SAFEXPLAIN

Safe and Explainable
Critical Embedded Systems based on AI

Ph2T0001 DL Requirements Specifications

Version 2.0

Documentation Information

Contract Number	101069595
Project Website	www.safexplain.eu
Contratual Deadline	04.12.2024
Dissemination Level	SEN
Nature	R
Author	Javier Fernández
Modified by	Lorea Belategi
Reviewed by	Irune Agirre
Approved by	Irune Agirre
Keywords	DL, Functional safety, DL requirements specifications

This project has received funding from the European Union's Horizon Europe programme under grant agreement number 101069595.

Table of Contents

1 Review / Modification History

Version	Date	Description Change
V2.0	15/02/2024	Changes Applied as a result of TÜV Review 2024-01-19
V1.0	18/12/2023	First version after complete internal review
V0.2	15/09/2023	Modifications and improvements based on internal review
V0.1	11/08/2023	First draft

Note. The paragraphs/name of the project/Rev./Ref./history table in ***blue*** *must be replaced with the information for the specific project. The paragraphs written in* ***red*** *are instructions that can be used as a guide, so they must be deleted.*

2 Objective

The aim of this document is to facilitate/guide the definition of the Deep Learning (DL) requirements specifications.

3 Scope

This template applies to the DL requirements specification of the System Architecture phase performed through the Artificial Intelligence - Functional Safety Management (AI-FSM). In fact, the documents generated at this step shall encompass safety, operational, functional and non-functional requirements specifications as well as interface requirements.

4 Requirements Features and Conventions

This section provides a set of characteristics and attributes that are required for each requirement.

4.1 Description of the Requirements

All the requirements of a safety-related project must be specified following the characteristics below:

- Shall be described in such a way that they are*:*
    - **Clear.** The requirement must be easy to understand and not misleading. The requirements are written in a way that allows them to be understood by all stakeholders in the project.
    - **Concise.** The requirement must not be too long. It must be easy to read and understand, and it must not contain definitions, descriptions of its use, or reasons for its need.
    - **Unambiguous.** The requirements can be interpreted only one way.
    - **Verifiable.** The implementation of the requirement can be determined through one of four possible methods: inspection, analysis, demonstration, or test.
    - **Traceable.** A good requirement is traceable: it must be easily traced through to specifications, design and testing. Besides, it must have a unique identity or number.
    - **Complete.** All conditions under which the requirement applies should be stated.
    - **Feasible.** The requirement can be implemented within the constraints of the project.
        - Shall be written to aid comprehension by those who are likely to use the information at any phase of the E/E/PE safety-related system.
        - Shall be expressed in natural or formal language and/or logic, sequence or cause and effect diagrams that define the necessary safety functions.

4.2 Requirements Specifications Table

The attributes required for each requirement to ease its identification and description are defined in the following table:

Table 1. Table of attributes for each requirement

<Identifier>	<Title>
Description	
Source	
Phase of the lifecycle	
Reference	
Type	
Validation criteria	
Date	
Version	

A brief description of each field of the previous table has been done below:

- *<Identifier>.* Each Safety Requirement must be identified with a code like **REQ-YYY-OOO-ZZZ** where **REQ** refers to *requirement* and **YYY** is a code that identifies the requirement category. In this case the code selected is **DLRS** (*Deep Learning Requirement Specification*). Depends on the project more subdivisions can be necessary to identify the requirements (The identifier "**OOO**" can be used to identify these subdivisions). Besides, the identifier "**ZZZ**" represents an incremental number used to identify the requirements in each sub-division. The last two identifiers are optional and they can be replaced for another approach.

- *<Title>.* Title of the requirement. The requirements can be divided into some more specific groups that can be directly related to the application such as components and devices of the system, the regulations, tools, programming languages, and others. In order to identify these specific sub-divisions it can be useful to define first the name of the sub-division and then the title of the requirement.

- *<Description>.* Brief description of the requirement. The description must be short (max. 2-3 lines), aligning with the characteristics outlined in Subsection 5.1, and it must only include one requirement. Besides, it must be defined by the following keywords:

    - ***MUST.*** This word, or the adjective "required", means that the definition is mandatory.
    - ***MUST NOT.*** This phrase means that the definition is an absolute prohibition of the specification.
    - ***SHOULD.*** This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully considered before choosing a different course. A requirement expressed as "should" could become a mandatory requirement in a future version of the document.

