Project TTC Toronto

Passenger Information System

Software Configuration Management Plan (SCMP)

CDRL:	Spec Refs:	
Q303E / 168	TS24.32 #168	

Prepared:	Marco Binz	*(Signature on file)*	signed: 2011-04-20
	Software System Integrator	Signature	yyyy-mm-dd
Verified:	Tilo Fuchs	*(Signature on file)*	signed: 2011-04-20
	Software Quality Manager	Signature	yyyy-mm-dd
Approved:	Martin Jaeggi	*(Signature on file)*	signed: 2011-04-20
	Managing Director	Signature	yyyy-mm-dd

This document and its contents are the property of ANNAX information systems AG or its subsidiaries. The reproduction, distribution, utilization or the communication of this document or any part thereof, without express authorisation is strictly prohibited. Offenders will be held liable for the payment of damages.	Identity Number:	29901605
	Revision:	05
	Status:	Approved
	Date (yyyy-mm-dd):	2011-04-20

Revision History

Revision	**Date** (yyyy-mm-dd)	**Initials**	**Description of Changes**
00	2010-04-26	MarBi	First issue
01	2010-07-20	MarBi	Section 3.3.1: Explanation regarding the use of Subversion added. Section 3.3.2: Description added what to track and report while baseline accounting Section 3.7 amended
02	2010-09-03	MarBi	Several Updates regarding review protocol from TTC: • Added TTC CDRL number on cover page • Added an explanation to the links in table 6
03	2010-11-05	MarBi	• Added an explanation regarding the use of [I1558] to Section 1.2 Scope • Removed references to IEEE Std. 730, 830 and 1058 from Table 4: List of Standards (references were not used in the document) • Table 5: List of ANNAX Project Documents: Sorted this list alphabetically • Corrected spelling of software tool “Mantis” • Table 1: List of Acronyms and Definitions: Added missing acronym “SCM” • Minor spelling corrections • Update of Table 9: Baseline Definition to be consistent with information in [ASPMP], Section 1.1.4 Schedule and Budget Summary • Correction in 6.3 SCMP Change Distribution
04	2011-03-03	MarBi	As a result of the SW audit held by Bombardier on 2010-12-03 in Gümligen: • Amended Section 3.6.2 to include stipulation regarding subcontractor’s use of Subversion Sections 3.1.1 and 3.1.2.3: Removed the note related to Comm./PIS PTE and BTE Included review comments from Bombardier review (dated 2011-01-12)
05	2011-04-20	MarBi	Bombardier review comments (document inspection checklist dated 2011-04-04) included

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

This document is the Software Configuration Management Plan (SCMP) for the development of the Passenger Information System (PIS) software for the TTC Toronto project.

The intended audience of this plan includes QA Car Builder for review, TTC for approval and ANNAX engineering, project management, QA members and subcontractors who are involved in this project.

1.2 Scope

The SCMP complies with IEEE Std 828 - IEEE Standard for Software Configuration Management Plans [I828], as modified by IEEE Std. 1558 – IEEE Standard for Software Documentation for Rail Equipment [I1558], Table A.3.

1.2.1 Software Configuration Items Covered

This SCMP covers all Software Configuration Items (SCI) of the Passenger Information System. The SCIs are listed in Section 1.1.1 of the ANNAX Software Project Management Plan [ASPMP]. Details of the SCIs are summarized in the ANNAX Software Configuration Items Summary Table [ASCIST].

1.2.1.1 Lowest Entity Covered

The lowest entity covered by this SCMP is the source code related to the PIS SCIs.

1.2.1.2 Other Entities Covered

Other entities covered by this plan are the software tools needed to generate the PIS software code.

1.2.2 SCM Relationship to Other Configuration Management

The SCMP should be used in conjunction with the Configuration Management Plan (CMP) [ACMP], covering system and hardware configuration management for the TTC Toronto project.

1.2.3 Life Cycle

This SCMP applies to all stages of the Software Development Life Cycle (SDLC) as defined in the ANNAX Software Project Management Plan for the TTC Toronto project [ASPMP], Section 6.1.

The formality of SCM is deeper than in most previous ANNAX projects because of extended documentation requirements. This also implies that the control of SCM activities is more rigor than in most previous ANNAX projects.

1.2.4 Specific Limitations and Constraints

None.

1.3 Assumptions

There is the assumption that tools used for configuration management, like SubVersion, will be highly available throughout the project.

It is assumed that SCI identification as expressed in the Software Configuration Items Summary Table [ASCIST] is confirmed by Bombardier/TTC in an early project phase.

1.4 Acronyms and Definitions

1.4.1 Acronyms

The following acronyms are used within this SCMP.

