Table of contents

1.0 Objectives/Purpose 3

2.0 Scope 3

3.0 Roles, Responsibilities, Rules and Risks 3

3.1 Key Roles and Responsibilities 3

3.2 Assumptions and Constraints 5

4.0 Software Safety Management 5

4.1 Project Organization, Inter-Group Coordination, Stakeholders List 5

4.2 Project Escalation 7

4.3 Project Communication 7

4.4 Project Dependencies 8

4.5 Project Deliverables 9

4.6 Team Competencies and Training Plan 9

4.7 Tools and Infrastructure 9

5.0 Safety Lifecycle 9

5.1 ISO 26262 Tailoring 12

5.1.1 Development Interface Agreement 12

5.1.2 ISO26262 Impact Analysis 12

5.1.3 ISO 26262 Low Level Tailoring 12

5.2 Product Development – SW Level 24

5.2.1 Initiation of SW Product Development 24

5.2.2 Software Safety Concept 26

5.2.3 Specification and Management of Safety Requirements 27

5.2.4 Software Architecture Design, Detailed Design and Safety Analysis 27

5.2.5 Software Unit Implementation 29

5.2.6 Software Unit Testing and Integration Testing 31

5.2.7 Release Activities 33

5.2.8 User Documentation 34

5.3 Supporting Processes 35

5.3.1 Interfaces with Distributes Development 35

5.3.2 Configuration Management 35

5.3.3 Change Management 36

5.3.4 Verification 37

5.3.4.1 Peer reviews 37

5.3.4.2 Release Readiness Review 37

5.3.4.3 SW Testing 37

5.3.4.4 Traceability Matrix 37

5.3.4.5 Static Code 37

5.3.4.6 Code Coverage 38

5.3.5 Confidence in Software Tools 38

5.3.6 Quality Assurance 39

5.3.6.1 Safety Case 39

5.3.6.2 Safety Confirmation Measures 40

5.3.7 Qualification of Software Components 41

5.4 RFE SW Project Planning 41

5.4.1 Project Planning and Tracking 41

6.0 Project Tailoring 42

7.0 Safety Plan Changes 44

8.0 Revision History 50

# Objectives/Purpose

This document aims to provide adequate visibility into the project activities performed during the software development of the RFE SW (safety-related product).

The document specifies the project management activities and deliverables, project organization, stakeholders, project communication, competence management, safety lifecycle covering product development sub-phases and support processes, ISO 26262 and AP SW process tailoring.

Note: From ISO 26262 standpoint, this document covers both Safety Plan and Project Plan (Refer ISO 26262 Part 2-6.4.6.4)

# Scope

The scope of this Project Management Plan (Safety Plan) consists of RFE SW components developed for all planned releases in roadmap. Refer RFE SW Architecture Specification [20] for RFE SW Components.

RFE SW project objectives are on-time delivery of the official release with the achievement of the required content, safety goals quality goals and project execution within the estimated costs. The Software Quality Assurance Plan contains the project quality goals and their targets.

# Roles, Responsibilities, Rules and Risks

## Key Roles and Responsibilities

Roles and their responsibilities, skills and authorities required to deploy this process, are documented in BL AP SW Roles Description document [6]. Specific responsibility and deliverable, when differ from the generic ones, are documented in this document, in this section, in the tables below.

Table 1 describes the roles assignment and contact information of the RFE SW internal stakeholders.

**Note**: “Integration Engineer” from AP SW process is named “Release Engineer” in this document and all project documents. This role is assigned to various members of the team for each release, assignment is documented in Gantt.

| **Name** | **Role** | **Responsibility/Output** |
| --- | --- | --- |
| Dalton Correya (BLR) | SW Team Manager | per BU Auto Roles Description Procedure [6] |
| Kishan Kumar (EHV) | SW Team Manager | per BU Auto Roles Description Procedure [6] |
| Samrat Guha Niyogi | SW Program Manager | per BU Auto Roles Description Procedure [6] |
| Vinoth Ranee Kumar | SW Configuration Manager | per BU Auto Roles Description Procedure [6] |
| Siddareddy Sadashivappa | Escalation level 0 for Quality issues  SW Quality Assurance Engineer | per BU Auto Roles Description Procedure [6] |
| Maulik Prabhudesai | SW Project Manager | per BU Auto Roles Description Procedure [6]  Plans and monitors also safety related activities, as part of the project plans. |
| Sandeep B B  Rashmi K C  Leena Mukherjee | SW Safety Architect  SW Project Safety Manager  SW Project Safety Manager(Shark Project) | per BU Auto Roles Description Procedure [6]  For the Safety Manager role: planning and monitoring of the safety related activities are delegated to Project Manager |
| Lars van Meurs  Alex K P  Anil Uke  Dilip Mannil  Hong Li  Hrudaynath Yelgudkar  Jagadesh S X  Maulik Prabhudesai  Mohammed Nabil Kunda  Pedro Ferrão  Rajiv Chithambaran  Ravikant Tiwari  Sandeep N V  Shrey Ravindra  Srinivasa Rao Hanumanthu  Srinivasa Rao Pentakota  Subodh Shukla  Sumit Jaiswal  Suramya Gupta  Yongyi Wei | SW Engineer | per BU Auto Roles Description Procedure [6] |
| Artur Burchard / Sri Krishnan M R | SW Architect  SW Requirements Manager | per BU Auto Roles Description Procedure [6] |
| Biju Ravindran | SW Test Architect | per BU Auto Roles Description Procedure [6] |
| Dongyu Gao | SW Test Engineer | per BU Auto Roles Description Procedure [6] |

Table 1 RFE SW Project Internal Stakeholders List

| **Name** | **Role/Team** | **Responsibility/Input for RFE SW** |
| --- | --- | --- |
| Tziki Oz-Sinay | PMO Manager | per BU Auto Roles Description Procedure  Escalation level 2 for PM issues |
| Sushil Kumar Gupta /  Vinod Pandey | SW Program Manager | Escalation level 1 for PM issues  O: Program schedule, meeting minutes/action items, status report, program risk management plan  I: silicon, reference manuals, erratas, evaluation boards |
| Melanie Schillinsky /  Suresh Adav | STRX Program Manager | Escalation level 1 for PM issues  O: Program schedule, meeting minutes/action items, status report, program risk management plan  I: silicon, reference manuals, erratas, evaluation boards |
| Mark Steigemann (HBG)  Henning Moeller (DRSD)  Joerg Fischer (DRSD)  Sridhar Ramaswami (BLR)  Raghavendra Chidri (BLR) | Development Manager /  Resource Manager | Points of contact for resourcing discussions |
| Cengiz Oezcan | Product Quality | per BU Auto Roles Description Procedure [6] |
| Michael Knebelkamp | Product marketing Manager | per BU Auto Roles Description Procedure [6] |
| Mark Steigemann | BL RFP Architect | per BU Auto Roles Description Procedure [6] |
| Marthijn de Man | STRX SW Architect | per BU Auto Roles Description Procedure [6] |
| Ronald van der Kraan | BL RFP Operation Manager | per BU Auto Roles Description Procedure [6] |
| Andreas Soltau | BL RFP R&D Manager | per BU Auto Roles Description Procedure [6] |
| Eleonora Gianfermi | SW Quality Assurance Officer | per BU Auto Roles Description Procedure [6] |
| Rolf Schlagenhaft | BL RFP Functional Safety Manager | per BU Auto Roles Description Procedure [6] |

Table 2 RFE SW External Stakeholders List

## Assumptions and Constraints

RFE SW relate to availability of results of other projects and to the quality of input documents for the projects: RSDK, FPGA (RFE digital), Simulation (AMS), Command dispatcher (part of IPCF) – core 2 core communication based on IPCF framework

# Software Safety Management

## Project Organization, Inter-Group Coordination, Stakeholders List

This section describes the following:

* The way RFE SW project development is being organized
* The list of internal stakeholders and the output deliverables/activities expected from them
* The list of external stakeholders and the input/output deliverables the RFE SW benefits from/delivers to
* The groups interactions during the project lifecycle

The project has a multi-site organization, as follows:

* RFE SW Development Team (EHV, DRS, HBG)
* RFE SW Independent Testing Team (BLR)
* Program Management (EHV, BLR, HBG)
* Apps & System team
* AP Tools team
* Product Engineering team
* Marketing – (HBG)
* ISO 26262 Functional Safety (BLR)
* Software Quality – (BLR)

The project organization and interactions with the groups/teams is described in [TeamForge : Document Details (nxp.com)](https://www.collabnet.nxp.com/sf/docman/do/viewDocument/projects.smarttrx/docman.root.es0.040_project_management.non_fdo_document/doc430793?nav=1)

Figure 1: RFE SW Project in context of Smart TRX Setup

**Sushil Kumar Gupta**

**Vinod Pandey**

STRX SW Program Manager

**RFE SW Project Setup**

**Dalton Correya**

**(BLR)**

Development Manager

**Rashmi KC**

**Leena M (Shark)**

Functional Safety

Manager

**Tejaswini X**

S&I Project Lead

**Samrat Guha Niyogi**

RFE SW Project Lead

**Harpreet Bhullar**

RFE HW Project

Manager

**Michalis Roppas**

System Validation

**Marthijn de Man**

STRX Software Architect

**Sandeep BB**

**FSA**

Functional Safety Architect/Manager

**Archs/POs**

**Artur Burchard**

**Lars van Meurs**

**Sri Krishnan M R**

**Sumit Jaiswal**

**Shrey Ravindra**

**Rajiv Chinthambaran**

**[FTEs](#_Key_Roles_and)**

**Siddareddy Sadashivappa**

Project QAE

**Yuanhao Bi**

System Integration

**Kishan Kumar**

**(EHV)**

Development Manager

**Vinoth Ranee Kumar**

SW Build CM

**Burkhard Bräuer** SW Doc CM

OSS Trusted Advisor

Naveen Phatak

CTO Functional Safety

Functional Safety Assessor

**RFE FW Software Team**

Figure 2: RFE SW Project Setup

## Project Escalation

* **Quality Issues Escalation Path**

RFE SW quality related issues shall first be addressed to project QAE (for the issues reported by the team) or to RFE SW Project Manager and Technical Lead (for the issues reported by the project QAE). The Safety Manager shall also be informed.

If the team in the context of the actual planning cannot deploy a suitable solution, or if the solution found impacts on the external stakeholders’ outcomes, the QAE shall escalate the issue to the Quality Manager.

* **Project Management Issues Escalation Path**

RFE SW project management related issues shall first be addressed to RFE SW Project Manager; the Safety Manager shall be informed.

If the team in the context of the actual planning cannot deploy a suitable solution, or if the solution found impacts on the external stakeholders’ outcomes, the Project Manager shall escalate the issue up to Software Program Manager and then to head of PMO.

* **Safety Issues Escalation Path**

RFE SW project issues related to ISO 26262 standard deployment or to achievement of safety goals shall first be addressed to Functional Safety Manager; the Project Manager shall be informed.

If the team in the context of the actual planning cannot deploy a suitable solution, or if the solution found impacts on the external stakeholders’ outcomes, the Functional Safety Manager shall escalate up to BL SW Safety Manager, and then to CTO Functional Safety. Safety issues and status at BL level are reported monthly to BL RFP Safety Manager.

The Project Manager notifies the Safety Manager in case a change request is impacting a safety feature.

## Project Communication

Project communications include, but are not limited to, face-to-face discussions and meetings (where applicable); Skype/phone conference calls; emails.

| Meeting Participants/Minutes Repository | Project Technical Status Meeting; dev team | Project Technical Status Meeting; dev – test team sync | Project Status Meeting – mgmt. sync |
| --- | --- | --- | --- |
| Scope | Discuss project technical topics and, when necessary, safety topics and quality topics within dev team.  Risks & issues, tickets. | Discuss project technical topics, and, when necessary, safety topics and quality topics. | 1. Management meeting  2. Roadmap meeting: project status, risks and issues, roadmap changes |
| Report /Minutes Repository | OneNote link:  [STRX\_RFE\_SW\_OneNote](onenote:https://nxp1.sharepoint.com/teams/206_21/Shared%20Documents/OneNote/STRX_RFE_SW_OneNote/) | OneNote link:  [STRX\_RFE\_SW\_OneNote](onenote:https://nxp1.sharepoint.com/teams/206_21/Shared%20Documents/OneNote/STRX_RFE_SW_OneNote/) | OneNote link:  [SmTRx](onenote:https://nxp1-my.sharepoint.com/personal/sushil_kumar_gupta_nxp_com/Documents/Notebooks/SmTRx/) |
| **Participants** |  |  |  |
| SW Safety Architect | As needed | As needed | Optional |
| RFE SW Architect | Always | Optional | Always |
| RFE SW PL | Always – chair, record | Optional | Always |
| Team Manager | Optional | Optional | Always |
| Project SQE | As needed | As needed | Never |
| Development Team | Always | Optional | Never |
| Test Manager | Never | Always | Never |
| Test Team | Never | Optional | Never |

|  | Safety Management Status Meeting | Quality Meeting |
| --- | --- | --- |
| Scope | Discuss safety related topics, issues, status | Discuss BCaM7 gates and project quality status and issues |
| Report /Minutes Repository | OneNote/ SharePoint | [BCaM7 statu](https://nxp1.sharepoint.com/sites/amp-rsdk/_layouts/15/guestaccess.aspx?guestaccesstoken=264B%2B1IHZTdDyfUJyGRhI0XItyhcg2Hmj5caQmdqxrY%3D&docid=2_09a15282277a34fc6b926549cfd94050e&rev=1&e=7a5b6e0305aa4aea96100cc530cbbd7d)s tracking file  OneNote/ SharePoint |
| **Participants** |  |  |
| Project Functional Safety Manager | Always – Chairs, Record | Always |
| SW Project Manager | Always | Always |
| SW Project QAE | Always - Record | Always - Record |
| SW Quality Assurance Officer | As needed | As needed |
| SW Safety Manager | Always | As needed |
| SW Team Manager | As needed | As needed |
| SW RFE SW Architect | As needed | As needed |

Table 3 Communication Matrix

RFE SW Project action items, owners and status are stored on link: [STRX\_RFE\_SW\_OneNote](onenote:https://nxp1.sharepoint.com/teams/206_21/Shared%20Documents/OneNote/STRX_RFE_SW_OneNote/).

