Project „Safety Function for Temperature Monitoring System “

Safety Plan Software Development

Modification history:

Version	Datum	Änderung	Ersteller	Prüfer
0.1	21.02.2020	Creation of Safety Plan	SaS	DS
0.2	29.01.2021	Modification of template: Additional notes for tools	SaS	DS
0.3	01.05.2023	Safety Plan for „Safety Function for Temperature Monitoring System “	Kaushik	

Successive document: SRS

Content

1. Safety – Plan

1.1 Project description

The goal of this project is to develop a safety function for the "Temperature Monitoring System" according to the software development approach prescribed by IEC 61508.

1.2 Project organization

1.2.1 Team members (CKN GmbH)

Name	Function/Task(s)	Department	Expertise/Competency	Phone / Email
David Wallace	Product Owner	Mgmt	Strategic Planning	wallace@de.ckn.com
Michael Scott	Safety Manager	FUSE	Functional Safety Expert	mscott@de.ckn.com
Toby Flenderson	HR SPOC	HR	Communication Skills	hr@de.ckn.com
Jim Halpert	SW-Architect	Dev	Functional Safety Professional	jhapert@de.ckn.com
Dwight Schrute	SW-Developer	Dev	Functional Safety Engineer	dschrute@de.ckn.com
Kevin Malone	SW-Developer	Dev	Functional Safety Engineer	kmalone@de.ckn.com

If further personnel are added to the list during project execution, this must be documented in the "modification history" of the document.

1.2.2 Participating external companies

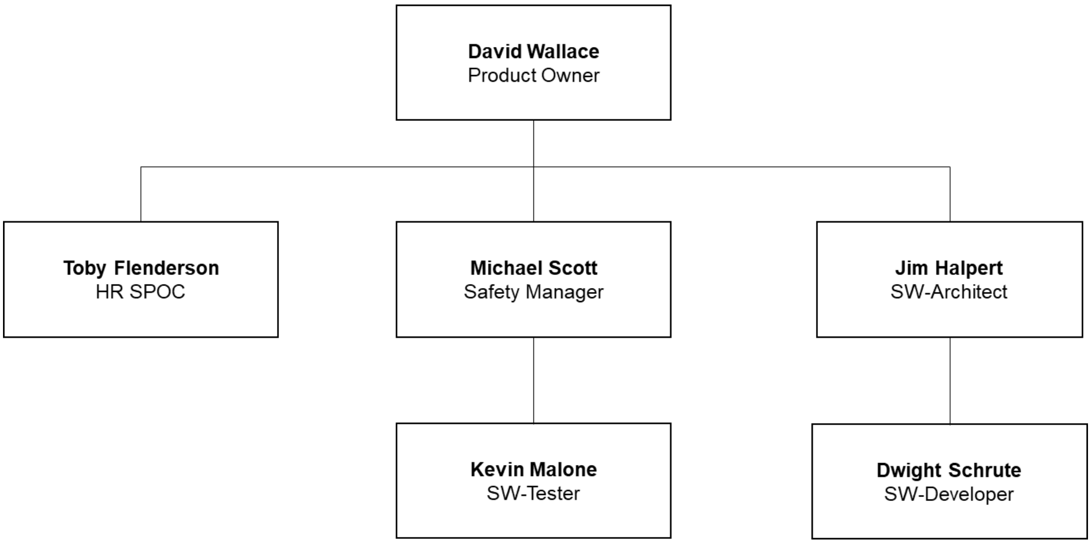
Unicorn Testing GmbH
DarrylStraße 10
Bonn 65543

"External company assigned to test the safety function as per IEC 61508"