Table 1: List of Acronyms and Definitions

Acronym	Definition
BT	Bombardier Transportation
BTE	Bench Test Equipment
CCB	Configuration Control Board
CDR	Conceptual Design Review
CDRL	Contract Deliverable Requirement List
CI	Configuration Item
CMP	Configuration Management Plan
Comm.	Communication
DBDD	Database Design Description
FAI	First Article Inspection
FDR	Final Design Review
ICD	Interface Control Document
LFLRV	Low Floor Light Rail Vehicle
LRV	Light Rail Vehicle
Mantis	Problem Reporting Database used by ANNAX development department
OFRIS	Open Framework for Railvox Information Systems
PDR	Preliminary Design Review
PIS	Passenger Information and Communication System
PRP	Project Realisation Plan
PTE	Portable Test Equipment
REC	Railvox Embedded Controller
RFS	Root File System
R&D	Research & Development
SCIST	Software Configuration Item Summary Table
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SDD	Software Design Description

Acronym	Definition
SDLC	Software Development Life Cycle
SDN	Software Delivery Note
SFD	System Functional Description
SPMP	Software Project Management Plan
SQAP	Software Quality Assurance Plan
SRS	Software Requirements Specification
STP	Software Test Plan
SUM	Software User Manual
SVN	Subversion
SVVP	Software Verification and Validation Plan
SVVR	Software Verification and Validation Report
SW	Software
SWATS	Software Acceptance Test Specification
TIR	Test Incident Report
TS	Technical Specification [TS]
TTC	Toronto Transit Commission
V&V	Verification and Validation

1.4.2 Definitions

The following definitions are used within this SCMP.

Table 2: List of Terms and Definitions

Term	Definition
anomaly	Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents. [I1012]
system testing	Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. [I1012]
test case	(A) A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. (B) Documentation specifying inputs, predicted results, and a set of execution conditions for a test item. [I1012]
test design	Documentation specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests. [I1012]

Term	Definition
test plan	(A) A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. (B) A document that describes the technical and management approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, responsibilities, schedules, and required resources for the testing activity. [I1012]
test procedure	(A) Detailed instructions for the setup, execution, and evaluation of results for a given test case. (B) A document containing a set of associated instructions as in (A). (C) Documentation that specifies a sequence of actions for the execution of a test. [I1012]
validation	(A) The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. (B) The process of providing evidence that the software and its associated products satisfy system requirements allocated to software at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions), and satisfy intended use and user needs. [I1012]
verification	(A) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. (B) The process of providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly). [I1012]

1.5 References

1.5.1 Internal Templates

All internal templates can be found under: G:\AXis\90_Internal\Vorlagen

1.5.2 Internal Guidelines

The following guidelines shall be applied. These documents can either be found

- in the document management system (Intranet: http://www.axis.annax.ch/dokverwaltung)
- on the ANNAX knowledge platform (Intranet: http://www.axis.annax.ch)
- in J:\80_Q-Management\20 Q-Dokumente\10 Freigegebene Dokumente\
  20 Prozesse\01 Allgemeine Prozesse Gümligen

Table 3: List of Guidelines

Doc ID	Document Description
[ACGUIL]	299003XX, ANNAX Coding Guidelines
[AHTUM]	100125XX, How to use the Problem Reporting System Mantis to generate a Test Incident Report
[ACLSRC]	155038XX, ANNAX Checklist “Software Release Creation” (German title: Software Release Erstellung Checkliste”)
[APIVWS]]	160474XX, ANNAX Process Instruction “OFRIS Versioning with Subversion”, (German title: “OFRIS Versionsverwaltung mit Subversion”)
[AUMMAN]	100125XX, ANNAX User Manual “Mantis”, (German title: “Bedienungsanleitung Mantis”)
[APISDAD]	VA09, Process Instruction “Storage of Documents and Data”, (German title: Verfahrensanweisung “Ablage von Dokumenten und Daten”)

Note: The most recent version of the documents is applicable.

1.5.3 Standards

The following standards were used as guides to develop this SCM process.

Table 4: List of Standards

Doc ID	Document Description
[I1558]	IEEE Std 1558-2004 - IEEE Standard for Software Documentation for Rail Equipment and Systems
[I1012]	IEEE Std 1012-2004 - IEEE Standard for Software Verification and Validation
[I828]	IEEE Std 828-2005 - IEEE Standard for Software Configuration Management Plans
[CMMI-DEV]	CMMI for Development, Version 1.2, CMU/SEI-2006-TR-008

1.5.4 ANNAX Project Documents

The following ANNAX project documents are referenced within this document:

Table 5: List of ANNAX Project Documents

Doc ID	Document Description
[ACMP]	299018XX, ANNAX Configuration Management Plan for the TTC Toronto project
[ADRP]	201004XX, MS Project file: Developer Resource Plan (ANNAX internal document; German title: Ressourcenplan)
[ASCIST]	299017XX, ANNAX Software Configuration Items Summary Table for the TTC Toronto project
[ASPMP]	299014XX, ANNAX Software Project Management Plan for the TTC Toronto project

Doc ID	Document Description
[ASPSW]	201003XX, MS Project file: Schedule Planning Software (ANNAX internal document; German title: Terminplanung Software)
[ASQAP]	ANNAX Software Quality Assurance Plan for the TTC Toronto project
[ASVVP]	299015XX, ANNAX Software Verification and Validation Plan for the TTC Toronto project

Note: The most recent version of the documents is applicable.

