. In addition, several informal baselines are usually established during the software development process.

Figure 16: Life Cycle Baselines
The Baseline Management Process is a sub-process of Configuration Management and is concerned with controlling changes to baselined WPs after they have been released. This is important because changes to released baselines are likely to require rework from later life cycle phases and follow-on projects.

The Baseline defines the set of all WPs which make up the product. A characteristic of these WPs is that a malfunction will occur in the field if the wrong version is used in the product development or delivery. Such WPs must be identified in the baseline with a specific, authorised version.

Other WPs which do not meet this criterion (e.g. reports, plans, schedules, metrics, etc) require a lower level of configuration control, and are referenced via the Project Index by taking the latest released edition.

Change Requests or Fault Reports are Change Control Forms. They are used to process changes to WPs in the baseline and record decisions made at a specific point in time. They are useful reference material, but are not maintained (they have no edition control) and so do not themselves form part of the baseline. Change Requests are a 'delta' on the product, while the Baseline defines the complete content of the product.

4.2 Scope

The purpose of the Baseline Management Process is to provide configuration control which meets the following expectations:

- The WPs to be placed under configuration control in the project are identified at the start of the project.
- The Baseline is maintained throughout the life-cycle of the project, and visible to all project team members and members of dependent/follow-on projects.
- Old versions of the baseline can be retrieved Changes to the baseline are controlled, through the approval of CCBs.
- Changes to the baseline are visible, understood and committed to by all impacted development phases and projects.
- Development work is conducted and products are released using only the official configuration specified in the Baseline.
- SCM audits periodically verify the integrity of the baseline

The Baseline Management Process is applicable to all released WPs in all projects within the TAF TSI deployment. It applies during the whole life time of the product.

The baseline is a full definition of the authorised product, and is the authority for creating a released product for use by another development phase or an end-customer. 'Full Definition' means specifying the precise variant and version which is authorised for a particular purpose.

This process encompasses setting up, publishing, authorisation and maintenance of the Baseline within one or more projects.

To baseline means to create (establish) a baseline. Because not all work products are created at the same time, the phrase to baseline a work product is sometimes used with the meaning of adding a work product to the baseline.

Not all work products are part of a baseline. Examples are: progress reports, meeting minutes, status reports.
Changes to the WPs in a baseline, or addition of new WPs is controlled via CR. A FR indicates that the WP in the baseline does not meet the requirements defined for it in the Baseline, and authorise an update to the WP.

Changes to baselines are authorised by Control Boards. There is a distinction between "functional" Control Boards and Project Control Boards.

4.3 Attributes of a baseline

Identification
Baselines use the defined numbering convention. They can be coded as Items or Documents. For Software or Project related baselines typically a baseline category is used.

Status
Baselines have status values described in the Coding System (see also 5):

- IP In Preparation
- PD Proposal Distributed
- RD Released for Design
- RP Released for Prototype
- RL Released

The status of the baseline can not be higher than the lowest status of its mandatory elements. E.g. If a document baseline contains one document in status IP, the status of the baseline can only be IP. A baseline in status RL will have all components in status RL. Status values RD and RP are not so much used in a document context. They refer more to items (for definition of “item” see chapter 5).

Version
In case a work product is:

- a document, it's version is called EDITION. When a document is in PD-status, an ITERATION is also part of the version: 01P02 would be edition 01, iteration 02, or in other words; the second proposal for the first edition.
- an item, it's version is called ICS.

Using versions, a history can be reconstructed:

<table>
<thead>
<tr>
<th>Ed of baseline</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work product 1</td>
<td>01</td>
<td>02</td>
<td>02</td>
<td>03</td>
</tr>
<tr>
<td>Work product 2</td>
<td>01</td>
<td>01</td>
<td>02</td>
<td>02</td>
</tr>
<tr>
<td>Work product 3</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
</tbody>
</table>

Rows are the work products and their versions (edition or ICS). Columns are the baseline version and the versions of the work products.

Control of a baseline
All changes to the baseline are subject to formal change control so that individual changes can be authorised and tracked at all times.
4.4 Process Description

**Input:**
SW Process Interface Model

**Output:**
Up-to-date Baseline Index (Template and instantiation for each project)
Baseline Notification List (per project)
Recorded CCB decisions
Change Notes showing the new functionality and fault corrections delivered

**Entry Criteria**
First WP has been released.

**Exit Criteria**
Maintenance Phase of the product has been completed.

**Tasks**

*Select WPs Forming the Baseline*
Process interfaces are recorded in a database, based on the list of procedures and processes authorised by the CCB. The CCB selects the WPs that are to be included in the standard Baseline Template, in conjunction with the Change Review Team and each competence area.

*Select WPs Relevant to a Project*
The list of WPs to be used in a particular project are tailored at the start of the project by the Configuration Controller based on the generic baseline, adapted to the scope and requirements of a particular project (e.g. a market which does not use ATOM does not need ATOM WPs listed in the Baseline).

*Identify WP Changes*
The detailed procedures for Change Management are identified for each type of work product in a database.

Two kinds of changes can be distinguished:
- Changes as a consequence of new or changed requirements. This may result in the creation of a variant of a version or variant of an existing work product or in the creation of an entirely new work product.
- Changes as a consequence of problems detected during the development and verification processes. This type of change management is described in detail in a separate Problem Management process. Fault corrections result in a new edition of a document or a new ICS of an item.

The change form must identify all impacted WPs and the nature of the change required.

*Approve WP Changes*
Competence-based Technical Review Boards are in charge of the individual Work Products. These boards are roughly organised by Competence Area, and authorise the changes to WPs in their Competence Area.

In principle, changes to work products must be authorised in two steps:
1. Acceptance of the Change Request.
2. Release of the Work Product(s) affected by this Change Request.

Each Control Board has the option to optimise this procedure by combining the two authorisations into one. This reduces the cycle time but increases the risk of rework in
subsequent phases because one of the changes does not meet the customers' needs. However, in many cases this risk is smaller than the overhead (in cost and time) caused by the standard two-step procedure.

