SW Requirements Management Plan (SWRMP)

|  |  |  |
| --- | --- | --- |
| **OP’nSoft Project information** | | |
| **Project ID** | **Project Name** | **Project Manager** |
| [Type Project ID] | [Type Project Name] | [Type Project Manager] |
| **Field of application:** [Type field of application] | | |
| **Customer Name** | **Project Start Date** | **OP’nSoft project ID** |
| [Type Customer Name ] | [Type Project Start Date ] | [Type OP’nSoft project ID] |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Function | Signature |
| Edited by |  |  |  |
| Reviewed by |  |  |  |
| Reviewed by |  |  |  |
| Reviewed by |  |  |  |
| Approved by |  |  |  |

# Template Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Version | Author | Section | Description / Task ID |
| 26.09.2023 | 1.0 | Maximilian Nagl | All | Initial creation of template |
|  |  |  |  |  |

# Document Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date  **(yyyy-dd-mm)** | Version  (x.y) | Status | Author | Section | Description / Task ID |
|  |  | Choose an item. |  |  |  |
|  |  | Choose an item. |  |  |  |
|  |  | Choose an item. |  |  |  |
|  |  | Choose an item. |  |  |  |

# Table of Contents

[Template Revision History 2](#_Toc148023364)

[Document Revision History 2](#_Toc148023365)

[Table of Contents 3](#_Toc148023366)

[1 Introduction 4](#_Toc148023367)

[1.1 Purpose and Scope 4](#_Toc148023368)

[1.2 Review and Approval 4](#_Toc148023369)

[1.3 Referenced documents 4](#_Toc148023370)

[1.4 Applicable documents 5](#_Toc148023371)

[1.5 Abbreviations 5](#_Toc148023372)

[1.6 Goals & Metrics 5](#_Toc148023373)

[2 Definition of Software Requirements 6](#_Toc148023374)

[2.1 Attributes of Software Requirements 6](#_Toc148023375)

[2.2 Examples of good/bad Software Requirements 8](#_Toc148023376)

[3 Ensuring Traceability and Consistency 9](#_Toc148023377)

[4 Process Tailoring 10](#_Toc148023378)

# Introduction

## Purpose and Scope

**Purpose:** The SW Requirements Management Plan (SWRMP) describes the way how software requirements are handled throughout the software engineering process (SW Requirements analysis, SW Architecture design, SW detailed design, and unit construction). It gives guidelines on how software requirements will be documented on the given project and how traceability and consistency will be ensured.

**Scope:** This document is focused on the management of software requirements across the processes SWE.1 Requirements Analysis, SWE.2 Software Architecture Design and SWE.3 Software Detailed Design and Unit Construction. For each process there will be additional guidelines for other topics than software requirements. These guidelines are not included in this document.

## Review and Approval

The SW Requirements Management Plan is intended for the project team and should be modified/adapted according to the project needs.

After the initial creation or any update afterwards, the document needs to be approved by all the following parties below:

* Software Product Owner

## Referenced documents

<In the table below, list the all the technical documents, norms, standards, etc, that were used to define this document>

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Storage path and link** |
| ASPICE documentation | 3.1 | 26.09.2023 | [Confluence-link](https://confluence.engine.capgemini.com/x/oFsoAQ) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Applicable documents

<In the table below, list the all the project specific documents that are cited in this document or that were used to define this document>

|  |  |  |  |
| --- | --- | --- | --- |
| **Document name** | **Version** | **Date** | **Storage path and link** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Abbreviations

<In the table below, list and describe the abbreviations that are used in the document>

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| SWE | Software Engineering |
| SWRMP | SW Requirements Management Plan |
| UID | Unique Identifier |
|  |  |
|  |  |

## Goals & Metrics

<Add goals and metrics regarding software specification for this project>

|  |  |
| --- | --- |
| **Metrics** | **At Release** |
| Software requirements traceability | 100% |
| <Optional: Add additional metrics> |  |
|  |  |

# Definition of Software Requirements

## Attributes of Software Requirements

The Software Requirements are captured in the Software Requirements Specification Template from which they can be imported to Polarion via the live-doc function. In the following table you will find the attributes to specify for each requirement.

