Hochschule Ruhr West
Institut für Naturwissenschaften
Sicherheitstechnik - Funktionale Sicherheit
i.A. von Prof. Dr.-Ing. David Schepers

Project „Safe Temperature Monitoring“

Validation Plan

Modification history:

Version	Date	Modification	Creator	Auditor
1.0	2020-09-28	First version	SaS	DS
1.1	2023-06-25	Safe Temperature Monitoring	CKN	DS

Content

1.0 Validation plan

Who shall perform the validation?

Company: Unicorn Testing GmbH
Name, first name: Martinez, Oscar
expertise: Certified Tester - Foundation
phone / e-mail: omartinez@de.unicorn.com

- Check whether all relevant standards have been consulted? (Each standard separately)
  - Yes. IEC 61508 standard has been referred for implementing this project

- Check all documentation for up-to-dateness and completeness?
  - Yes. All IEC 61508 specified documents are well maintained.

- Checking verification activities for completeness with clear description of activities?
  - Yes

- Which analysis and test methods were used?
  - Black box tests, White box tests, Negative tests, Boundary condition tests, statistical testing and code coverage tests are used

- Are all safety functions according to the SRS are implemented and executable without faults?
  - Yes. All SRS requirements are implemented and the failed tests during integration testing have been noted and the review points were shared to the SW-Development team

- Were all operating parameters according to the SRS considered and have they been implemented and verified?
  - Yes. The parameters are validated for its physical limits and plausible values.

- Verification of forward and backward traceability for all requirements.
  - Yes. Requirement traceability can be found in this document (00_Requirement_Tracking_Template_V01_2).

- Are the output data values and all test results, correct?
  - No. 24/30 tests were passed.
  - Failed tests include Black box, Negative tests and code coverage tests.

- Description of the validation procedure including justification.
  - The justification must include:
    - The choice of manual or automated techniques or both;
    - The choice of static or dynamic techniques or both;
    - The selection of analytical or statistical techniques or both;
    - The selection of acceptance criteria, based on objective factors or judgments by experts or both.

- Listing of tools and equipment incl. calibration data
  - Eclipse for SW-Development and SW-testing
  - Microsoft Excel for requirement tracking

- Microsoft Word for the remaining IEC 61508 documents

- Documentation of the results of the validation activities
  - Documented in the company server

- Listing of discrepancies between expected and actual results.
  - Plausibility of 'maxDisCr' was not implemented
    - Issue found during the integration testing and the review points were shared to the SW-Development team
- Does the subsystem meet the systematic capability? (SIL, independence of functions)
  - Yes. The review points are fixed now and the system meets the SIL3 requirements now.

1.1 Process simulation

Process simulation shall test the function of a software system, together with its interface to the outside world, without allowing it to modify the real world in any way. The creation of a system, for testing purposes only, which mimics the behaviour of the equipment under control (EUC).

The simulation may be software only or a combination of software and hardware. It must

– provide inputs, equivalent to the inputs which will exist when the EUC is actually installed;

– respond to outputs from the software being tested in a way which faithfully represents the controlled plant;

– have provision for operator inputs to provide any perturbations with which the system under test is required to cope.

When software is being tested the simulation may be a simulation of the target hardware with its inputs and outputs.

1.2 Animation of specification and design

This shall guide the software verification by means of a systematic examination of the specification.

A representation of the software that is more abstract than the executable code (i.e. a specification or a high level design) is examined to determine the behavior of the eventual executable software. The examination is automated in some way (depending on the possibilities afforded by the nature and level of abstraction of the higher level representation) so as to simulate the behavior and outputs of the executable software. One application of this approach is to generate tests (or "oracles") that can be later applied to the executable software, thus automating to some degree the testing process. Another application is to animate a user interface so that non-technical end-users can gain some appreciation of the detailed meaning of the specification to which the software developers will work. This provides a valuable method of communication between the two groups.

2.0 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.