SAFEXPLAIN
Safe and Explainable
Critical Embedded Systems based on AI

SAFEXPLAIN – Safe and explainable critical embedded systems based on AI

Irune Yarza

ikerlan
MEMBER OF BASQUE RESEARCH & TECHNOLOGY ALLIANCE

Funded by European Union

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

Safe and explainable critical systems based on Artificial Intelligence (AI)

SAFE

- Increasing level of autonomy requires higher complexity
  - Exhaustive development **processes**
  - **Safety architecture** and **safety measures** to cope with increasing HW/SW complexity

EXPLAIN

Several challenges to bridge the gap between functional safety and AI

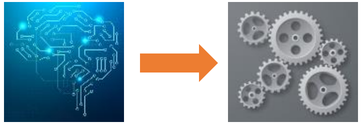
- **Explainability** an essential property

AI

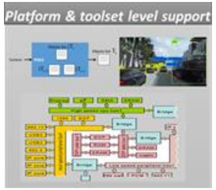
- Critical Embedded Systems increasingly rely on **AI**:
  - E.g., automotive, space, railway, avionics, robotics, etc.
  - Rapidly growing research field not usually aligned with **functional safety practices and standards**

SAFEXPLAIN ambition

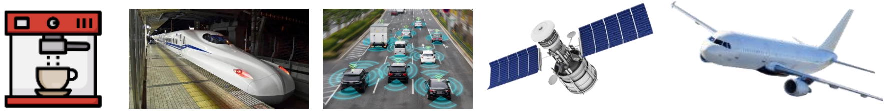
- Architecting DL solutions **enabling certification/qualification**
  - Making them **adhere to "safety culture"**



  - Preserving **high performance**



  - Tailoring solutions to **varying safety requirements** (e.g., different safety needs for a coffee machine and a plane)



OCT 2022 – SEP 2025

SAFEXPLAIN

Safe and Explainable
Critical Embedded Systems based on AI

BARCELONA SUPERCOMPUTING CENTER (BSC)
https://www.bsc.es/

IKERLAN, S. Coop (IKR)
https://www.ikerlan.es/

AIKO SRL (AIKO)
https://www.aikospace.com/

RISE RESEARCH INSTITUTES OF SWEDEN AB (RISE)
https://www.ri.se/

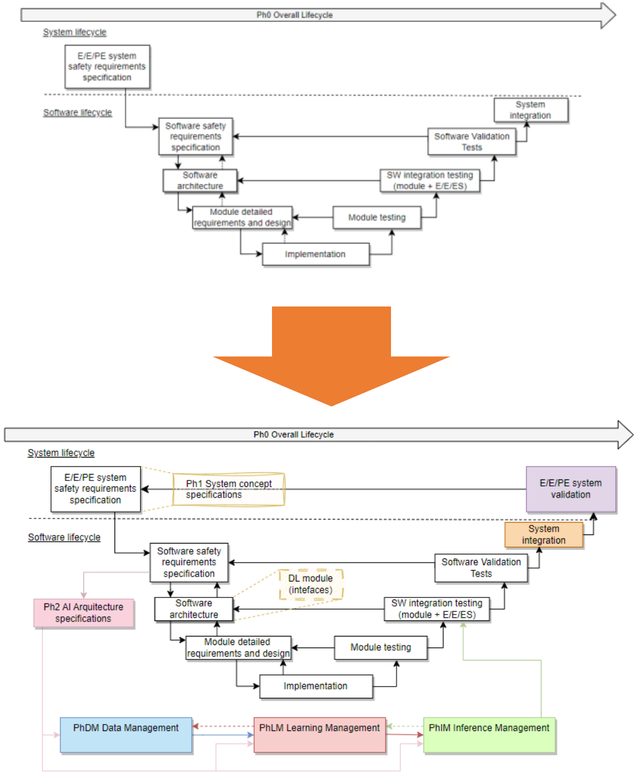
NAVINFO EUROPE BV (NAV)
https://www.navinfo.eu/

EXIDA DEVELOPMENT SRL (EXI)
https://www.exida-eu.com/

Jaume Abella
Project Coordinator

Ambition/objectives

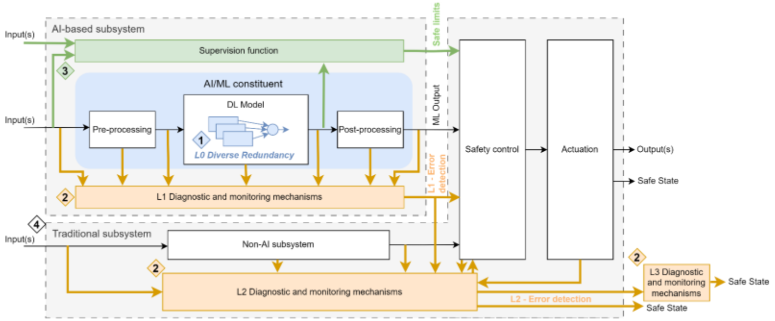
- Re-think safety lifecycle
  - **Keep principles** but with AI implementation in mind
  - Enable the **use of some AI models** first, and generate requirements, goals, unit testing, etc. from there (**bottom-up approach instead of top-down**)
  - Specific steps to **manage data, learning** and **inference**