Follow the functional procedures to create or update the WP, as required in the CR or FR.

For the evaluation of the technical quality of the work products, the competence-based boards rely on the results of activities such as Technical Reviews, Inspections, Code Reading, Module and subsequent Test activities.

WP Identification

Work Products to be included in the baseline must be uniquely identified with edition, variant and document type within the CM Library, by the numbering scheme.

Store WPs in CM Libraries

CM Libraries provide the facilities required to effectively manage WPs which form part of the Baseline, including edition numbering, history, access to old versions, status recording, etc. The steps up to inclusion in an CM library are generally covered in the Work Product Management. WPs are normally set to Released status in the CM library before being released to the Baseline.

Traceability

Traceability is the ability to identify the links between one WP and those on which it is based. This is needed to:

- Confirm that all customer requirements have been completely implemented in the product
- Identify the version in which a particular change was incorporated
- Ensure that no unauthorised changes have been made to the product
- Assist identification the impact of an updated WP into the baseline on dependent WPs
- Identify all of the low-level changes implemented in the delivery of a composite WP

WP Change Notification

The delivery of a new version of a work product must be notified to all affected parties. This includes the owners (implementation authorities) of work products which are based on the new or modified work product(s).

Maintain Competence-Based Baselines

Once released by a CA-based board, the current authorised version of each identified WP must be specified in the Baseline Index.

The TPM selects the Project Team to represent all CAs impacted by the project. This member of the project team is responsible to inform the project team of all baseline changes arising from that CA.

Maintain Project Baselines

Based on the updated baseline index, the Project Control Board reviews impacts on the project, and approves inclusion of the updated baseline into the project.

The Release Authority, acting on behalf of the Project Control Board is the Configuration Controller.

The Configuration Controller includes the newly authorised index into the project baseline.
Baseline Change Notification

An update to a Baseline must be notified to all affected parties. This includes the project teams of all impacted projects, owners (implementation authorities) of other baselines and work products which are based in the newly delivered or updated work product.

Baseline Audits

The CM group periodically audits the integrity of the Baseline, CM Libraries and change control procedures.

The QA group periodically audits that the CM processes are being followed, and that the baseline is being maintained.

Actors

CM Council

- Oversees the content and structure of the Baseline, and relationships between different components of the Baseline, across all competence centres.
- Co-ordinates the other CM Groups in maintaining the baseline.

Implementation Authority (IA)

The implementation authority of a work product (the author of a document) is in charge of the implementation of authorised changes in the work product.

Software Configuration Control Boards (SCCB / CCB)

Responsible for authorising changes in baselines, in consultation with all affected parties.

Technical Review Boards (TRBs)

Responsible for authorising changes to WPs within a CA. Distinguished from CCBs in that TRBs:

- have narrower representation, primarily within one CA, rather than all impacted parties
- review the details of a proposed change
- work at the level of individual WPs rather than a complete baseline

Configuration Controller (CC)

The Configuration Controller is a nominated member of the project team responsible for the establishment and maintenance of the Project Baseline, based on authorisation provided by the Project Control Board. The CC calls the meetings, updates the results, and verifies the overall integrity of the project baseline.

Quality Authority (QA)

QA is responsible to periodically audit the activities of the CM Group to ensure that the baseline is correctly maintained.

Exception Handling

For urgent changes, the Project Control Board and TRB co-ordinate their activities through the member of the project team representing that CA.

For urgent changes, it is admissible to accept into the Baseline WPs which are not at Released Status, as the basis of work in later phases. However, the Project Manager must evaluate the risk of instability and approve the potential rework involved. If unreleased WPs are included in the Baseline Index, they must be replaced with a Released version as soon as one is available, since unreleased WPs are not protected against deletion from official archives.

For projects in sustaining phase, a single board may control all field maintenance corrections, in consultation with the external customer.
5 Some special CM Procedures

5.1 Procedure for Configuration Identification

5.1.1 Background/Purpose

This procedure outlines the activities involved in Configuration Identification. An identification scheme reflects the structure of the product and identifies components/items and their type, making them accessible in some form. The objective of configuration identification is to uniquely identify the components of a product and all versions and variants of the product throughout its life to allow the building or rebuilding of any version/variant with the correct components and to facilitate the control of changes to the product. The key to effective configuration management is unambiguous identification of the parts of the software. Every configuration item shall have an identifier that distinguishes it from other configuration items with different:

- Requirements, especially functionality and interfaces
- Implementation.

5.1.2 Roles and Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM Manager</td>
<td>Oversee the establishment of the configuration identification scheme and the identification of the components/items and baselines.</td>
</tr>
<tr>
<td>CM Organisation</td>
<td>Issue the configuration identifier to the components/items and related technical data. Verify that the correct identifiers have been used. Identify the baselines.</td>
</tr>
<tr>
<td>CCB</td>
<td>Approve baselines and authorise changes to the approved baselines.</td>
</tr>
</tbody>
</table>

5.1.3 Activities

5.1.3.1 Procedure Overview

<table>
<thead>
<tr>
<th>Entry Criteria</th>
<th>• New project/version/release is identified or need to add new work products is identified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>• Allocated system requirements (e.g. from Change requests)</td>
</tr>
<tr>
<td></td>
<td>• Identified work products and defined baselines (from Project Planning)</td>
</tr>
<tr>
<td>Steps</td>
<td>1. Select, identify, and classify the components/items to be placed under configuration control.</td>
</tr>
<tr>
<td></td>
<td>2. Create an identification scheme that reflects the structure of the product.</td>
</tr>
<tr>
<td></td>
<td>3. Create a labelling scheme for the storage of the work products (e.g., file names)</td>
</tr>
<tr>
<td></td>
<td>4. Determine the baselines and their contents</td>
</tr>
<tr>
<td></td>
<td>5. Determine the releases and their contents</td>
</tr>
<tr>
<td></td>
<td>6. Establish the CM library</td>
</tr>
<tr>
<td>Outputs</td>
<td>• Items to be placed under configuration control identified</td>
</tr>
<tr>
<td></td>
<td>• CM Identification Scheme</td>
</tr>
<tr>
<td></td>
<td>• CM Labelling Scheme</td>
</tr>
<tr>
<td></td>
<td>• Baseline descriptions</td>
</tr>
<tr>
<td></td>
<td>• Initial Version Description Document (VDD) for each release</td>
</tr>
<tr>
<td></td>
<td>• Initial CM Library Structure</td>
</tr>
<tr>
<td></td>
<td>• Updated CM Plan (CMP)</td>
</tr>
</tbody>
</table>
5.1.3.2 Detailed Procedural Steps

The following paragraphs provide guidance for Configuration Management Identification.