1.5.5 Other References

Table 6: Other References

Doc ID	Document Description
[SUBCL]	http://subclipse.tigris.org/
[COLNX]	http://ccpis.rd.ascom.ch/tikiwiki/tiki-index.php?page=coLinux
[DD]	http://ccpis.rd.ascom.ch/tikiwiki/tiki-index.php?page=dd
[ICDIPPIS]	590017128, Bombardier IP Interface Control Document TCMS - PIS Controller

Note: [COLNX] and [DD] are internal documents to ANNAX and contain confidential information which we are unable to release to Bombardier and TTC. They will be made available for review at our site.

1.6 Maintenance

After internal review and approval, this SCMP will be submitted to Bombardier/TTC for official approval. After Bombardier's/TTC's approval, the SCMP will be placed under configuration management. Subsequent changes to the SCMP will follow the stipulations in this SCMP. The SCMP will be updated by the Software System Integrator as needed.

2 SCM Management

2.1 Organization

Refer to the ANNAX Software Project Management Plan [ASPMP], Section 4.2.

2.2 SCM Responsibilities

There are two main entities involved in performing SCM activities:

- Software System Integrator
- Configuration Control Board (CCB)

2.2.1 SCM Roles

2.2.1.1 Software System Integrator

2.2.1.1.1 Purpose and Objectives

The Software System Integrator will plan and monitor SCM activities and will prepare and update, as needed, the SCMP.

2.2.1.1.2 Membership and Affiliations

The Software System Integrator is member of the ANNAX TTC Toronto project and is an ANNAX employee of the development department.

2.2.1.1.3 Period of Effectivity

This role is effective from start to end of the project.

2.2.1.1.4 Scope of Authority

The Software System Integrator has the overall SCM responsibility and authority for the TTC Toronto project.

2.2.1.1.5 Operational Procedures

The Software System Integrator is integrated in the operational procedures of change management, configuration status accounting, baselining etc. as described in the subsequent sections of this document.

2.2.1.2 Configuration Control Board

2.2.1.2.1 Purpose and Objectives

The CCB is responsible of approving proposed changes. The composition of the CCB will depend on the severity of the proposed change as described in Table 8.

Severity categories are defined in Table 7.

Table 7: Change Severity Classes

Proposed Change Severity	Description
Minor	Estimated impact of less than 3 person days, e. g. bug fixes
Medium	Estimated impact of more than 3 person days.

Proposed Change Severity	Description
High	Impact of more than 1 person month

2.2.1.2.2 Membership and Affiliations

Table 8: CCB Composition, Membership and Affiliation

Proposed Change Severity	Mandatory CCB Members	Membership	Affiliation
Minor	Software System Integrator	See Section 2.2.1.1.2	See Section 2.2.1.1.2
Medium	Software System Integrator, Project Manager	The Project Manager is member of / leads the TTC Toronto project	The Project Manager belongs to the development department
High	Software System Integrator, Project Manager, Business Manager	The Business Manager is member of the steering committee of the TTC Toronto project	ANNAX Business Manager

2.2.1.2.3 Period of Effectivity

The CCB's period of effectivity is from date of first change request to end of project.

2.2.1.2.4 Scope of Authority

See Section 2.2.1.2.1

2.2.1.2.5 Operational Procedures

The CCB is integrated in the operational procedures of change management.

2.3 Applicable Policies, Directives, and Procedures

The following documents apply to SCM within this project:

- ANNAX Process Instruction “Storage of Documents and Data” [APISDAD]
- ANNAX Process Instruction “OFRIS Versioning with Subversion" [APIVWS]
- ANNAX User Manual “Mantis” [AUMMAN]
- ANNAX Checklist “Software Release Creation” [ACLSRC]

2.4 Management of the SCM Process

Development is responsible for the overall SCM Process.

2.4.1 Anticipated Cost of the SCM Process

Anticipated cost of the SCM process was estimated as described in Section 5.1.1 of the ANNAX Software Project Management Plan [ASPMP] and documented in the Project Realisation Plan [APRP].

2.4.2 Independent Surveillance of SCM Activities

SCM activities will be surveyed by the Software Quality Manager as described in the ANNAX Software Quality Assurance Plan [ASQAP].

2.4.3 Risk Identification

Risk identification is described in the ANNAX Software Project Management Plan [ASPMP], Section 5.4. Risks are documented in the Project Realisation Plan [APRP].

2.4.4 Risk Management

Risk identification is described in the ANNAX Software Project Management Plan [ASPMP], Section 5.4.

3 SCM Activities

3.1 Configuration Identification

3.1.1 Identifying Configuration Item

3.1.1.1 Configuration Items

Configuration items for the project are:

- Software documentation
- Software code
- Software parameters
- Software environment.

Software Configuration Items are identified in the ANNAX Software Project Management Plan [ASPMP]. More detailed information related to software code and software environment CIs is provided within in the ANNAX Software Configuration Items Summary Table [ASCIST]. For a list of software documentation SCIs, refer to the SQAP [ASQAP], Section 4.

ANNAX software consists of

- Reused components contained in the software framework OFRIS
- Product specific components
- Customer specific parameters

It is the ANNAX Software System Integrator's responsibility to update the CI lists as appropriate.