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Attribute description** |
| UID | Yes | Every requirement will have a Unique Identifier (UID) which ensures to avoid confusion or unintentional overwriting from other requirements.  When importing to Polarion the UID will be created automatically. In case Polarion is not available as ALM-tool for the project, the UID has to be filled in manually. |
| Description | Yes | The description is an explanation of the requirement. If possible, it is recommended to follow the standardized requirements pattern:  *<Subsystem> <shall> <have> <a: Optional> <PHYSICAL\_PROPERTY>*  *<OPERATOR> <Quantitative value>*  Example: The Power\_Supply shall have an Availability greater than or equal to 98%. |
| Rationale | No | Rationale states the reason for the requirements existence. It defines why the requirement is needed and other information to better understand the reason for and intent of the requirement. The rationale can also record and assumptions that were made when writing the requirement and the source of any numbers in the requirement. |
| Owner | Yes | This attribute specifies who is responsible that the requirement is defined and verified |
| Assigned to | (Without tool) No | The person who is currently working on the requirement (specification or verification). |
| Status | Yes | A requirement can have the following status:   * *Under construction* (During this state the requirement can be changed) * *Ready to Review* (Requirement is ready to be reviewed by authorized person) * *Reviewed* (Requirement has been accepted and is ready for implementation) * *Obsolete* (Requirement is correct but no need for implementation anymore) * *Rejected* (Requirement need rework) * *Implemented* (Requirement has been developed and integrated in the product) * *Validated* (Requirement has been tested and validated)   Illustration for the Lifecycle of a requirement: |
| Requirements type | Yes | There are two options for the requirement type:   * Functional Requirement * Non-functional requirement |
| Functional Safety | Yes | The ASIL is established by performing a risk analysis at system level. Software requirements which are derived from a functional safety critical system requirement take over the ASIL level (except if lowering of ASIL level can be justified). The following options for ASIL level are available:   * *QM* (Development supported by establish Quality Management is sufficient) * *A* (lowest level of ASIL, low risk reduction necessary) * *B* (second lowest level of ASIL) * *C* (second highest level of ASIL) * *D* (high level of ASIL, high risk reduction necessary) |
| Cybersecurity | Yes | This field indicates if the requirement has a relation to Cyber Security topics and if the Cyber Security process needs to be applied. |
| Target Release | Yes | This attribute indicates in which release it is planned to deliver this requirement to the customer/division. It should be aligned with the Release Plan. |
| Satisfies | Yes | In case of system development, every software requirement should be based on a systems requirement or a component of the system architecture which should be linked in this attribute. |
| Satisfied by | Yes (for release) | Link to the SW elements of Software Architecture Design. |
| Feature | No | This field indicates the high-level system function that this requirement will satisfy. |
| Verification criteria | Yes | The verification criteria define the qualitative and quantitative criteria for verification of a requirement. Verification criteria demonstrate that a requirement can be verified within agreed constraints. |
| Verification method | No | This field describes the proposed method to provide proof that the designed and build product meets the requirement. The following options are available:   * *Test* (verification of a product or system using a controlled and predefined series of inputs, data, or stimuli to ensure that the product or system will produce a very specific and predefined output as specified by the requirements) * *Demonstration* (manipulation of the product or system as it is intended to be used to verify that the results are as planned or expected) * *Inspection* (nondestructive examination of a product or system using one or more of the five senses (visual, auditory, olfactory, tactile, taste). It may include simple physical manipulation and measurements.) * *Analysis* (verification of a product or system using models, calculations and testing equipment. Analysis allows someone to make predictive statements about the typical performance of a product or system based on the confirmed test results of a sample set or by combining the outcome of individual tests to conclude something new about the product or system) |
| Verification approach | No | Specific approach to verify a requirement (depends on the verification method chosen above). |
| Tested by | (Without tool) No | Reference to the software qualification test which will test this requirement |
| Variant | No | Indicate if this requirement applies to all variants or only to specific variants. The list of variants needs to be defined specifically to the project. |
| Additional information | No | This field gives possibilities to add any further content to the requirements (comments, diagrams, models, etc.) |