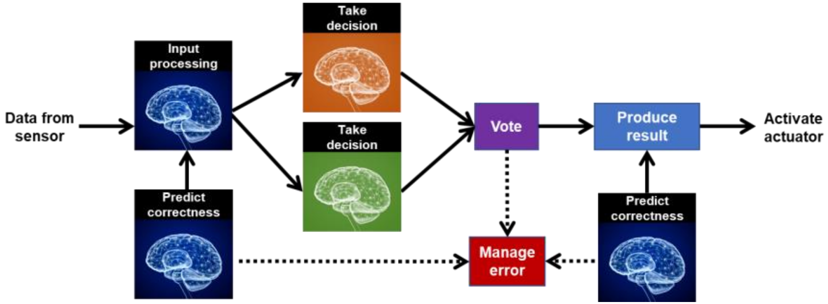


Ph0 Overall Lifecycle
System lifecycle
E/E/PE system safety requirements specification
Software lifecycle
System integration
Software safety requirements specification
Software Validation Tests
Software architecture
SW integration testing (module + E/E/ES)
Module detailed requirements and design
Module testing
Implementation
Ph0 Overall Lifecycle
System lifecycle
E/E/PE system safety requirements specification
Ph1 System concept specifications
E/E/PE system validation
Software lifecycle
System integration
Software safety requirements specification
Software Validation Tests
DL module (intefaces)
Ph2 AI Arquitecture specifications
Software architecture
SW integration testing (module + E/E/ES)
Module detailed requirements and design
Module testing
Implementation
PhDM Data Management
PhLM Learning Management
PhIM Inference Management

Ambition/objectives

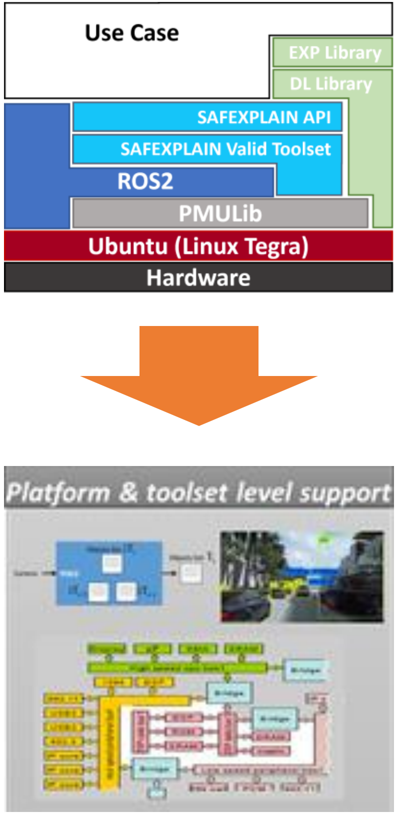
- Re-think AI software
  - Move **from “black-box” to “gray-box”** exposing intermediate behavior
  - Realize AI solutions **following safety principles** (redundancy, monitoring, etc.)
  - Make **AI decisions explainable** (be able to understand why a given decision has been taken)





Ambition/objectives

- Preserve performance and accuracy
  - Keep **high accuracy**
  - Keep **high performance**
  - A **safe AI solution taking too long or with little accuracy is of no use**

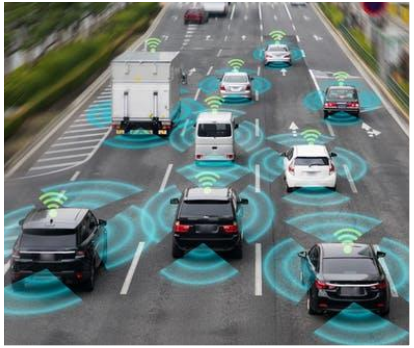


Ambition/objectives

- Assess findings in three key domains

Railway

a relatively controlled scenario with more limited driving options



Automotive

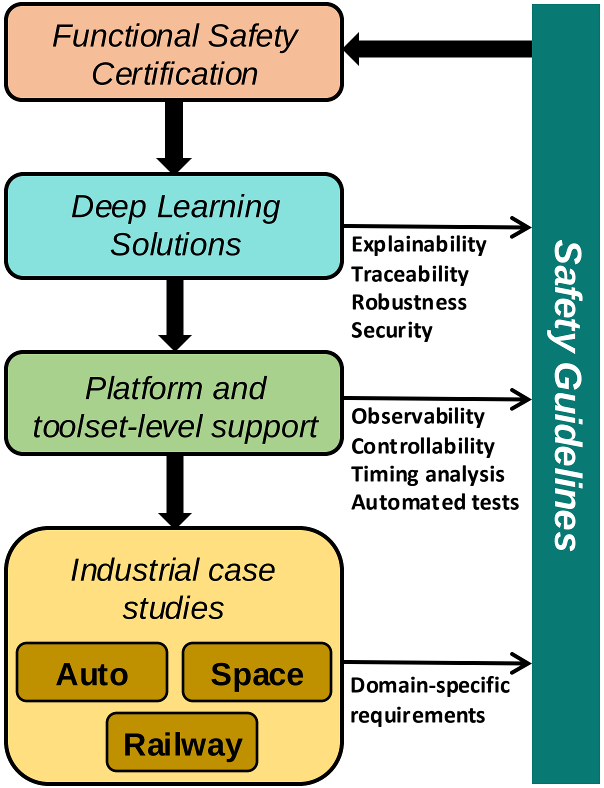
a more complex scenario with ability to change lanes and move freely

Space

the **most complex scenario involving** 3D navigation and extreme lighting conditions