3.1.1.2 Baseline Definition

Table 9: Baseline Definition

Baseline	Documentation	Code/Code Environment
PDR	Software Plans SFD ICDs SRS, SDD, STPr documents (preliminary)	None
FDR	Software Plans (updated if necessary) SFD (updated if necessary) ICDs (updated if necessary) SRS, SDD, STPr documents (final) All other SW documents excluding reports and SUM	None
pre-FAI Release (BT)	All SW documents above (updated if necessary)	None

Baseline	Documentation	Code/Code Environment
pre-FAI Release (customer)	All SW documents above (updated if necessary)	None
FAI Release 1.1.0	All SW documents above (updated if necessary) SUM	None
TBD - Releases for Comm./PIS PTE	TBD – Information from BT missing	TBD
TBD - Releases for Comm./PIS BTE	TBD – Information from BT missing	TBD
Release 1.0.0 – Release of SW for Level 3 Test	All SW documents (updated if necessary) excluding reports and SUM	Source code of entire PIS software Related development environment
Release 1.1.0 – Release of SW for Static Vehicle Test	All SW documents (updated if necessary)	Source code of entire PIS software Related development environment
Release 1.2.0 – Release of SW for Dynamic Vehicle Test	All SW documents (updated if necessary)	Source code of entire PIS software Related development environment
Release 1.3.0 – Release of SW for Site Acceptance Test	All SW documents (updated if necessary)	Source code of entire PIS software Related development environment
Final Release 1.4.0 – Release of final vehicle software	All SW documents (updated if necessary)	Source code of entire PIS software Related development environment
As-built 1.5.0	All SW documents (updated if necessary)	Source code of entire PIS software Related development environment

3.1.2 Naming Configuration Item

3.1.2.1 Documentation

All project documentation files will be named by a unique prefix consisting of an 8-digit number created by the document management system “Dokumentenverwaltung”, succeeded by an underscore sign (“_”).

The first digit of the prefix will be “2” for documentation.

The last two digits constitute the version of the document, starting with “00” for the first version, followed by “01” for the first revision. There are no specific rules for the remainder of the file name.

Example:

29901600_SCMP_TTC Toronto.doc.

3.1.2.2 Source Code

In general, source code will be identified by a unique software product name, e.g. 'ProdLion'. Upon release the trunk of all sources will be tagged in the Subversion repository. The tag consists of the release version in the format xx.yy.zz where

xx = Major Release: Major new functionality added/ implementing of additional mayor requirements, new interfaces; Impact of about one month or more for coding and testing. Full regression test required.

yy = Minor Release: Enhancements or changes in existing requirements or bug fixing, with an impact of less than one month for coding and testing; Full regression test required

zz = Revision: Bug fixing with an impact of several days at maximum for coding and testing; Reduced regression test, not all requirements need to be re-tested, the scope of testing can be defined by the test team

3.1.2.3 Executable Code

The compiled executables of all used software products will be assembled to form a project specific collection, referred to as "Image". The Image also contains the project specific release notes that refer to the names and versions of the used software products, stored in the Subversion repositories. The Image is stored in a folder identified by a unique prefix consisting of an 8-digit number.

The first digit of the prefix will be “4” for software.

The last two digits of the 8-digit number constitute the major version of the software, starting with “00” for the initial major software release, followed by “01” for the second software release.

Software releases will be identified by the 8-digit (4nnnnnxx) number described above, extended by 4 digits yy and zz, as follows:

kkknnnnxx.yy.zz

With kkk as defined in the table below

Table 10: Software Identification

Software for	Consists of	Identification kkknnnnxx.yy.zz with kkk =
PSC	Application, libraries, operating system	413
PCU-Box	Application, libraries, operating system	414
CCU	Application, libraries, operating system	414
Displays	Application, libraries, operating system	414
Comm./PIS PTE	TBD	TBD
Backoffice PC	Application, libraries, operating system	460

Software for	Consists of	Identification kkknnnnxx.yy.zz with kkk =
Comm./PIS BTE	TBD	TBD

3.1.2.4 Media

A Flash Card will be used to transfer the PIS software to the PIS components. The Flash Card will be labelled with the part number of the Flash Card.

Furthermore, all software will be stored on CD-ROM especially for use during the Escrow validation activity. This includes all documentation, source code, executables, development environment and manuals.

3.1.2.5 Subcontracted or Commercial Software Identification

Commercial software will be named using the name used by the software's vendor. Subcontracted software naming will be the same as naming of software developed by ANNAX. Identification will be supported by the tool Subversion, since subcontractors will be obliged to use this tool in the same way as ANNAX.

3.1.2.6 Other Configuration Item Identification

Software generation tools will be named using the name used by the software generation tool vendor.

3.1.3 Acquiring Configuration Items

3.1.3.1 Documentation

After their ANNAX internal approval the respective author will store documents in the transfer directory J:\50_Entwicklung\Copy\databank.

The Software Test Manager will weekly move all documents in the directory to the read-only document management system “Dokumentenverwaltung”.

3.1.3.2 Source Code

Source code will be stored in the version-control system “Subversion”.