## Examples of good/bad Software Requirements

### Example of good requirement

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Attribute value** |
| UID | Yes | REQ-SDT-03101 |
| Description | Yes | Req1 : The Heat controller will control the temperature inside the car between 16°C and 28°C. |
| Rationale | No | For the maximum passengers confort, we need a heat controller to control the temperature of the car |
| Owner | Yes | Sébastien LOISEL |
| Assigned to | No | Sébastien LOISEL |
| Status | Yes | Ready for Review |
| Requirements type | Yes | Functional |
| Functional Safety | Yes | QM |
| Cybersecurity | Yes | No |
| Target Release | Yes | Sample A |
| Satisfies | Yes | REQ-SDT-0021 |
| Satisfied by | Yes (for release) | <Add reference to SW element of Software Architecture Design > |
| Feature | No |  |
| Verification criteria | Yes | <Add verification criteria of requirement> |
| Verification method | No | Choose an item. |
| Verification approach | No |  |
| Tested by | (Without tool) No |  |
| Variant | No |  |
| Additional information | No |  |

### Example of bad requirement

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Mandatory** | **Attribute value** |
| UID | Yes | 000  [Requirements has no proper UID] |
| Description | Yes | For the maximum confort of the occupants, we need a heat controller to control the temperature of the car and the Heat controller is able to control the temperature between a maximum and a minimum using an hysteresis following this algortihm :  heatPointLow = Max°C  heatPointHeight = Min°C  heater = off  while(true){  if(temperature < heatPointLow)  heater = on  if(temperature > heatPointHeight)  heater = off  }  And the Heat Controller will be controlled by a panel with buttons  [This requirements actually consist of multiple requirements, it is not clear due since no explanation for ‘Min’ and ‘Max’, contains already elements of technical solution, contains its rationale in description, contains interface elements, etc.] |
| Rationale | No |  |
| Owner | Yes | Jeff  [The owner should be clearly designated with full name or company ID] |
| Assigned to | No |  |
| Status | Yes | In work  [This is not an official status as defined in requirements management plan, only use values from scroll-down bar] |
| Requirements type | Yes | Non-functional  [Wrong selection of requirement type, this is an example for a functional requirement] |
| Functional Safety | Yes | QM |
| Cybersecurity | Yes | No |
| Target Release | Yes | ASAP  [The target release of the requirement should be aligned with release plan. It can be a date, a sprint, a milestone but should be time specific] |
| Satisfies | Yes | Customer RFQ  [This field requires a reference to a specific system requirement or a system architecture element] |
| Satisfied by | Yes (for release) | User Story  [This field requires a reference to a specific User Story] |
| Feature | No |  |
| Verification criteria | Yes | Requirement to be tested by system team  [Every requirement should be tested internally at software level before delivering it to the division or customer] |
| Verification method | No | Choose an item. |
| Verification approach | No |  |
| Tested by | (Without tool) No |  |
| Variant | No |  |
| Additional information | No |  |

# Ensuring Traceability and Consistency

The Traceability of the requirements will be done automatically in Polarion. Further details will be provided one Polarion is in place at OPnSoft

# Process Tailoring

Tailoring of Requirements Management process should be documented in this chapter. The following tailoring has been defined:

* <Describe tailoring of process here>

The tailoring has been approved by:

* Software Quality Manager
* Software Product Owner
* Software Process Owner