Any other project stakeholder that is not mentioned in the tables above may participate to the meetings, as needed. Regarding the Change Control Board (CCB), refer to RFE SW Software Configuration Management Plan [5].

## Project Dependencies

The RFE SW Project Dependencies are documented in the RFE SW Dependencies list [14].

## Project Deliverables

Supported platforms: STRX, Onechip, Shark, Remote. This document primarily scopes in the STRX RFE SW development for the STRX OneChip which is a complete reuse for Remote and Shark (additionally applicable change requests and bring up activities would be planned)

Export Control classification for the Software shared with customers is available at: <https://www.collabnet.nxp.com/sf/go/doc364183>

Deliverables are defined within table below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deliverable** | **Description** | **Integrity Level** | **Owner** | **Code** |
| RFE software | Calibration,  Control,  mmWave HAL,  RFE driver | **ASIL B**  **(process)** | BL-RFP | production |

RFE SWpackage; Quality package (for PRC and RFP); Safety package (for PRC and RFP); documentation; all uploaded in Flexera*.* For details please see [RFE SW SCMP](https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Change_and_Configuration_Management/RFE_Configuration_Management_Plan.docx) document.

Licenses in use per SW component: Basic Proprietary Commercial License

## Team Competencies and Training Plan

The competencies/skills matrix (necessary vs. available) and the training plan for the engineers involved in RFE SW project are created and maintained by each Team Manager (see [18]).

Training plan for STRX SW team members (including RFE SW team) is maintained at:

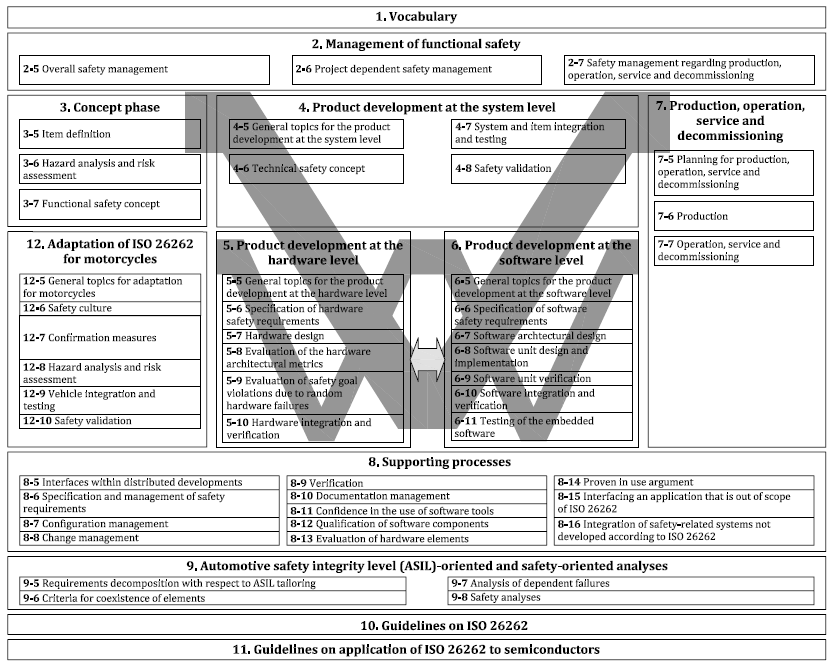
<https://www.collabnet.nxp.com/sf/go/doc406698>

## Tools and Infrastructure

The list of the software tools used during RFE SW project lifecycle and their versions are listed within the link: [RFE\_SW\_Tool\_Eval\_Qualificaion.xlsx](https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Functional_Safety/Tool_Evaluation_Qualification/RFE_SW_Tool_Eval_Qualificaion.xlsx)

# Safety Lifecycle

ISO 26262 standard addresses the development of Electrical and/or Electronic (E/E) systems within road vehicles. The RFE SW software contributes to the development of such an E/E system, and Figure 3 depicts which parts and clauses of ISO 26262 are applicable – yellow highlights.



***SW*** *Component Developed as* ***SEooC***

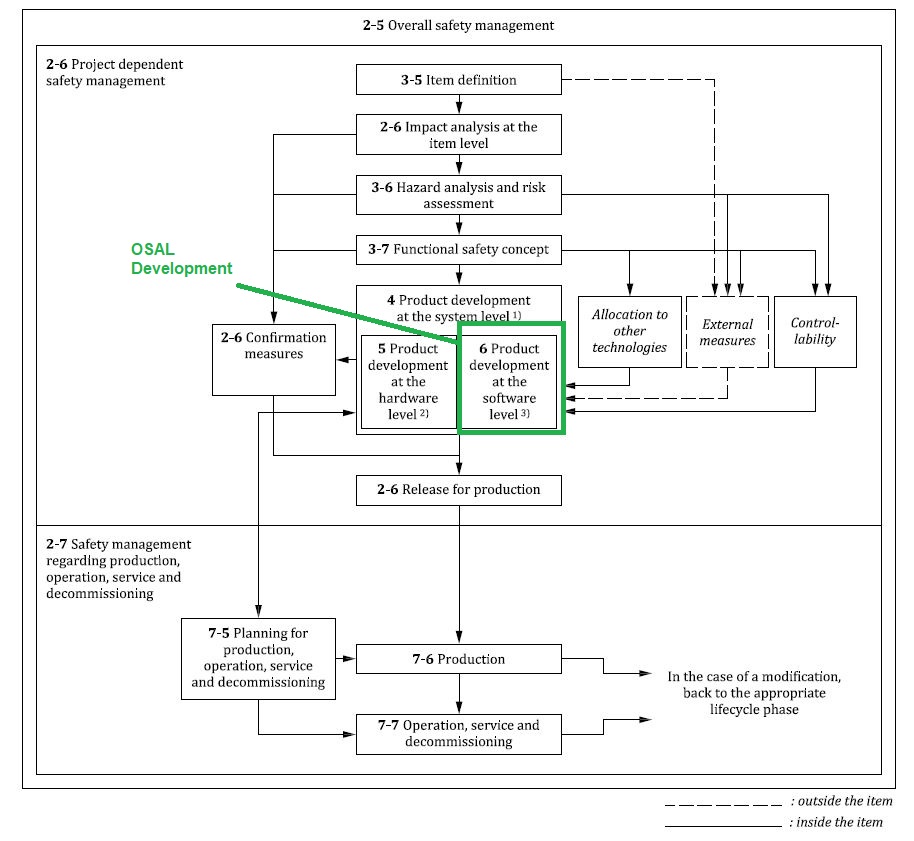
*Reference ISO 26262-10:2018*

*Applicable to Component developed as SEooC*

Figure 3: ISO 26262 Applicability for RFE SW Software Development

Project Safety case covers detailed information regarding ISO 26262 requirements applicability for RFE SW Software.

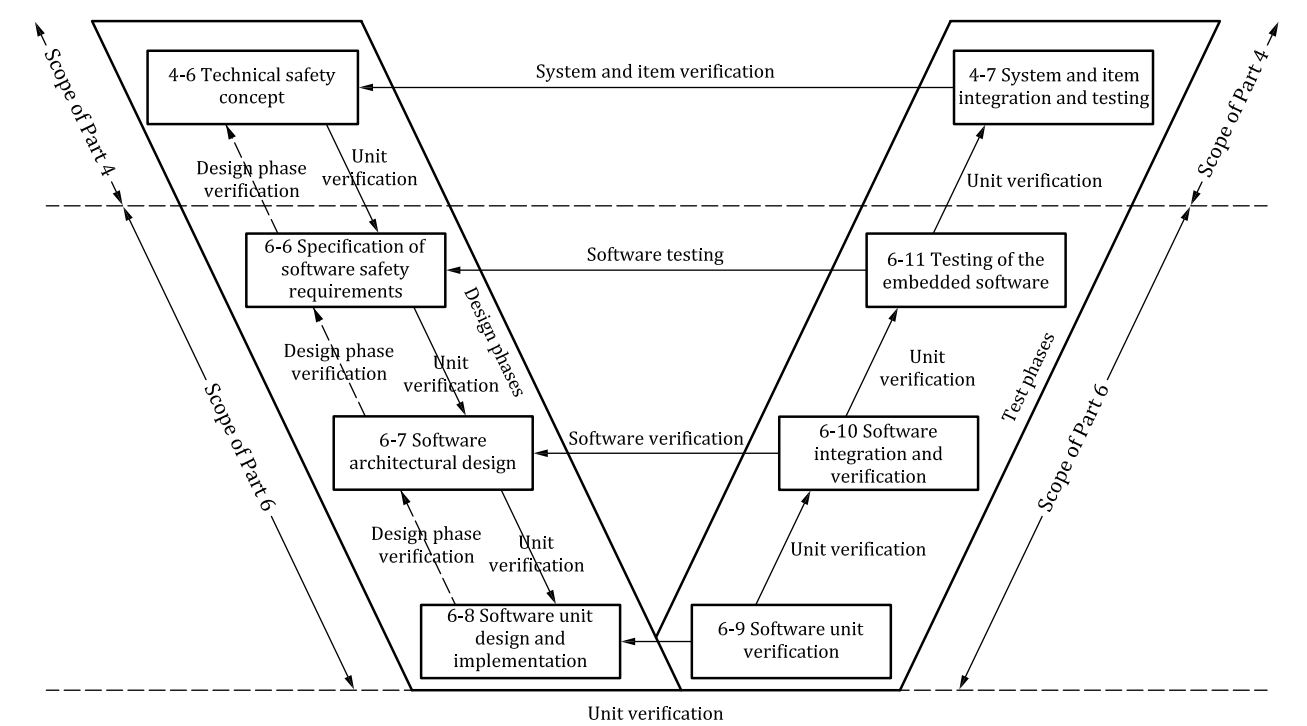
Figure 4 depicts the part of the Item Safety Lifecycle, which is applicable for RFE SW development, highlighted in the green box. The reason for tailoring the item safety lifecycles is the scope of the project. Thus, the part of the Item Safety Lifecycle applicable for RFE SW development is ISO 26262 – Part 6.



**RFE SW**

**Development**

Figure 4: Management activities in relation to the safety lifecycle



**RFE SW**

Figure 5: SW Safety Lifecycle (ISO 26262 - Part 6) for RFE SW

## ISO 26262 Tailoring

### Development Interface Agreement

There is no customer level Development Interface Agreement (DIA) for RFE SW project level.

Development Interface Report is defined with link: <https://www.collabnet.nxp.com/sf/go/doc364314>

### ISO26262 Impact Analysis

RFE SW is developed according to ASIL B ISO 26262:2018 version requirements as Safety Element out of Context and is an Explicit Safety project.

### ISO 26262 Low Level Tailoring

**ISO26262 Part 2:**

The only tailoring performed per ASIL is for Confirmation Measures. These are tailored depending on ASIL target. NXP performs ISO 26262 Confirmation Reviews (CR), Audit and Assessment as required by ISO 26262 for SeooC development. All checks are executed with required level of independence.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Confirmation Measures** | **QM** | **ASIL A** | **ASIL B** | **ASIL C** | **ASIL D** |
| CR Impact Analysis | Yes | Yes | Yes | Yes | Yes |
| CR Safety Plan | - | Yes | Yes | Yes | Yes |
| CR Technical Safety Concept | - | Yes | Yes | Yes | Yes |
| CR Safety Analysis | - | Yes | Yes | Yes | Yes |
| CR Safety Case | - | Yes | Yes | Yes | Yes |
| Audit | - | - | - | Yes | Yes |
| Assessment | - | - | - | Yes | Yes |

*Note: The following confirmation reviews are not applicable: hazard analysis and risk assessment, functional safety concept, item integration and testing, validation plan.*

**ISO26262 Part 3:**

* Not applicable.

**ISO26262 Part 4:**

* Not applicable in general due to SEooC development.
* NXP contributes to the item HW-SW Interface with HW Reference Manual.

**ISO26262 Part 5:**

* Not applicable.

**ISO26262 Part 6:** Applicable with the following tailoring:

* ISO 26262 Part 6 Section 5: General topics for product development at the software level
* ISO 26262 Part 6 Section 5.4.3 Table 1: Topics to be covered by modelling and coding guidelines

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | | Comments and Tailoring |
| **A** | B | **C** | **D** |
| 1a | Enforcement of low complexity | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered : Done by measuring cyclomatic complexity |
| 1b | Use of language subsets | ++ | ++ | + | + | Applicable without tailoring  Covered : By enforcing compliance to MISRA rules. |
| 1c | Enforcement of strong typing | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered : By compile time checks, avoiding void pointers, following coding guideline. |
| 1d | Use of defensive implementation techniques | + | + | ++ | ++ | Applicable without tailoring.  Covered:  1. failure to check return value of any function  2. failure to check for null in case of pointer  3. Avoid Usage of return value in a function without explicitly checking  Remarks: Range check is done at the entry point(RFE API ensures that it passes a valid parameter) |
| 1e | Use of well-trusted design principles | + | + | + | + | Applicable without tailoring.  Covered: Review of Requirement Specification + linking of requirements |
| 1f | Use of unambiguous graphical representation | + | ++ | ++ | ++ | Applicable without tailoring.  Covered: By using Behavioral & Structural UML diagrams in Architecture Specification |
| 1g | Use of style guides | + | ++ | ++ | ++ | Applicable without tailoring.  Covered: By following RFE coding guideline. [10] |
| 1h | Use of naming conventions | ++ | ++ | ++ | ++ | Applicable without tailoring.  Covered: By following Section 5.3 RFE coding guideline. [10] |
| 1i | Concurrency aspects | + | + | + | + | Not used.  RFE SW is single threaded except for HW error handling via ISR. ISR is not nested. |