- *<Source>.* Source of information relevant to the requirement, i.e. department, contact person, etc.

- *<Reference>.* References relevant to the requirement, i.e. documents, files, figures, emails, etc.

- *<Type>.* Mandatory/Desirable/Optional.

- <Validation Criteria> The requirements will be validated using one, or several of the following methods (at different process phases), depending on the attributes assigned to them:

    - ***Inspection***: It takes place when a direct examination of the design documentation is enough to show that the requirement has been implemented. The requirements must be traced to the proposed design for the system. This design should be defined in the documents.
    - ***Analysis***: This is a non-functional validation, which may include simulation, quantitative analysis, statistical analysis and comparison between the results obtained analytically and numerically. The requirements will be tested by a dedicated analysis.

- ***Test***: Implies a functional validation, which can verify the suitability of the implemented requirement by direct measurement. The requirements will be verified using a test procedure. A test is defined by a set of input parameters, an environment, a procedure and a set of outcomes or outputs.

- *<Phase>:* Specification/Architecture/Detail. Phase in which the requirement is defined as specified in the V-model. For the DL requirements case, DL architecture specifications.
- *<Date>:* Date of creation of the requirement version.
- *<Version>:* Requirement version. All versions are included in the report in a consecutive order.

5 DL Requirements Specifications

This section provides guidance for DL requirement specification. DL requirements shall specify the functional and non-functional requirements associated with the DL constituent, defining its functionality, safety functions and the adopted strategies to achieve the desired Safety Integrity Level (SIL) (Systematic Capability).

In general, the following listed items should be under consideration when creating the DL specification:

- ***Functional.*** *The DL requirement specifications contains the requirements of all software functions implemented by the DL constituent, describing what each function does and if it is safety-related e.g.:*
  - *Safety functions*
    - *Functions that enable the EUC to achieve or maintain a safe state.*
    - *Conditions for the execution of the safety functions, including relation with the ODD and Operational Scenarios*
  - *Non-safety functions*
    - *Display elements*
    - *Processing of non-safety inputs/outputs*
    - *…..*

  Note: All software shall be treated as safety related, unless suitable design measures ensure that the failures of non-safety functions are interference free.

- ***Non-Functional – Characterizing properties.*** *An adequate specification of non-functional properties may include requirements for:*
  - *accuracy*
  - *timing and performance*
  - *capacity*
  - *robustness*
  - *overload tolerance*
  - *…*
- ***Software Systematic Capability.*** *The requirements necessary to achieve the desired SIL specified for each safety function allocated in the DL constituent.*
  - *Requirements related to the techniques, measures, and processes to control and avoid systematic faults so that systematic safety integrity of the software elements meets the requirements of the SIL.*
- ***Operation modes.***
- ***Interfaces.*** *Interaction between different elements such as:*

- *Elements with different systematic capability.*
- *Non safety-related elements.*
- *Safety-related elements and the environment.*
- *Between the DL constituent and other system SW components and HW platform.*

- ***Diagnostics.***
  - *Fault detection:*
    - *DL constituent software and hardware faults.*
    - *DL model insufficiencies.*
  - *Periodic testing:*
    - *Off-line tests.*
    - *On-line tests.*
  - *Software self-monitoring.*
  - *Monitoring of the DL constituent.*
  - *….*

6 DL Requirements Overview

To obtain a comprehensive overview of all the requirements established in this phase, it is essential to collect them in the following table:

Table 2. Data requirements overview

<Identifier>	Description

***Reminder**:*

- Update the state of REF_Ph0D0003_AI Document List.docx when a document is generated or modified, including the last version generated.

- The tools and frameworks employed must be listed in REF_Ph0D0011_AI_Tools_Selection.docx.

- The traceability between the traditional software requirements specifications and DL requirements must be updated in REF_Ph0D0013_AI_Traceability_Matrix.docx

7 Acronyms and Abbreviations

Below is a list of acronyms and abbreviations employed in this document:

- AI - FSM – Artificial Intelligence - Functional Safety Management
- DC – Diagnostic Coverage
- DL – Deep Learning
- EUC – Equipment Under Control
- FPS – Frames Per Second
- SIL – Safety Integrity Level
- ODD – Operational Design Domain

8 Bibliography

Add here the reference to used bibliography / references (if any).