Software Unit Verification – Component [insert component name] (SWDD)

|  |  |  |
| --- | --- | --- |
| **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 |
| 26/10/2023 | 0.1 |  |  |  |
|  |  |  |  |  |

# 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](#_Toc149232970)

[Document Revision History 2](#_Toc149232971)

[Table of Contents 3](#_Toc149232972)

[1 Introduction 5](#_Toc149232973)

[1.1 Purpose and Scope 5](#_Toc149232974)

[1.2 Review and Approval 5](#_Toc149232975)

[1.3 Referenced documents 5](#_Toc149232976)

[1.4 Applicable documents 6](#_Toc149232977)

[1.5 Glossary & Abbreviations 6](#_Toc149232978)

[2 Detailed design definition strategy 8](#_Toc149232979)

[2.1 Document adaptation to the project 8](#_Toc149232980)

[2.2 Naming convention 8](#_Toc149232981)

[2.3 Resource consumption criteria 8](#_Toc149232982)

[2.4 Detailed design evaluation 8](#_Toc149232983)

[2.4.1 Evaluation Criteria 8](#_Toc149232984)

[2.4.2 Detailed design evaluation method 9](#_Toc149232985)

[2.5 Detailed Design requirements 10](#_Toc149232986)

[3 Software architecture overview 11](#_Toc149232987)

[4 Component overview 12](#_Toc149232988)

[4.1 Description 12](#_Toc149232989)

[4.2 External interfaces 12](#_Toc149232990)

[5 Static view 13](#_Toc149232991)

[5.1 Illustration 13](#_Toc149232992)

[5.2 Units description 13](#_Toc149232993)

[5.3 Interfaces between units 14](#_Toc149232994)

[6 Component dynamic view 15](#_Toc149232995)

[6.1 States 15](#_Toc149232996)

[6.1.1 Illustration 15](#_Toc149232997)

[6.1.2 Description 15](#_Toc149232998)

[6.2 Use cases 16](#_Toc149232999)

[6.2.1 Illustrations 16](#_Toc149233000)

# Introduction

## Purpose and Scope

**Purpose:** The purpose of the Software Unit Verification (SWUV) is to describe the Software Unit Verification tests of the Software components defined in the Software Detailed Design (SWDD) of the project[Type Project Name]**.**

The Software Unit Verification contains project specific information and is part of the OP’nSoft Software Unit Verification construction process. Hence, this document cannot be seen independently because further information, valid for all projects, is not part of this plan, but part of the process.

**Scope**: This Software Unit Verification is valid for the project [Type Project Name]and affects the Software Development activities and their supplementary work products. It aims to:

* Describe the software units tests of the components

## Review and Approval

The Software Architectural Design is intended for all project members involved in the development and should be modified/adapted according to the project organization needs.

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

* Software Product Owner
* Software Development Engineer
* Software Quality Engineer