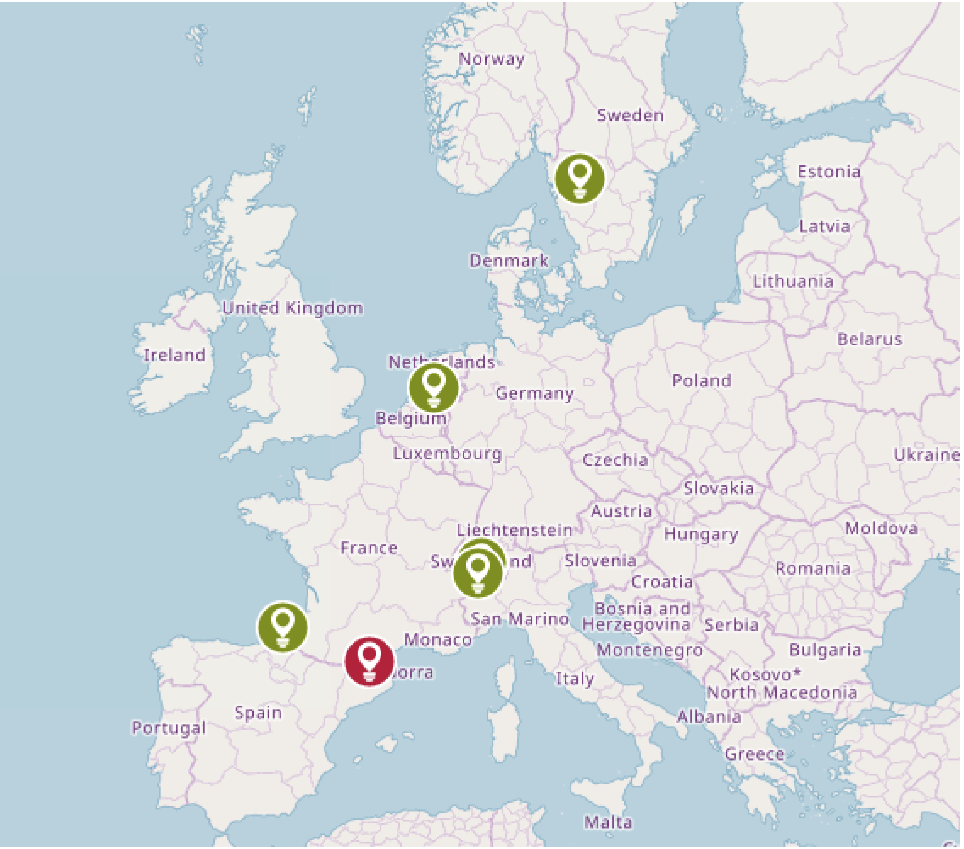
Ambition/objectives

- Ambition: architecting DL solutions **enabling certification/qualification**
  - Making them **explainable** and **traceable**
  - Preserving **high and predictable performance**
  - Tailoring solutions to varying safety requirements by means of **different safety patterns**
  - Evaluation in three **industrial case studies**



Project Consortium

- **BARCELONA SUPERCOMPUTING CENTER (BSC)**
  - https://www.bsc.es/
- IKERLAN, S. Coop (IKR)
  - https://www.ikerlan.es/
- AIKO SRL (AIKO)
  - https://www.aikospace.com/
- RISE RESEARCH INSTITUTES OF SWEDEN AB (RISE)
  - https://www.ri.se/
- NAVINFO EUROPE BV (NAV)
  - https://www.navinfo.eu/
- EXIDA DEVELOPMENT SRL (EXI)
  - https://www.exida-eu.com/



THANK YOU !

SAFEXPLAIN

Safe and Explainable
Critical Embedded Systems based on AI

Follow us on social media:

www.safexplain.eu

Funded by
the European Union

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

SAFEXPLAIN

Safe and Explainable
Critical Embedded Systems based on AI

HiPEAC

Towards functional safety management for AI-based critical systems

Javier Fernández

ikerlan
MEMBER OF BASQUE RESEARCH
& TECHNOLOGY ALLIANCE

Funded by European Union

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

Agenda

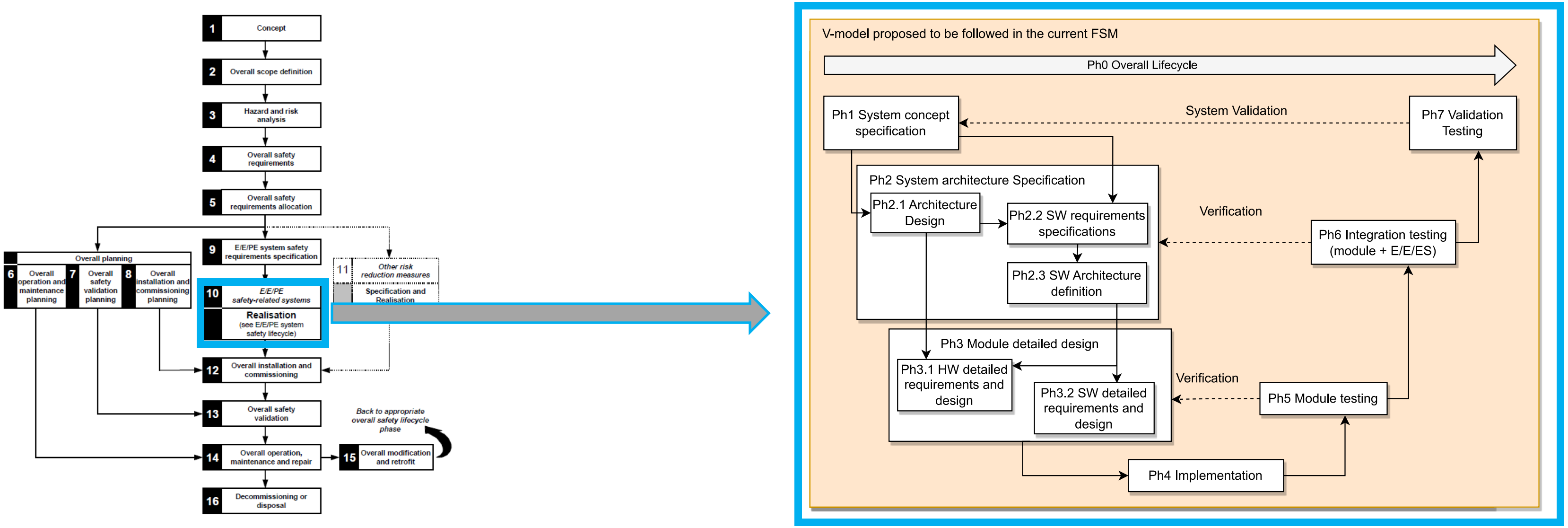
- Contextualization
- Proposed Lifecycle
- AI-FSM
- AI-FSM in-depth
- Safety Technical Assessment

Contextualization

Contextualization

Functional Safety Management (FSM): encompasses all essential activities throughout the Functional Safety lifecycle phases, as mandated by IEC 61508-1. FSM is designed to **prevent errors during specification, design, development, manufacturing, and commissioning**.