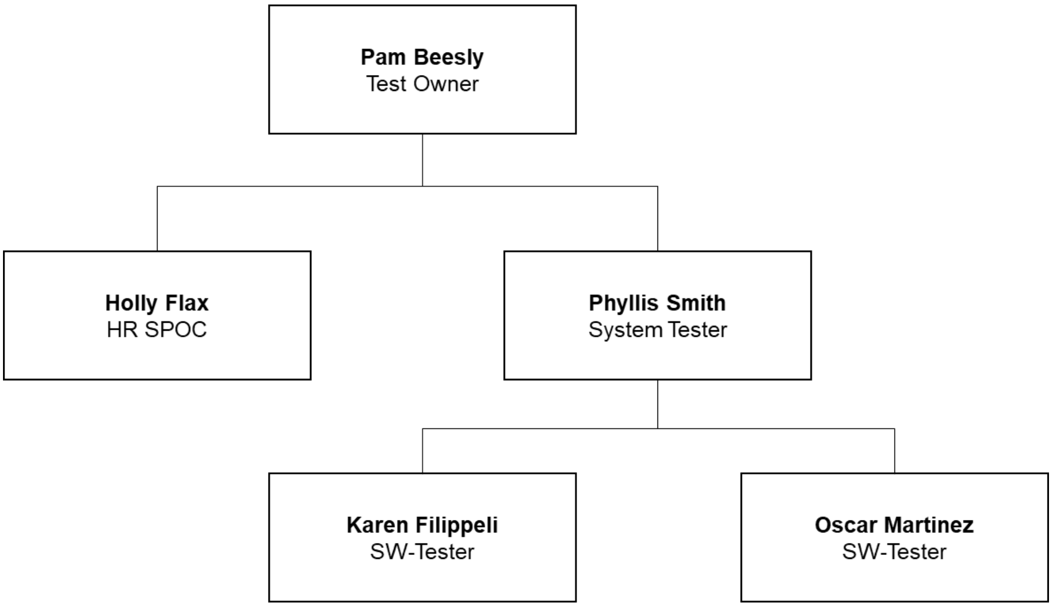
Name	Function/Task	Company/ Department	Expertise, Competency	Phone / Email
Pam Beesly	Test Owner	Mgmt	Test Manager	pbeesly@de.unicorn.com
Holly Flax	HR SPOC	HR	Communication Skills	hflax@de.unicorn.com
Phyllis Smith	System Tester	Test	Functional Safety Tester	psmith@de.unicorn.com
Karen Filippeli	SW-Tester	Test	Certified Tester - Foundation	kfilippeli@de.unicorn.com
Oscar Martinez	SW-Tester	Test	Certified Tester - Foundation	omartinez@de.unicorn.com

1.2.3 Company organigram (own and external responsibilities)

1.2.3.1 CKN GmbH



1.2.3.2 Unicorn Testing GmbH



1.3 Communication

The stakeholders involved in this project (i.e., CKN GmbH and Unicorn Testing GmbH) has agreed to follow the below mentioned communication protocols during the entire development phase of this project.

- Meetings
    - Kickoff meeting (10-05-2023)
    - Weekly sync meeting
        - Between Product Owner and Test Owner
        - Between CKN GmbH employees
        - Between Unicorn GmbH employees
    - Monthly sync meeting
        - Between CKN GmbH & Unicorn GmbH
    - On demand meeting
        - To discuss critical issues or other important topics
- Telephone conferences/video conferences
    - The above-mentioned meetings can be conducted either in office premises or virtually in the official MS Teams
- Storage location of meeting protocols
    - The Product Owner and the Test Owner shall be responsible for storing meeting protocols and circulate it to the respective employees
- Access control/Authorization for project data access
    - Project data (like SW and Test artifacts) are stored in the Git repository (Professional Version) to enable version control and parallel development activities
- Location/address of data server
    - Decided by the Product Owner and the Test Owner

1.4 Requirement Tracking

The necessary requirement tracking according to IEC 61508 shall be realized by using **MS Excel** tool.