3.1.3.3 Library and Component Code

The used libraries form an integral part of the ANNAX software and are identified, documented and stored like the source code in the version-control system “Subversion”. The used libraries include:

- LibSstDeviceManagement
- LibSstIntercom
- LibSstMatrixDisplay
- LibZip
- LibSQLite

3.1.3.4 Physical Storage

Configuration items will we physically stored on company file servers which are maintained and daily backed up by the provider Swisscom. Due to service level agreements, it is the responsibility of Swisscom, not ANNAX, to maintain and replace if appropriate storage devices.

Remark: Swisscom is the biggest telecommunication and IT company in Switzerland. The agreement with Swisscom covers all aspects of professional IT services, including protection from viruses, worms, trojans, and replacement of data in case of loss or corruption.

3.1.3.5 Configuration Item Retention and Recovery

Retention of configuration items is described in Section 3.5 of the Process Instruction “Storage of Documents and Data” [APISDAD].

3.1.3.6 Configuration Item Distribution

Configuration items are extracted from their storage location by the Software System Integrator and distributed via email or ftp-Server. The executable configuration items are installed over the network using a service laptop.

3.2 Configuration Control

3.2.1 Requesting Changes

Change Request (CRs) may be external (customer) or internal Change Requests. All CRs will be managed within the bug tracking/change request tool Mantis.

It is the Project Manager’s responsibility to collect and enter external CRs in Mantis. Internal CRs may be entered by any ANNAX staff and will be evaluated, tracked and controlled by the Software System Integrator.

3.2.2 Evaluating Changes

The Software System Integrator will act as the “Manager” within Mantis and will evaluate the CR. He will determine the severity of the change according to Table 7.

3.2.3 Approving or Disapproving Changes

3.2.3.1 Change Approval

All changes will be approved or disapproved by the CCB as described in Section 2.2.

3.2.3.1.1 Software Configuration Control Board

Refer to Section 3.2.3.1.

3.2.3.1.2 Product Configuration Control Board

Refer to Section 3.2.3.1.

3.2.3.1.3 Documentation Configuration Control Board

Refer to Section 3.2.3.1.

3.2.3.1.4 Other Configuration Control Board

Not applicable.

3.2.4 Implementing Changes

3.2.4.1 Change Process

After a change has been implemented, the implementing person will mark the CR as resolved and provide in Mantis additional information related to the CR implementation, including the affected items. The verification and release dates as well as the responsible parties are automatically recorded by Mantis. As soon as the next baseline is created, the Software System Integrator will add the identifier of the new version to the Mantis entry of each CR implemented since the last baseline.

In the course of processing changes, numerous information items are recorded within Mantis, e. g.

- The associated change request(s);
- The names and versions of the affected items;
- Verification/test date and responsible party;
- Release or installation date and responsible party;
- The identifier of the new version.

3.2.4.2 Release Process

The Project Manager together with the involved project team will update the release plan, taking correlation with other CRs into consideration. The release will be carried out according to [ACLSRC].

3.3 Configuration Status Accounting

3.3.1 SCI Status Accounting

Status accounting will be performed on two different levels:

On a higher level, configuration status accounting will be performed using an MS Excel worksheet. This worksheet will be used to capture the status of software documentation, software releases, SCM tools, and of software releases.

On a lower level, configuration status accounting for source code will be through Subversion. Subversion handling is described in Section 3.1.2 and in [APIVWS].

The following data elements will be tracked and reported for each CI and baseline: its initial approved version, the status of requested changes, and the implementation status of approved changes. Reporting will be both on an as needed basis as well as in the context of configuration audits as described in Section 3.4.

Information will be collected by the Software System Integrator and will be used for project progress review and CCB decisions.

Access to Subversion status data is granted for all members of the development team. The MS Excel worksheet is handled as any other project document in the ANNAX document management system “Dokumentenverwaltung”. Only the Software System Integrator is allowed to change the information in the MS Excel Worksheet and project related data in Subversion.

There is no automated system is used for status accounting activities.

3.3.2 SCI Baseline Accounting

Baseline accounting will be performed for all baselines defined in 3.1.1.2 Section using the instruments described in Section 3.3.1.

As a minimum, the following information will be tracked and reported for the SCIs

- Initial approved version

- Status of requested changes
- Implementation status of approved changes

3.4 Configuration Evaluation and Reviews

3.4.1 Formal SCM Audits

Formal SCM audits will occur as Functional Audits and Physical Audits described in Section 6.2.5 and 6.2.6 of the ANNAX Software Quality Assurance Plan [ASQAP]:

Table 11: Functional Audits

Software for	Consists of
Objective	A functional audit will be held prior delivery of the software release for Factory Acceptance Test. The functional audit verifies that requirements specified in the SRS have been met. The SRTM will serve as a tool to check which requirements and changes have been implemented.
Schedule	A functional audit will be held prior delivery of the software release for Factory Acceptance Test.
Procedures	Functional Audits will be held according to Section 6.2.5 of the ANNAX Software Quality Assurance Plan [ASQAP]. The functional audit will be performed as a meeting, witnessed and documented by the Software Quality Manager.
Participants	Software Quality Manager, Project Manager, Software System Integrator, Software Tester
Documentation required to be available for review	As a minimum, the following documents will be reviewed: • Software Requirements Specification • Software Design Description • Software Requirements Traceability Matrix • Software Acceptance Test Procedure • Software Version Description
Procedure for recording deficiencies	The audit will be documented by the Software Quality Manager. The Software Quality Manager will enter deficiencies as change requests into the external Mantis.
Approval criteria	The audit will be considered to be passed only when all corrections have been implemented.
Actions to occur upon approval	Upon approval, the baseline will be finalized and frozen, i.e. will be protected against change.