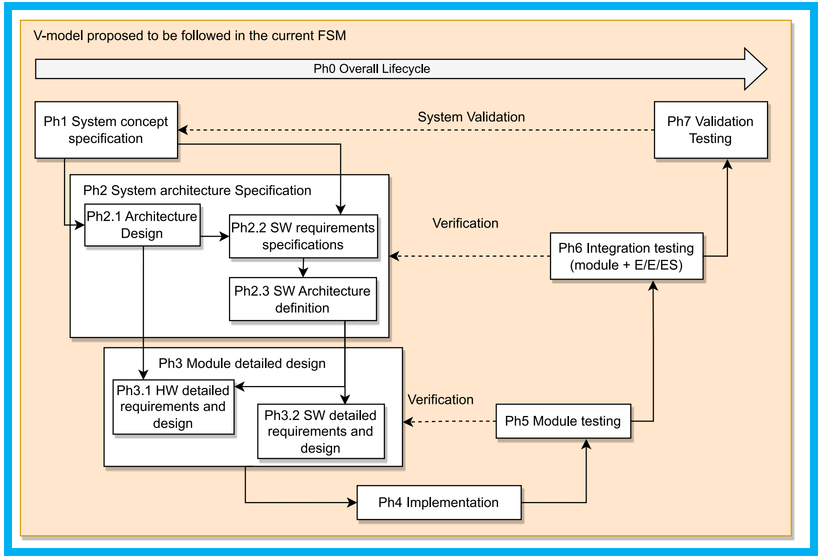
- In context, IKR has its own **FSM for safety systems up to SIL 3** according to IEC 61508.


1 Concept
2 Overall scope definition
3 Hazard and risk analysis
4 Overall safety requirements
5 Overall safety requirements allocation
Overall planning
6 Overall operation and maintenance planning
7 Overall safety validation planning
8 Overall installation and commissioning planning
9 E/E/PE system safety requirements specification
10 E/E/PE safety-related systems
Realisation (see E/E/PE system safety lifecycle)
11 Other risk reduction measures
Specification and Realisation
12 Overall installation and commissioning
13 Overall safety validation
Back to appropriate overall safety lifecycle phase
14 Overall operation, maintenance and repair
15 Overall modification and retrofit
16 Decommissioning or disposal
V-model proposed to be followed in the current FSM
Ph0 Overall Lifecycle
Ph1 System concept specification
System Validation
Ph7 Validation Testing
Ph2 System architecture Specification
Ph2.1 Architecture Design
Ph2.2 SW requirements specifications
Ph2.3 SW Architecture definition
Verification
Ph6 Integration testing (module + E/E/ES)
Ph3 Module detailed design
Ph3.1 HW detailed requirements and design
Ph3.2 SW detailed requirements and design
Verification
Ph5 Module testing
Ph4 Implementation

Contextualization

SIL 3 FSM (IKERLAN): Development process

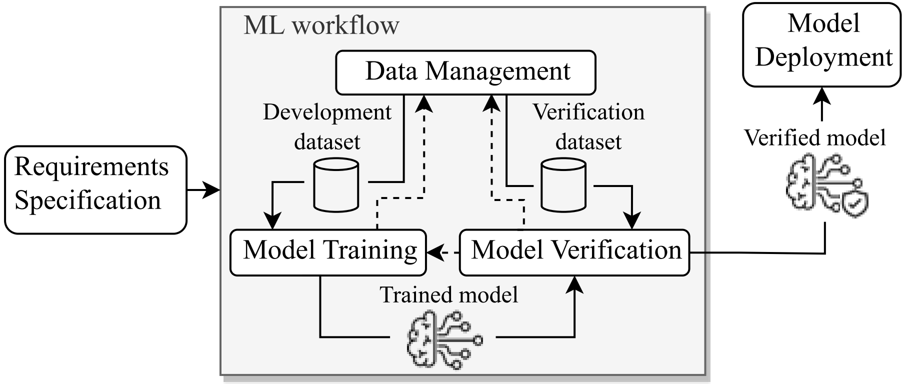
- Traditional lifecycle is based on the V-model development process and structured in the following lifecycle phases:
  - Ph0 Overall Life Cycle
  - Ph1 System Concept Specification
  - Ph2 System Architecture Specification
  - Ph3 Module Detailed Design
  - Ph4 Implementation
  - Ph5 Module Testing
  - Ph6 Integration Testing
  - Ph7 Validation Testing


V-model proposed to be followed in the current FSM
Ph0 Overall Lifecycle
Ph1 System concept specification
System Validation
Ph7 Validation Testing
Ph2 System architecture Specification
Ph2.1 Architecture Design
Ph2.2 SW requirements specifications
Ph2.3 SW Architecture definition
Verification
Ph6 Integration testing (module + E/E/ES)
Ph3 Module detailed design
Ph3.1 HW detailed requirements and design
Ph3.2 SW detailed requirements and design
Verification
Ph5 Module testing
Ph4 Implementation

