|  |
| --- |
|  |
|  |
|  |
| Software development Planning Document for Personnel Safety Systems 0 |
|  |

|  | Name | **Role/Title** |
| --- | --- | --- |
| **Author** | Denis Paulic | Deputy Group Leader for Protection Systems Group, ICS |
| **Owner** | Stuart Birch | Senior Engineer – Personnel Safety Systems, ICS |
| **Reviewer** | Morteza Mansouri  Manuel Zaera-Sanz | Lead Integrator Engineer for Safety Critical Systems, ICS  Senior Engineer for Mission Critical PLC Based Interlock Systems, ICS |
| **Approver** | Annika Nordt | Group Leader for Protection Systems Group, ICS |

|  |  |
| --- | --- |
| Table of content | Page |

1. Introduction 4

1.1. Scope 4

1.2. Objectives 4

1.3. IEC 61511 5

2. PSS0 software development lifecycle 6

2.1. Application program safety requirements specification 8

2.1.1. Objectives 8

2.1.2. Inputs 8

2.1.3. Requirements 8

2.1.4. Activity steps 8

2.1.5. Methods 9

2.1.6. Verification activities 9

2.1.7. Tools 9

2.1.8. Outputs 9

2.1.9. Phase exit criteria 9

2.1.10. Roles 9

2.2. Application program development and design 9

2.2.1. Objectives 9

2.2.2. Entry criteria 9

2.2.3. Inputs 10

2.2.4. Requirements 10

2.2.5. Activity steps 10

2.2.6. Methods 10

2.2.7. Verification activities 11

2.2.8. Tools 11

2.2.9. Outputs 11

2.2.10. Phase exit criteria 11

2.2.11. Roles 11

2.3. Application program implementation 11

2.3.1. Objectives 11

2.3.2. Entry criteria 11

2.3.3. Inputs 11

2.3.4. Requirements 12

2.3.5. Activity steps 12

2.3.6. Methods 13

2.3.7. Verification activities 13

2.3.8. Tools 13

2.3.9. Outputs 13

2.3.10. Phase exit criteria 14

2.3.11. Roles 14

3. Abbreviations 15

4. References 16

Document Revision history 17

list of Figures

Figure 1: IEC 61511 V-model lifecycle for software development (Figure A.1 [3]) 5

Figure 2: PSS0 software development and testing lifecycle. 7

# Introduction

## Scope

The scope of this document is limited to the Personnel Safety System 0 (PSS0). It addresses the requirements of IEC 61511-1 [1] for software development.

## Objectives

The objectives of this document are the following:

* To describe the PSS0 software development strategy that shall follow the IEC 61511 lifecycle for application program (AP) development. This is covered in the left side of the V-model from Figure 1. The phases on the right side address the planning for software testing, verification and validation and that is covered in the verification and validation plan [2].
* To specify techniques and resources for developing the PSS0 software and extract high-level software planning requirements.
* To define documents to be created for each software development lifecycle phase.
* To address measures and good practices that shall be taken into account during software development.
* To define the responsible organisation and roles for each life-cycle phase.

This document shall be used as a support during the PSS0 software design and give guidelines on important aspects of IEC 61511 .

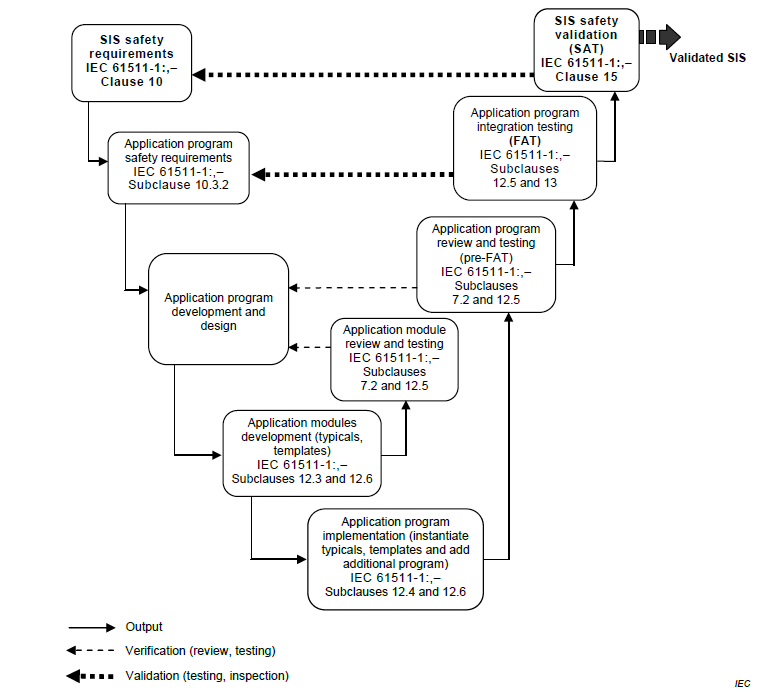


Figure 1: IEC 61511 V-model lifecycle for software development (Figure A.1 [3])