* ISO 26262 Part 6 Section 6: Specification of software safety requirements
* There are safety requirements received from upper level and SW–FMEA.
* Tailoring of the inputs: System design is not visible to NXP internal development; HW-SW Interface is avalaible in DOORSng.
* ISO 26262 Part 6 Section 7: Software architectural design
* Tailoring of the inputs: System design is not visible to NXP internal development.
* ISO 262626 Part 6 Section 7.4.1 Table 2 Notations for software architectural design.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | | Comments and Tailoring |
| **A** | **B** | **C** | **D** |
| 1a | Natural language | ++ | ++ | ++ | ++ | Applicable without tailoring.  Covered: Natural language is used in RS and AS. In case where additional information is needed in UML diagrams, they are placed as notes inside diagrams. |
| 1b | Informal notations | ++ | ++ | + | + | Applicable without tailoring  Covered: Software architectural design is described using a combination of informal notation (free-style diagrams) and semi-formal notation (standard UML diagrams – activity, sequence etc.) . Details in RFE SW Design Guideline [7]. |
| 1c | Semi-formal notations | + | + | ++ | ++ |
| 1d | Formal notations | + | + | + | + | Not used  Mathematic notations are not used to describe architecture.  RFE SW is of low complexity(no concurrency, limited set of simple state machines), the costs of going for formal notations far outweighs the benefits. |

* ISO 26262 Part 6 Section 7.4.3 Table 3 - Principles for software architectural design

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Methods*** | | ***ASIL*** | | | | ***Comments and Tailoring*** |
| ***A*** | ***B*** | ***C*** | ***D*** |
| 1a | Appropriate hierarchical structure of software components | ++ | ++ | ++ | ++ | Applicable without tailoring.  Covered: BL-RFP Architecture Template enforcing this method is used In RFE SW AS. |
| 1b | Restricted size and complexity of software components | ++ | ++ | ++ | ++ | Applicable without tailoring.  Covered: In RFE SW AS document |
| 1c | Restricted size of interfaces | + | + | + | ++ | Applicable without tailoring.  Covered: In RFE SW AS document, |
| 1d | Strong cohesion within each software component | + | ++ | ++ | ++ | Applicable without tailoring.  Covered: By “separation of concerns” in |
| 1e | Loose coupling between software components | + | ++ | ++ | ++ | Applicable without tailoring.  Covered: By having data structures per component in RFE SW, restricting the use of global data. Covered in RFE SW unit design document |
| 1f | Appropriate scheduling properties | ++ | ++ | ++ | ++ | Not used.  No scheduler is present. Execution is sequential |
| 1g | Restricted use of interrupts | + | + | + | ++ | Applicable without tailoring.  Covered: Minimum subset of interrupts is identified and interrupt priority is set. Details in RFE SW AS. |
| 1h | Appropriate spatial isolation of the software components | + | + | + | ++ | Applicable without tailoring.  Covered: Spatial isolation is provided by defining different memory regions and enforcing access via MPU as described in RFE SW AS. |
| 1i | Appropriate management of shared resources | ++ | ++ | ++ | ++ | Not used.   RFE SW is single threaded, hence concurrent access of resources does not arise. System level shared resources used by RFE SW (sRAM by IPCF) are described in overall AS. |

* ISO 26262 Part 6 Section 7.4.14 Table 4 – Methods for the verification of the software architectural design.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Methods*** | | ***ASIL*** | | | | ***Comments and Tailoring*** |
| ***A*** | ***B*** | ***C*** | ***D*** |
| 1a | Walk-through of the design | ++ | + | o | o | Applicable without tailoring.  Covered: Review process defined AP SW Review Guideline [14] |
| 1b | Inspection of the design | + | ++ | ++ | ++ | Applicable without tailoring.  Covered: Review process defined AP SW Review Guideline [14] |
| 1c | Simulation of dynamic parts of the design | + | + | + | ++ | Applicable without tailoring.  Covered: By simulation/emulation models used to test RFE SW.  Note: Not using Model Based Design. |
| 1d | Prototype generation | o | o | + | ++ | Applicable without tailoring.  Covered : Prototype for check some functionality (ex dynamic chirping) will be used. |
| 1e | Formal verification | o | o | + | + | Not used.   Requirement based Test Generation is considered enough. |
| 1f | Control flow analysis | + | + | ++ | ++ | Not used.  It is expected to be done at application level |
| 1g | Data flow analysis | + | + | ++ | ++ | Applicable without tailoring.  Covered: In Component integration testing |
| 1h | Scheduling analysis | + | + | ++ | ++ | Not used.   RFE SW is singlethreaded hence there is no scheduler. |

* ISO 26262 Part 6 Section 8: Software unit design and implementation
* ISO 26262 Part 6 Section 8.4.3 Table 5 – Notations for software unit design.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | | Comments and Tailoring |
| **A** | **B** | **C** | **D** |
| 1a | Natural language | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered**:** Natural language is used to compliment UML diagrams. They are added as notes. Documented in RFE SW Unit Detailed Design |
| 1b | Informal notations | ++ | ++ | + | + | Applicable without tailoring  Covered: Detailed design is documented using a combination of semi-formal notations (UML), and informal notations (free-style diagrams). Documented in RFE SW Unit Detailed Design |
| 1c | Semi-formal notations | + | + | ++ | ++ |
| 1d | Formal notations | + | + | + | + | Not used. |

* ISO 26262 Part 6 Section 8.4.5 Table 6 – Design principles for software unit design and implementation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | | Comments and Tailoring |
| **A** | **B** | **C** | **D** |
| 1a | One entry and one exit point in subprograms and functions | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance to MISRA-C 2012 checks |
| 1b | No dynamic objects or variables, or else online test during their creation | + | ++ | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance to MISRA-C 2012 checks |
| 1c | Initialization of variables | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered : by enforcing compliance to MISRA-C 2012 & CWE checks |
| 1d | No multiple use of variable names | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance to MISRA-C 2012 checks |
| 1e | Avoid global variables or else justify their usage | + | + | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance to MISRA-C 2012 checks |
| 1f | Restricted use of pointers | + | ++ | ++ | ++ | Applicable without tailoring  Covered:The abuse in using pointers is identified during code review and anomalies are detected during MISRA rules verification. |
| 1g | No implicit type conversions | + | ++ | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance to MISRA-C 2012 & CWE checks |
| 1h | No hidden data flow or control flow | + | ++ | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance to MISRA-C 2012 Section 8.15 rules which identifies control flow discrepancies |
| 1i | No unconditional jumps | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: Usage of *break, continue, goto, return* is prohibited or controlled when required. In case they need to be used, it will be explicity documented in the code. |
| 1j | No recursions | + | + | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance to MISRA-C 2012 checks |

* ISO 26262 Part 6 Section 9: Software unit verification
* ISO 26262 Part 6 Section 9.4.2Table 7 - Methods for software unit verification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Methods** | | **ASIL** | | | | **Comments and Tailoring** |
| **A** | **B** | **C** | **D** |
| 1a | Walk-through | ++ | + | o | o | Applicable without tailoring  Covered: Review process defined AP Software Review Guideline [14] |
| 1b | Pair-programming | + | + | + | + | Not used |
| 1b | Inspection | + | ++ | ++ | ++ | Applicable without tailoring  Covered: Review process defined AP Software Review Guideline [14] |
| 1c | Semi-formal verification | + | + | ++ | ++ | Not used.  Requirement based verification is used.  Semi-formal Verification = Simulation of Requirements |
| 1d | Formal verification | o | o | + | + | Not used  Requirement based verification is used.  Semi formal and Formal verification are applicable for Model based development. |
| 1e | Control flow analysis | + | + | ++ | ++ | Not used.  It is expected to be done at application level |
| 1f | Data flow analysis | + | + | ++ | ++ | Not used.  It is expected to be done at application level |
| 1g | Static code analysis | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: by enforcing compliance MISRA & CWE checks. |
| 1h | Static analysis based on abstract interpretation | + | + | + | + | Not used  There is no plan to extend the compiler parse tree. |
| 1j | Requirements-based test | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: Requirement linking to Unit Test |
| 1k | Interface test | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: Requirement linking to Unit Test |
| 1l | Fault injection test | + | + | + | ++ | Not used  In the SmartTRX software test plan, FIT is done only at Component Integration Test level |
| 1m | Resource usage test | + | + | + | ++ | Applicable without tailoring  Covered: SmartTRX software test plan at Component Unit Test Spec level |
| 1n | Back-to-back comparison test between model and code, if applicable | + | + | ++ | ++ | Not used  Model based development is not used. |

* ISO 26262 Part 2 Section 9.4.3Table 8 - Methods for deriving test cases for software unit testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Methods** | | **ASIL** | | | | **Comments and Tailoring** |
| **A** | **B** | **C** | **D** |  |
| 1a | Analysis of requirements | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: In SmartTRX software test plan [19] |
| 1b | Generation and analysis of equivalence classes | + | ++ | ++ | ++ | Applicable without tailoring  Covered: In SmartTRX software test plan [19] |
| 1c | Analysis of boundary values | + | ++ | ++ | ++ | Applicable without tailoring  Covered: In SmartTRX software test plan [19] |
| 1d | Error guessing | + | + | + | + | Applicable without tailoring  Covered: In SmartTRX software test plan [19] |

* ISO 26262 Part 6 Section 9.4.4 Table 9 - Structural coverage metrics at the software unit level

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | **ASIL** | | | | **Comments and Tailoring** |
| **A** | **B** | **C** | **D** |
| 1a | Statement coverage | ++ | ++ | + | + | Applicable without tailoring |
| 1b | Branch coverage | + | ++ | ++ | ++ | Applicable without tailoring  Covered: In SmartTRX software test plan [19].Will be reported in Code coverage report generated by LTB tool |
| 1c | MC/DC (Modified Condition/Decision Coverage) | + | + | + | ++ | Applicable without tailoring  Covered: In Structural tests detailed in SmartTRX software test plan [19]. Will be covered LTB tool. |

* ISO 26262 Part 2 Section 10: Software integration and verification
* ISO 26262 Part 6 Section 10.4.2Table 10 - Methods for verification of software integration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Methods** | | **ASIL** | | | | **Comments and Tailoring** |
| **A** | **B** | **C** | **D** |
| 1a | Requirements-based test | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: In Component Integration testing detailed in detailed in SmartTRX software test plan [19]. |
| 1b | Interface test | ++ | ++ | ++ | ++ |
| 1c | Fault injection test | + | + | ++ | ++ | Applicable without tailoring  Covered: In Component Integration testing detailed in detailed in SmartTRX software test plan [19] |
| 1d | Resource usage evaluation | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: by bandwidth, stack, code and data Memory usage detailed in STRX SW Overall test plan |
| 1e | Back-to-back comparison test between model and code, if applicable | + | + | ++ | ++ | Not used  Model Based development is not used. |
| 1f | Verification of the control flow and data flow | + | + | ++ | ++ | Not used.  Strategies for control flow monitoring are implemented at application level. |
| 1g | Static code analysis | ++ | ++ | ++ | ++ | Applicable without Tailoring  Static code analysis is done on combination of units. Covered already at unit level. |
| 1h | Static analysis based on abstract interpretation | + | + | + | + | Not used  There is no plan to extend the compiler parse tree. |

* ISO 26262 Part 6 Section 10.4.3Table 11 - Methods for deriving test cases for software integration testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | | Comments and Tailoring |
| **A** | **B** | **C** | **D** |  |
| 1a | Analysis of requirements | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: In Component Integration testing detailed in SmartTRX software test plan [19]. |
| 1b | Generation and analysis of equivalence classes | + | ++ | ++ | ++ | Applicable without tailoring  Covered: In Component Integration testing detailed in SmartTRX software test plan [19]. |
| 1c | Analysis of boundary values | + | ++ | ++ | ++ | Applicable without tailoring  Covered: In Component Integration testing detailed in SmartTRX software test plan [19]. |
| 1d | Error guessing based on knowledge or experience | + | + | + | + | Applicable without tailoring  Covered: In Component Integration testing detailed in SmartTRX software test plan [19]. |

* ISO 26262 Part 6 Section 10.4.5Table 12 - Structural coverage metrics at the software architectural level - no tailoring

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | **ASIL** | | | | **Comments and Tailoring** |
| **A** | **B** | **C** | **D** |
| 1a | Function coverage | + | + | ++ | ++ | Applicable without tailoring  Covered: In SmartTRX software test plan [19] |
| 1b | Call coverage | + | + | ++ | ++ | Applicable without tailoring  Covered: In SmartTRX software test plan [19] |

* ISO 26262 Part 6 Section 11: Testing of the embedded software
* ISO 26262 Part 6 Section 11.4.1 Table 13 -Test environments for conducting the software safety requirements verification

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | | Comments and Tailoring |
| **A** | **B** | **C** | **D** |
| 1a | Hardware-in-the-loop | ++ | ++ | ++ | ++ | Applicable w/ithout tailoring  Covered**:** in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |
| 1b | Electronic control unit network environments | ++ | ++ | ++ | ++ | Not used.  NXP does not have access to final system design |
| 1c | Vehicles | + | + | ++ | ++ | Not used.  NXP does not have access to vehicle |

* ISO 26262 Part 6 Section 11.4.2 Table 14 – Methods for tests of the embedded software

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | **ASIL** | | | | **Comments and Tailoring** |
| **A** | **B** | **C** | **D** |
| 1a | Requirements-based test | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |
| 1b | Fault injection test | + | + | + | ++ | Applicable without tailoring  Covered: in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |

* ISO 26262 Part 6 Section 11.4.3 – Table 15 – Methods for deriving test cases for the test of the embedded software

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | | Comments and Tailoring |
| **A** | **B** | **C** | **D** |
| 1a | Analysis of requirements | ++ | ++ | ++ | ++ | Applicable without tailoring  Covered: in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |
| 1b | Generation and analysis of equivalence classes | + | ++ | ++ | ++ | Applicable without tailoring  Covered: in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |
| 1c | Analysis of boundary values | + | + | ++ | ++ | Applicable without tailoring  Covered: in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |
| 1d | Error guessing based on knowledge or experience | + | + | ++ | ++ | Applicable without tailoring  Covered: in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |
| 1e | Analysis of functional dependencies | + | + | ++ | ++ | Applicable without tailoring  Covered: in Component Integration testing detailed in SmartTRX software test plan. RFE SW is a closed system in the case of Onechip, hence component level tests are the same as testing RFE SW in the “target environment” The exception to this is, in the case of Shark (when SPI, GPIO interfaces are considered and in the case of cascsading). |
| 1f | Analysis of operational use cases | + | ++ | ++ | ++ | Not used  No plan to do software update in the field, end-of-line-specific test bench mode for safeguarding production personnel etc |

- Annex C – the software configuration and calibration data are not in scope

**ISO26262 Part 7:**

* Not applicable.