**Select, Identify, and Classify the Components/Items To Be Placed Under Configuration Control**

A `configuration item' (CI) is a collection of elements, treated as a unit, for the purpose of configuration management. Several factors may be relevant in deciding where to draw the boundaries of a configuration item. The primary factor should be to attempt to isolate those parts of the system which are likely to change from those parts of the system which are likely to remain static. In addition, it is worthwhile to isolate those parts of the system which have different requirements for formality (e.g., safety or security critical versus non-critical). A configuration item may be any kind of item, for example: a module, a collection of modules, a document, or a set of CIs.

At a minimum, each component defined in the design process shall be identified as a configuration unit and possess an identifier.

Although the key component to be managed is the source code, all related items, for example documentation, object or reusable code, executable code, files, tools, test software and data, shall be identified and placed under CM control. It is important to store and track all environment information and support tools used throughout the life cycle to ensure that the product can be reproduced.

**Create an Identification Scheme That Reflects the Structure of the Product**

The identifier shall include a number or a name related to the purpose of the CI. The identifier shall include an indication of the type of processing the CI is intended for (e.g., file type information). The identifier shall also include a means to indicate the version of the CI. The term `version' is used to define a stage in the evolution of a CI. Each stage is marked by a `version number'. When the CI changes, the version number changes. The ‘version number’ includes an issue number and a revision number. Issue numbers are used to mark major changes and revision numbers are used to mark minor changes. The issue number and revision number together mark the version of the document. If the product is expected to be produced with different variants, then a provision in the identification scheme should also include a method to identify the variant within the version. The simplest way to accomplish this is to add a third component to the ‘version number’ to identify the variant.

The configuration identification method shall be capable of accommodating new CIs, without requiring the modification of the identifiers of any existing CIs.

The CI naming scheme is used to both identify, display, and maintain relationships between components

- Model the structure of the product via a system model that represents the inventory of components for that product
- Defining relationships and interfaces between the various products
- Identify way of partitioning the product for limiting the effects of changes to it
As part of the configuration identification method, a module shall have a header that at a minimum includes:

- System of which it is a part (if applicable)
- Configuration item/unit identifier (name, type, version)
- Original author
- Creation date
- Change history (version/date/author/description).

Create a Labelling Scheme for the Storage of the Work Products

A labelling scheme for the storage of work products (e.g., file names) and which may be used for organisational look-up and for CI identification needs to be created. This scheme may include the need for a cross reference file to accommodate short file names or organisational CI numbering schemes which are not description of the contents of the file names.

All documentation and storage media shall be clearly labelled in a standard format, with at least the following data:

- Project name
- Configuration item identifier (name, type, version)
- Date
- Content description.

Determine the Baselines and their Contents

A ‘baseline’ is a work product or collection of work products that has been formally reviewed and agreed upon, and is a basis for further development. A baseline is an assembly of configuration items. When the baseline is established, its component item versions are frozen. Formal change control procedures are then required to apply changes to a baseline thus creating new versions of the components.

Baselines should generally be established at selected life cycle phase transition milestones. For example, a requirements baseline would be established at the end of the requirements analysis and specification phase after the SRS has been reviewed and approved. This initiates formal change control on the SRS and any changes to the requirements must go through the change control process.

Supporting work products also need to be baselined, generally as a part of the work products baselines, but, if important enough, may be baselined on their own. These include tools, compilers, operating systems, simulators, etc. They are an integral part in the creation of the product, and since part of the reason for baselining is to be able to recreate releases, access to the specific tools and supporting products may be vital.

Versions of the work products may already have been baselined for other projects, and if a project intends to use them ‘as is’ they may be referenced as having been baselined and including the identifier of the baselined work product. If the intent is to modify the support work products, then they will have to be baselined as a part of the new project.

Along with identifying the baseline contents, the creation criteria for the baselines needs to be determined and recorded. For example, the SRS needs to be formally reviewed and
approved, or for a module, the module test plan must have been executed with the actual results equalling the expected results and the module must have successfully exited a formal inspection.

The identification of the baselines, their contents, and their creation criteria shall be recorded in the CMP.

**Determine the Releases and Their Contents**

A release is ‘version’ of the product that will be ‘released’ to the customer and the work products used to produce it. The supporting work products may or may not be delivered to the customer depending upon the arrangements which have been made. A release is composed of a specific versions of the components which make up the product and the support work products. A VDD is used to describe the contents of the release and is used as a part of the configuration control mechanism for the release. Every component is identified in the VDD (this identification includes the version of the component; for example it should be identified as MS Windows v3.1 rather than simply MS Windows).

**Establish the CM library**

In this step the CM library is established. The library is the heart of the CM system. It serves as the repository for the work products created during the life cycle. Repositories or libraries are needed to store and capture CM information as well as the different kinds of components such as source and object code, executables, diagrams, documentation and baselines.

The three main types of software libraries which are maintained by CM are the:

- Development (or Dynamic) library
- Master (or Controlled) library
- Static (or Archive) library

Software is coded and tested as a set of modules in the development library. After unit tests, modules are transferred (sometimes called promoted) to master libraries for integration testing and system testing. When changes to master library modules are necessary, the appropriate modules are transferred back to a development library from the master library after the appropriate authorisation has been given. When a baseline is established, the master libraries for the baseline are copied into the static libraries for archival purposes. The copies in the static library are never changed (hence the designation static).

