

VALIDATION PLAN

**Failsafe Network Multiplexer (FNMux)**

**Document ID:TE/FNMux/VaP**

**Version 1.0**

Approval History

|  |  |  |  |
| --- | --- | --- | --- |
|  | Prepared By | Reviewed By | Approved By |
| Name |  |  |  |
| Signature |  |  |  |
| Date |  |  |  |

Copyright Notice © All rights reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from Team Engineers.

All copyright, confidential information, patents, design rights and all other intellectual property rights of whatsoever nature contained herein are and shall remain the sole and exclusive property of Team Engineers. The information furnished herein is believed to be accurate and reliable.

However, Team Engineers assumes no responsibility for its use, or for any infringements of patents or other rights of third parties resulting from its use.

All other trademarks are the property of their respective owners.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version (x. y) | Date of Revision | Description of Change | Reason for Change |
| 1.0 | 11th Apr 2024 | Baseline version | First Release |
|  |  |  |  |

TABLE OF CONTENTS

[1. Preface 5](#_Toc163813123)

[1.1. Introduction 5](#_Toc163813124)

[1.2. Purpose 5](#_Toc163813125)

[1.3. Scope 6](#_Toc163813126)

[1.4. Acronyms and Abbreviations 7](#_Toc163813127)

[1.5. References 10](#_Toc163813128)

[2. System Description – FNMUX 12](#_Toc163813129)

[3. FNMux project Organization structure 13](#_Toc163813130)

[3.1. Organization Structure 13](#_Toc163813131)

[3.2. Roles and Responsibilities 13](#_Toc163813132)

[3.2.1. Validation & Safety Manager 13](#_Toc163813133)

[3.2.2. Software Tester 14](#_Toc163813134)

[3.2.3. Hardware Tester 15](#_Toc163813135)

[4. Environment 16](#_Toc163813136)

[4.1. Testing Environment 16](#_Toc163813137)

[4.1.1. Validation activities to test FNMux system 16](#_Toc163813138)

[4.1.2. Bug Tracking in each phase in testing 16](#_Toc163813139)

[4.1.3. Software changes for Impact Analysis 17](#_Toc163813140)

[4.2. Configuration Environment 17](#_Toc163813141)

[4.3. Manufacturing Environment 17](#_Toc163813142)

[4.4. Validation Procedure 17](#_Toc163813143)

[4.4.1. Entry Criteria 17](#_Toc163813144)

[4.4.2. Controls 17](#_Toc163813145)

[4.4.3. Validation Plan Review 17](#_Toc163813146)

[4.4.4. Validation Plan Approval 18](#_Toc163813147)

[4.4.5. Exit Criteria 18](#_Toc163813148)

[4.4.6. Process Improvement on Creating/Maintaining Validation Plan Procedure 18](#_Toc163813149)

[4.4.7. Hardware Validation Activities 19](#_Toc163813150)

[4.4.8. System Validation Activities 19](#_Toc163813151)

[5. Hardware validation Tasks of FNMUX 20](#_Toc163813152)

[5.1. Design and Implementation Phase 20](#_Toc163813153)

[5.1.1. Hardware Validation 20](#_Toc163813154)

[5.1.2. Following are the deliverables of Validation of Design and Implementation Phase 23](#_Toc163813155)

[5.2. Software 23](#_Toc163813156)

[5.2.1. Software Validation 23](#_Toc163813157)

[5.2.2. Software Validation 24](#_Toc163813158)

[5.3. FNMux System Validation 27](#_Toc163813159)

[5.3.1. System Validation Activities 27](#_Toc163813160)

[5.3.2. Verification of System Validation phase output 28](#_Toc163813161)

[6. ANNEXURES 30](#_Toc163813162)

[6.1. Annexure – A Personnel Identified with Roles 30](#_Toc163813163)

List of Figures

[**Figure 1: System Overview** 12](#_Toc163813164)

[**Figure 2: Organization Structure** 13](#_Toc163813165)

List of Tables

[**Table 1: Acronyms and Abbreviations** 10](#_Toc163813246)

[**Table 2: References** 11](#_Toc163813247)

[**Table 3: Roles & Responsibilities of Validator** 14](#_Toc163813248)

[**Table 4: Roles & Responsibilities of Software Tester/Validator** 14](#_Toc163813249)

[**Table 5: Roles & Responsibilities of Hardware Tester/Validator** 15](#_Toc163813250)

[**Table 6: Personnel Identification with Roles** 30](#_Toc163813251)

# Preface

Fail-safe Network Multiplexer (FNMux) developed by Team Engineers (TE), is required to meet the Technical & Operational requirements of the RDSO specification “RDSO/SPN/11/2022” for transporting vital signaling information from interlocking to field using dual redundant OFC media in a fail-safe manner and driving the relays/end equipment in the field.