**ISO26262 Part 8:**

* Applicable with the following exception: Evaluation of HW elements and Proven in use argument.

**ISO26262 Part 9:**

* Applicable with the following exceptions: ASIL decomposition and criteria for coexistence.

## Product Development – SW Level

### Initiation of SW Product Development

RFE SW activities are performed according to the SW Quality Process Baseline [1], covering by means of global procedures, guidelines, templates, checklists, training materials and product specific plans and guidelines, the ISO 26262 standard requirements applicable for the AP Software Projects.

Deviations from the SW Quality Process Baseline are described in the <https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Quality_Assurance/RFE_FW_RFE_GUI_Quality_Assurance_Plan.docx> section below. The Project QAE approves the deviations, during the current document review.

Figure 6 describes the deliverables of the RFE SW project mapped on the development phases.

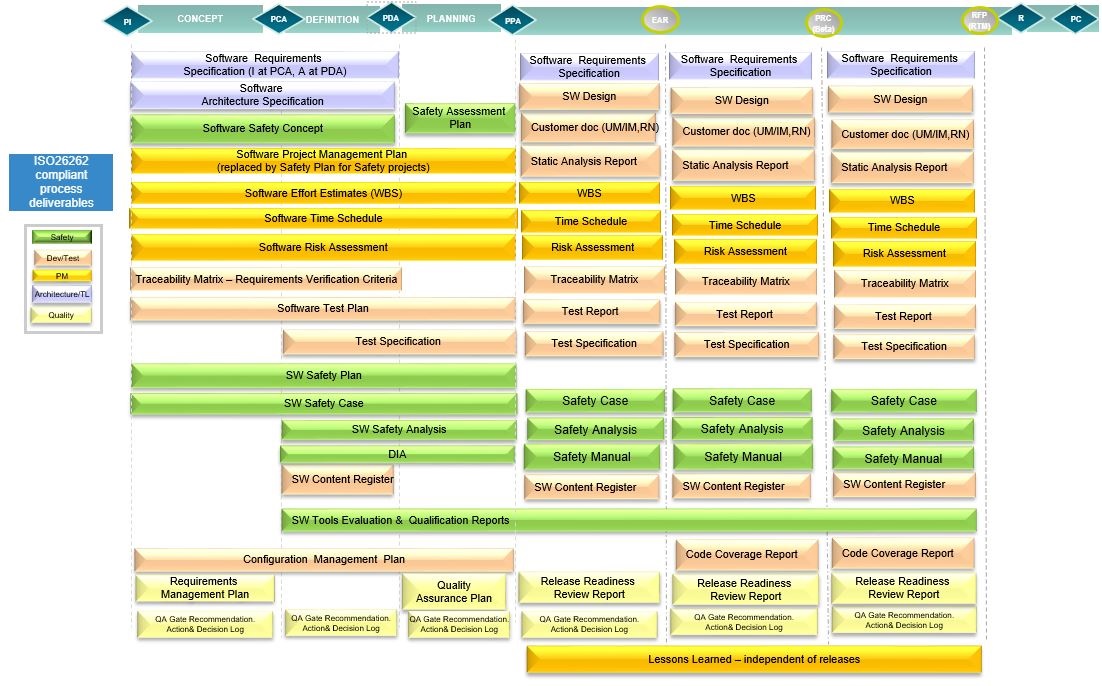
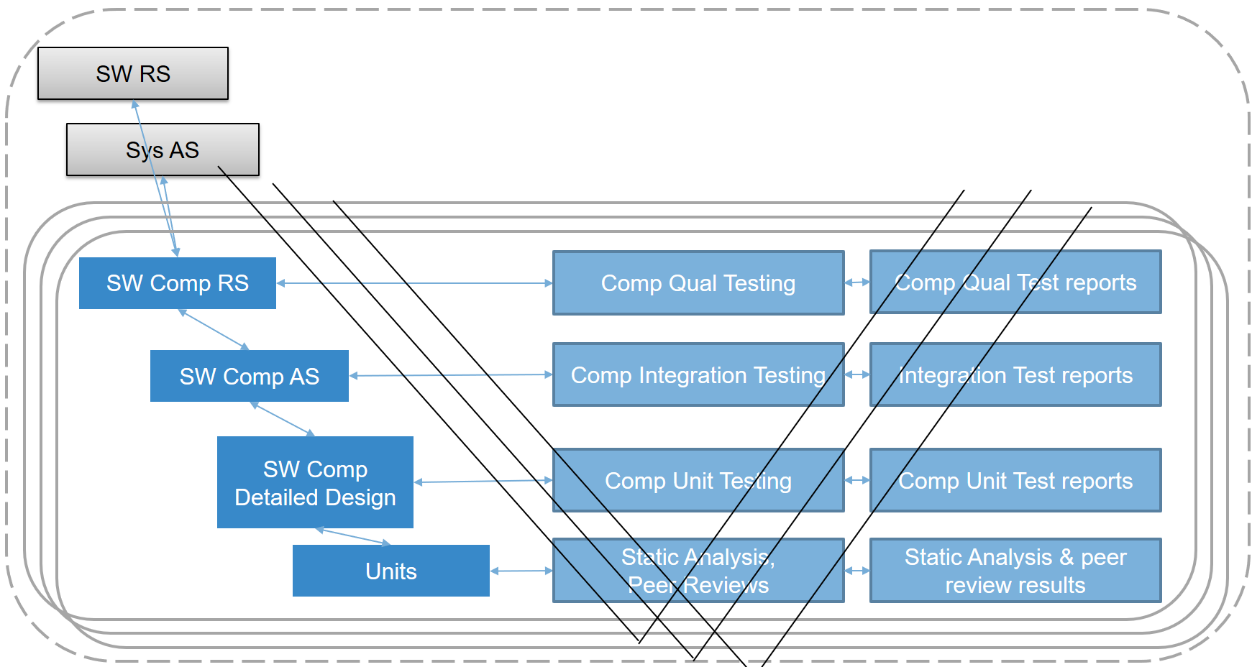


Figure 6 Product Lifecycle Deliverables

For each software development sub-phase (requirements, development, design, coding, testing), the reference model used is the V-model. The approach of applying the V-Model in the RFE SW project is described in Figure 7



Sys = RFE SW + HW

RFE SW = Calibration, Control, mmWave HAL and RFE driver

Figure 7 V-Model applicable for RFE SW SW Product Development

From A-SPICE perspective, the System is considered as the product obtained based on RFE SW and Hardware; it is out of scope of RFE SW project described in this Project Management Plan.

From ISO 26262 perspective, the Element is the RFE SW component, which consist of units: Calibration, Control, mmWave HAL and RFE driver. The interaction between units is documented in the [Architectural Specification of RFE SW](https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Software_Architecture/RFE_SW_Architectural_Specification.docx) [20].

RFE SW Test Strategy describes in detail the testing approach, test levels, definitions, as well as the test methodologies, this document will be available at execution phase (after PDA). For pre-EAR and EAR releases, a selected, reduced set of component tests is executed to verify the features implementation. The complete testing on unit, component and integration levels is executed for PRC and RFP releases.

Next chapters describe in detail the software development activities in the sub-phases of the V-cycle. The following items are specified for each sub-phase: the activities that shall be performed and the deliverables that shall be produced as results of these activities, the role responsible for creation, review and approval of the deliverables, the inputs/pre-requisites required, the outputs and their location, the verification planning, specification and reporting.

### Software Safety Concept

The Software Safety Concept is created as documented in BL AP Software Development Generic Plan [7], section 5.1.2, updated below with project specific information.

The AP Software Review Guideline [14] is applicable for rules on how to perform reviews.

| Activity | Software Safety Concepts Creation |
| --- | --- |
| **Objective** | Create the Software Safety Concepts  Identify the safety related functionalities and the architectural solution of a given product with respect to ISO 26262 requirements |
| **BCAM Phase** | Definition |
| **RASCI** | R – Software Safety Architect; A – n/a; S – n/a; C – n/a; I - Project Manager, Software Tech Lead, BU Software Safety Manager, project Quality Assurance Engineer |
| **Entry Criteria** | Concept Phase started. Activity planned. |
| **Inputs** | Software Product Requirements; inputs to HW Safety Manual; NPI Safety Concept; Software Architectural Design |
| **Tasks** | Software Safety Architect creates the Software Safety Concepts, containing: element definition, safety related faults (handling systematic and hardware random faults), safety requirements (recorded in Requirements Specification), safety measures (to be implemented or passed to the integrator as assumptions in the Safety Manual), safety architecture (for explicit safety projects), integration environment (FTE, GTE, NTE, safety assumptions) |
| **Exit Criteria** | Software Safety Concepts reviewed and approved. |
| **Outputs** | Software Safety Concepts . |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan [13] |
| **Template** | Software Safety Concept template [18] |
| **Verification Criteria and Planning** | Review type: Inspection or Walkthrough review; reviewed and approved by another Safety Software Architect or Software Safety Expert  CR of Technical Safety Concept at PPA |
| **Verification Specification** | In the peer review, verify that the template is followed  For FSA, use checklist |
| **Verification Result** | Crucible review  CR of Technical Safety Concept report |

Table 4 Software Safety Concepts Creation

### Specification and Management of Safety Requirements

The requirements specification activity is performed according to the BL AP SW Addendum to BU Requirements Management Procedure [12], containing generic description of activities, deliverables, roles, responsibility and skills required for requirements development and management, process workflow, requirements verification methods, change control and traceability.

Project specific information is documented in the STRX SW Requirements Management Plan [4] - detailed description of the way the RFE SW requirements are derived, analyzed and verified (activities and deliverables), including traceability matrix.

| Activity | SW Requirements Specification |
| --- | --- |
| **Objective** | Create component SW Requirements Specification containing functional and non-functional requirements |
| **Product Phase** | Primary analysis and allocation in Concept Phase  Detailed analysis in Definition Phase; updated thru Change Control process when necessary |
| **Responsible** | Component Technical Leads |
| **Reviewers** | Technical Lead, other Component Technical Leads, SW Safety Architect, SW Test Lead, Project Manager, QAE |
| **Approver** | Technical Lead, SW Safety Architect (for safety requirements) |
| **Inputs** | Sys Requirements Spec, Sys Arch Spec  HW Reference Manual as Hardware-Software interface specification for SEooC |
| **Outputs** | RFE SW Requirements Specification (at component level)  Updates in Risk Register and SW FMEA, if necessary |
| **Outputs Location** | DOORS NG RFE SW project |
| **Verification Planning** | 1.Verification method: Walkthrough for PDA and major changes after PDA (having impact on other modules Requirements Spec or Arch Specs); Walkthrough for changes not having impact on other modules or Arch Spec  2.Planning: before PDA and for each change  3. Tools used: Crucible |
| **Verification Specification** | See also the RFE SW Requirements Management Plan |
| **Verification Results** | Crucible record |

Table 5 Software Requirements Specification

There are **assumed** software safety requirements, documented in the Safety Concept and there are safety requirements derived from the Safety Analysis. The safety requirements follow the requirements management process described in the RFE SW Requirements Management Plan [4].

### Software Architecture Design, Detailed Design and Safety Analysis

The Software Architecture, Detailed Design and Safety Analysis are performed as documented in AP Software Development Generic Plan [7], sections 5.1.1, 5.1.3 and 5.2.1, updated below with project specific information.

The AP SW Review Guideline [19] is applicable for rules on how to perform reviews.

| Activity | Software Architectural Design |
| --- | --- |
| **Objective** | Design the software architecture and create the Software Architectural Specifications |
| **BCAM Phase** | Initiated in Concept, approved in Definition |
| **RASCI** | R – Software Architects; A – Software Tech Leaders; S – Software Tech Leaders, SoftwareSafety Architect, Software Test Lead; C – n/a; I - Project Manager, SoftwareSafety Manager |
| **Entry Criteria** | Concept Phase started. Activity planned. |
| **Inputs** | Software Requirements; HW Reference Manual; RFE SW Design Guideline [7]. |
| **Tasks** | The Software Architects create Software Architectural Specifications. Those contain the software components and their interactions with one another in a hierarchical structure; static aspects, such as interfaces and data paths of the software components, resource consumption objectives, as well as dynamic aspects, such as process sequences and timing behavior.  The requirements shall be traced in Software Architectural Specification as input for Traceability Matrix generation, as documented in RFE SW Requirements Management Plan. |
| **Exit Criteria** | Software Architectural Specifications approved |
| **Outputs** | Software Architectural Specifications |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan [13] |
| **Template** | SW Architectural Specification Template, translated into DOORS NG module & attributes. |
| **Verification Criteria and Planning** | Review type: Inspection  Control Flow Analysis and Data Flow Analysis performed using static analysis and design review.  Reviewed and approved by another Software Architect, Software Tech Lead and Software Safety Architect; reviewed by Software Test Lead |
| **Verification Specification** | Checklist for Architectural Spec review |
| **Verification Result** | Crucible review  Checklist under Configuration Management per project SCMP. |

Table 6 Software Architectural Design

| Activity | Safety Analysis of the Software Architecture |
| --- | --- |
| **Objective** | Perform the safety analysis of the Software Architecture in order to identify or confirm the safety-related mechanisms of software components. |
| **BCAM Phase** | Initiated in Planning, approved in Execution at EAR, PRC/BETA, RFP/RTM |
| **RASCI** | R – Software Safety Architect; A – other Software Safety Architect; S – Software Architect, Software Tech Leaders; C – n/a; I - Project Manager, SoftwareSafety Manager |
| **Entry Criteria** | Activity planned. Inputs available (see below) |
| **Inputs** | Software Requirements; Software Architectural Specification; Software Safety Concept; HW Architectural Specification/HW Safety Manual/HW FMEDA/HW Saftey Concept/SoC |
| **Tasks** | Software Safety Architect with contribution of Software Architect performs the Safety Analysis and determines the safety mechanisms and/or measures. The Software Safety Architect adds the safety mechanisms and measures to Software Requirements Specification, as “safety requirements”, or to Safety Manual as “assumptions”.  The following methods are used: Fault Tree Analysis (FTA); for each subsystem; SW Failure Mode and Effect Analysis (SW FMEA) for RFE SW component consist of units: Calibration, Control, mmWave HAL and RFE driver. |
| **Exit Criteria** | Safety Analysis approved |
| **Outputs** | Safety Analysis  Safety measures as: safety requirements in Software Requirements Specification and/or assumptions in the Safety Manual |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan |
| **Template** | SW DFMEA template [20] |
| **Verification Criteria and Planning** | Review type: Walkthrough; reviewed and approved by other Software Safety Architect, reviewer Software Architect  Confirmation review at EAR |
| **Verification Specification** | In the peer review, verify that the template is followed, and the document is complete, verify safety measures are documented in Requirements Spec and Safety Manual  For the confirmation review, use checklist |
| **Verification Result** | Crucible review  Confirmation review report |