## IEC 61511

To use IEC 61511 the following needs to be fulfilled:

* All hardware components for implementing PSS0 safety functions [4] shall be certified/qualified commercial-of-the-shelf (COTS) components.
* The programming language for developing PSS0 software shall be a limited variability language (LVL) [5].
* The software development lifecycle shall follow the V-model lifecycle for AP development.

This document shall describe the objectives, requirements, inputs, outputs, required tasks, roles and documents for each phase of software development lifecycle.

# PSS0 software development lifecycle

Figure 2 shows a simplified flow-chart of the planned software development process for PSS0. The software development requirements shall be extracted from the documents listed below, where all procedures for system’s operation, safety analysis and specifications for chosen hardware components are documented.

* Overall SIL Assessment Report for Personnel Safety System 0 [6]
* Safety Requirements Specification Document for PSS0 (SRS) [4]
* Concept of Operations for Personnel Safety Systems 0 (ConOps) [7]
* Hardware Design Specifications for Personnel Safety System 0 (HWDS) [8]

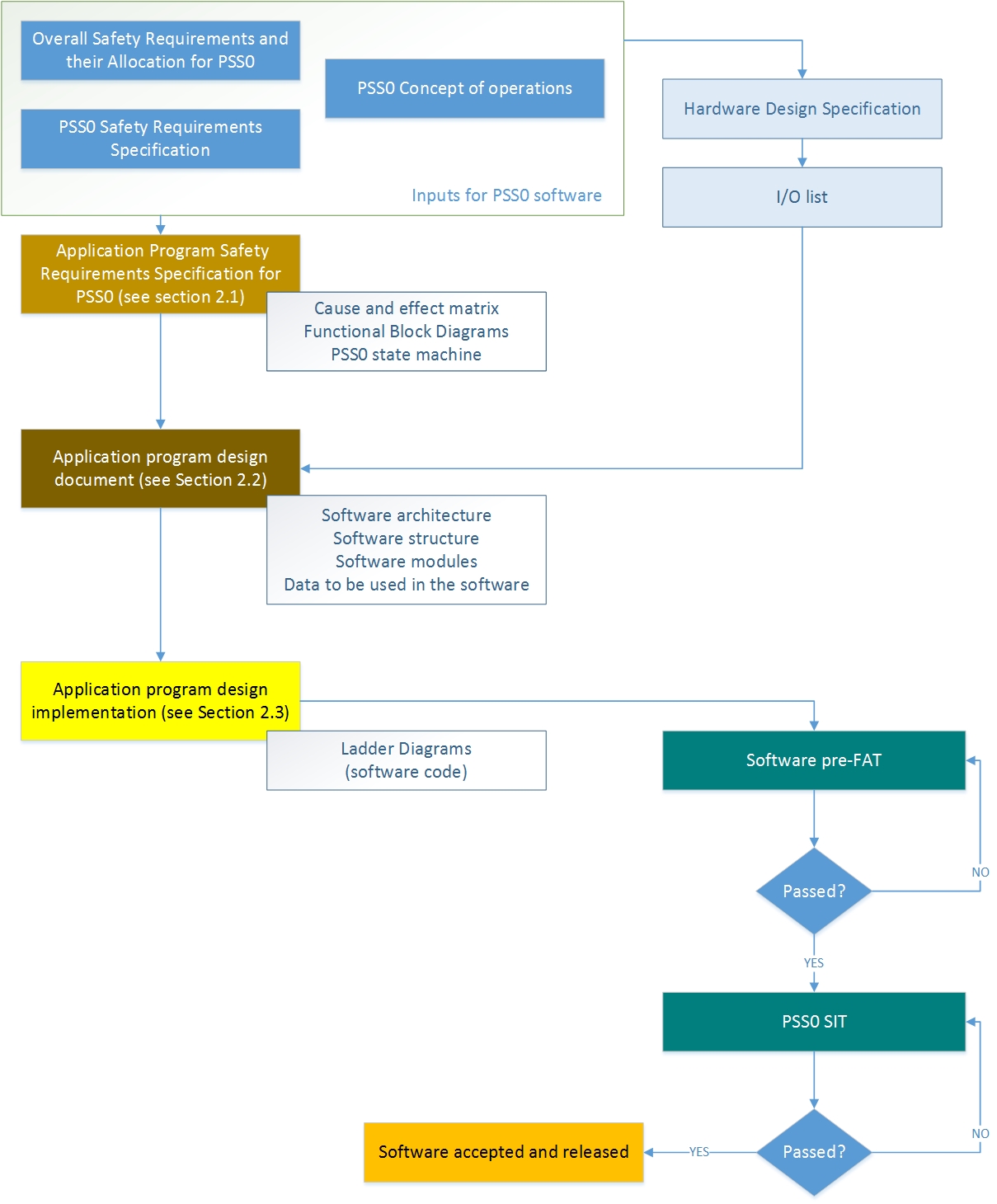


Figure 2: PSS0 software development and testing lifecycle.

## Application program safety requirements specification

### Objectives

The objectives of this document are:

* To specify the software safety requirements:
  + for each safety instrumented system (SIS) necessary to implement the required safety instrumented function (SIF);
  + for each safety instrumented function allocated to that SIS.
* To specify software performance requirements and design constraints.
* To specify the functional software requirements for the SIS and each SIF.

### Inputs

* SIS architecture defined in SIL Assessment Report for PSS0 [6]
* PSS0 SRS [4]
* PSS0 HWDS [8]

### Requirements

* The software safety requirements shall be specified for each programmable SIS device necessary to implement required SIF.
* It shall be sufficiently detailed.
* The requirements shall be clear, understandable, verifiable, modifiable and traceable back through all deliverables, including the design documents and analysis documents. [9]

### Activity steps

In order to compile the document, the following is necessary:

* Collect and review all software related requirements for each (programmable) SIS subsystem from the SRS [4], SIL Assessment Report [6] and HWDS [8] for PSS0:
  + Identify the concerned programmable SIS elements;
  + Identify safety requirements for each safety function, including sensors voting;
  + Identify the requirements resulting from the SIS architecture.
* Write the Software Safety Requirements Specification for PSS0 (SSSRS)
  + Describe all PSS0 modes of operation, including start-up, operation, shutdown and optionally the periodic testing of the safety functions.
  + Describe the behaviour of SIS elements.
  + Specify the safety requirements for each safety function during all modes of operation.
  + Specify the non-safety-related software requirements for implementing safety functions.
  + Specify the requirements on operator interfaces.