Contextualization

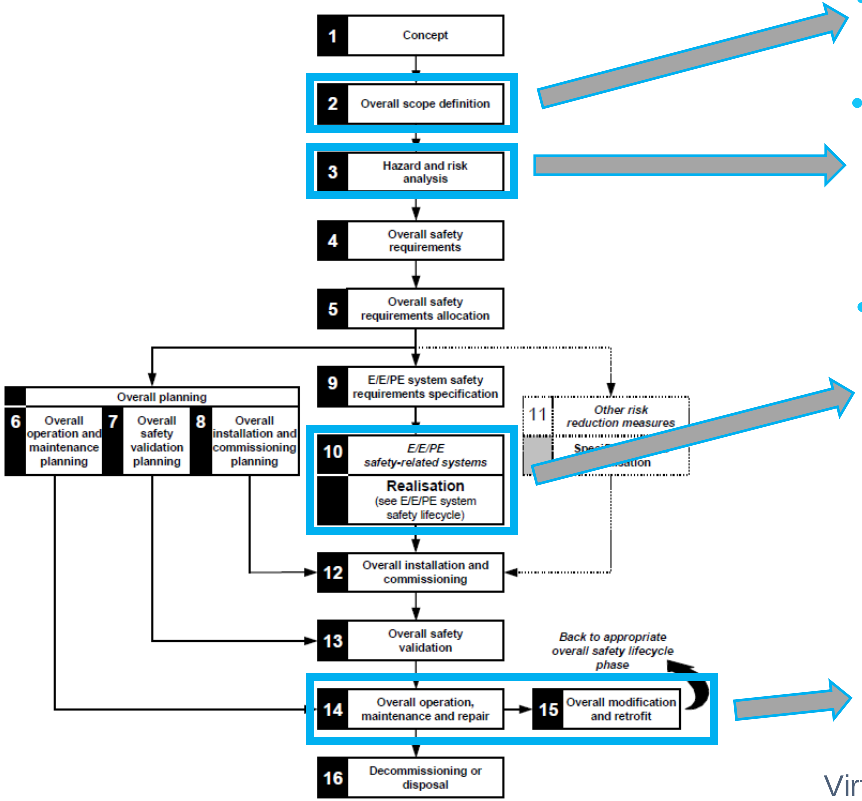
AI lifecycle phases

- Five main stages:
  - Requirements Specification
  - Data Management
    - Development dataset
      - Training + Validation* dataset
    - Verification dataset
  - Model Training
    - Trained model
  - Model Verification
    - Verified model
  - Model Deployment
    - Inference model



Contextualization

Phases affected by including DL



1 Concept
2 Overall scope definition
3 Hazard and risk analysis
4 Overall safety requirements
5 Overall safety requirements allocation
Overall planning
6 Overall operation and maintenance planning
7 Overall safety validation planning
8 Overall installation and commissioning planning
9 E/E/PE system safety requirements specification
10 E/E/PE safety-related systems
Realisation (see E/E/PE system safety lifecycle)
11 Other risk reduction measures
12 Overall installation and commissioning
13 Overall safety validation
Back to appropriate overall safety lifecycle phase
14 Overall operation, maintenance and repair
15 Overall modification and retrofit
16 Decommissioning or disposal

- Definition of the ODD and operational scenarios

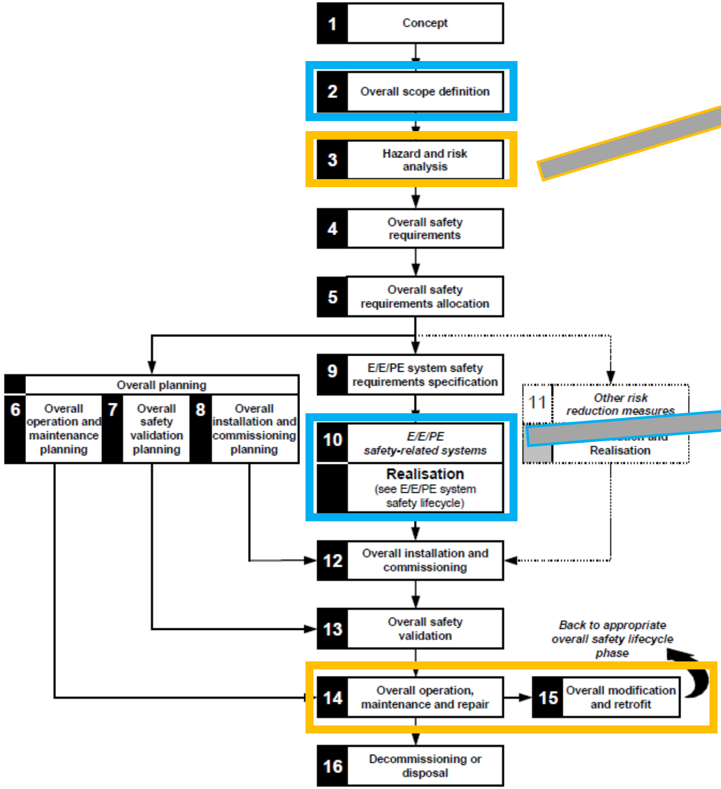
- HARA shall identify potential hazards caused by the DL-based systems. The ODD and operational scenarios are used as input for this stage.

- New phases not contemplated by the traditional V-model:
  - Data management
  - Learning management
  - Inference management

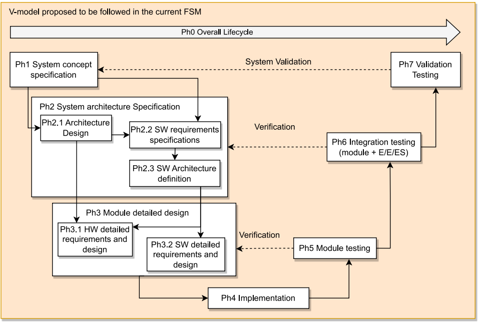
- In traditional software development, updating a product after its release typically involves a lengthy re-assessment process. This can be particularly challenging for DL models, as their product lifecycles often require more frequent updates.