Table 7 Software Safety Analysis

| Activity | Software Component Detailed Design |
| --- | --- |
| **Objective** | Create the component detailed design. |
| **BCAM Phase** | Execution, approved at EAR, PRC/BETA, RTM/RFP |
| **RASCI** | R – Software Engineer; A – Software Tech Lead; S – n/a; C – n/a; I - Project Manager, Software Tech Lead, SoftwareSafety Manager |
| **Entry Criteria** | Execution phase started. Activity planned. |
| **Inputs** | Software Requirements; Software Architectural Specification; Software Safety Concept; HW Reference Manual; Design Guidelines |
| **Tasks** | Software Engineer creates the detailed design documents describing in UML language static and dynamic views, as follows:   * Static View: file hierarchy, units and units interfaces, interaction with external components, data types * Dynamic View: dynamic behavior of the units, using activity diagrams and sequence diagrams   The requirements and/or architectural spec shall be traced in software component detailed design as input for Traceability Matrix generation, as documented in project specific Requirements Management Plan. |
| **Exit Criteria** | Software Component Detailed Design document approved |
| **Outputs** | Software Component Detailed Design (UML) |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan |
| **Template** | SW Detailed Design Specification template UML |
| **Verification Criteria and Planning** | Review type: Inspection  Review Moderator: Software Tech Lead in case of first complete version or major changes, another Software Engineer otherwise.  Reviewers: Software Architect, Software Safety Architect, Software Test Engineer  Before proceeding to the software unit testing phase |
| **Verification Specification** | Checklist for Detailed Design review [9], used for review once the design is complete or for major changes in design (optional to be used for minor changes) .  Verify that requirements derived from Safety Concept are implemented into the detailed design  The static analysis tool (MISRA checkers) used to verify design principles like initialization of variables, limited use of pointers (after implementation)  Verify the traceability and consistency between requirements, architectural spec and detailed design. |
| **Verification Result** | Crucible review  Checklist under Configuration Management per project SCMP. |

Table 8 Software Component Detailed Design

### Software Unit Implementation

The Software Unit Implementation is performed as documented in AP Software Development Generic Plan[7], section 5.3, updated below with project specific information.

The AP SW Peer Review Guideline [14] is applicable for rules on how to perform reviews.

Project specific information:

* code review is performed upon code merge from development branches to the main branches (per SCMP).

| Activity | Software Unit Implementation |
| --- | --- |
| **Objective** | Implement the software unit according to the software component detailed design and requirements. |
| **BCAM Phase** | Execution |
| **RASCI** | R – Software Engineer; A – Software Technical Lead; S – n/a; C – n/a; I - Project Manager, SoftwareSafety Manager |
| **Entry Criteria** | Activity planned. Inputs available |
| **Inputs** | Software Component Detailed Design; Software Component Requirements Specification; Reference Manuals; RFE SW Coding Guideline[10]; Compiler Release Notes |
| **Tasks** | Software Technical Lead decides and documents the coding style rules specific to the project, additional rules and static analysis checkers (if any), the compiler (compiler, linker, library manager, exe file format conversion tools) options and language extensions, the selected target, the optimizations (aggressive optimizations may expose defects in compiler), the warning level. Software Technical Lead analyzes the known issues of the compiler (input to FMEA and to Tools Evaluation process); in case new issues in compiler get detected, these are included in the known issues list and in the above analyses, communicated to compiler vendor and tracked as dependencies (if new versions of compiler are expected). The software engineers update the build control files (makefiles etc.) based on the decided compilation options.  Software Technical Lead decides and documents in the project specific coding style the deviations from the MISRA rules; the justification for deviation is reviewed by Software Engineers and project QAE.  Software Engineers implement in software code the component detailed design specification, at the same time ensuring that component requirements are fulfilled. In case there are requirements not captured in the design, they shall be fulfilled.  Software Engineers analyze and fix the errors and warnings reported during compilation, before committing the code to integration branches (as documented in the project specific Software Configuration Management Plan). Warnings that cannot be fixed are justified in the Complier Warnings Report.  Source code is written according to the Coding Guidelines.  The requirements, architectural specification and/or component detailed design are traced in software implementation as input for Traceability Matrix generation, as documented in project specific Requirements Management Plan. |
| **Exit Criteria** | Detailed Design is implemented, and source code is reviewed and reworked |
| **Outputs** | Software Code (.c, .h, .xml, .cfg files) and/or binaries (library), baselined per project SCMP  Updated Coding Guideline (compiler options, language extensions, assembly language usage)  Compiler Warnings Report (for RFP/RTM) |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan |
| **Template** | Source code templates referenced in link for RFE SW Coding Guideline [10]  Best Practices for Compiler Warning Reports are available at [AP Software - Architecture, Design, Coding](https://nxp1.sharepoint.com/:f:/s/ampsoftware/EnqSERUv1NtIsuyGUxvnCBQBmb9RYmr4ydCl235Ji25tAw?e=U5tcIQ) |
| **Verification Criteria and Planning** | Static analysis is performed using Coverity and is launched in execution by the continuous integration framework.  Source code review (inspection) by another Software Engineer, performed upon code merge from development branches to the main branches (per RFE SW SCMP)  Control Flow Analysis and Data Flow Analysis are performed by running static analysis tool.  Compiler Warnings Report (at RFP/RTM) – reviewed by another Software Technical Lead |
| **Verification Specification** | Static analysis executed based on checkers documented in Coding Guidelines.  Code Review Checklist, defined based on the RFE SW Design Guidelines and in Coding Guidelines, used for review once the code is complete (before starting the test campaign for PRC) or for major changes (optional to be used for minor changes) [11].  Review traceability and consistency between requirements or design to code  Review the reason for accepting compilation warnings. |
| **Verification Result** | Static Analysis Report  Code review in Bitbucket pull request/Crucible; checklist attached to Crucible record or under Configuration Management per project SCMP.  Crucible review for Compiler Warnings Report |

Table 9 Software Unit implementation

### Software Unit Testing and Integration Testing

Note:

1. The **units** are defined in the RFE SW Architectural Specification.

The Software Unit Testing and Integration Testing are performed as documented in AP Software Generic Test Plan[8], sections 5.1 to 5.10, updated below with project specific information.

The AP SW Peer Review Guideline[19] is applicable for rules on how to perform reviews.

| Activity | Test Plan (Test Strategy) Creation |
| --- | --- |
| **Objective** | Create the RFE SW Test Plan (Test Strategy) |
| **BCAM Phase** | Initiated in Concept, updated in Definition, approved in Planning and updated during the Execution phase, if needed |
| **RASCI** | R –Test Tech Lead; A – Project Manager; S –Test Architect, Test Engineers; C – SW Tech Lead, SW Architects, SW Safety Architect; I - project Quality Assurance Engineer, Safety Manager |
| **Entry Criteria** | Concept Phase started; initial requirements and architectural specification are available |
| **Inputs** | Software Requirements; Software Architectural Specifications; AP SW Generic Test Plan; project schedule |
| **Tasks** | Software Test Lead creates the project specific Test Plan and documents the following:   * Scope of testing - what is in /out of scope; * Quality objectives for all testing levels, if different from generic; * Assumptions and constraints for all testing levels, if different from generic; * Testing methodology – applicable test levels (unit, integration, qualification), containing test types, test techniques, test case derivation methods, tests organization; methods and techniques selected based on ASIL (with support from Test Architect); * Regression strategy for all testing levels (with support from Test Architect); * Testing activities schedule (high level); * Testing deliverables for all testing levels, if different from generic; * Resource and environment needs in terms of software and hardware resources, testing equipment, staffing and training needs, if different from generic. |
| **Exit Criteria** | RFE SW Test Plan (Test Strategy) approved |
| **Outputs** | RFE SW Test Plan (Test Strategy) |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan |
| **Template** | [Test Plan (Test Strategy) template](https://nxp1.sharepoint.com/:u:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AMP%20Software%20-%20Testing/BU%20Auto%20-%20SW%20Test%20Plan%20Template.url?csf=1&web=1&e=BmXngP) |
| **Verification Criteria and Planning** | Review type: Inspection  Reviewed and approved by Project Manager |
| **Verification Specification** | Verify that the test strategy is complete and consistent, contains all testing methods recommended by the Generic Test Plan or justification for methods not used. |
| **Verification Result** | Crucible review |

Table 10 Test Strategy Creation

| Activity | Software Test Specification |
| --- | --- |
| **Objective** | Create the test specification |
| **BCAM Phase** | Execution – first approved at EAR; updated and approved PRC and RFP |
| **RASCI** | R – Test Engineer; A – Test Tech Lead; S – n/a; C – Sw Engineer, Test Architect; I –Software Safety Manager |
| **Entry Criteria** | Concept Phase started |
| **Inputs** | Project Specific Test Plan (Test Strategy); Detailed Design; Arch Specification; Software Requirements (component level); HW Reference Manual; List of known defects |
| **Tasks** | The recommended methods for testing and for deriving test cases and the coverage methods are defined in the BL AP Software Generic Plan (Test Strategy) .  Test Engineer creates the Test Specification and documents the following:   * the test environment and constraints and, for each test case, name/ID, description, test level, technique, preconditions, expected result, pass/fail criteria, reference back to the tested detailed design or component requirements; * test cases added/updated for verifying that defects have been removed.   The detailed design shall be traced in unit test cases as input for Traceability Matrix generation, as documented in RFE SW Requirements Management Plan [4]**.**  The architectural specification shall be traced in integration test cases as input for Traceability Matrix generation, as documented in RFE SW Requirements Management Plan. |
| **Exit Criteria** | Test Specification reviewed and approved |
| **Outputs** | Test Specification |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan |
| **Template** | [SW Test specification Template](https://nxp1.sharepoint.com/:w:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20TEMPLATES/BCaM7%20Automotive%20SW%20Test%20Specification%20Template.docx?d=w9ba0a11179bc4f1681a6f0ce8f3a8cf4&csf=1&web=1&e=e185zx) |
| **Verification Criteria and Planning** | Review type: Walkthrough or Mini-Walkthrough review  Reviewed and approved by Test Tech Lead |
| **Verification Specification** | Verify that the unit test spec:  - complies with the unit detailed design;  - complies with specification of the hardware-software interface;  - verifies the non-functional requirements;  - proves the absence of unintended functionality;  - proves robustness (example: the absence of inaccessible source code, effectiveness of error detection and error handling mechanisms).  Verify that the integration test spec:  - complies with the architectural spec and hardware – software interface;  - proves robustness (example: the absence of inaccessible source code, effectiveness of error detection and error handling mechanisms). |
| **Verification Result** | Crucible review |

Table 11 Software Test Specification

| Activity | Software Test Implementation |
| --- | --- |
| **Objective** | Implement the test code based on the test specifications |
| **BCAM Phase** | Execution |
| **RASCI** | R – Test Engineer; A – another Test Engineer; S – n/a; C – n/a; I - Project Manager, Software Safety Manager |
| **Entry Criteria** | Test Specifications available |
| **Inputs** | Test Specifications; Coding Guideline |
| **Tasks** | Test Engineer implements the test specification in test code. |
| **Exit Criteria** | Test Code reviewed and approved |
| **Outputs** | Test Code (.c, .h, .xml, .cfg files) |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan |
| **Template** | [test code templates](https://nxp1.sharepoint.com/:f:/s/ampsoftware/EuzKA1FymFhFnxqvOvjxcqUBw5wMXaTm3Kp1tXsXKUsFug?e=BOX9oI) |
| **Verification Criteria and Planning** | Review type: Walkthrough or Mini-Walkthrough  Review Moderator: another Software Test Engineer. |
| **Verification Specification** | Verify that test code implementation complies with test specification |
| **Verification Result** | Bitbucket pull request |

Table 12 Software Test Implementation

| Activity | Software Test Execution |
| --- | --- |
| **Objective** | Execute tests as described in Test Plan and in Test Specification |
| **BCAM Phase** | Execution |
| **RASCI** | R – Test Engineer; A –Test Tech Lead; S – n/a; C – n/a; I - Project Manager, Software Safety Manager |
| **Entry Criteria** | Execution phase started  Testing Environment is setup and available  Source Code build to be tested is available |
| **Inputs** | Release Test Plan; Tests Specification; Test Code; Source Code build to be tested |
| **Tasks** | Test Engineers execute the unit testing per Test Plan  The detailed tasks/steps are documented in the RFE SW Test Plan (Test Strategy) |
| **Exit Criteria** | Test Execution Suspension and Resumption Criteria:   * Defect(s) block the test execution (e.g. catastrophic severity); * Defects produce a situation where they cumulatively mean testing has no value at a given point in time (little or no sense to continue testing).   Test Completion Criteria:   * Unit tests executed per Test Plan; * Blocked and Not executed tests documented in Test Report; * Failed tests reviewed with developers and defect submitted in system; * Code coverage target mentioned in Release Test Plan for the specific test campaign is achieved.   The Test Tech Lead has the authority to suspend/resume/complete/stop the test execution before criteria are met, with provided rationale. |
| **Outputs** | Test Report containing PASS/FAIL results per executed test case (may be an independent document, or included in the overall test report together with the rest of the test levels) and BLOCKED/NOT EXECUTED for not-executed test cases (with reason) |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan |
| **Template** | [Test Report template](https://nxp1.sharepoint.com/:x:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Testing/Best%20Practices/AMP%20Tools%20-%20Product%20Test%20Report%20Template.xlsx?d=wcb613a4721e74ab587c6f66968b9f29f&csf=1&web=1&e=YhyZRW) |
| **Verification Criteria and Planning** | Review type: Walkthrough review  Reviewed and approved by Test Tech Lead  At Release Readiness Review, reviewed by Quality Assurance Engineer |
| **Verification Specification** | Verify that   * Test execution is compliant with Release Test Plan and Test Spec; * Failed tests have corresponding defects submitted in JIRA; verify that defects have the appropriate severity, version, component; * All “not executed” test cases have documented reason. |
| **Verification Result** | Crucible review |

Table 13 Software Test Execution

### Release Activities

The release activities are performed as documented in AP [Software Generic Release Plan](https://nxp1.sharepoint.com/:x:/r/teams/206_16/Shared%20Documents/STRX/ProjectPlanning/STRX_SW_Release%20Plan.xlsx?d=w3e56ecd62ff7491b9dd8eda8f33f2987&csf=1&web=1&e=0i8rqX) **.**, updated below with project specific information.