Table 12: Physical Audits

Software for	Consists of
Objective	A physical audit will be held to verify internal consistency of the software and its documentation, and their readiness to release.

Software for	Consists of
Schedule	Physical audits will be held whenever an official configuration management baseline is drawn.
Procedures	All configuration items (e.g. documents, code) to be placed in the respective baseline shall be checked for consistency. The Software Quality Manager shall be invited to attend the audit, and shall receive and store minutes of the audit meeting. The minutes shall define corrective actions.
Participants	Software Quality Manager, Project Manager, Software System Integrator, Software Tester
Documentation required to be available for review	As a minimum, the following documents will be reviewed: • Software Requirements Specification • Software Design Description • Software Requirements Traceability Matrix • Software Acceptance Test Procedure • Software Version Description
Procedure for recording deficiencies	The audit will be documented by the Software Quality Manager. The Software Quality Manager will enter deficiencies as change requests into the external Mantis.
Approval criteria	Approval may occur as conditional approval or full approval. Conditional approval requires correction of deficiencies found prior to full approval. Full approval will only be granted if no deficiencies are found during the audit.
Actions to occur upon approval	Upon approval, the baseline will be finalized and frozen, i.e. will be protected against change.

3.4.2 In-process SCM Audits

Table 13: In-Process SCM Audits

Software for	Consists of
Objective	To verify the consistency of the design.
Schedule	In-process audits will be held at the discretion of the Software Quality Manager on an as-needed basis.
Procedures	In-process audits will be performed as informal meetings at the developer`s desk.
Participants	Software Quality Manager, other roles as needed.
Documentation required to be available for review	As a minimum, the following documents will be reviewed: • System Functional Description • Software Design Description • other documents as needed

Software for	Consists of
Procedure for recording deficiencies	The audit will only be documented if deficiencies have been found. The Software Quality Manager will enter potential deficiencies as change requests into the external Mantis.
Approval criteria	Approval will only be granted if deficiencies have been corrected.
Actions to occur upon approval	No special actions.

3.5 Interface Control

3.5.1 Dependencies

3.5.1.1 System Software

Dependencies of the ANNAX configuration items from other system software, hardware, and support software outside the scope of this plan are shown in the context diagram in Section 1.1.1 of the ANNAX Software Project Management Plan [ASPMP]. Interfaces are described in the Interface Bombardier IP Interface Control Document TCMS - PIS Controller [ICDIPPIS]. This document describes the nature of the Ethernet interface and the affected organization.

Changes to the project SCIs with changes to interfacing hardware, system software, and support Software outside the subcontractor's scope of work are coordinated by the ANNAX Software System Integrator on a peer-to-peer basis. It is the Software System Integrator’s responsibility to control interface code, documentation and data and to ensure that those are approved and released into the specified baselines.

3.5.1.2 Hardware

The ANNAX PIS has an Ethernet interface to Bombardier’s TCMS. This interface is governed by the Bombardier IP Interface Control Document TCMS - PIS Controller [ICDIPPIS]. This document is controlled like any other requirements document. Should changes to this document occur, a change request must be entered into Mantis, and the CCB will decide upon release of the ICD Support Software

3.5.1.3 Other

Not applicable.

3.5.2 Interface Configuration Control

The Software System Integrator will check CRs for possible effects on interfaces described in Interface Control Documents.

3.6 Subcontractor/Vendor Control

3.6.1 Third-party Software

Third-party software like for example software for the TCMS Simulator and related licences will be placed under configuration management in the same way as software developed by ANNAX. Changes to these software items will be handled by the ANNAX Software System Integrator, who will be notified by any project member of (potential) changes.

3.6.2 Subcontracted Software Design

Subcontracted software will be placed under configuration management in the same way as software developed by ANNAX. Suppliers will deliver source code, documentation and related information (version description, licences, description of development environment) for storage on ANNAX systems. Suppliers will also use Subversion.

Since stages and deliveries for subcontractors are almost equal to the internal development stages and deliveries (refer to [ASPMP], Section 7.7), subcontractors do not need a separate SCMP and will – even though working remotely most of the time – have access and use the ANNAX internal tools for test, release and Change Request Management (Mantis).

Because subcontractors are integrated into ANNAX' development process, no additional SCM controlling other than described in other sections of this document will be performed. For the same reason, no explicit separate acceptance of delivered by subcontractors will be performed.

As deemed appropriate, software integration workshops will be performed when major functionality of software provided by subcontractors has been implemented.

Documents provided by suppliers will be reviewed by ANNAX personnel according to the rules defined in [ASQAP].

3.6.3 Temporary Personnel

Not applicable.

3.7 Release Management and Delivery

Delivery of releases requires formal checks based on checklists. The rigor and depth of such checks depends on the release type: The later the release, the more rigorous and the deeper the check will be.