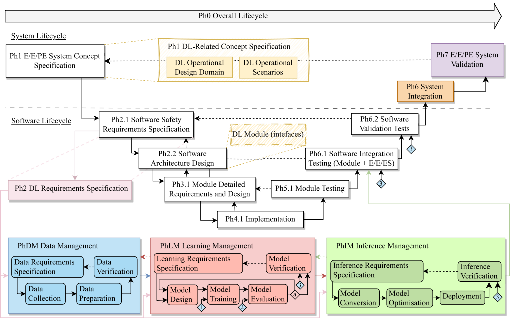
Contextualization

Current state of the AI-FSM


1 Concept
2 Overall scope definition
3 Hazard and risk analysis
4 Overall safety requirements
5 Overall safety requirements allocation
Overall planning
6 Overall operation and maintenance planning
7 Overall safety validation planning
8 Overall installation and commissioning planning
9 E/E/PE system safety requirements specification
10 E/E/PE safety-related systems
Realisation
(see E/E/PE system safety lifecycle)
11 Other risk reduction measures
Realisation
12 Overall installation and commissioning
13 Overall safety validation
Back to appropriate overall safety lifecycle phase
14 Overall operation, maintenance and repair
15 Overall modification and retrofit
16 Decommissioning or disposal

- Not contemplated in the current version. The following version will consider recommendations from standards such as SOTIF.


V-model proposed to be followed in the current FSM
Ph0 Overall Lifecycle
Ph1 System concept specification
System Validation
Ph7 Validation Testing
Ph2 System architecture Specification
Ph2.1 Architecture Design
Ph2.2 SW requirements specifications
Ph2.3 SW Architecture definition
Verification
Ph6 Integration testing (module + E/E/ES)
Ph3 Module detailed design
Ph3.1 HW detailed requirements and design
Ph3.2 SW detailed requirements and design
Verification
Ph5 Module testing
Ph4 Implementation

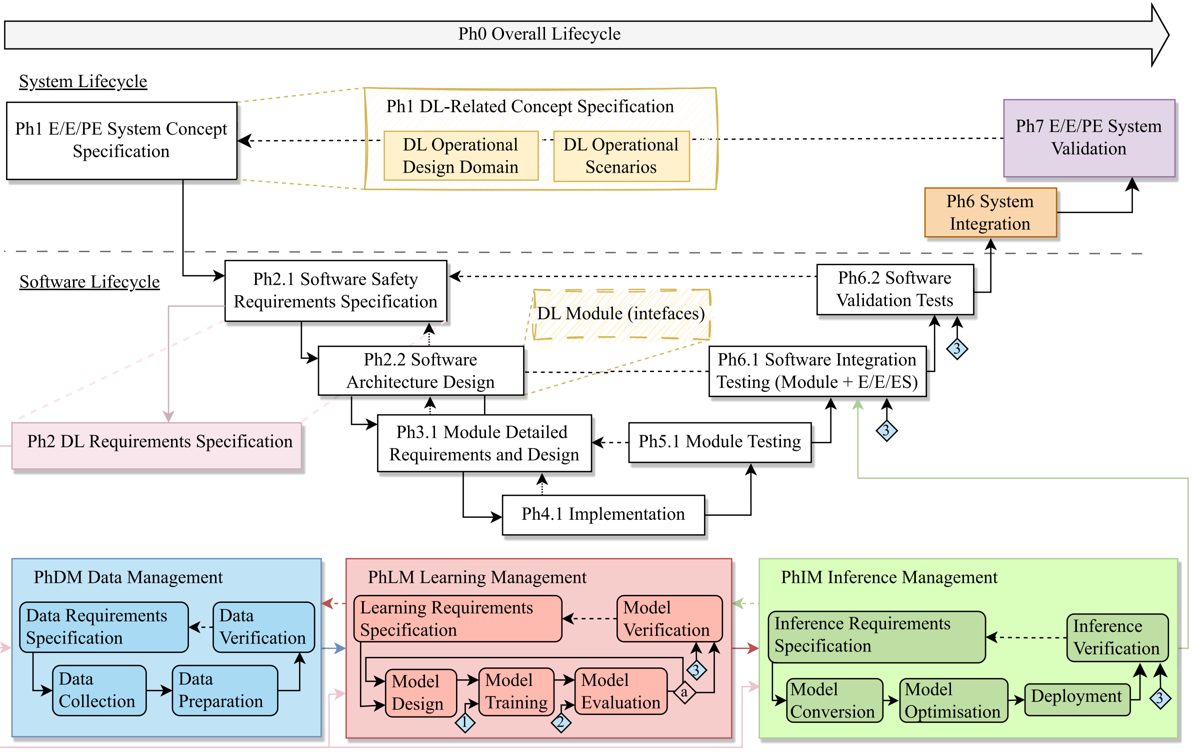

Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
3
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
3

- The current version does not address this challenge.