The structure of the library should handle currently planned baselines and easily support the addition of new baselines throughout the life of the product. The libraries should be structured to accommodate the need to promote work products from the essentially unrestricted development library, through ever increasingly controlled environments, until they are finally (with CCB authorisation) promoted into the baseline library, where no further changes that version/variant are allowed.
5.1.4 Checklists

<table>
<thead>
<tr>
<th>No.</th>
<th>Activity</th>
<th>Yes</th>
<th>N/A</th>
<th>Signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All known components/items (including related items) have been identified.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The identification scheme is documented.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The identifiers include both an issue (for major changes) and a revision (for minor changes, which together create the version number).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Baselines have been identified and their contents have been determined.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>All deliverable items and related items necessary to produce the deliverables are allocated to baselines.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Baseline creation criteria have been documented</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>The release and their contents have been determined and a VDD for each release has been created</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>The CM library has been established.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The above checklist contains some questions that can be used to ensure compliance with this procedure. This checklist is meant to be a starting point and should be modified to meet specific organisational and project needs.

5.2 Procedure for Configuration Status Accounting

5.2.1 Background/Purpose

This procedure outlines the activities involved in performing configuration status accounting. Configuration status accounting is an element of Configuration Management (CM) that consists of the recording and reporting of information needed to manage a configuration effectively. This process helps to maintain configuration integrity during change control periods by ensuring that the status accounting records, documents, and software listings are compatible. This process also ensures that the Configuration Status Accounting Report (CSAR) reflects the current approved and validated versions of project documentation and the CM library inventory.

5.2.2 Roles and Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Management</td>
<td>Performs configuration status accounting and publishes a report that is distributed to all project management</td>
</tr>
<tr>
<td>Software Manager</td>
<td>Reviews CSAR</td>
</tr>
<tr>
<td>Project Manager</td>
<td>Reviews CSAR</td>
</tr>
</tbody>
</table>

5.2.3 Activities

5.2.3.1 Procedure Overview

<table>
<thead>
<tr>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Document library</td>
</tr>
</tbody>
</table>
5.2.3.2 Detailed Procedural Steps

The following paragraphs provide guidance for performing Configuration Status Accounting.

**Create the Configuration Status Accounting Report**

The CSAR is a periodic report that consists of several different configuration management reports, based on items under configuration control, collected in one document. The contents of the CSAR varies from project to project and depending upon the purpose for the report. The frequency of the CSAR is defined in the Configuration Management Plan (CMP).

Typically the CSAR contains the status of the CM library and tracking system over a specific period of time. Example of types of information that appear in a CSAR are: Action Item status reports, Submittal and Release reports, Library Inventory reports, Problem Report (PR) and Change Request (CR) status reports. The actual content of the CSAR is defined in the CMP.

**Action Item Status Report**

An action item status report consists of the status of all action items that are under configuration control. Usually the report is in tabular form that lists the originator, date submitted or opened, the action, the action item number, suspense date, the person the action is assigned to, and the date the action item is closed. Typically, the action item report lists closed action items for two reporting periods. This report should be generated from the CM tracking system.

**Submittal and Release Report**

A submittal and release report consists of the number of submittals and releases CM has processed, and a description of their contents. The report is in tabular form and contains requester's name, date of request, type of request (submittal or release), date request was
fulfilled, and the contents of the request. This report should be generated from the CM tracking system.

**Library Inventory Report**

The Document Library report lists all project documents; identifies specifications, design documents, and drawings; identifies projected changes; tracks implemented changes; and identifies the compatibility between the documentation and the work product version. This report should be generated from the CM tracking system.

**PR and CR Status Report**

The PR and CR status report summarises all PRs and CRs associated with developmental configuration; provides statistical summary of total PRs and CRs, open PRs and CRs, and closed PRs and CRs; identifies reported problems against each version; tracks each PR and CR against specified functions; identifies modules and documentation affected by each PR and CR; tracks version in which trouble was corrected; identifies the originator of the PR and CR; and identifies the project activities affected by the PR and CR.

**Distribute the Configuration Status Accounting Report**

CM distributes the CSAR to all necessary personnel. The distribution list for the CSAR at a minimum includes the Project Manager and Quality Assurance. The project distribution list is defined in the Development Plan (DP) and the CMP.

**5.2.4 Checklists**

<table>
<thead>
<tr>
<th>No.</th>
<th>Activity</th>
<th>Yes</th>
<th>N/A</th>
<th>Signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create a Configuration Status Accounting Report.</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Collect and document appropriate metrics.</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Collect and document lessons learned</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Distribute Configuration Status Accounting Report to all in project distribution list.</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

The above checklist contains some questions that can be used to ensure compliance with this procedure. This checklist is meant to be a starting point and should be modified to meet specific organisational and project needs.

**5.3 Procedure for Conducting Configuration Audits**

**5.3.1 Background/Purpose**

A Configuration Audit verifies that the product is built according to requirements, standards, and/or contractual agreement. Test reports and documentation (such as the Requirements Traceability Matrix) are used to verify that the system, hardware or software meets the stated requirements. The goal of a Configuration Audit is to verify that all work products have been produced, correctly identified and described, and that all change requests have been resolved in accordance with the Configuration Management (CM) Plan. Informal audits are conducted at key phases of the life cycle.
Configuration Management

5.3.2 Roles and responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Control Board (CCB)</td>
<td>Reviews and approves the result of the audit as a part of the process to establish a production baseline and deliver the product</td>
</tr>
<tr>
<td>Configuration Manager</td>
<td>Responsible for the conduct of the configuration audits – that it is planned, that adequate staff is available, and that the procedures are followed. Reports the results to the CCB</td>
</tr>
<tr>
<td>Configuration Management Staff</td>
<td>Conducts the configuration audit according to the plan and the procedures. Prepares the audit report for the SCM Manager</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>Audits the audit to ensure that all procedures have been followed and that the findings have been accurately reported</td>
</tr>
</tbody>
</table>