FNMux consists of the following functions

Exchange of vital signalling digital I/O information from interlocking to field using the dual redundant OFC

Driving the relays / end equipment in the field

For detailed explanation of each of the above functions and supporting functions refer RDSO Specification RDSO /SPN /211/2022, Effective Date: 24.11.2022 [Ref 1]

## Introduction

The FNMux system is proposed to be developed in compliance with CENELEC standards. FNMux System Verification and Validation plan shall be applicable to Life Cycle Phases starting from Phase 1-Concept to Phase 10-System Acceptance as shown in Figure 6 of the EN 50126-1:2017 [Ref. 2].

## Purpose

The purpose of Validation Plan (VaP) is to plan activities to examine and arrive at judgement based on the evidence that output items such as the processes, documentation, software, hardware, and application of each phase-specific development fulfils the specified requirements with respect to completeness, correctness, and consistency.

The Validation activity will also evaluate and establish that all phases of FNMUX lifecycle under the scope of this document fulfil the requirement of

1. Quality Management Condition
2. Safety Management Condition
3. Functional and Technical Safety
4. Fulfilment of the declared Tolerable Hazard Rate (THR).

The above activities will lead to providing a judgement whether the FNMux System Hardware and Software including its documentation fulfills the respective phase requirements to proceed to next phase.

Validation team will conduct the validation activity to provide supporting evidence that the FNMux is in-line with system requirements allocated to hardware and software. As part of Validation, in addition to analysis, the Validator shall witness the complete/part of the system testing/simulated testing to ensure that the system is in-line with Technical Specification for FNMux issued by RDSO. The deliverable at the end of validation is a Validation Report of FNMux hardware and software.

## Scope

The Scope of Validation Plan provides procedures and deliverable for each applicable phase of FNMux system life cycle for hardware and software as per the functional requirements as elaborated in FNMux Specifications [Ref 1]. The Design of FNMux System shall meet CENELEC requirements to achieve Safety Integration Level - 4 (SIL-4) as elaborated in FNMux Specifications [Ref 1].

The design part of the system life cycle viewed as “top-down” sequence of phases followed by integration, the validation part of the system life cycle viewed as “bottom-up” phases this is called as “V” representation of Design and Validation portion of safety life cycle.

The Software life cycle Validation conducted as per EN50128:2011+A1 2020 [Ref. 5]. The Software design and implementation Validation conducted as per Clause 5.9 of this document.

Hardware Validation is performed in parallel to the SW validation, and System Validation is conducted in system validation Phase after integration phase.

This plan is developed for Generic Application (class of application) for safety acceptance and approval process as per Figure 9 of EN 50129:2018[Ref. 6].

## Definition

|  |  |
| --- | --- |
| Terms | Definitions |
| Configuration Manager | Entity that is responsible for implementing and carrying out the processes for the configuration management of documents, software and related tools including change management |
| Component | A constituent part of software which has well-defined interfaces and behaviour with respect to the software architecture and design |
| Requirements Management | The process of eliciting, documenting, analyzing, prioritizing and agreeing on requirements and then controlling change and communicating to relevant stakeholders. It is a continuous process throughout a project |
| Safety-related Software | Software which performs safety-related functions |
| Software | Intellectual creation comprising the programs, procedures, rules, data and any associated documentation pertaining to the operation of a system |
| Software Baseline | Complete and consistent set of source code, executable files, configuration files, installation scripts and documentation that are needed for a software release. Information about compilers, operating systems, preexisting software and dependent tools is stored as part of the baseline. This will enable the organization to reproduce defined versions and be the input for future releases at enhancements or at upgrade in the maintenance phase |
| Software Deployment | Transferring, installing and activating a deliverable software baseline that has already been released and assessed |
| Software Life-cycle | Those activities occurring during a period of time that starts when software is conceived and ends when the software is no longer available for use. The software lifecycle typically includes a requirements phase, design phase, test phase, integration phase, deployment phase and a maintenance phase |
| Software Maintainability | Capability of the software to be modified; to correct faults, improve performance or other attributes, or adapt it to a different environment |
| Software Maintenance | Action, or set of actions, carried out on software after Deployment with the aim of enhancing or correcting its functionality |
| Software Safety Integrity | Level classification number which determines the techniques and measures that have to be applied to software |
| System Safety Integrity Level | Classification number which indicates the required degree of confidence that an integrated system comprising hardware and software will meet its specified safety requirements |
| Verification | Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled |
| Verifier | Entity that is responsible for one or more verification activities |
| Validation | Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled |
| Validator | Entity that is responsible for the validation |
| Opensource Software | Source code available to the general public with relaxed or non-existent copyright restrictions |
| Tool class T1 | Generates no outputs which can directly or indirectly contribute to the executable code (including data) of the software |
| Tool class T2 | Supports the test or verification of the design or executable code, where errors in the tool can fail to reveal defects but cannot directly create errors in the executable software |
| Tool class T3 | Generates outputs which can directly or indirectly contribute to the executable code (including data) of the safety related system |