### Methods

* The state-machine diagram for describing PSS0 operational modes, with clear descriptions of each mode and transitions between the modes.
* The semi-formal method (cause and effect charts or logic diagrams) for documenting the safety functions.

### Verification activities

* The SSSRS shall be reviewed (visual inspection) by the software developer and at least one competent person not involved in original development.

### Tools

* Microsoft Office Tools (Microsoft Word, Excel and Visio).

### Outputs

* SSSRS

### Phase exit criteria

* Reviewed and approved SSSRS, stored in CHESS.

### Roles

* A software developer [10] from Protection Systems Group (PSG) shall deliver the SSSRS.
* A software verifier [10] from PSG that is not involved in original development shall review the SSSRS.

## Application program development and design

### Objectives

The objectives of this phase are:

* To create the PSS0 software architecture, which fulfils the specified software safety requirements.
* To review and evaluate the software requirements placed by PSS0 hardware architecture.
* To specify the procedures for PSS0 software development.
* To develop the PSS0 software design document [11].
* To identify a suitable set of configuration, library, management, and simulation and test tools, over the safety lifecycle for PSS0 software.

### Entry criteria

* Reviewed and approved SSSRS and HWDS for PSS0.

### Inputs

* PSS0 Software Safety Requirements Specification [4]
* PSS0 HWDS [8]
* Manuals of the SIS programmable devices
* Manufacturer guidelines on safety system development and programming

### Requirements

* The requirements for items to be included in software design [12] are listed below (to be applied as appropriate):
  + The software architecture with clearly separated areas of the software for non-safety and safety program (the safety software shall not depend on any non-safety variables to perform the safety functions).
  + The software structure (decomposition into modules) to show how the safety functions are implemented and how the safety integrity is achieved.
  + Specification of all identified software components, and the description of connections and interactions between identified components.
  + All SIS logic including all process operating modes for each safety function (including SIS start-up actions and reaction to power loss).
  + Procedures for modifications, versioning, back up and restoration of the software.
  + Detailed description of the functions and function blocks being used in safety program, information interchange with real I/O-s, timing constraints, memory allocation (process safety time shall be considered)
  + All non-safety parts of the software and interfaces between safety and non-safety parts of the software.
  + List of inputs, outputs, and associated data types.
  + Details of data exchanged between the software and operator interfaces and diagnostic data processing and logging.
  + Description of maintenance and operation interfaces.
  + Description of integration test requirements to be included in.
  + Functions related to periodic testing and maintenance.