The AP SW Peer Review Guideline is applicable for rules on how to perform reviews.

| Activity | Software Release Activities |
| --- | --- |
| **Objective** | Create, release and distribute RFE SW Package, Quality Package, Safety Package |
| **BCAM Phase** | Execution |
| **RASCI** | R – Release Engineer; A – Project Manager; S – Test Engineer, Configuration Manager; C – n/a; I - Software Technical Lead, Software Safety Manager |
| **Entry Criteria** | Source Code and reports available  Quality GO from Release Readiness Review |
| **Inputs** | Source code, user documentatation  Traceability matrix, test specs, test reports, static analysis reports, code coverage reports, size reports for Quality Package  SW FMEA reports and Safety Manual for Safety Package  For release - Quality decision GO after Release Readiness Review |
| **Tasks** | Release Engineer creates the packages.  Test Engineer performs smoke tests |
| **Exit Criteria** | Package available, smoke testing pass |
| **Outputs** | RFE SW Release Package, Release Notes  Quality Package (PRC/BETA, RFP/RTM)  Safety Package (initial at EAR, updated at PRC/BETA, final at RFP/RTM) |
| **Output Location** | Flexera |
| **Template** | Not applicable |
| **Verification Criteria and Planning** | Smoke tests done on package, per Release Test Plan |
| **Verification Specification** | Per Release Test Plan  Verify in addition that:   * Release Notes are updated with latest release content * RFE SW Package is generated based on the latest approved release candidate in Release Readiness Review * Quality Package and Safety Package are complete and contain the reviewed, final versions of the documents, referring to the latest approved release candidate in Release Readiness Review |
| **Verification Result** | Notification of smoke test result by email |

Table 14 Software Release Activities

### User Documentation

The user documentation is developed as documented in AP Software [Development Generic Plan](https://nxp1.sharepoint.com/:x:/r/teams/206_16/Shared%20Documents/STRX/ProjectPlanning/STRX_SW_Release%20Plan.xlsx?d=w3e56ecd62ff7491b9dd8eda8f33f2987&csf=1&web=1&e=0i8rqX) **.**, section 5.3.5, updated below with project specific information.

The AP SW Peer Review Guideline is applicable for rules on how to perform reviews.

| Activity | Software User Manual and Release Notes Creation |
| --- | --- |
| **Objective** | Elaborate the User Manual, the Product Brief and the Release Notes |
| **BCAM Phase** | Execution (pre-EAR, EAR, PRC/BETA, RFP/RTM) |
| **RASCI** | R – Software Engineer; A – Software Technical Lead; S – n/a; C – n/a; I - Project Manager, SoftwareSafety Manager |
| **Entry Criteria** | Activity planned. Inputs available. |
| **Inputs** | Software Requirements Spec; Architectural Specification; Detailed Design; Test Reports; defects database. |
| **Tasks** | Software Engineer elaborates the User Manual, Product Brief and Release Notes |
| **Exit Criteria** | User Manual, Product Brief and Release Notes approved |
| **Outputs** | User Manual, Product Brief and Release Notes |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan in Release Package |
| **Template** | [User Manual template](https://nxp1.sharepoint.com/:w:/s/ampsoftware/EXeg4qaGV4tNgjSIHQPOBVMBMVIgX5Yodo6cfv3xgLYs3Q?e=IyhW8t), [Product Brief](https://nxp1.sharepoint.com/:w:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AMP%20Software%20%20-%20Development/BL%20AMP%20Software%20Product%20Brief%20(Template).docx?d=w6bef26a9a20742d98bfc473fd6ae1056&csf=1&web=1&e=yxb1D5) and [Release Notes template](https://nxp1.sharepoint.com/:w:/s/ampsoftware/ESl5qJS_5IlCrwl_Su99cW8BV5rxmRX1EbV5zxoV3y5iKw?e=dcfZqJ) |
| **Verification Criteria and Planning** | Review type: Walkthrough, reviewed and approved by Software Technical Lead, reviewer: project Quality Assurance Engineer |
| **Verification Specification** | Verify that the content of the User Manual is consistent with the release type and scope.  Verify that the content of the Release Notes is complete and consistent with the release type and scope. |
| **Verification Result** | Crucible review |

Table 15 Software User Manual, Product Brief and Release Notes Creation

| Activity | Software Safety Manual Creation |
| --- | --- |
| **Objective** | Elaborate the Software Safety Manual |
| **BCAM Phase** | Execution - initial at EAR, updated at PRC/BETA, approved at RFP/RTM |
| **RASCI** | R – Software Architect; A – Software Safety Manager; S – Software Tech Leaders; C – n/a; I - Project Manager |
| **Entry Criteria** | Activity planned. Inputs available (see below) |
| **Inputs** | Software Safety Concept; External Assumptions; Safety Analysis |
| **Tasks** | Software Safety Architect elaborates the Safety Manual |
| **Exit Criteria** | Software Safety Manual approved |
| **Outputs** | Software Safety Manual |
| **Output Location** | Per RFE SW Configuration Items Baseline Plan [13] in Safety Package |
| **Template** | Software Safety Manual template [21] |
| **Verification Criteria and Planning** | Review type: Walkthrough, reviewed and approved by Safety Manager, reviewers: Software Tech Lead, project Quality Assurance Engineer |
| **Verification Specification** | Verify that Safety Manual is complete and consistent with SW Safety Concept and Safety Analysis |
| **Verification Result** | Crucible review |

Table 16 Software Safety Manual Creation

## Supporting Processes

### Interfaces with Distributes Development

RFE SW is tested within NXP SW Validation Team in BGL.

### Configuration Management

From configuration management standpoint, RFE SW software development is performed according to the [RFE SW Configuration Management Plan](https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Change_and_Configuration_Management/RFE_Configuration_Management_Plan.docx) **.**, in terms of branching, folders, labeling strategy.

|  |  |
| --- | --- |
| **Activity** | **Define and document Configuration Management environment and process** |
| **Objective** | Define the Configuration Management strategy, environment to be used in the project and create SCMP. |
| **BCaM Phase** | PCA to PPA |
| **RASCI** | R – SW Configuration Manager, SW Engineer, Quality Assurance Engineer; A – Project Manager; S- SW Engineer(s); C – n/a; I – SW Engineer(s) |
| **Entry Criteria** | Project concept phase started |
| **Inputs** | Project and process requirements; AP SW Change, Configuration Management and Reviews area |
| **Tasks** | Project and Team Manager assigns project Configuration Manager.  Project Configuration Manager together with project team, define the environment including the configuration control tool to be used.  Project Configuration Manager identifies the criteria for selecting the items to be placed under configuration control.  Project Configuration Manager, together with project team, defines configuration management strategy in terms of branching, folders structure, labelling/tagging, naming conventions, etc.  Project Configuration Manager or Quality Assurance Engineer creates project software Configuration Management Plan and document configuration management strategy.  The Configuration Management Plan must be reviewed and approved before PPA gate. The author of the SCMP plans the review and creates review record.  After the Configuration Management Plan is approved, the author places final, approved version in the corresponding repository, according to Project Software Configuration Management Plan and Configuration Items Plan. |
| **Exit criteria** | Project Configuration Manager is allocated. Project Configuration Management environment system and strategy are defined. SCMP is available. |
| **Outputs** | Project Configuration Management Plan |
| **Output Location** | As defined in Project Configuration Items Plan |
| **Template** | [Configuration Management Template](https://nxp1.sharepoint.com/:u:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AMP%20Software%20-%20Change%20Configuration%20Management%20and%20Reviews/BU%20Auto%20-%20Configuration%20Management%20Plan%20Template.url?csf=1&web=1&e=tmBlc8) |
| **Verification Criteria and Planning** | Review type: Inspection, Walkthrough or Mini-walkthrough review  Reviewed and approved by the Project Manager |
| **Verification Specification** | Project change and configuration management strategy is documented. BU Auto Configuration Management Template is used for project SCMP and all sections are filled-in. |
| **Verification Results** | Crucible/Pull Request review |

Table 17 Configuration Management Plan definition

### Change Management

From change management standpoint, RFE SW software development is performed according to the [RFE SW Configuration Management](https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Change_and_Configuration_Management/RFE_Configuration_Management_Plan.docx) Plan **.**, in terms of change management rules.

|  |  |
| --- | --- |
| **Activity** | **Define Change Management environment and process** |
| **Objective** | Define change management process and set up environment and customize tool schma per project needs |
| **BCaM Phase** | PCA and PDA |
| **RASCI** | R – Quality Assurance Engineer, Tool Admin, Configuration Manager; A- SW Quality Manager; S – Project Team (SW Engineer); C – n/a; I – Project Manager |
| **Entry criteria** | Project start-up |
| **Inputs** | Project and process requirements;  AP SW Change, Configuration Management and Review area |
| **Tasks** | Identify Change management related process and project requirements.  Create and configure the project in Atlassian JIRA.  Define the way of working, process particularities, applicable workflow, project specific rules and JIRA schema changes needed.  Set-up project change control board (CCB or scrum/synch meeting) |
| **Exit Criteria** | Change management process is established and JIRA project is created |
| **Outputs** | JIRA project set-up |
| **Output Location** | Atlassian JIRA |
| **Template** | n/a |
| **Verification Criteria and Planning** | n/a |
| **Verification Specification** | n/a |
| **Verification Results** | n/a |

Table 18 Define Change Management environment

### Verification

Verification activities are planned, specified, executed and reported as follows.

#### Peer reviews

See the AP SW Review Guideline [14].

See also RFE SW Software Quality Assurance Plan [6].

#### Release Readiness Review

The RFE SW deliverables and release criteria are verified during Release Readiness Reviews using specific checklist.

| Activity | Release Readiness Review |
| --- | --- |
| **Objective** | Performed RRR for <Project Name> before each delivery |
| **Product Phase** | Execution phases – updates |
| **Responsible** | Project Quality Assurance Engineer |
| **Reviewers** | Other SW Quality Assurance Engineer/ SW Quality Assurance Manager |
| **Informed** | SW Project Manager, SW Quality Assurance Manager, SW Safety Manager; RFE SW team, RFE SW Team Manager |
| **Approver** | SW Quality Assurance Manager/Corporate Quality Systems Manager during ISO 26262 assessments |
| **Inputs** | Project documentation  Attlassian JIRA database |
| **Outputs** | RRR Report |
| **Outputs Location** | Per CI Baseline plan |
| **Verification Planning** | n/a |
| **Verification Spec** | n/a |
| **Verification Results** | n/a |

Table 19: Release Readiness Review

#### SW Testing

Testing activities are estimated and scheduled like all other activities in the project, and can be found in schedule. Test cases are described in detail in Test Specification. Tests are executed as documented in the Test Specification and the results are reported in the test summary reports.

#### Traceability Matrix

The traceability matrix is used as a method of software verification, through which one checks that the requirements have been implemented in code, and that those requirements have been tested.

#### Static Code

The code is verified using static analysis tool (e.g Coverity). The conformance with MISRA & CWE standards is verified using the same tool. The adherence to the coding guideline is verified using peer reviews. The data and control flows are verified using the static analysis tool and by peer reviews.

#### Code Coverage

Measuring the number of statements and branches covered by test execution is a method for verification, highlighting the code that is not verified through test cases and that must be verified by other means (like reviews). Gcov may be used in early testing stages for measuring the statement and branch coverage; for PRC and RFP releases, a functional-safety certified tool will be used (Lauterbach T32).

### Confidence in Software Tools

The tools used during RFE SW products development are included in the Tools Qualification Program as documented in AP SW Generic Safety Management Plan [10] – for generic description of the qualification activities, deliverables, roles, responsibility and skills required for performing this specific activity, process workflow, qualification methods.

The results of this process are listed below:

* RFE SW Tools Evaluation and Qualification Report [17] – list of tools assessed in terms of tool confidence level actual and targeted, action plan for increasing the confidence level; for capturing the results of the tool qualification; goes thru Confirmation Review at EAR.
* Tools User Manual – for explaining the way tools should be used, when the need was decided as result of the evaluation.