1.5 Definition of software lifecycle phases

Lifecycle phases	**Responsible Person**	**Input**	**Executing Person**	**Output**	**Verifiying Person**	**Evaluation**
SRS	„Michael Scott“	Protocols of customer meetings […], Safety Plan […] etc.	„Michael Scott“	SRS […], Requirement tracking protocol […], validation plan […]	„David Wallace“	OK / not OK Evaluation protocol […]
Concept phase	Jim Halpert	Protocols of customer meetings […], Safety Plan […], SRS […] etc.	Dwight Schrute	SW – detailed specification […], SW – integration test plan […]	Michael Scott	…

SW system design	Jim Halpert	SW – detailed specification […]	Dwight Schrute	SW – system specification […], SW – system test plan […]	Jim Halpert	
SW module design	Dwight Schrute	SW – system specification […]	Kevin Malone	SW – module specification […], SW – module test plan […]	Dwight Schrute	
SW module testing	Kevin Malone	SW – modul specification […], SW – module test plan […]	Phyllis Smith	SW – module test protocol […]	Kevin Malone	
SW system testing	Phyllis Smith	SW – system specification […], SW – system test plan […]	Karen Filippeli	SW – system test protocol […]	Phyllis Smith	
SW integration testing	Karen Filippeli	SW – detailed specification […], SW – integration test plan […]	Oscar Martinez	SW – integration test protocol […]	Karen Filippeli	
Validation testing	Oscar Martinez	All documents	Phyllis Smith	Validation test protocol […]	Oscar Martinez	

All documents must be provided with a unique number and have to be recorded in the document list (chapter 3.0).

The executing person and the verifying person never must be the same person (“4-eyes principle”).

Above table shall be considered as “alive”, i.e. it shall finally completed in the last phase of the development lifecycle. But nevertheless it must be considered, that at all times the actual and succeeding lifecycle phase must be clearly defined.

1.6 Planned measures for fault avoidance

In order to fulfill the requirements of the functional safety standards with the aim of avoiding errors and (systematical) faults in every phase of the development lifecycle, appropriate and suitable measures for fault avoidance and for guaranteeing a high-quality standard have to be selected and must be efficiently applied with the necessary intensity.

The measures will be selected from IEC 61508-3 (functional safety standard for software development).

Detailed information, which measures and techniques have to be applied during the different development life cycle phases, will be specified within the verification and validation plan (see chapter 2).

1.7 Procedure in case of modifications

A request for modification may be initiated by the customer, sales department or internal/external personnel (developers, production etc.). The request may contain a fault description, desired additional features or requirements for improvement.

As a basis for decision, the project leader must report about the following topics:

- Impact on costs
- Impact on schedule
- Expected improvements
- Impact on design and development

Analysis:
A modification request will be examined regarding the impact on schedule, costs and technical quality.

Postponement:
A modification request with low priority must be postponed. Until the final decision, if the desired modification will be accepted or refused, the modification request must be discussed in every review meeting (in order to avoid, that open modification requests will not be considered at all).

Realization of modifications:
Modifications will be carried out by the corresponding technical department. By means of an influence or impact analysis it will be examined, which impact the modifications might have (regarding safety and technical issues).

Evaluation of the modifications:
As a result of the influence analysis testing and verification/validation measures have to be performed in order to proof, that all modifications were successful and were carried out as specified. Thus, it will be shown, that the product further fulfills all technical and safety requirements.

Requirement Tracking:
All modifications must be documented, i.e. the relevant documents must be updated accordingly. For this purpose, it is necessary, that all documents underly the overall requirement tracking process. Requriement tracking may be supported by appropriate software tools.

1.8 Configuration Management

At this point it must be described how

- the different documents during project execution will be handled and administrated (compilation, verification/review process, actualization, archival storage etc.)
- the documents will be uniquely identified (name, version number, date, author etc.) and how it will be ensured, that all team members will always be able to access to the latest and up-to-date version of the documents.

2.0 Verification and Validation Plan (V&V-Plan)

The V&V-Plan describes the planned measures for fault avoidance during software design and development.

The V&V-Plan is no test plan, which defines the tests to be conducted in detail. It is a plan which shall describe the planning of the general verification and validation measures, which shall be conducted for software testing. That means, the V&V-Plan contains a list of the general measures to be conducted and describes in which phase these tests have to be conducted. The detailed definition of the tests will be described in the corresponding test plans.