## 

## Acronyms and Abbreviations

|  |  |
| --- | --- |
| ABBREVIATIONS | DESCRIPTION |
| **ADAD** | Application Data Architecture and Design |
| **ADAVR** | Application Data/Algorithm Verification Report |
| **ADPP** | Application Data Preparation Plan |
| **ADRS** | Application Data Requirements Specification |
| **ADTR** | Application Data Test Report |
| **ADTS** | Application Data Test Specification |
| **APVR** | Application Preparation Verification Report |
| **CENELEC** | European Committee for Electrotechnical Standardization |
| **CM** | Commissioning Manual |
| **CMP** | Configuration Management Plan |
| **CPS** | Communication Protocol Specification |
| **CS** | Coding Standards and Guide lines |
| **CU** | Central Unit |
| **DP** | Document Plan |
| **EL** | Field Trial Records |
| **EN** | European Norm |
| **ESS** | Electronic Support System |
| **ESSR** | Electronic Support System Report |
| **FAT** | Factory Acceptance Test |
| **FMECA** | Failure Modes Effects and Criticality Analysis |
| **FNMUX** | Failsafe Network Multiplexer |
| **FRACAS** | Failure Reports and Corrective Actions |
| **FTA** | Fault Tree Analysis |
| **FTP** | Functional Test Procedure |
| **FTR-SL** | Functional Test Reports at System Level |
| **FTR-CL** | Functional Test Reports for Card Level |
| **FTS** | Fail Safety Test Specification and Report |
| **FU** | Field Unit |
| **GASC** | Generic Application Safety Test |
| **HAS** | Hardware Safety Analysis |
| **HDD** | Hardware Design Description |
| **HRS** | Hardware Requirements Specification |
| **HVR** | Hardware Validation Report |
| **HZA** | Hazard Analysis |
| **HZL** | Hazard Log |
| **IM** | Installation Manual |
| **ISO** | International Organization for Standardization |
| **MFPR** | Manufacturing Process Record |
| **MIP** | Manufacturing and Inspection Plan |
| **MM** | Maintenance Manual |
| **OSTR** | Overall Software Test Report |
| **OSTS** | Overall Software Test Specification |
| **PCCL** | Pre-Commissioning Check List |
| **PHA** | Preliminary Hazard Analysis |
| **PMP** | Project Management Plan |
| **QMS** | Quality Management System |
| **QP** | Quality Plan |
| **RAM** | Reliability Availability and Maintainability |
| **RAMA** | RAM Analysis |
| **RAMP** | RAM Plan |
| **RDSO** | Research Design and Standards Organization |
| **RNDV** | Release Note and Deployment Plan |
| **RNVP** | Release Note and Validation Plan |
| **SAD** | System Architecture Description |
| **SADVR** | Software Architecture and Design Verification Report |
| **SCAD** | Source Code of Application Date |
| **SCMP** | Software Configuration Management Plan |
| **SCR** | Software Change Record |
| **SCTR** | Software Component Test Report |
| **SDM** | Software Deployment Manual |
| **SDP** | Supplies Development Plan |
| **SDR** | Software Deployment Records |
| **SDVR** | Software Deployment Verification Report |
| **SHITR** | Software/Hardware Integration Test Report |
| **SHITS** | Software/Hardware Integration Test Specification |
| **SIL** | Safety integrity level |
| **SIS** | Software Interface Specification |
| **SITR** | Software Integration Test Report |
| **SITS** | Software Integration Test Specification |
| **SIVR** | Software Integration Verification Report |
| **SMP** | Software Maintenance Plan |
| **SMR** | Software Maintenance Records |
| **SMVR** | Software Maintenance Verification Report |
| **SPI** | Serial Peripheral Interface |
| **SPN** | Specification Number |
| **SQAP** | Software Quality Assurance Plan |
| **SQAPVR** | Software Quality Assurance Plan Verification Report |
| **SRS** | System Requirement’s Specification |
| **SRSVR** | System Requirement’s verification Report |
| **SSCD** | Software Source Code and Supporting Documentation |
| **SSCVR** | Software Source Code Verification Report |
| **SSP** | System Safety Plan |
| **SSVR** | System Requirement’s Specification |
| **STS** | System Test Specification |
| **STVR** | Software Tools Validation Report |
| **SVR** | Software Validation Report |
| **TTL** | Traceability Table |
| **UM** | User Manual |
| **VAP** | Validation Plan |
| **VP** | Verification Plan |