5.3.3 Activities

5.3.3.1 Procedure Overview

| Entry Criteria                             | • Approved Release Request  
|                                           | • Approval of the CCB to proceed   |
| Inputs                                    | • Version Description Document (VDD)  
|                                           | • Requirements Traceability Matrix (RTM)  
|                                           | • All life cycle products and documents  
|                                           | • Test Plans, Procedures, and Reports  
|                                           | • Verification and Validation Reports  
|                                           | • Configuration Status Accounting Reports  |
| Steps                                     | 1. Compare the products produced during the life cycle to those listed in the VDD  
|                                           | 2. Check the requirements against the subsequent life cycle products  
|                                           | 3. Check the design against the product  
|                                           | 4. Trace Change Requests (SCR)/ Problem Reports (PR) into the affected work products  
|                                           | 5. Examine verification and validation reports  
|                                           | 6. Examine the expected versus actual test results  
|                                           | 7. Examine the configuration status accounting reports  
|                                           | 8. Prepare the Configuration Audit report  |
| Outputs                                   | • Configuration Audit Report  
|                                           | • Problem Reports  |
| Metrics                                   | • Number of items (e.g., requirements, design component) checked  
|                                           | • Number of defects by product type and by item  |
| Exit Criteria                             | • Completed and signed Configuration Audit Report |

5.3.3.2 Detailed Procedural Steps

The following paragraphs provide guidance for conducting Configuration Audits.
Compare the Products Produced During the Life Cycle To Those Listed In the VDD

This step ensures that all the products which were to be produced for delivery have actually been produced and are either in the master baseline library or in the static baseline library and that they are the correct versions. As a part of this comparison, the auditor will examine each product to determine its actual version and compare that to what is listed in the VDD. The auditor will also examine supporting documents to ensure that for each of the products the appropriate process was followed (e.g., all reviews were performed, the CCB (or appropriate authority) approved the promotion of the product to the master library, etc.).

Check the Requirements Against the Subsequent Life Cycle Products

Check the requirements against the subsequent life cycle products (e.g., documents) using the RTM to determine that all the work products are consistent with the RTM and thus with each other. This involves first examining the RTM to ensure that all requirements trace through all work products to the deliverable product. Although it would be best if every requirement could be verified that it was correctly addressed by the work products, there are normally not enough resources for that except in very small systems. Therefore, we have the following two ways of selecting requirements for verification. For each critical requirement, each work product shall be examined to verify that the requirement is addressed by the work product (e.g., the design should be checked to verify that the requirement is incorporated in the design, the test cases include cases to test the requirement, etc.). For the non-critical requirements a random sample of the requirements shall be verified against the work products. The following table is given as a guide to the number of requirements to be randomly selected.

<table>
<thead>
<tr>
<th>Number of Requirements</th>
<th>Sampling Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 100</td>
<td>&gt; 20 %</td>
</tr>
<tr>
<td>100 - 500</td>
<td>10 %</td>
</tr>
<tr>
<td>500-1000</td>
<td>5 %</td>
</tr>
<tr>
<td>1000 - 10000</td>
<td>1 %</td>
</tr>
<tr>
<td>&gt; 10000</td>
<td>&gt; 0.1 %</td>
</tr>
</tbody>
</table>

You want to select the largest sample size which your resources (time and staff) will allow. The table gives recommended minimum sample sizes.

Check the Design Against the Product

In this step it is being verified that the design was implemented as specified. This step should answer the questions, have all the specified components been built and have they been integrated into the product. This is accomplished by comparing the design specifications against the components that have been built and integrated into the product. The person performing this activity will need to understand the code in order to determine if it accurately reflects what is in the design specification. They will also need to check the architecture as implemented versus the designed architecture. Additionally, the test plan and case need to be verified to ensure they test the design (while not forgetting that they must also test to ensure that the requirements are implemented), and the documentation such as User’s Manuals also need to be verified to see that they accurately reflect the design of the system and the as built system.
Trace CRs/PRs into the Affected Work Products

In this step, all closed CRs/PRs are traced into the affected work products to ensure that the changes and/or corrections have been made to the correct version. Additionally, they should be checked to determine that all of the needed work products were changed and that all procedures were followed.

Examine Verification and Validation Reports

In this step the verification and validation reports are examined to determine if all defects found have been addressed. Additionally, these reports are verified against the Verification and Validation Plan to ensure that all V&V activities have been performed consistently with the plans, that the results are as expected, and where the results were not as expected, adequate measures were taken to address the deficiencies.

Examine the Expected Versus Actual Test Results

In this step the expected versus actual test results for module, integration, regression, and other tests are examined to help determine if the system operates as specified. Additionally, the test reports are compared to the Test Plans to ensure that all tests have been run, that the results are as expected, and where results were not as expected, Problem Reports (PRs) were generated to address the deficiencies. The handling of the PRs will be addressed in the next step.

Examine the Configuration Status Accounting Reports

In this step the Configuration Status Accounting Reports are examined to determine what actions/changes/problems are unresolved at audit time and to ensure that the unresolved issues have CCB authorised waivers before delivery to the customer. The goal of this step is not to ensure that there are no known defects in the system to be shipped, but that the CCB is aware of the remaining known defects and has authorised shipment with the knowledge of those defects.