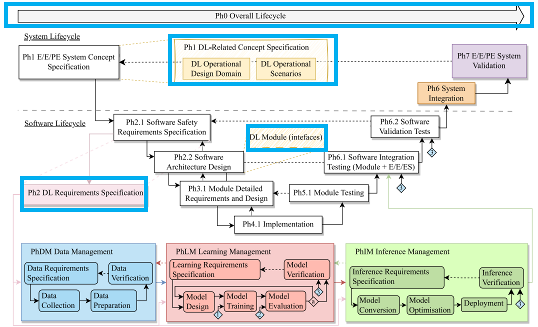
Proposed Lifecycle

Proposed lifecycle

- IEC 61508 traditional functional safety lifecycle (Software V-model) + AI lifecycle

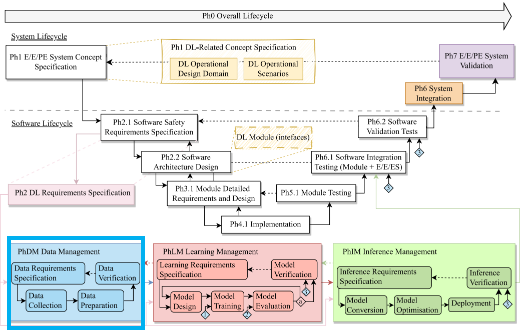

Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
3
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
3

Proposed lifecycle: phases' objectives


Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
DL Module (intefaces)
Ph6.2 Software Validation Tests
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
3
3
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
3
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
3

- **Ph0 Overall Lifecycle:** It is a transversal phase that ***collects*** all the ***generic project information***
  - Documents generated
  - Organization chart
  - Tools selection
- **Ph1 DL-Related Concept Specification:** This phase encompasses the ***definition*** of the ***DL Operational Design Domain (ODD)*** and ***operational scenarios*** in which the DL will operate. In the case the safety-related system entails the use of DL, these definitions are required in addition to the traditional use case description and operation definition outlined in the requirements.
- **DL Modules (interfaces):** This box highlights that Ph2.2 shall define all the interfaces of the DL modules.
- **Ph2 DL Requirements Specification**: This phase ***allocates*** the ***software requirements to DL*** constituents and ***refines them***:
  - Safety, operation, functional and non-functional requirements specification (among others)

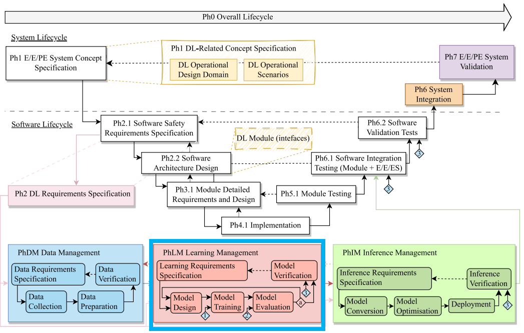
Proposed lifecycle: phases' objectives


Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
1
2
3
a

- **PhDM Data Management**. It is responsible for collecting and preparing the datasets. Four steps:
  - Data req. Specifications. It allocates the DL req. to the data req. and refine them. It shall collect:
    - Data and datasets req.
    - Req. Associated with the collection and preparation steps.
    - Data filename policy.
    - Degree of differentiation.
  - Data collection. It involves collecting all the data to generate the datasets:
    - Data gathering. It involves gathering data from different sources.
    - Data generation. It relates to generating new data to complete the data gathering.
  - Data preparation. In this step, the previous data is cleaned, processed, or annotated to meet the reqs.
  - Data Verification. This phase checks if the datasets meet the data req. specification.
  - Inputs:
    - DL reqs specifications
    - ODD
    - Operational scenarios
  - Ouputs generated:
    - Development dataset (training + validation*)
    - Verification dataset

All actions and decisions taken shall be documented

Proposed lifecycle: phases’ objectives


Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
3
3
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
3
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
3

- **PhLM Learning Management**. It is responsible for generating a DL model that meets the DL req. specification. Five steps:
  - Learning req. Specifications. It allocates the DL req. to learning reqs. and refine them. It shall collect:
    - Qualitative and quantitative learning reqs.
    - Model post-training selection criteria.
    - Reqs. associated with the model design and training.

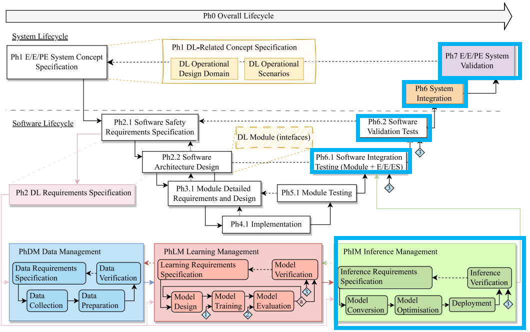
All actions and decisions taken shall be documented

  - Model design. It focuses on the specification of a set of DL models that best suit the application.
  - Model training. In this step, the specified models are generated employing the training dataset.
  - Model evaluation. Once the model(s) are trained, they are evaluated employing the validation dataset.
  - Model verification. This phase not only evaluates the generalization capabilities and identifies potential issues using the verification dataset but also checks if the reqs. are met.

- Inputs:
  - Development dataset (training + validation*)
  - Verification dataset
  - DL req. specification

- Ouputs:
  - Trained model
  - Evaluated model
  - Verified learning model

Proposed lifecycle: phases’ objectives



Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
DL Module (intefaces)
Ph6.2 Software Validation Tests
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
3
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment

- **PhIM Inference Management**. Its purpose is to adapt the verified model for its deployment on the target HW while ensuring that it still meets the DL reqs. after converting and even optimising it. Five stages:
  - Inference req. specification. It allocates the DL and learning reqs. to inference reqs. and refine them. It shall collect:
    - Inference reqs.
    - Req. associated with the model conversion, optimization and deployment
  - Model conversion. The model is transformed into a format suitable for deployment that must ensure compatibility with the specific target inference platform.
  - Model optimisation. the model may undergo optimization to enhance its performance, reduce its size, or adapt it for resource-constrained environments.
  - Deployment. This steps entails the implementation of the model in the target platform.
  - Inference verification. This phase not only evaluates the generalization capabilities and identifies potential issues using the verification dataset but also checks if the reqs. are met.
  - Input:
    - Verified learning model from PhLM
    - Verification dataset from PhDM
    - Learning and DL req. specification
  - Ouput:
    - Verified inference model

AI-FSM

AI-FSM

Definition:

AI-FSM refers to all essential activities to be performed throughout the functional safety lifecycle phases to avoid systematic errors in the development of AI constituents. It is an annex to traditional FSM to be employed when a safety-critical systems involves the use of AI. AI-FSM maps the content of the AI development process with the traditional safety development process.