For all releases later than FAI, releases will be delivered together with a Software Delivery Note.

3.7.1 Releasing a Customer Application

Releasing a customer application consists of 4 steps:

1. Defining the ReleaseConfig.txt
2. Copying the root file system
3. Building the customer application
4. Making the flashcard

The process requires the installation and use of Subversion, colinux and php. All the software versions are managed with Subversion. The OFRIS applications are versioned according the stipulations in section 3.1.2. More details on using Subversion with Eclipse are given in [SUBCL]. The software is compiled on the Linux virtual machine, colinux, which is installed on the user's PC [COLNX].

A customer application is based on a number of software modules and libraries, assembled together. Which version of these software modules to take is defined in a configuration file named 'ReleaseConfig.txt'. There is one ReleaseConfig.txt file for each customer application release.

The ReleaseConfig.txt file is saved under the directory 'Build', on the customer's repository in Subversion. Each customer has its own repository on Subversion. The repository name must begin with the project number and may contain the project name, for example: 'SvnRepositories/411001xx.FzpfSbb-5.x. The customer's repository contains all customer specific files.

The operation of copying the root file system, building the customer application and making the flashcard are performed using php scripts. These php scripts are saved on the repository \\SvnRepositories\CommonTools\LinuxBuildScripts.

3.7.1.1 ReleaseConfig.txt

The ReleaseConfig.txt defines all the software modules versions and settings used to build the customer application.

All the versions specified in the ReleaseConfig.txt must have a corresponding tag, i.e. released version, on the relevant subversion repository.

3.7.1.2 Root File System

The root file system contains the basic file system modules needed to run a Linux application. The root file system is copied with the script **copy_rfs.php**.

A customer application has a dedicated root file system (RFS) on colinux, defined in the ReleaseConfig.txt with 'RFS_NAME'. When starting from scratch, the base RFS must be copied to this application RFS.

Which version of the base RFS to take is specified in the ReleaseConfig.txt file with 'RFS_BASE_VERSION'.

Attention must be paid that the script copy_rfs.php does not delete any directory or file from an existing RFS on colinux. If a brand new RFS is to be copied, local RFS files must be manually deleted before starting copy_rfs.php.

3.7.1.3 Customer Application

A customer application consists of binaries (compiled software modules, libraries, etc) and customer specific files (configurations, etc).

Binaries for the customer application are built and copied to the customer RFS using the script **build_appl.php**. The source files are exported from the SVN repositories, compiled on the local user machine (colinux) and installed on the application RFS.

The customer specific files are directly copied from the customer SVN repository using the script **copy_specific.php**.

Which software version to take for performing the operation is specified in the ReleaseConfig.txt. For example, 'OFRIS_VERSION' specifies the OFRIS library version to use.

3.7.1.4 Flashcard

The customer software application must finally be copied on a Flashcard, to be run on the target device.

The flashcard is divided in two partitions: the first partition is used to start the Linux kernel and the second partition is used for the customer RFS and application.

The script **make_flash.php** makes a flash card image on the user’s PC. This image can then be burned on the flash card using the tool ‘dd’ [DD].

Which version to take is specified in the ReleaseConfig.txt file.

4 SCM Schedule

4.1 SCM Sequence and Dependencies

SCM sequence and dependencies are incorporated/referenced in Section 5.3.1 of the ANNAX Software Project Management Plan [ASPMP].

4.2 SCM Milestones and Schedule

Refer to the ANNAX Software Project Management Plan [ASPMP].