Prepare the Configuration Audit Report

In this step, the Configuration Audit Report is prepared. The primary purpose of this report is to inform the CCB of the status of the system in order that they may make the decision on whether or not to approve the production baseline and authorise delivery to the customer. That is why this report spans all of the activities in the production of the system. In order for the CCB to authorise the baseline and delivery, the report needs to be able to state that:

- The system as built is consistent with the Requirements Specification
- All work products are consistent with each other and the system as built
- All known deficiencies/defects in this release have CCB authorised waivers

The reports shall discuss the activities which were performed during the audit. The report shall note any defects or deviations which were found during the course of the audit. The report shall contain a discussion of any potential areas for improvement, either in the process or the product. The report shall recommend whether or not the CCB should approve the production baseline and authorise delivery to the customer.
### 5.3.4 Checklists

<table>
<thead>
<tr>
<th>No.</th>
<th>Activity</th>
<th>Yes</th>
<th>N/A</th>
<th>Signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>There is an approved release request.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The master library components for this release are consistent with those listed in the VDD.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The requirements are traceable through all of the life cycle products and into the deliverable product</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>The system as built is consistent with the design specifications.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>All closed CRs/PRs have been traced into their affected work products.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>All outstanding defects or deficiencies have CCB authorised waivers.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>The Configuration Audit Report has been produced and addresses all of the required topics.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The above checklist contains some questions that can be used to ensure compliance with this procedure. This checklist is meant to be a starting point and should be modified to meet specific organisational and project needs.

### 5.4 Procedure for Library Maintenance

#### 5.4.1 Background/Purpose

The maintenance of the Software Configuration Management Library and the baselines contained are at the heart maintaining the control and integrity of configuration items. These procedures describe the processes to control access to the configuration items and baselines. A number of short procedures have been combined under library maintenance since they use the same authorisation and control process. The individual actions covered under this procedure are submit material, promote modules, check in, check out, and archive/backup. The procedure for product release is more complex and is covered under a separate procedure.

#### 5.4.2 Roles and Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>Approve request</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Check for appropriate authorisation, obtain authorisation from Configuration Control Board (CCB) if necessary, perform request</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>Perform process audits and independent reviews to ensure all steps have been accomplished in the prescribed manner</td>
</tr>
<tr>
<td>Configuration Control Board</td>
<td>Approve the actions and changes to the baseline</td>
</tr>
</tbody>
</table>
5.4.3 Activities

5.4.3.1 Procedure Overview

| **Entry Criteria** | • Established library of project items  
• Problem Report (PR), Change Request (CR), or any other item that changes the CM baselines  
• Request for action by CM library |
| **Inputs** | • Request for library action  
• Project materials to be baselined or controlled (CM products): Code, CMP, QAP, DP, Test Plan, Problem Reports  
• Request supporting material (CCB authorisation, Audit reports, test reports, etc.) |
| **Steps** | 1. Receive request  
2. Analyse request  
3. If appropriate obtain CCB authorisation  
4. Perform appropriate procedure  
5. Close the request |
| **Outputs** | • Updates to the action tracking system, CM baseline information  
• CM products  
• Current technical baseline |
| **Metrics** | • Number of requests closed  
• Number of requests opened  
• Number of hours spent controlling CM baselines  
• Amount of time an open request was in the system |
| **Exit Criteria** | • QA review, as required  
• CM and project files updated  
• Lessons learned collected and documented  
• Metrics collected and documented |

5.4.3.2 Detailed Procedural Steps

The following paragraphs provide guidance for

**Receive Request**

The request of which there are several types (e.g., submit material, promote modules, check in, check out, baseline a work product, and product release) is submitted to CM. When received, the request will first be entered into the action tracking system. It will then be checked to ensure that all appropriate material and authorisations have been submitted with the request and that all requested items are available in the required status. If any required material is missing, the submitter will be notified and requested to provide the required material.

<table>
<thead>
<tr>
<th><strong>Type of Submission</strong></th>
<th><strong>Required Material</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Submit Material</td>
<td>Material Submission Form signed by the authority designated in the Software Configuration Management Plan (CMP), material in both hard copy and electronic media where appropriate</td>
</tr>
<tr>
<td>Promote Modules</td>
<td>Module Promotion Form signed by the authority designated in the CMP, copy of module test results</td>
</tr>
</tbody>
</table>
| Check-Out              | For a baselined item - CCB authorisation to change module (form – may be part hardcopy and part electronic media, see Change Control Procedures for more details)  
For a non-baselined item - Authorisation from the authority designated in the |
Check-In

For a baselined item - CCB authorisation to change module (see note on form above), signed by the authority(s) designated in the CMP as completed, copy of test results
For a non-baselined item - signed authorisation from the authority designated in the CMP

Baseline a Work Product

The work product (can be code, an OS, a compiler, an SRS, etc.), request for baselining signed by the authority designated in the CMP

Product Release

Release Request for a product signed by the authority designated in the CMP

**Analyse Request**

The request is analysed to determine the appropriate course of action. For example, in the submit materials, if the materials will not be incorporated into a baseline (e.g., Development Plans, CM Plans, etc.) but are being submitted for purposes of version control, no CCB decision is necessary, and the material may be placed under CM control. If however, a product release is being requested, or the material submitted is to be baselined, then CCB authorisation is needed. If the submission is a module promotion or a check-in, the test results shall be examined to determine if the module is in condition to be promoted/checked-in. If the submission is a check-in, the documentation shall be checked to ensure that all the changes have been made to all the documentation (not just the code).

**If Appropriate Obtain CCB Authorisation**

One of the results of the analysis step shall be whether or not CCB authorisation is needed. If it is not, go to the next step. If this is a non-emergency request, place the request on the agenda for the next CCB. If it is an emergency request, then an emergency CCB meeting needs to be held to discuss the emergency request. If the CCB authorises the action, then go to the next step, else, notify the submitter that the CCB has not approved their request.

**Perform Appropriate Procedure**

The table below provides the procedure for each submittal type.

<table>
<thead>
<tr>
<th>Type of Submission</th>
<th>Procedure</th>
</tr>
</thead>
</table>
| Submit Material    | • If there is no proper directory to hold the submitted material, create a directory  
                    • Copy the material into the proper directory assigning the user designated version  
                    • Enter the reference to the material into the index of library components |
| Promote Modules    | • If the directory for this module has not been created, create it  
                    • Copy the module into the directory, assigning the correct version to resulting copy  
                    • Enter the reference to the module into the index |
| Check-Out          | • Provide a read/write copy or give read/write access to the module(s) to the submitter  
                    • Log the check-out in the check-out log |
| Check-In           | • Examine the checked-in material to ensure that all changes have been made and tested  
                    • Examine related documentation to ensure that it has been modified to reflect the changes and if it has not note it in the index that changes remain undone  
                    • Copy the module into the directory, assigning the correct version to resulting copy |

**Configuration Management**
Version 1.0 12.10.2004
Baseline a Work Product

- Copy the version to the appropriate directory and mark it in the index as a baselined version
- Write protect the version
- Ensure the VDD indicates that the work product has been baselined
- Ensure a backup/archive is executed

Product Release

- Perform the Procedure for Product Release

**Close the Request**

Log the completion in the action tracking system. Notify the submitter on the action taken.