### Activity steps

In order to compile the document, the following is necessary:

* The software developer shall review information in the SSSRS [13] and HWDS [8].
* The software developer shall create the PSS0 software design document [11], which should include the items listed in requirements.

### Methods

* Graphical models, showing the architectural design, the internal design of software modules and interactions between these modules.
* Tables that include the software components, inputs, outputs and data types.

### Verification activities

* Critical Design Review with design walkthrough.
* The PSS0 software design document shall be reviewed (visual inspection) by the SIS designers and at least one competent person not involved in the original development.

### Tools

* Microsoft Office Tools (Microsoft Word, Excel and Visio)
* Siemens Totally Integrated Automation (TIA) Portal

### Outputs

* PSS0 software design document [11], which includes:
  + Description of the software architecture and structure.
  + Description of software design.
  + Procedures and functions to be used during programming.
  + Integration test requirements.
* Information for PSS0 software verification.

### Phase exit criteria

* Reviewed and approved software design document, stored in CHESS.

### Roles

* A software developer [10] from PSG shall deliver the software design document.
* A software verifier [10] from PSG that is not involved in original development shall review the software design document.

## Application program implementation

### Objectives

The objectives of this phase are:

* To develop the software and its modules.
* To implement the software that fulfils the specified requirements.
* To use the appropriate procedures, support tools and programming languages during software implementation.

### Entry criteria

* Reviewed and approved PSS0 software design document, stored in CHESS.

### Inputs

* Description of the software design and architecture.
* Vendor manuals, guidelines and procedures for the selected PSS0 logic solver.

### Requirements

* Development methodology shall comply with the development tools and restrictions given by the manufacturer of PSS0 programmable electronics (recommended good practices in the manufacturer’s manuals and guidelines).
* The toolset for programming the PSS0 programmable electronics shall include the capabilities listed in [14]
* The following information shall be contained either directly in the software or in supporting documentation for software implementation [15]
  + The software originator;
  + Description of the purpose of the software (can be described in software design document [11]);
  + Version of safety manuals that were used;
  + Firmware versions of all programmable equipment;
  + Identification of SIFs and their SILs;
  + Identification of used symbols, logic conversions and library functions;
  + Identifications of input and output signals;
  + Description of information flows, if applicable;
  + Description of program structure;
  + Cyber security measures (if applicable);
  + Version identification and history of changes;
  + If previously developed software library functions are used, their compliance to IEC 61511.
* Software shall be produced in structured way, to achieve [15]):
  + Modular decomposition and functionality;
  + Keep the complexity on a minimum level;
  + Testability of functionality;
  + Traceability to software functions (and associated constraints);
  + Traceability to hardware architecture.
* The recommendations on security measures from the Siemens safety programming guideline [16] shall be followed as applicable.

### Activity steps

In order to implement the software, the following is necessary:

* The software developer shall review the software design document.
* The software developer shall configure all the modules and implement the ladder logic for PSS0 software and store required information in software comments. This typically includes the following major steps:
  + Configure the I/O modules and memory variable data areas.
  + Develop the tag names for all the I/O and memory variables.
  + If the override is required, develop the technique for maintenance override.
  + Define the communication variables to other systems. Variables that can be modified by other systems should be carefully defined and are typically placed in a special read/write area of memory.
  + Define where and how the sequence of events is recorded and understand its impact on the safety program.
  + Separate the safety and non-safety functions so that the emphasis can be placed on the safety critical programs.
  + Develop the software code.
  + Determine the proper execution order of the networks and logic.
* The software developer shall create the PSS0 software configuration document, which includes the list of firmware versions of all configurable equipment, all configured parameters for such equipment, at least one checksum for a safety program, time stamps when program was compiled and safety related settings that are enabled in the programming tool.
* The software developer shall prepare the PSS0 software pre-FAT document, which will consist of the verification checklists for testing the PSS0 software.
* The software developer shall generate the safety summary report provided by the programming tool. This report should be used during software verification.
* The software developer shall version the software that is ready for the pre-FAT, and store it on the access-protected private ESS repository.