## 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** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

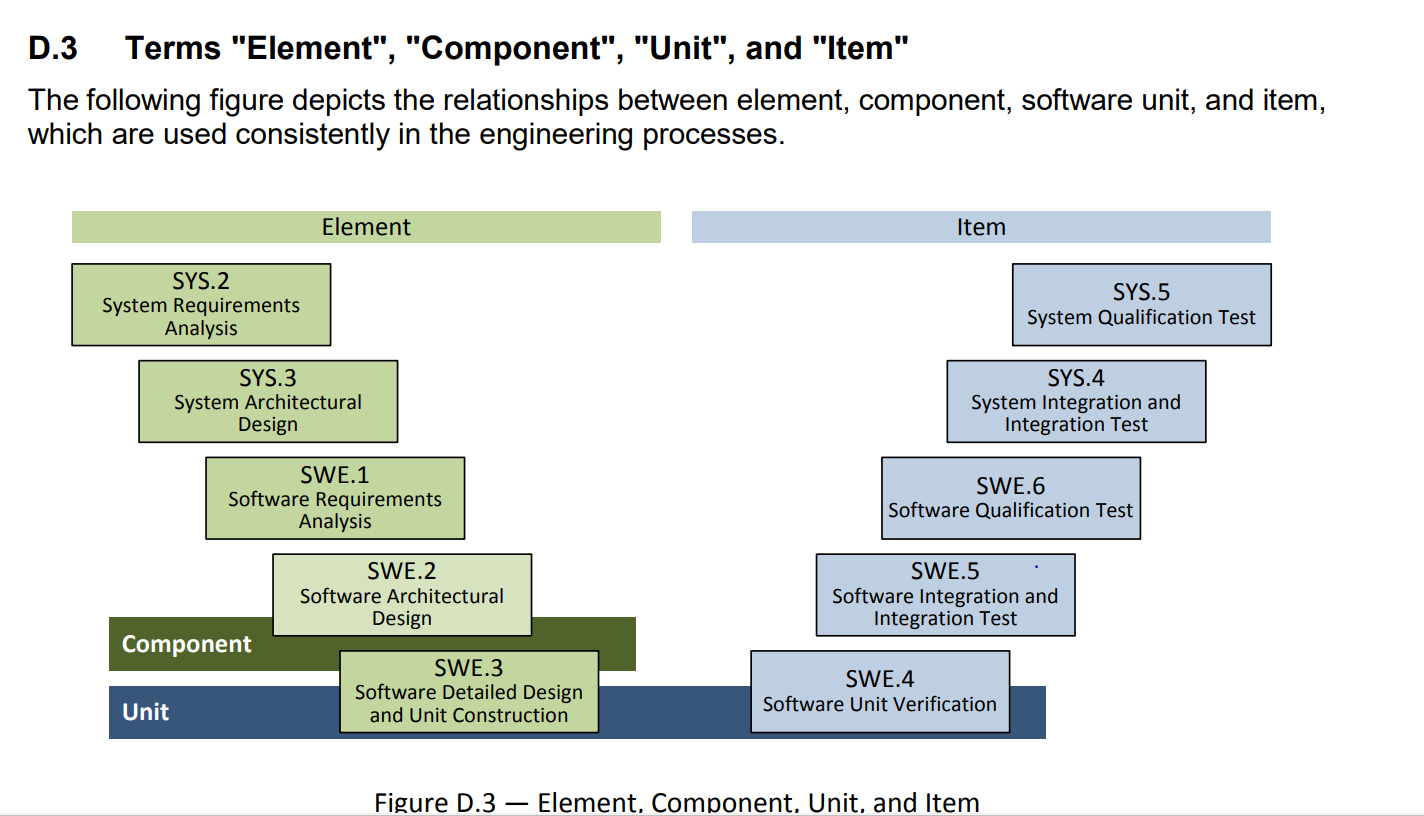
## Applicable documents

<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** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Glossary & Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Elements** | Elements are all structural objects on architectural and design level on the left side of the "V". Such elements can be further decomposed into more fine-grained sub-elements of the architecture or design across appropriate hierarchical levels |
| **System elements** | A member of a set of elements that constitutes a system. A system element is a discrete part of a system that can be implemented to fulfill specified requirements. A system element can be hardware, software, data, humans, processes (e.g., processes for providing service to users), procedures (e.g., operator instructions), facilities, materials, and naturally occurring entities (e.g., water, organisms, minerals), or any combination. (ISO/IEC 15288:2015) |
| **Software Element** | Result of the decomposition of the architecture on software level: The software is decomposed into elements of the software architecture across appropriate hierarchical levels down to the software components (the lowest level elements of the software architecture). |
| **Software Component** | Software-Components (SW-C) are architectural elements that provide and/or require interfaces and are connected to each other through the Virtual Functional Bus to fulfill architectural responsibilities. The Software Component is the central structural element used when building a system at the VFB-level. A SW-C has well-defined ports, through which the component can interact with other Software Components.  A Software Component has a formal description defined by the Software Component template. Software Components can be abbreviated as SW-Cs.  A SW-C contains a Software Component Description (SW-CD) and the implementation.  In Automotive SPICE V3.1 the term "software component" is used for the lowest level elements of the software architecture for which finally the detailed design is defined. A software "component" consists of one or more software "units". |
| **Unit** | Part of a software component which is not further subdivided. |



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

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Units verification tests