### 5.4.4 Checklists

<table>
<thead>
<tr>
<th>No.</th>
<th>Activity</th>
<th>Yes</th>
<th>N/A</th>
<th>Signed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Does all the necessary material accompany the submittal?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Has the appropriate authorisation been obtained?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Has the appropriate action been taken?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Has the submitter been notified of the results?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>If it was a CCB Authorised action, has the CCB been notified of the results?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>If material has been accepted, has the backup/archive been scheduled?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The above checklist contains some questions that can be used to ensure compliance with this procedure. This checklist is meant to be a starting point and should be modified to meet specific organisational and project needs.
6 Appendix A: Definition

Definition of Work products

The result of an activity is a work product. Examples are design documents, program sources, development plans, progress reports, test specifications, executable, etc. A work product can be an item (an item can be logical, physical or electronic) or a document (typically an information item intended to be read by humans -'documents' in the classical sense of the word- or machines -'data'-).

A registered work product is work product identified with a unique identification, typically by the use of a numbering convention or another depending on the CM library.

Work product attributes

Different attributes are given to a work product:

- A type: for example a source for an electronic item or a design document for a document. Several aspects of an item can be covered with several documents associated to this item. They may have the same number as the item and may be distinguished from each other by an additional document type indicator.
- A version: this is a change attribute. This term can also involve the Item Change Status (ICS) term, the edition/iteration term, and the project term…
- A status: for example IP (In Preparation), or RL (Re-Leased, so can be used) …

Change Notes

Change Notes are used to announce changes and to list the work products affected (for trace ability). They register the delivery of a new type or version of a work product.

CM libraries

Because of the diversity in work products and associated procedures, a range of CM Libraries is used to manage them. Services are offered by these libraries as implementation of access restrictions on assigned authorisation, status control, verification of consistency …

Work Product Life Cycle

The life cycle of work products involves the following steps:

- Creation of the Work product, individually or by recording the results of workshops. This step may involve informal verification activities.
- Formal technical verification (Peer Review, Inspection, Test). During this step the work product must be formally identified and available in a library. It is under developmental change control.
- Appraisal (by the Control Board) and formal release (by the Release Authority)
- Change and Problem Management is performed
- End of life by reaching a withdrawn status (WD or WR status)
Problem

A problem arises if an observation is in conflict with expectations. Observations are made during reviews, inspections, code reading, testing, customer acceptance etc. Expectations are based on specifications, interpretations of specifications or sometimes even personal opinions.

Problems can be caused by a fault (defect) in the system or by a shortcoming in its description. They can also result from wrong expectations. Conversely, if defects are never detected they do not cause a problem. Problems are a measure for the quality of the work product and the process, in particular the number of problems detected by the customer.

Fault Reports and Change Request

Fault Reports are used to register and track faults or problems. Change Requests are used to register changes against work products. Change forms (FR/CR) attributes: Change forms have status values, and receive a severity level (among other characteristics).

Control Board / Change Review Board

It is recommended to use a global change and problem management system with as much as possible the same set of forms for all kinds of work products (items and documents).

Changes forms (FR/CR) are recorded into a database. The relevant control board assigns them to a person or a group. The status of the progress and the elements for the evolution are visible in the record.

Comment facilities

In many cases a comment facility is integrated with the CM-library. A comment mechanism can be a valid alternative to formal change control in the following cases: no CRB decision is needed, stand-alone documents not linked to items (studies, reports, internal documents…)

Change Form (FR/CR) Life Cycle

The change form life cycle usually involves the following steps:

- Creation phase
- Analysis phase
- Correction phase
- Verification phase

Product Control Plan and Category

Items are organised in a Product Control Plan (PCP) and category within a Product Control Plan. A work product belongs to only one PCP/CATEGORY. A category is containing a group of work products of the same type. (expl: objects/executables are all grouped together in a category named OBJ). Documents belonging to that work product have the same PCP/CATEGORY as that work product. Each number can contain a set of document types that must (may) be associated with an work product of that category.
For example, a Process has the category PROCSS. The Configuration Management Process is an instance of this category. The description of the process is mandatory, the process presentation is optional.

A Product Control Plan (PCP) defines the structure of a product and its documentation. It is a structured collection of categories, each with a list of mandatory and optional document types. The categories may be structured hierarchically by means of reference categories. Each item has one reference item (parent) in one of its reference categories.

Typically a Product Control Plan defines following category groups:

- Product Management
- Hardware
- Software
- Hardware Dependent Software
- Customer Documentation

**Work Product Version Management**

In order to address every individual version of a work product, it is not enough to give the work product a unique identification number in the Product Control Plan.

Each version gets a unique identification in the form of the work product number as it is defined in the PCP, extended with a Version identification. This identification is built up from a "Variant" code and a "Product Change Status" PCS.

**Product Control Database**

Most attributes or 'meta data' of work products (Number, version, title, status,...) are recorded in the Product Control Database.

**Variant**

A dedicated Variant code is assigned to every "functionally" different version of a work product. This Variant code makes it easy to recognise which projects share the same and which projects apply different functional versions of a given work product.

In practice this means that when a work product is initially created or when a functionally different version is derived from an existing version in order to implement the requirements of a project, a functional variant is assigned to the new version.

However, when an update is made to an existing version that does not alter the functional behaviour (e.g. error fix), the created version will keep the same Variant code as its predecessor. In this case, the new version will get a unique identification through the PCS, which is described below.