For safety projects it is very important to select the proper tools that will be used during project lifecyle because activities performed for software safety product may rely on the correct functioning of the tool.

| **Activity** | **Tool Qualification method selection** |
| --- | --- |
| **Objective** | Select the appropriate SW tool qualification method according to ISO 26262 - 8, clause 11.4.6.1 requirement. |
| **BCAM Phase** | PCA to PPA |
| **RASCI** | R – Safety Architect and SW Quality Engineer ; A – Safety Manager ; S – Project Manager ; C – n/a ; I – Project Team (SW Engineer) ; |
| **Entry Criteria** | The TCL is established for each software tool. |
| **Inputs** | Tools evaluation result from SW Tool Evaluation and Qualification report. |
| **Tasks** | * Select a tool qualification method, according to the ISO 26262 criteria (ISO 26262 - 8, clause 11.4.6.1 , table 4 and table 5) and depending on the TCL determined in the previously steps. * A qualification method is selected only for tools that have TCL 2 and TCL 3. * A tool classified as TCL 1 does not require the implementation of qualification method. For SW tools classified at TCL 2 or TCL 3, at least one dedicated tool qualification method shall be implemented. * For SW tools classified at TCL 3, the following qualification methods are recommended: * Increased confidence from use; * Evaluation of the tool development process; * Validation of the software tool; * Development in accordance with a safety standard; * The method selection is also impacted by the maximum ASIL targeted in the safety project; * Record the chosen method into Tool Evaluation Qualification Report and document the reason why the method was selected. * **Note**: For AP safety software projects it is recommended that the SW Tools are qualified by implementing a method required for ASIL D, if applicable. |
| **Exit Criteria** | SW Tool Qualification method is selected and documented in SW Tool Evaluation and Qualification report. |
| **Outputs** | SW Tool Evaluation and Qualification report. |
| **Output Location** | Per project Configuration Items Plan. |
| **Template** | BL AP - SW Tool Evaluation Qualification template [22] |
| **Verification criteria and planning** | Review type: Inspection, Walkthrough review.  Reviewed and approved by the Safety Manager. |
| **Verification Specification** | Verify if tool qualification method is correctly chosen and documented in SW Tool Evaluation Qualification. Verify if the defined template is used. |
| **Verification Results** | Crucible/Pull Request review. |

Table 20 Confidence in SW Tools

### Quality Assurance

The following generic plan, templates, guidelines shall be followed:

* AP Software [Quality Assurance Generic Plan](https://nxp1.sharepoint.com/sites/ampsoftware/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAP%20Software%20%2D%20Quality%20Assurance%2FBL%20AP%20Software%20Generic%20Quality%20Assurance%20Plan%2Epdf&parent=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAP%20Software%20%2D%20Quality%20Assurance) **.** – for generic description of quality assurance activities, deliverables, roles, responsibility and skills required for performing this specific activity, process workflow;
* BU Auto [Software Safety Assessments Procedure](https://nxp1.sharepoint.com/:u:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Safety%20Management/BL%20AP%20SW%20Safety%20Plan%20Template.url?csf=1&web=1&e=NYsc7w) **.**;
* RFE SW [Software Quality Assurance Plan](https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Quality_Assurance/RFE_FW_RFE_GUI_Quality_Assurance_Plan.docx) **.** – for detailed description of the way the RFE SW quality assurance activities shall be performed/available (SQAP creation; Release Readiness Review; BCaM7 Compliance Audits).

Software Quality Assurance Plan (SQAP) creation and Metrics Plan creation activities are described in the table below:

| Project  Activity | **SQAP Creation and Metrics Plan Creation** |
| --- | --- |
| Objective | Crete Quality Assurance Plan (SQAP) and Metrics plan for RFE SW.  Plan quality assurance activities and document them in SQAP. |
| Product Phase | Official, approved at PPA, updated if needed for EAR, PRC, RFP |
| Responsible | Project Quality Assurance Engineer |
| Reviewers | SW Project Manager, SW Safety Manager, (Project Team) SW Engineer |
| Approver | SW Project Manager, SW Quality Assurance Manager |
| Inputs | RFE SW project plans (including release plan).  AP SW quality goals. |
| Outputs | RFE SW SQAP.  RFE SW Metrics Plan  Review Records (Atlassian Bitbucket PullRequest/Crucible reviews). |
| Outputs Location | Per CI Baseline plan |
| Verification Planning | 1.Verification method: Peer Review according to the defined criteria in section 5.3.4.1  2.Planning: in Project Schedule, taking in account peer review steps and roles involved as documented in the AP SW Review Guideline  3. Tools used: Atlassian Bitbucket PullRequests/ Crucible reviews |
| Verification Spec | Verify that the content of the SQAP is complete and consistent with RFE SW project goals and plans. |
| Verification Results | Atlassian Bitbucket PullRequest review record/ Crucible review record |

Table 21 SQAP Creation

#### Safety Case

The creation of the Safety Case is documented in the AP Safety Management Procedure [4], updated with project specific information below.

The AP SW Review Guideline [14] is applicable for rules on how to perform reviews.

| Activity | Safety Case Creation |
| --- | --- |
| **Objective** | Create the Safety Case |
| **Product Phase** | Preliminary version at PCA+PDA, incrementally updated at PPA, EAR, PRC  Final, approved version at RFP |
| **Responsible** | SW Quality Engineer |
| **Reviewers** | Project Manager, Safety Manager, Quality Assurance Manager, RFE SW Tech Leads, Test Technical Lead, Safety Architect |
| **Approver** | Safety Manager |
| **Inputs** | ISO 26262 Standard  RFE SW Project Documentation  AP SW Process Documentation |
| **Outputs** | RFE SW Safety Case  Peer Review Records (Crucible/ Atlassian Bitbucket Pull request) |
| **Outputs Location** | In Bitbucket Per RFE SW Configuration Items Baseline Plan |
| **Verification Planning** | Confirmation review at RFP |
| **Verification Spec** | In the checklist |
| **Verification Results** | Checklist |

Table 22 Safety Case Creation

#### Safety Confirmation Measures

There are three types of Confirmation Measures, which are described in the Automotive Functional Safety Assesment Procedure [3]: Safety Confirmation Reviews; Safety Audit; Safety Assessment.

The confirmation reviews and functional safety audits can be merged and combined with the functional safety assessment to support the handling of comparable variants of an item, per ISO 26262 requirements.

Safety Confirmation Measures Plan for RFE SW project can be found in RFP SW Assessment Plan [17].

| Activity | Safety Confirmation Reviews |
| --- | --- |
| **Objective** | Perform Safety Confirmation Reviews for: Safety Plan (Project Management Plan); Safety Analysis; Tools Qualification Documentation; Safety Case, as required by ISO 26262.  The confirmation reviews verify the correctness with respect to formality, contents, adequacy and completeness regarding the requirements of ISO 26262 of the above listed documents. |
| **Product Phase** | Planning and Execution Phase – safety related PPA, PRC, RFP |
| **Responsible** | I3; AP Safety Assessor + Independent FS assessor (from CTO Functional Safety team), for all except Tools Qualification, where responsible is member of QAE team other than RFE SW assigned QAE |
| **Inputs** | RFE SW project and process documentation in scope for review |
| **Outputs** | Confirmation Review Reports |
| **Outputs Location** | As per AP SW Generic Safety Management Plan [10] |

Table 23 Safety Confirmation Reviews

| Activity | Safety Audit |
| --- | --- |
| **Objective** | Perform Safety Audit, for verifying the implementation of the processes against the definitions of the activities referenced or specified in the safety plan  TUV Sud certification is an alternative for the safety audit. |
| **Product Phase** | Execution Phase - RFP |
| **Responsible** | I3; AP Safety Assessor + Independent FS assessor (CTO Functional Safety Team) |
| **Inputs** | Global procedures, RFE SW specific plans, project work products |
| **Outputs** | Safety Audit Report |
| **Outputs Location** | As per AP SW Generic Safety Management Plan [10] |

Table 24 Safety Audit

| Activity | Safety Assessment |
| --- | --- |
| **Objective** | Perform Safety Assessment, for verifying the work products required by the safety plan, the implementation of the required processes and for reviewing the implemented safety measures. |
| **Product Phase** | Execution Phase - RFP |
| **Responsible** | I3; AP Safety Assessor + Independent FS assessor (CTO Functional Safety Team) |
| **Inputs** | Global procedures, RFE SW specific plans, project work products |
| **Outputs** | Safety Assessment Report |
| **Outputs Location** | As per AP SW Generic Safety Management Plan [10] |

Table 25 Safety Assessment

### Qualification of Software Components

* NA

## RFE SW Project Planning

### Project Planning and Tracking

RFE SW Project Lead handles the 1st level of planning, long term (6 -12 months) release plan, schedules all RFE SW releases in context of STRX SW program and sets release dates (official customer delivery dates).

The RFE SW Project Lead schedules internal RFE SW releases, plans the entire development and testing team activities. The release dates are agreed with STRX SW Program Manager and all internal stakeholders (marketing, systems, apps, tools).

This high-level release schedule is documented in the “[Product SW Roadmap](https://nxp1.sharepoint.com/:x:/r/teams/206_16/Shared%20Documents/STRX/ProjectPlanning/STRX_SW_Release%20Plan.xlsx?d=w3e56ecd62ff7491b9dd8eda8f33f2987&csf=1&web=1&e=0i8rqX)” **.**.

The high-level resource planning, effort estimations, resources load leveling is done considering the complexity of each release, public holidays and available engineering bandwidth.

There is a schedule used for the entire RFE SW project, available in the above Product SW Roadmap.

If not included in RFE SW Master Plan, a separate Gantt chart must be prepared for testing activities, the high-level milestones per releases should be like the ones from RFE SW Master Plan.

The project team members (Engineers, SW Architect, project SQE, Safety Architect, integrator, etc.) provide effort estimates for each project milestone.

# Project Tailoring

RFE SW Project is developed following a “Class A” development process, according to AP SW Tailoring Guideline [21]. The reason for “Class A” selection is the customer’s ISO26262 compliance request.

This chapter contains the list of deviations from the global generic plans: area, tailored items and the section in the tailoring guideline which specifies the allowed/not allowed deviations. Functional safety specific tailoring is available in Safety Case argumentation.

Deviations list is approved by project SQE.

| **Area In-Scope for Tailoring** | **Tailored Items** |
| --- | --- |
| Product Lifecycle | RFE SW project follows the BCaM7 phases – Concept Definition, Planning, Execution.  There are presilicon releases (pre-EAR), delivered before EAR (which is on silicon).  The project SQE performs BCaM7 Phase Gate reviews based on AMD and AMC from BU Auto level. |
| SW Development Lifecycle | Acceptance Testing Phase and System Integration testing are out of scope. |
| Project Management | - Inputs and outputs of the process deliverables, applicable for RFE SW project, are documented in the current plan, not in the global procedure;  - Project planning process is described in the current project plan, chapter 5.4;  - Allocated resources for the next BCaM7 phase can be seen in RFE SW schedule Gantt.  Estimation deviations:  - Only effort/ schedule estimation is performed; it is done using expert opinion;  - Size estimation based on LOC and number of requirements is not performed;  - Critical resources (e.g. ROM, RAM, MIPS) are not estimated;  - Effort (estimated and actual) used as input for project Gantt is based on project tasks.  Project milestones are estimated based on the set of tickets planned to be implemented for a release; effort estimation of a development activity contains requirements analysis and/or updates, design development and/or updates, implementation, verification (static analysis, peer reviews), configuration management and change management related activities;  - Estimating and tracking the costs are performed at BL AP SW level, outside the scope of RFE SW project.  - The WBS and estimations at PDA are documented in Gant; the WBS and estimations for the next milestone are detailed and complete at PPA;  - The WBS and estimations are collected using excel based WBS, not using JIRA fields;  - Reviews of the WBS, estimations and schedule are performed in bi-weekly meetings, no Crucible review;  - Reviews of the risk register are performed in monthly meetings, no Crucible review;  - Cost related estimations are out of scope, they are performed at BL AP SW level.  Risk management: the Risk Register is used to capture risks and issues at all levels within RFW SW project scope. |
| Change Management  and Problem Resolution | JIRA workflow:  - CCB state can be skipped for tickets that do not require decision from RFE SW CCB  - Effort -related fields are not used for estimation and tracking. |
| Quality Assurance | - Quality review is done as part of the site Quality Management Review and AMP SW management reviews;  - RRR meetings are held only there is NO GO decision and requested waiver/escalation;  - Customer Satisfaction survey is not sent, feedback is received thru Marketing channels;  - Metrics plan is BL AP SW global, no specific metrics for RFE SW project;  Lessons Learned Process |
| SW Metrics | The information in the RFE SW project metrics plan is included in RFE SW SQAP and in BL AP SW global metrics program. Metrics analysis is performed only for deviations from targets at RRR. |
| Requirements Engineering | The product requirements and system architectural specifications are managed in DOORS NG system, and the RFE SW requirements specifications are managed in DOORS.  The PRD and MRD are owned by the Marketing team and are global for entire program, they are outside of the RFE SW scope and we consider them as inputs into RFE SW project (various baselines).  Product Requirements Collection:   * It is out of scope of the RFE SW project.   Software Requirements Analysis:   * Preliminary (section 1): the preliminary software requirements allocation activity is out of scope of the RFE SW project. The preliminary analysis activity is in scope; no tailoring. DOORS used as SWRA database * Detailed (section 2): no tailoring. DOORS used as SWRA database   Requirements Change Control:   * It refers to PRD only and is out of scope of RFE SW project. * However, if the SDK Tech Leads receive/identify a change request for the software requirements specifications, they record it in Jira and go thru the Change Management flow (impact analysis, CCB, implementation, verification)   Requirements Traceability:   * The creation of the traceability matrix is documented in the RFE SW Requirements Management Plan. * The review of the traceability matrix is recorded in Crucible * The Requirements Review Checklist is not used for review |
| SW Design (Technical Solution) | - Activities inputs and outputs applicable for RFE SW project are documented in the current plan, not in the global procedure.  - SW Architectural Spec and Component Detailed Design are documented in the same document (Enterprise Architect project); there is no other document that describes the Architecture or the Detailed Design.  - Unit Design as required by ISO26262, is part of the SW Architecture & Detailed Design Process. |
| SW Construction (Technical Solution) | - Coding may start before design is completed; requirements, architectural specification, APIs and detailed design may change, change will be controlled thru Change Requests (Jira tickets) in case those have been baselined before, otherwise the only change record will be in version control system  - Activities inputs and outputs applicable for RFE SW project are documented in the current plan, not in the global procedure.  - The allowed deviations from MISRA required rules for RFE SW are documented in Coding Guideline. The reason for accepting the violations are documented in MISRA report.  - Code review based on checklist is performed at least once before PRC and upon major changes to a previously reviewed code base. |
| Software Testing | - Activities inputs and outputs applicable for RFE SW project are documented in the current plan, not in the global procedure.  - Acceptance testing is out of scope. |
| Peer Review | * Reviews for Risk Register, WBS, estimations, project schedule are done in meetings; there is no need for formal approval; the fact that the RFE SW Team Manager communicates the baselines of the roadmap implicitly represents his approval; * Reviews performed by project SQE for documentation like: User Manuals, Traceability Matrix Report, Test Reports, Code Coverage Report, MISRA report, Release Notes may be documented in the Release Readiness Review report. |

Table 26 Tailored Items Table

# Safety Plan Changes

Any event specified in the table below triggers changes to this document

|  |  |  |
| --- | --- | --- |
| **Trigger Type** | **Value** | **Action** |
| Change of process | Project specific or global process changes, significant enough to make the existing documents incompatible with the new ones | Impacted document(s) and SPMP (Safety Plan) to be updated |
| Changes in project organization, stakeholders, communication channels. | Any change | SPMP (Safety Plan) specific sections |

Table 27 Evolution of Safety Plan

1. Abbreviations and terminology

| Abbreviation | Description |
| --- | --- |
| AP | Automotive Processors |
| BCaM | Business Creation and Management; Product Creation Process in NXP |
| CCB | Change Control Board |
| CWE | Common Weakness Enumeration |
| EAR | Early Access Release |
| ES | Explicit Safety |
| FAE | Field Application Engineer |
| FS | Functional Safety |
| NXP | NXP Semiconductor |
| HW | Hardware |
| ISO | International Standards Organization |
| MISRA | Motor Industry Software Reliability Association |
| N/A | Not Applicable |
| PM | Project Manager |
| PRC | Product Release Candidate; also known as BETA |
| PRD | Product Requirements Document |
| R&D | Research & Development |
| QAE | Quality Assurance Engineer |
| RRR | Release Readiness Review |
| RFP, RTM | Release for Production; also known as Release To Market (RTM) |
| SC, DC, MC/DC | Statement Coverage, Decision Coverage, Modified Condition/Decision Coverage |
| SCMP | Software Configuration Management Plan |
| SOW | Statement of Work |
| SPMP | Software Project Management Plan |
| SQAP | Software Quality Assurance Plan |
| SW | Software |
| TBD | To Be Defined |
| UML | Unified Modeling Language |
| WBS | Work Breakdown Structure |

1. Referenced documents

Table 28 References Table – Process Artifacts contains references to process documentation like global procedures, generic plans, guidelines.