**Table 1: Acronyms and Abbreviations**

## References

The following are the reference documents for Verification and Validation of FNMux:

|  |  |  |
| --- | --- | --- |
| **Reference No.** | **Document Title** | **Document Description** |
|  | **RDSO /SPN /211/2022,**  **Date Effective:24.11.2022** | Specification for Failsafe Network Multiplexer (FNMux). |
|  | **EN 50126-1:2017**  **EN 50126-2:2017** | Railway Applications- Specifications and demonstration of Reliability, Availability, Maintainability & Safety. |
|  | **50128-2011+A1:2020** | Railway Applications-Communications, Signalling and processing systems-Software for Railway Control and Protection Systems. |
|  | **EN 50129:2018** | Railway Applications-Communications, Signalling and processing systems- Safety Related Electronics Systems for Signalling. |
|  | **EN50159:2010+A1:2020** | Railway Applications-Communications, Signalling and processing systems - Safety related communication in closed transmission systems. |
|  | **RDSO/SPN/144/2012** | Safety and reliability requirement of electronic signalling equipment. |
|  | **ISO 9001:2015** | Quality Management Systems – Requirements |
|  | TE/FNMux/VaP | Validation Plan |
|  | TE/FNMUX/SQAP | Software Quality Assurance Plan |
|  | TE/FNMUX/QP | Quality Plan |
|  | TE/FNMUX/SAD | System Architecture Description |
|  | TE/FNMUX/HDD | Hardware Design Description |
|  | TE/FNMUX/SyRS | System Requirements Specifications |
|  | TE/FNMUX/HwVR | Hardware Validation Report |
|  | TE/FNMUX/SwRS | Software Requirements Specification |
|  | TE/QSM/001 |  |

**Table 2: References**

# System Description – FNMUX

Fail safe Network Multiplexer system will consist of a distributedmultiplexer. modules, connected in a network, constituting a network of fail-safe multiplexer modules for exchange of vital signaling information among fail-safe multiplexer modules.

The system architecture shall allow the formation of a scalable centralized unit of modules (FNMux Central Unit -CU) to concentrate I/O from the distributed field modules (FNMux Field Unit -FU). Furthermore, the network protocol and addressing technique adopted shall be such that any pair of vital modules, either in the central unit or in the field unit can be virtually connected from any point to any point. The FNMux Central unit shall also be able to communicate with Data Logger

The main purpose of FNMux is to transfer vital signaling information from FU to CU and from CU-to-CU meeting SIL4

A picture containing text, screenshot, diagram, colorfulness

Description automatically generatedThe Figure 1 below gives the FNMux top level block diagram.

**Figure 1: System Overview**

# FNMux project Organization structure

## Organization Structure

******

**Figure 2: Organization Structure**

## Roles and Responsibilities

### Validation & Safety Manager

|  |  |
| --- | --- |
| Role | Validator |
| Responsibilities | - Shall transform the System, Hardware, and software requirements into Validation Requirements  - Shall apply appropriate validation principles and standards as per E.8 of EN50129[Ref 4] and B.7 of EN50128[Ref 3]  - Shall evaluate all applicable phases of life cycle to validate that the evidence is adequate to fulfil the FNMux Specification.  - Shall generate a Validation Report based on which Hardware and Software Release is done  - Shall ensure the FNMux system design is safe as per EN50126[Ref 2]  - Shall ensure that the techniques and methods selected for Software development of FNMux are adequate for the SIL-4. |
| Key Competencies | - Shall be competent in System/Hardware and Software Validation  - Shall have exposure to safety requirements  - Shall be competent in various validation methods and ability to choose the right method for the validation  - Shall be capable of deriving the types of validation tools, techniques, and methods from the given specification  - shall understand relevant parts of EN 50126[Ref2], EN 50128[Ref 4], EN 50129[Ref 5] and ISO 9001:2015  - Shall be graduate with min 5 years of experience in System Validation. |