2.1 Planned measures for fault avoidance and their implementation

2.1.1 Safety Requirement Specification (SRS)

Table B.1 - Techniques and measures to avoid mistakes during specification of E/E/PE system design requirements (see 7.2 / IEC 61508-2)

	Technique/measure	See IEC 61508-7	SIL1	SIL2	SIL3	SIL4	Verification of technique/measure		
							Applied	Description	Result
	Project management	B.1.1	M low	M low	M medium	M high			
	Documentation	B.1.2	M low	M low	M medium	M high			
	Separation of E/E/PE system safety functions from non-safety functions	B.1.3	HR low	HR low	HR medium	HR high			
	Structured specification	B.2.1	HR low	HR low	HR medium	HR high			
▨	Inspection of the specification	B.2.6	- low	HR low	HR medium	HR high			
▨	Semi-formal methods	B.2.3, see also table B.7 of IEC 61508-3	R low	R low	HR medium	HR high			
▨	Checklists	B.2.5	R low	R low	R medium	R high			
▨	Computer aided specification tools	B.2.4	- low	R low	R medium	R high			
▨	Formal methods	B.2.2	- low	- low	R medium	R high			

All techniques marked “R” in the grey shaded group are replaceable, but at least one of these is required.

For the verification of this safety lifecycle phase, at least one of the techniques or measures shaded grey in this table or listed in Table B.5 shall be used.

NOTE 1 For the meaning of the entries under each safety integrity level, see the text preceding this table.

NOTE 2 The measures in this table can be used to varying effectiveness according to Table B.6, which gives examples for low and high effectiveness. The effort required for medium effectiveness lies somewhere between that specified for low and for high effectiveness.

NOTE 3 The overview of techniques and measures associated with this table is in Annex B of IEC 61508-7.Relevant subclauses are referenced in the second column.

2.1.2 Fault Avoidance Software

See Document „03_Fault_Avoidance_Software_Template_V00_1.docx“.

2.2 Tools

Note: If the applied tools are not completely known during concept phase, further tools may be added in the architecture specification, see template for architecture specification.

Every tool, which will be applied during specification, development, verification and validation, must be listed below.

Tool No.	Tool Type	Manufacturer	Tool Name	Version	Classification
1	Requirement Tracking	Microsoft	MS Excel	2019	
2	Integrated Development Environment	IBM	Eclipse	2019-12	
3	Test	QA-Systems	Cantata	2019-12	

"Classification": Definition, whether the tool complies to T3, T2 or T1 (see IEC 61508-3, 7.4.4)

For every tool, the following information must be available:

- Specification of the applied tool
- Bug lists/errata sheets/version information must be available in order to guarantee, that known faults and information about restricted use of tool functions may be considered
- Proof of tool application (including tool revision) in previous projects (if available)
- If proprietary software tools (i.e. tools for company internal use) are applied, the revisions of these tools must be administrated by means of an appropriate configuration management tool
- If necessary, number of shipped tools/releases and supporting libraries/software elements must be evaluated by estimating the hours of operation, see IEC 61508-7, Table D.1.

3.0 List of documents

Due to the high number of documents and their different revisions, which have to be handled during a safety software development process, it is highly recommended to sum up all documents (including document title, file name, date, revision) in the table below.

This document list is „alive", i.e. that many revisions might arise and be processed during project execution. The document list is to be updated if new documents are added or new versions are released.

It is recommended to use the following file name format:

{document number}_{short document title}_{version}_{date}.file_extension

No.	Title	File Name	Version	Status	Last Modification
SC	Software Concept (Customer Document)	SC_Task_Safe_Temperature_Monitoring_Concept_V01_2021-01-29_.docx	0.1	Released	2021-01-29
00	Requirement Tracking	00_Requirement_Tracking_Template_V01_2.xlsx	1.2	Template	

4.0 Note

This template is **not considered complete, but applicable**. The template contains the requirements of IEC 61508 for the development of software functions. The template is to be regarded as alive, i.e. during the development process, possibly issues may be discovered, which are not contained in the template. These should be added to the template at the correct place, in order to be able to consider them during next use. The more diverse software is created by means of this template, the more completely and exactly can the development process for software be realized in the future.