### Methods

* Ladder Diagram or Functional Block Diagram for developing the PLC software.

### Verification activities

* Code inspections to ensure compliance with Siemens programming guidelines and software design document.
* Simulation using the Siemens TIA Portal PLC Sim and Watch tables.
* The PSS0 software shall be reviewed (visual inspection) by the competent person not involved in original development.
* Software pre-FAT verification.

### Tools

* Siemens TIA Portal.
* Test environment programmable hardware.
* Microsoft Office tools to create required documents.
* ESS GitLab[[1]](#footnote-1), Git[[2]](#footnote-2) repository with possibility to have the access-protected location to store the software.

### Outputs

* PSS0 software logic (ladder diagrams).
* PSS0 software pre-FAT checklist (simulation test checklists).
* PSS0 software configuration document.
* Information for verification activities (inputs for software FAT – integration tests).

### Phase exit criteria

* Software reviewed by a competent person not involved in original development, versioned and stored on the ESS GitLab.
* Reviewed and approved checklist to be used for PSS0 software pre-FAT, stored in CHESS
* Reviewed and approved PSS0 software configuration document, stored in CHESS.

### Roles

* A software developer [10] from PSG shall deliver:
  + the software for the project
  + a checklist to be used for PSS0 software pre-FAT
  + the PSS0 software configuration document
* A software verifier [10] from PSG that is not involved in original development shall review:
  + the software
  + the PSS0 software configuration document

# Abbreviations

AP Application Program

ConOps Concept of operations

COTS Commercial-Of-The-Shelf

ESS European Spallation Source

FAT Factory Acceptance Test

HWDS Hardware Design Specifications

IEC International Electrotechnical Commission

I/O Inputs and Outputs

ICS Integrated Control Systems

IT Information Technology

OSRTA Overall safety requirement and their test allocation

PLC Programmable Logic Controller

PSS Personnel Safety Systems

PSS0 Personnel Safety System 0

SIF Safety Instrumented Function

SIL Safety Integrity Level

SIS Safety Instrumented System

SRS Safety Requirement Specification

SSSRS Software Safety Requirements Specification

TIA Totally Integrated Automation

# References

|  |  |
| --- | --- |
| [1] | “Functional safety - Safety instrumented systems for the process industry sector (IEC 61511-1),” 2016. |
| [2] | “Verification and validation for PSS0 (ESS-0233615)”. |
| [3] | Functional safety - Safety instrumented systems for the process industry sector (IEC 61511-2), 2016. |
| [4] | “Safety Requirements Specification Document for PSS0 (ESS-0238059)”. |
| [5] | “IEC 61131: Programmable controllers,” 2013. |
| [6] | SIL Assessment Report for PSS0 (ESS-0231390), 2018. |
| [7] | “Concept of Operations For the Accelerator Personnel Safety System 0 (PSS0) (ESS-0134492)”. |
| [8] | Hardware Design Document for PSS0 (ESS-0237967), 2018. |
| [9] | Functional safety - Safety instrumented systems for the process industry sector (10.3 in IEC 61511-1), 2016. |
| [10] | 2018, Systems Engineering Management Plan for Personnel Safety Systems (ESS-0454273)\*. |
| [11] | “PSS0 Software Design Document (ESS-0364105),” [Online]. |
| [12] | Functional safety - Safety instrumented systems for the process industry sector (12.3 in IEC 61511-1), 2016. |
| [13] | Application program safety requirements specification for PSS0 (ESS-0407101), 2018. |
| [14] | *Functional safety - Safety instrumented systems for the process industry sector (clause E1 in IEC 61511-2), 2016..* |
| [15] | Functional safety - Safety instrumented systems for the process industry sector (12.4 in IEC 61511-1). |
| [16] | Siemens, “Safety Programming Guideline for SIMATIC S7-1200/1500,” 2017. |

Document Revision history

| Revision | Reason for and description of change | Author | Date |
| --- | --- | --- | --- |
| 1 | First issue  Implemented comments from functional safety assessment   * IEC 61508 is not mentioned anymore. * PSS0 modes moved to Software SRS. * Software versioning and backup (procedures) removed from this document. * Software pre-FAT removed and will be documented in V&V plan. * References to specific clauses of IEC61511 standard added. | Denis Paulic | 2018-08-23 2018-09-27 |
|  |  |  |  |

1. https://gitlab.esss.lu.se/ [↑](#footnote-ref-1)
2. Open source distributed version control system [↑](#footnote-ref-2)