A Variant has four characters of which

- First 2 characters: functional variant (e.g. changed functionality in a project)
• Next 2 characters: realisation variant (e.g. same functionality, but in another language, other platform, ...)

**Product Change Status (PCS)**

To identify all work product versions that share the same Variant code, the Variant identification is extended with a "Product Change Status" (PCS).

Within the collection of work product versions having the same Variant, each one gets a unique PCS.

The PCS is built up out of a major number and a minor number: xxy

- **xx: Major number** -- Is upgraded with every incremental feature addition during the development of one specific functional variant of the work product. Typically, a feature change is described in a Detailed Design Change Request (DD-CR), which is linked to a project's increment. In this case the module is listed as reportable item in the DD-CR.

  The major number is a two digit decimal number, starting from ONE ("01", "02", ... "99").

- **y: Minor number** -- Is stepped up for every bugfix.

  When the major number is incremented, the minor number is blanked out. For further minor deliveries, which are related to later bugfix deliveries within the same major number, the minor number is incremented.

  When not blank, the minor number is an uppercase letter (<blank>, "A", "B", ... "S").

**Status of work products**

**Items** have typically following status's:

- IP: In Preparation, the item is probably being prepared
- PD: Proposal Distributed, the item can be used for review
- RD: Released for design, the item can be used in a design environment
- RP: Released for prototype, the item can be used as a prototype
- RL: Released, the item can be used
- WD: Withdrawn and deleted
- WR: Withdrawn and replaced (by other item)

A second attribute of an item is the version or ICS: the Item Change Status. A higher ICS indicates a bug fix of the workproduct.

**Documents** have typically following status's

- IP: In Preparation, the document is probably being prepared,
- PD: Proposal Distributed, the document can be only be used for review
- RL: Released, the item can be used
- WD: Withdrawn and delete
A second attribute of a document is the version or edition.

CM Authorities

Three basic types of CM authorities can be distinguished:

<table>
<thead>
<tr>
<th>IA</th>
<th>Implementation Authority</th>
<th>The IA is the owner or author of a work product. IA's are recorded in the Product Control Database.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA / CRB</td>
<td>Appraisal Authority/Change Review Board</td>
<td>The AA authorises changes to work products and also authorises the release of work products. The AA is usually the chairperson of the CRB. A CRB is also referred to as Control Board.</td>
</tr>
<tr>
<td>RA</td>
<td>Release Authority</td>
<td>The RA implements the decisions of the AA.</td>
</tr>
</tbody>
</table>

The CM authorities are listed together with the other project responsibilities in the Project Plan.

All work products under formal configuration control are released in a project baseline. The Baseline is a list of one or more work product versions that serve as the basis for work. For more details refer to "Baseline Management".

Control Boards

Baselines and changes to baselines are co-ordinated in the "SW Configuration Control Boards".

Formal Change Control

Three types of forms are used as a vehicle for formal change control on work products:

\{TR\} -- Trouble Reports
\{CR\} -- Change Requests
\{DN\} -- Delivery Notes

Trouble Reports and Change Requests may have more than one affected item. Delivery Notes have one and only one affected item.

Trouble Reports are used to register and track faults or problems, a difference between observed and expected behaviour. There is normally no restriction on who is allowed to register trouble reports.

Change Requests are used to register request for changes in the functionality. There is normally no restriction on who is allowed to submit CRs.
Delivery Notes are used to register the delivery of a new variant or version of a work product. Delivery Notes have one or more accepted CRs or FRs as parent. Change Notes are approved by the Release Authority of a work product.

**Traceability**

The parent-child relationship between TR/CRs and DNs provide bi-directional traceability from problem report and change request to implementation and back.

```
TR/CR (Trouble Report / Change Request)  
|   |   |   |   |   |   |   |   |
 v   v   v   v   v   v   v   v
DN (Delivery Note)  
|   |   |   |   |   |   |   |
 v   v   v   v   v   v   v
WP (Work Product)  
```

To indicate relationships between successive development phases, also parent-child links between the CRs of these phases are defined.

```
CR ---> CR ---> TR/CR  
|   |   |   |   |   |   |
 v   v   v   v   v   v   
DN  DN  DN  
|   |   |   |   |   |   |
 v   v   v   v   v   v   
WP (TLD)  WP (DD)  WP (code)  
```

**Change Management**

Change Management is the authorisation of changes. Refer to the "Change Management process".

**Problem Management**

Problem Management is how we deal with problems reported against work products. Refer to the "Problem Management process" for details.

**Documentation Management**

A document is typically an information item intended to be read by humans. ('documents' in the classical sense of the word) or by machines ('data'). For more details, refer to "Documentation Management”

**CM Libraries**

Because of the diversity in work products and associated procedures, a range of CM libraries is used to accommodate the needs of each process. Services offered by these libraries are:

- Check-in/check-out with locking of work products.
- Check-out for revision (proposal), variant and approval (official edition).
• Interface to an access control system to implement access restrictions based on assigned authorisations.
• Logging of actions.
• Automatic generation of Change Notes in the Change Control System.
• Generation of selection list of approved Change Control Forms for inclusion in Change Note.
• Verification of consistency with Product Control Plan (see Planning) and Number register.
• Interface to electronic mail services for notification of library events.
• Status control of work products.
• Generation of status reports on work products and baselines.
• Registration of ownership.
• Support for automatic generation of derived products.
• Easy access to the presentation form of work products

Build Management

Industrialisation ensures that all SCM-related records are retained through the life of the project. Per produced Build Load, this department keeps track of which work product versions were combined into the Load. How a Load is composed and built must be described in process documents.

Planning

For each project, the PL establishes and maintains a SW Configuration Management Plan. The CM plan formulates how the Configuration Management process is applied in a project and is published through the project's Project Index Page.

Developmental Configuration Management

Work Products are already placed in CM Libraries prior to release. This is necessary to support unique identification, version control and status control during the development process that precedes the release. Formal Change Control is not applied during this phase. The work products are said to be under "Developmental Configuration Management".

END OF DOCUMENT