**Table 3****: Roles & Responsibilities of Validator**

### Software Tester

|  |  |
| --- | --- |
| Role | Software Testing |
| Responsibilities | - Shall transform the System/Software requirements into Test Plan and Test specifications  - Shall test the system/Software as per the Test Plan, Test Specification  - Shall apply appropriate testing principles and standards  - Shall ensure traceability of requirements to Test Cases  - Shall develop and maintain Test documents and implement change control for the same |
| Key  Competencies | - Shall be competent in System/Software testing and implementation  - Shall have exposure to safety requirements  - Shall be competent in testing principles and test methods and ability to choose the right method for testing process as per SIL-4  - Shall have the analytical thinking ability and good observation skills  - shall understand relevant parts of EN 50128[Ref 3] and ISO 9001:2015.  - Shall be graduate with min 1 years of experience in Software testing |

**Table 4****: Roles & Responsibilities of Software Tester/Validator**

### Hardware Tester

|  |  |
| --- | --- |
| Role | Hardware Testing |
| Responsibilities | - Shall transform the System/Hardware requirements into Test Plan and Test specifications  - Shall implement as per the Test Plan, Test Specification  - Shall apply appropriate testing principles and standards  - Shall ensure traceability of system/hardware requirements to Test Cases  - Shall develop and maintain Test documents and implement change control for the same |
| Key  Competencies | - Shall be competent in System/Hardware testing and implementation  - Shall have exposure to safety requirements  - Shall be competent in testing principles and test methods and ability to choose the right method for testing process for SIL-4  - Shall have the analytical thinking ability and good observation skills  - shall understand relevant parts of EN 50126[Ref 2], EN 50129[Ref 5] and ISO 9001:2015  - Shall be graduate with min 1 years of experience in Hardware testing |

**Table 5****: Roles & Responsibilities of Hardware Tester/Validator**

# Environment

The Validation Process uses the FNMux development environment, Software Testing environment, Manufacturing Environment, Lab simulation set up for in-house trials, Software and Hardware Configuration environment to perform the Validation activities in each lifecycle phase of FNMux.

## Testing Environment

### Validation activities to test FNMux system

The testing environment is used for Validation activities to test the FNMux system with reference to the FNMUX Specification [Ref 01]. Following is the test environment used

1. FNMux unit
2. Simulators
3. Digital multi-meter
4. Digital Oscilloscope
5. AC- DC Power Supplies
6. Digital Temperature Measuring Unit

### Bug Tracking in each phase in testing

Bug tracking shall be maintained for each phase of testing to ensure closure of issues raised during testing cycles/Validation. As per SQAP [Ref. 4], the impact of corrective actions is analysed and each of the component/subsystem/system affected are addressed. Detailed Procedure for Software Change Requirement Management and configuration data are addressed as per Software Configuration Management Plan [Ref. 8]. For Hardware changes, based on the process defined in Quality System Manual (QSM) TE/QSM/001 [Ref. 18], change management is addressed through raising of Engineering Change Notices. Testing and re-verification is carried out on modified components/Module for hardware.

### Software changes for Impact Analysis

For Software changes, the Configuration Management Tool (Jira) is used to track and close the issues raised. Impact of the corrective actions is analysed based on impact analysis, relevant software modules are modified, and regression testing performed.

## Configuration Environment

The Configuration Environment is defined in the Configuration Management Plan [Ref. 53] under “Configuration Management Library”, which contains all the RAMS related documentation.

## Manufacturing Environment

The Validation activities in the Manufacturing Environment as stated in the Quality Plan TE/FNMUX/QP [Ref. 19]shall be progressed using Team Engineers QMS, ISO 9001:2015 Quality System Manual, Procedures, Work Instructions and Formats TE/QSM/001 [Ref. 18].

## Validation Procedure

### Entry Criteria

Entry criteria are defined as the elements and conditions necessary to be in place to begin a procedure. The entry criteria for a phase starts with the deliverables mentioned in the respective phase.

### Controls

Controls are defined as the data that regulates a process activity. Controls regulate the transformation of inputs into outputs.

### Validation Plan Review

The TE/FNMux/VaP shall be reviewed by all the project people identified to be part of this process. This allows project personnel to submit comments and feedback on the role, responsibility, resources, and activities mentioned in this document. This document shall be reviewed by the Validator for crosschecking of Validation activities before implementing the validation activity and before the final review of the document by the Safety Manager for approval. During the Project, this document shall be reviewed at least once in six months.

### Validation Plan Approval

On completion of required reviews and acceptance by the Validator, this document shall be approved and released by Head Technical/CEO.