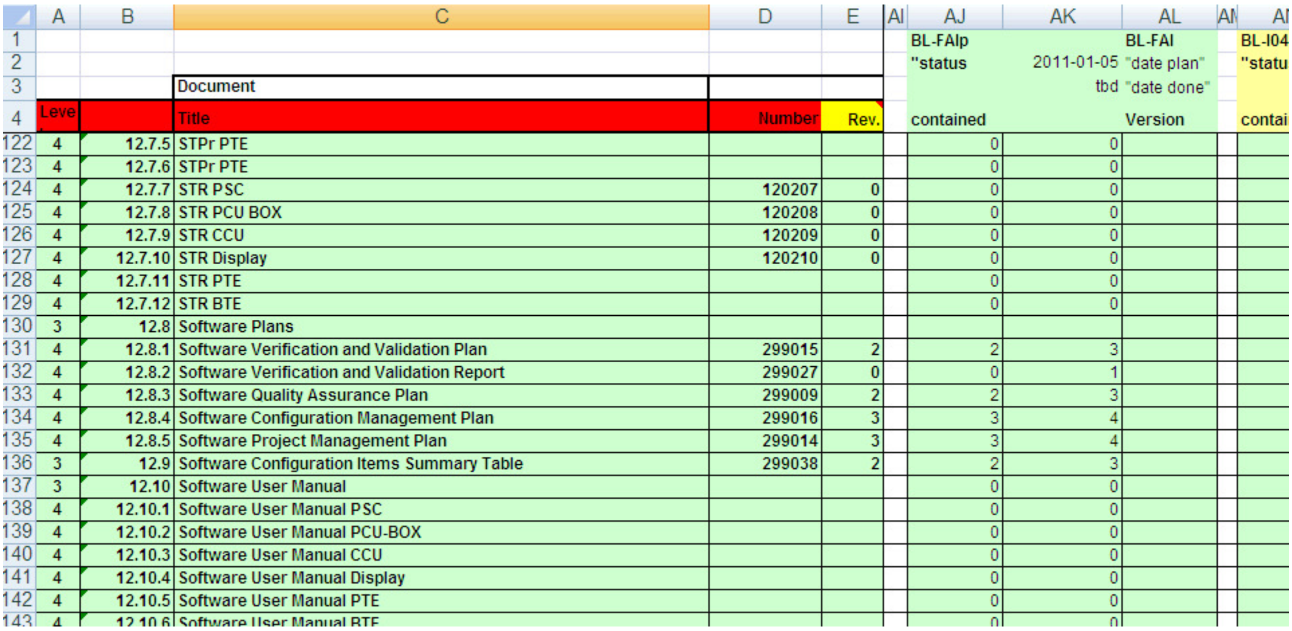
5 SCM Resources

5.1 Tools

The following tools will be used for SCM:

Table 14: SCM Tool

SCM Tool Name	Purpose
MS Excel	Worksheet for configuration status accounting and baseline accounting. File name: TTC_ConfigurationStatsuAccounting_<yyyy-mm-dd.xls>, see screenshot below.
Subversion	Version control
Mantis	Change request and bug tracking system
Dokumentenverwaltung	Electronic document management



	A	B	C	D	E	AI	AJ	AK	AL	AM	AN
1							BL-FAIp		BL-FAI		BL-I04
2							"status	2011-01-05	"date plan"		"statu
3			Document					tbd	"date done"		
4	Leve		Title	Number	Rev.		contained		Version		contai
122	4	12.7.5	STPr PTE				0	0			
123	4	12.7.6	STPr PTE				0	0			
124	4	12.7.7	STR PSC	120207	0		0	0			
125	4	12.7.8	STR PCU BOX	120208	0		0	0			
126	4	12.7.9	STR CCU	120209	0		0	0			
127	4	12.7.10	STR Display	120210	0		0	0			
128	4	12.7.11	STR PTE				0	0			
129	4	12.7.12	STR BTE				0	0			
130	3	12.8	Software Plans								
131	4	12.8.1	Software Verification and Validation Plan	299015	2		2	3			
132	4	12.8.2	Software Verification and Validation Report	299027	0		0	1			
133	4	12.8.3	Software Quality Assurance Plan	299009	2		2	3			
134	4	12.8.4	Software Configuration Management Plan	299016	3		3	4			
135	4	12.8.5	Software Project Management Plan	299014	3		3	4			
136	3	12.9	Software Configuration Items Summary Table	299038	2		2	3			
137	3	12.10	Software User Manual				0	0			
138	4	12.10.1	Software User Manual PSC				0	0			
139	4	12.10.2	Software User Manual PCU-BOX				0	0			
140	4	12.10.3	Software User Manual CCU				0	0			
141	4	12.10.4	Software User Manual Display				0	0			
142	4	12.10.5	Software User Manual PTE				0	0			
143	4	12.10.6	Software User Manual BTE				0	0			

Figure 1: Screenshot of the Configuration Status Accounting Worksheet

5.1.1 Configuration Management of Tools

Tools for software development will be listed in the Software Configuration Items Summary Table [ASCIST].

Versions of other SCM tools will be tracked in the MS Excel status accounting worksheet.

As soon as the development environment has been installed, normally no changes to tools used for the project will occur. Should versions of tools change, the change need to be requested and approved like a software change.

5.2 Techniques

No specific SCM techniques will be used.

5.3 Methodologies

No specific SCM methodologies will be used.

5.4 Personnel

SCM tasks will be performed by all project members. The Software System Integrator, Mr. Marco Binz, has the overall SCM responsibility. He acts as the Software Configuration Manager.

5.5 Training

The TTC Toronto project team members have received a training to carry out the SCM tasks, especially software status accounting, described in this SCMP. This training was held on June 23, 2010.

5.6 Environment and Infrastructure

Neither special environment nor infrastructure is required for SCM tasks for this project.

5.7 SCM Cost Management

SCM costs are managed as other project costs. Refer to Sections 5.2.4 and 5.3.3 of the ANNAX Software Project Management Plan [ASPMP]

6 SCMP Maintenance

6.1 SCMP Monitoring

It is the Software System Integrator’s responsibility to monitor the SCMP and update the document whenever necessary. The necessity of updating the plan shall be checked at least every 6 months.

6.2 SCMP Revision and Approval

The Software System Integrator is responsible of revising the SCMP. Internal approval and release of the SCMP shall follow the stipulations in the ANNAX Software Quality Assurance Plan [ASQAP], Section 4. Changes require formal approval by both Bombardier/TTC.

6.3 SCMP Change Distribution

After changes, the new SCMP revision will be distributed by the Project Manager via email to:

- Bombardier/TTC
- Subcontractors
- Project team members.

The distribution to Bombardier will be via official letter.