Table 29 References Table – RFE SW Project Documents contains references to RFE SW project specific documents.

| **Process Artifacts – Title** | **Document Location** |
| --- | --- |
| 1. AP SW Quality Process | <https://nxp1.sharepoint.com/:f:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL?csf=1&e=6WpjLG> |
| 1. BU Auto Roles Description Procedure | <https://nxp1.sharepoint.com/sites/OMS/_layouts/15/DocIdRedir.aspx?ID=NXPOMS-999116894-2733> |
| 1. Automotive Functional Safety Asesment Procesure | <https://nxp1.sharepoint.com/:w:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Safety%20Management/BCAM%203.3%20Confirmation%20Review%20Templates/Automotive%20Functional%20Safety%20Assessment%20Procedure%E2%80%93BCaM7.docx?d=w8c534aad4ebc46fdb9999299b2fcffed&csf=1&web=1&e=XAA9dL> |
| 1. AP Safety Management Procedure | https://nxp1.sharepoint.com/sites/OMS/\_layouts/15/DocIdRedir.aspx?ID=NXPOMS-1719007347-3873 |
| 1. BU Auto Risk Management Procedure | https://nxp1.sharepoint.com/sites/ampsoftware/Shared%20Documents/Forms/AllItems.aspx?csf=1&e=6WpjLG&cid=976ce917%2Dbcd7%2D4182%2Da82f%2D3dd395fc4580&FolderCTID=0x012000E3E903685BFC06428116C9F69BE1633E&viewid=c92d7639%2D95d3%2D4c37%2D96dc%2D0bfdc419fb02&id=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAP%20Software%20%2D%20Risk%20Management |
| 1. AP SW Roles Description | <https://nxp1.sharepoint.com/:x:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/BL%20AP%20SW%20-%20%20Roles%20Description.xlsx?d=we58d24ddadb24312b9d03dddce2571d1&csf=1&web=1&e=9e0h9E> |
| 1. BL AP Software Development Generic Plan | [https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20%20-%20Development/BL%20AP%20Software%20Generic%20Development%20Plan.pdf?csf=1&web=1&e=LmihUC](https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20%20-%20Development/BL%20AP%20Software%20Generic%20Development%20Plan.pdf?csf=1&web=1&e=LmihUC%20) |
| 1. BL AP Software Generic Plan (Test Strategy) | <https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Testing/BL%20AP%20Software%20Generic%20Test%20Plan%20(Test%20Strategy).pdf?csf=1&web=1&e=DeaM8t> |
| 1. BL AP Software Generic Release Management Plan | <https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Release%20Management/BL%20AP%20Software%20Generic%20Release%20Management%20Plan.pdf?csf=1&web=1&e=TQifxb> |
| 1. AP SW Generic Safety Management Plan | https://nxp1.sharepoint.com/:u:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Safety%20Management/BL%20AP%20SW%20Safety%20Plan%20Template.url?csf=1&web=1&e=NYsc7w |
| 1. AP Software Generic Quality Assurance Plan | <https://nxp1.sharepoint.com/sites/ampsoftware/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAP%20Software%20%2D%20Quality%20Assurance%2FBL%20AP%20Software%20Generic%20Quality%20Assurance%20Plan%2Epdf&parent=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAP%20Software%20%2D%20Quality%20Assurance> |
| 1. AP SW Generic Requirement Engineering Plan | https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Requirements%20Engineering/BL%20AP%20Software%20Generic%20Requirements%20Engineering%20Plan.pdf?csf=1&web=1&e=sMq1M8 |
| 1. AP Software Generic Program and Project Management Plan | <https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Program%20and%20Project%20Management/BL%20AP%20Software%20Generic%20Project%20and%20Program%20Management%20Plan.pdf?csf=1&web=1&e=y7K7MV> |
| 1. AP Software Review Guideline | <https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Change%20Configuration%20Management%20and%20Reviews/AP%20SW%20Review%20Guideline.pdf?csf=1&web=1&e=Zujjc0> |
| 1. AP Software JIRA Guideline | <https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Change%20Configuration%20Management%20and%20Reviews/AP%20SW%20JIRA%20Guideline.pdf?csf=1&web=1&e=GOVCwQ> |
| 1. AP Software Confirmation Reviews Templates | <https://nxp1.sharepoint.com/sites/ampsoftware/Shared%20Documents/Forms/AllItems.aspx?csf=1&e=6WpjLG&cid=c9ad7542%2Ddefc%2D4e9a%2Db650%2D1966a14533e9&FolderCTID=0x012000E3E903685BFC06428116C9F69BE1633E&viewid=c92d7639%2D95d3%2D4c37%2D96dc%2D0bfdc419fb02&id=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAP%20Software%20%2D%20Safety%20Management> |
| 1. RFE SW Assessment plan | https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Project\_Management/Safety\_Assessment\_Plan\_STRX\_RFE\_SW.xlsx |
| 1. AP SW Safety Concept Template | <https://nxp1.sharepoint.com/:w:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/TEMPLATES_AP_SW_4.3_BCaM_CTO_3.4/BL%20AP%20SW%20-%20Safety_Concept_Template.docx?d=w3aa62048184849f9ba8a64d24ea1e273&csf=1&web=1&e=LK3Myq> |
| 1. AP SW Review Guideline | <https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Change%20Configuration%20Management%20and%20Reviews/AP%20SW%20Review%20Guideline.pdf?csf=1&web=1&e=vt0OhK> |
| 1. DFMEA Template | https://nxp1.sharepoint.com/:x:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Risk%20Management/Auto%20template%20FMEA.xlsx?d=w6bd3f426f9e1480880d12780e6669c40&csf=1&web=1&e=kCyGWI |
| 1. Safety Manual Template | <https://nxp1.sharepoint.com/:w:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/TEMPLATES_AP_SW_4.3_BCaM_CTO_3.4/Auto%20BCaM7%20SW%20Safety%20Manual.docx?d=wfa1bb847581c4e41abc186676e36c0cf&csf=1&web=1&e=SyCbjV> |
| 1. BL AP – SW Tool Evaluation Qualification template | https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Functional\_Safety/Tool\_Evaluation\_Qualification |

Table 28 References Table – Process Artifacts

| **Project Document – Title** | **Location** |
| --- | --- |
| 1. BCaM7 Gate Reports | [Quality\_Assurance](https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Quality_Assurance) |
| 1. HW Reference Manual | <https://www.collabnet.nxp.com/sf/go/doc364199> |
| 1. RFE SW CCB dashboard | [STRX JIRA](https://jira.sw.nxp.com/secure/RapidBoard.jspa?rapidView=6897&projectKey=STRX) (no dedicated RFE SW CCB before PPA |
| 1. RFE SW Requirements Management Plan | <https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Requirements_Management/RFE_SW_Requirement_Specification.xlsx> |
| 1. RFE SW Software Configuration Management Plan | <https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Change_and_Configuration_Management/RFE_Configuration_Management_Plan.docx> |
| 1. RFE SW Software Quality Assurance Plan | <https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Quality_Assurance/RFE_SW_Quality_Assurance_Plan.docx> |
| 1. RFE SW Design Guideline | [RFE SW Design Guideline](https://nxp1.sharepoint.com/:b:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AMP%20Software%20%20-%20Development/Best%20Practices/Best%20Practice_Design_Guideline_SMCAL.pdf?csf=1&web=1&e=lG7OnS) |
| 1. RFE SW Detailed Design UML template | [RFE SW Detailed Design UML template](https://nxp1.sharepoint.com/sites/ampsoftware/Shared%20Documents/Forms/AllItems.aspx?originalPath=aHR0cHM6Ly9ueHAxLnNoYXJlcG9pbnQuY29tLzpmOi9zL2FtcHNvZnR3YXJlL0VnbzFIVWpiTkJwTXZqYkpNVndjN1VjQk9ETEtnSmN6cWdRYnJYUkVBTXFEaUE%5FcnRpbWU9bW00RktFbi0xMGc&id=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAMP%20Software%20%20%2D%20Development%2FBest%20Practices%2FBest%20Practice%5FUML%5FDesign%5FMCAL%5FASR%5F40%2Eeap&parent=%2Fsites%2Fampsoftware%2FShared%20Documents%2FAMP%20SW%20Process%20%2D%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL%2FAMP%20Software%20%20%2D%20Development%2FBest%20Practices) |
| 1. RFE SW Design review checklist | [RFE SW Design review checklist](https://nxp1.sharepoint.com/:x:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AMP%20Software%20%20-%20Development/Best%20Practices/Checklist_for_Design_Review_(ClassA)_November_2016.xlsx?d=wd9fb81fcd1f94f30b3a2963da16bec19&csf=1&web=1&e=H86hM3) |
| 1. RFE SW Coding Guideline. | [RFE SW Coding Standard](https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Coding_Standard) |
| 1. RFE SW Code review Checklist | [RFE SW Code review Checklist](https://nxp1.sharepoint.com/:x:/r/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AMP%20Software%20%20-%20Development/Best%20Practices/Best%20Practice_Code_Review_Checklist_Autosar_OS_V1.0.xls?d=wa8aa18cc8dd94ffbadbfc2561766d081&csf=1&web=1&e=gbS4JK) |
| 1. RFE SW Test Strategy | <https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Software_Testing/Docs/RFE_STRX_TS.docx> |
| 1. RFE SW Configuration Items Baseline Plan | https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe\_sw\_process/browse/Change\_and\_Configuration\_Management |
| 1. RFE SW dependencies | https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe\_sw\_process/browse/Project\_Management |
| 1. RFE SW Master Schedule (Gantt chart) | https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe\_sw\_process/browse/Project\_Management |
| 1. RFE SW Product Roadmap | [Project\_Management](https://bitbucket.sw.nxp.com/projects/STRX/repos/rfe/browse/docs/Project_Management) |
| 1. RFE SW Tools Evaluation and Qualification Reports | [RFE SW Tools Evaluation and Qualification Reports](https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Tool_Evaluation_Qualification/RFE_Tool_Eval_Qualificaion.xlsx?at=refs%2Fheads%2Ffeature%2FSTRX-491-strx-rfe-fs-rfe-tool-qualification-template-formation-and-update) |
| 1. Team Competencies and Training Plans | <https://www.collabnet.nxp.com/sf/go/doc406698> |
| 1. SmartTRX software test plan | <https://www.collabnet.nxp.com/sf/docman/do/viewDocument/projects.smarttrx/docman.root.es0.105_verification/doc405691?_message=1598336262870> |
| 1. RFE SW Architecture Specification | <https://bitbucket.sw.nxp.com/projects/STRX/repos/strx/browse/rfe/docs/Software_Architecture/RFE_SW_Architectural_Specification.docx> |
| 1. AP and EP Software Generic Waiver and Tailoring Plan | <https://nxp1.sharepoint.com/sites/ampsoftware/Shared%20Documents/AMP%20SW%20Process%20-%20BU%20BCaM7%20Process%20Adoption%20in%20AMP%20BL/AP%20Software%20-%20Process%20Governance%20and%20Tailoring/BL%20AP%20Software%20Generic%20Waiver%20and%20Tailoring%20Plan.pdf?CT=1604490389963&OR=ItemsView> |

Table 29 References Table – RFE SW Project Documents

# Revision History

| **Document Author** | **Date** | **Description of Change** | **Document Owner** |
| --- | --- | --- | --- |
| Grzegorz Kulik | 20.06.2020 | Initial version created | Grzegorz Kulik |
| Mathew P Joseph | 08.09.2020 | Updated Section 5.1 ISO 26262 Tailoring | Grzegorz Kulik |
| Mathew P Joseph | 11.11.2020 | Updated document with review comments resolution | Grzegorz Kulik |
| Mathew P Joseph | 3.12.2020 | Updated references | Maulik Prabhudesai |
| Maulik  Prabhudesai | 10.12.2020 | Updated Project Management sections | Maulik Prabhudesai |
| Mathew P Joseph | 15.12.2020 | Fixed the assessor review comments | Maulik Prabhudesai |
| Sandeep B B | 30.07.2021 | Fixed assessor review comments | Maulik Prabhudesai |
| Maulik Prabhudesau | 16.06.2022 | Fixed assessor review comments | Maulik Prabhudesai |
| Samrat Guha Niyogi | 13.03.2023 | Updated Section 4.1 and 4.2 and Fixed assessor review comments | Samrat Guha Niyogi |
| Rashmi K C | 21.03.2023 | Fixed assessor review comments Sec 5.3.6.2 | Rashmi K C |
| Rashmi K C | 04.04.2023 | Fixed assessor review comments | Rashmi K C |
| Leena Mukherjee | 23/05/23 | Added my name as FSM for Shark project | Leena M |
| In case of questions or change proposals please contact the latest document author or owner. | | | |