### Exit Criteria

Exit criteria defined as elements and/or conditions necessary to be in place to complete a procedure. The exit criteria for this process activity are that this document is reviewed, changes incorporated, approved and placed under configuration control.

### Process Improvement on Creating/Maintaining Validation Plan Procedure

#### The following shall be ensured during validation.

Validator checks that verification process is complete as per Clause 6.3.4.9 of EN:50128:2020[Ref. 5 ]

1. The software validation report shall fully state the software baseline that has been validated.
2. At any stage of lifecycle validation result shall demonstrate that the system developed fulfils user requirement in the defined environment with respect to safety as per clause 5.3.11 of EN50129:2018 [Ref. 6]

#### A review on how closely project personnel follow this procedure to create and maintain a Project and the areas for improvement shall be monitored during reviews conducted for adherence to the process. Based on these reviews, requirement of any changes in this document shall be made and base lined to improve the process.

#### The checklist for each of the Phase Deliverable is given in Annexure B in this document.

### Hardware Validation Activities

Hardware validation activities are carried out in Design and Implementation phase for producing hardware validation report.

### System Validation Activities

System Validation Activities are performed as per Clause 5.10 of this document.

# Hardware validation Tasks of FNMUX

## Design and Implementation Phase

The System Validation is performed as per EN 50126-1:2017 [Ref. 2 ]clause no. 6.7 Validation, 5.3.11 safety validation. The intended design shall be validated with the techniques/measures provided in Table E.8 of the standard EN 50129:2018 [Ref. 6].

### Hardware Validation

#### Input documents for Design and Implementation phase

1. Hardware design Documents
2. BOM
3. Schematics
4. PCB Layouts
5. FMECA
6. FTA
7. MTBF and THR Calculations
8. Functional safety Test Report
9. Type Test Plan and Type Test Report
10. Manufacturing Plan
11. User Manual

#### Process for validation of Hardware during Design and Implementation phase

1. To check that verification process for the hardware during design and implementation phase is complete.
2. To check that the TE/FNMUX/SAD [Ref. 25] document description regarding architecture ensures the following
3. Safety related Modules are isolated from non-safety modules
4. Dual electronic structure based on composite fail safety with failsafe comparison is implemented.
5. Wherever composite fail safety is not implemented, reactive fail safety shall be implemented.
6. Evaluation of Justification provided for the architecture of the FNMUX for fulfilling the SIL-4 requirement.
7. To check that the design document TE/FNMUX/HDD [Ref. 29] has taken care of the following
8. Validation of design phase documentation to verify that the design document provides graphical description of module, description of interfaces, measures for EMI/EMC, vibration, temperature are addressed, procedure for modification provided, user manual developed and complies to the requirement of TE/FNMUX/SyRS [Ref. 14].
9. Manufacturing process and documentation are developed and are in line with the RAMS requirements.
10. Application-level documentation prepared considering the H/W layout specific installation site.
11. The design document should also take care of implementation of structure design that are traceable to the requirement specification, modularity of the sub-system/module are functionally independent.
12. Requirement of Plausibility check on each input data type is possible to be implemented in the design there by ensuring testability
13. Evaluation of design document for the protection adopted against sabotage to the hardware addressed and the method is adequate.
14. Method of detection and mitigation of single component failure are incompliance with the requirement of SIL-4 for discrete and digital components. No single component failure can lead to an unsafe status of safety function of FNMux.
15. Inter channel isolation is implemented and described in the design.
16. Time taken for detection and mitigation of single fault meets the SIL -4 requirements (evaluated from the frequency of periodic built in self-test (BIST)).
17. Test cases prepared for testing and verification of hardware are complete and traceable to TE/FNMUX/SyRS [Ref. 14] and the result of testing and verification assures correct functional operation of the systems fulfilling the requirement of FNMux Technical Specification
18. Validation that integration of hardware has been carried out as mentioned in the TE/FNMUX/HDD [Ref. 29].
19. Fail safety testing of the H/W demonstrates no single component failure leads to an unsafe failure of safety function.
20. Single component failure leads to retention of safe state.
21. Analysis of FTA document for compliance to SIL requirement to safety function of FNMux is evident.
22. Analysis of FMECA document regarding procedure adopted for development of FMECA and outcome of FMECA analysis for compliance to SIL-4 requirement is evident.
23. Validation of MTBF, THR, failure rate of components is carried out based on part stress method, procedure of MTBF, THR and failure rate are as per MIL Handbook 217F [Ref. 32]
24. Validation of Type test report to evaluate that the type test results are in compliance to FNMux Technical specifications [Ref. 1].
25. Validation of Manufacturing Plan for addressing the requirement of FRACAS and ESS test reports are in compliance to FNMux specification requirement.
26. Quality management system audit for generation of quality management report
27. Safety management audit carried out for generation of safety management report.
28. Evaluation of functional and technical safety based on type test report and functional test report for compliance to SIL-4

### Following are the deliverables of Validation of Design and Implementation Phase

1. Generic application safety case
2. Validation test report for test cases generated by Validator.
3. Hardware release note by Validator
4. Hardware validation report TE/FNMUX/HwVR [Ref. 33].

## Software

### Software Validation

The Software Validation is to ensure that the Software Life Cycle processes adopt the procedures as defined in EN50128:2020 [Ref. 5**]**, and the software complies with the requirements defined in Software Requirements Specification TE/FNMUX/SwRS[Ref. 37] with traceability to System Requirements Specification TE/FNMUX/SyRS[[Ref.14](#bookmark=id.upglbi)] and fulfills the requirement of Safety Integrity Level 4.

The Software Validation planning and activities are carried out as per clause 6.3-Software Validation of EN50128 [Ref. 5]. The project phases and activities are as defined in Software Quality Assurance Plan TE/FNMUX/SQAP [Ref. 4].

The Software Validation Plan specify all the criteria, techniques and tools used in the validation process. The Software validation Plan adopts techniques and measures as described in Software Quality Assurance Plan TE/FNMUX/SQAP [Ref. 4] chosen based on Table A.5, Table A.6, Table A.7 and Table A.8 of EN 50128[Ref. 5].

The objective of software validation is to demonstrate that the processes and their outputs are such that the software is of the defined software safety integrity level, fulfills the software requirements and is fit for its intended application. This activity is performed by the Validator (Cl 6.3 of EN 50128 [Ref. 5]).

The validation activities are performed to demonstrate by analysis and or testing that all the software requirements are specified, implemented, tested, and fulfilled as required by the SIL-4, and to evaluate the safety criticality of all anomalies and non-conformities based on the results of reviews, analysis, and tests.

Validator checks that verification process is complete.

At any stage of lifecycle validation result shall demonstrate that the FNMux system developed fulfillsFNMux requirements Specification [Ref. 1].

### Software Validation

#### Following are the input documents for this Phase.

* System Requirements Specification
* System Architecture Description
* System Safety Plan
* Software Quality Assurance Plan
* Configuration Management Plan
* Verification Report of SQAP
* Verification Report on SCMP
* Software Requirements Specification
* Overall Software Test Specification
* Verification Report of Software Requirements Specification
* Verification Report on Overall Software Test Specification
* Software Architecture & Design Specification
* Software Interface Specification
* Software/ Hardware Integration Test Specification
* Coding Standards
* Verification Report of Software Architecture and Design Specification, including Software Interface Specification.
* Verification Report on Software Hardware Integration Test Specification
* Verification Report on Coding Standards
* Software Component Design Specification
* Software Component Test Specification
* Verification Report of Software Component Design Specification
* Verification Report on Software Component Test Specification
* Source code and Supporting
* Software Component Test Report
* Verification Report of Source Code and Support Documentation
* Verification Report on Software Component Test Report including test coverage for each component.
* Software Hardware Integration Test Report, which includes Software Integration Test Report
* Verification Report of Software Hardware Integration Test Report.

#### Validation activities of this phase are as given below.

* Check the completeness of Verification activities as defined in each phase.
* The adequacy of the Software Requirements Specification, the Software Architecture and Design Specification and the Software Module/component Design Specification in fulfilling the safety requirements set out in the System Safety Requirements Specification established through walk through, analysis and traceability.
* Analysis of tools, hardware, or software, used for verification to establish suitability for demonstration to software requirement specification.
* Additional testing of the software if required by analysis and by simulation of input signals present during normal operation, through Simulator, recording anticipated occurrences and determining if any undesired conditions requiring system action.
* Identification of deficiencies, if any, in the software realized through simulation or analysis and the impact on the system on the use of such software with reference to Software Requirements Specification that complies system requirement specification (system safety requirement specification in particular) .
* Analysis and adequacy of Hazard log for implementation of mitigation measures and closure of all Risk
* Evaluation of data preparation plan and analysis of application data as per FNMux specification
* Evaluation of Test coverage and Identification of Additional Test Requirement from analysis of Overall Software Test Report with reference to Software Requirements Specification.
* Evaluation of Methods used for tracking and closure of deviations observed during the software development lifecycle of FNMux.
* All documents mentioned above shall be reviewed for readability, adequacy, consistency, and completeness.
* Validation of identity of documentation and configuration of software.
* Assessment of the adequacy of the methods, tools and techniques used within each phase of software development life cycle for SIL – 4 requirements as per Software Quality Assurance Plan and EN50128.
* Assessment of adequacy of the methods and techniques used within each phase of software development life cycle for SIL– 4 requirements as per EN50159[ Ref. 7].
* Assessment of adequacy of safety techniques used in design.
* Assessment of competency of all personnel undertaking tasks within this phase

#### Following are the deliverables of this Phase Validation Report.

* Updated Overall Software Test Report
* Software Validation Report
* Tools Validation Report.
* Release Notes for the software fulfilling SIL-4 requirement as per FNMux Specification [Ref. 1].

## FNMux System Validation

Hardware validation is performed in parallel to the Software validation. The process and deliverables for hardware and for software detailed in section 5 of this document.

### System Validation Activities

1. System Validation Activities are performed after integration and testing in the Lab and system validation report shall be generated for system acceptance.
2. The project manager shall ensure that the review comments by Validator if any are to be attended by design team and corrective actions to be completed for acceptance.
3. Details of System Validation activities are given as per Table E.8 – Verification and Validation of System and Product design of EN50129 [Ref. 6].

#### Input documents for System Validation phase

* System Requirements Specifications
* Verification and Validation Plan for System Validation Phase
* Generic Application Safety Case
* User Manual

#### Validation activities of System Validation phase

* Evaluation of Data, statistics used for verification of validation phase of FNMux system required for safety acceptance for compliance to System Requirement Specification.
* Validation, by analysis and test that the Integrated system meets RAMS requirements.
* Assessment of the adequacy and effectiveness of the operational data collection system.
* Assessment of the adequacy of the methods, tools and techniques used in System Validation phase.
* Assessment of competency of all personnel undertaking tasks within this phase

#### Outputs of System Validation Phase

* System validation report
* RAM analysis Report
* Recommendation for Safety Acceptance based on evidence of Generic Application safety case, system validation report which includes safety assessment report.
* Updated Generic Application Safety Case

### Verification of System Validation phase output

#### Input documents for System Validation phase

* System validation report
* Updated Generic Application Safety Case
* RAM analysis Report
* Recommendation for Safety Acceptance based on evidence of Generic Application safety case, system validation report which includes safety assessment report.

#### Verification activities of System Validation phase

* Assessment of the adequacy of the information, and where appropriate, data and other statistics, used as input to tasks within this phase.
* Verification, by analysis and test of FNMux meets RAMS requirements.
* Assessment of the adequacy of the methods, tools and techniques used in System Validation phase.
* Assessment of competency of all personnel undertaking tasks within this phase
* Verification of Generic Application Safety Case
* Verification of RAM analysis Report
* Review of Recommendation for Safety Acceptance based on evidence of Generic Application Safety Case

#### Outputs of System Validation Phase pertaining to Verification activities are given below. Outputs of Validation are detailed in.

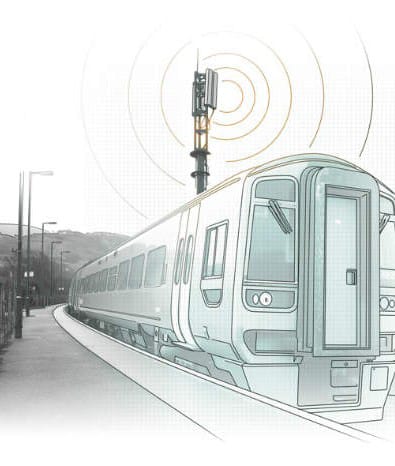
* Verification Report on RAM analysis Report
* Agreement to Recommendation for Safety Acceptance based on evidence of Generic Application Safety Case

# ANNEXURES

## Annexure – A Personnel Identified with Roles

|  |  |  |
| --- | --- | --- |
| S. No | Role | Name |
|  | Project in-charge |  |
|  | Project Manager |  |
|  | Validation In-charge |  |
|  | Software Development Manager |  |
|  | Hardware Manager |  |
|  | Verification Manager |  |
|  | Validator Engineer |  |
|  | Verification Engineer |  |
|  | CM In charge (SCM & HCM) |  |
|  | Hardware Engineer |  |
|  | Hardware Technician |  |
|  | Software Engineers |  |
|  | System integration Tester |  |

**Table 6****: Personnel Identification with Roles**

******