Scope:

The current version of this AI-FSM is restricted to DL constituents with the following features:

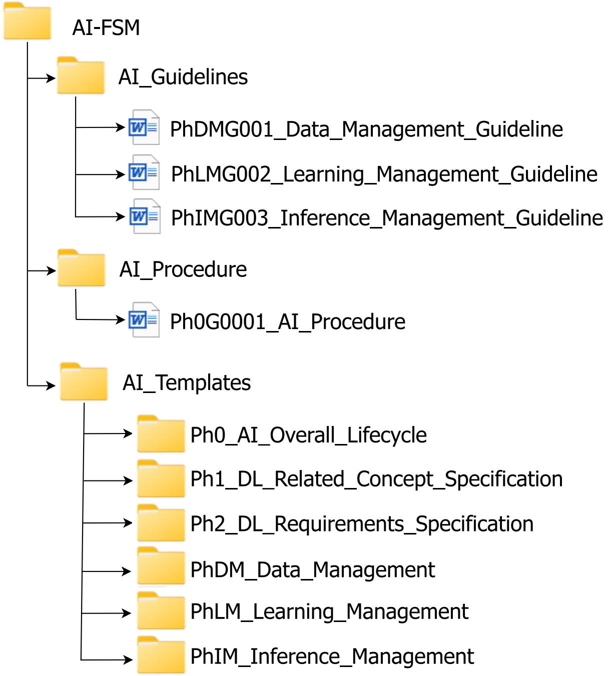
- DL algorithms based on supervised learning for visual perception classification tasks.
- Applications based on offline learning processes in which the model remains fixed at approval time, while excluding online learning processes.

AI-FSM

Types of documents:

- Main procedure: It provides a set of steps required to generate the basic structure for a specific safety-related project. It serves as an internal guideline for fulfilling the procedure template.
- Procedure template: This document compiles how functional safety has been assessed within the organization.
- Guidelines: These documents offer additional guidance for specific processes.
- Templates: Standard documents used to collect the information consistently. They often include examples and tables to be completed.
- Internal Reviews (IRs): reviews based on the activities of the left side of the safety lifecycle. Objective: Check that the activities defined in each phase have been properly carried out.
  - Quality Assurance

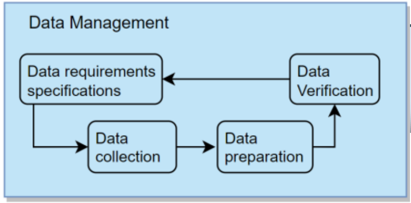
Folder Structure proposed:



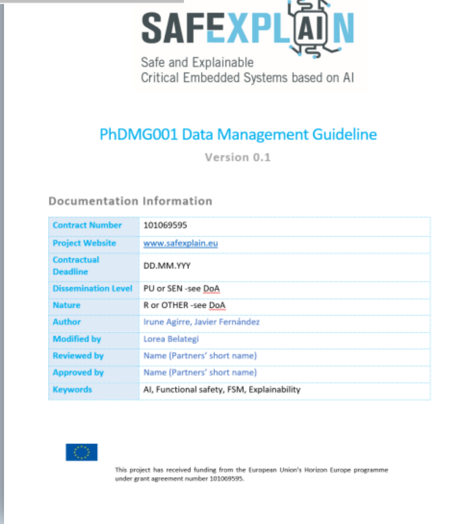
- AI-FSM
  - AI_Guidelines
    - PhDMG001_Data_Management_Guideline
    - PhLMG002_Learning_Management_Guideline
    - PhIMG003_Inference_Management_Guideline
  - AI_Procedure
    - Ph0G0001_AI_Procedure
  - AI_Templates
    - Ph0_AI_Overall_Lifecycle
    - Ph1_DL_Related_Concept_Specification
    - Ph2_DL_Requirements_Specification
    - PhDM_Data_Management
    - PhLM_Learning_Management
    - PhIM_Inference_Management

AI-FSM

- PhDM Data Management



Guideline



SAFEXPLAIN
Safe and Explainable
Critical Embedded Systems based on AI

PhDMG001 Data Management Guideline

Version 0.1

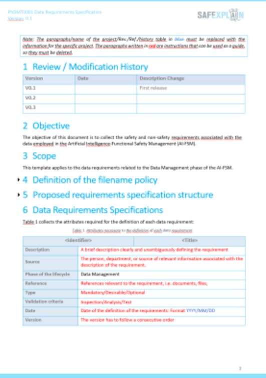
Documentation Information

Contract Number	101069595
Project Website	www.safexplain.eu
Contractual Deadline	DD.MM.YYY
Dissemination Level	PU or SEN -see DoA
Nature	R or OTHER -see DoA
Author	Irune Agirre, Javier Fernández
Modified by	Lorea Belategi
Reviewed by	Name (Partners' short name)
Approved by	Name (Partners' short name)
Keywords	AI, Functional safety, FSM, Explainability

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

Templates

Data reqs.



PhDMT0001 Data Requirements Specification
Version 0.1

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.

1 Review / Modification History

Version	Date	Description Change
V0.1		First release
V0.2		
V0.3		

2 Objective

The objective of this document is to collect the safety and non-safety requirements associated with the data employed in the Artificial Intelligence Functional Safety Management (AI-FSM).

3 Scope

This template applies to the data requirements related to the Data Management phase of the AI-FSM.

4 Definition of the filename policy

5 Proposed requirements specification structure

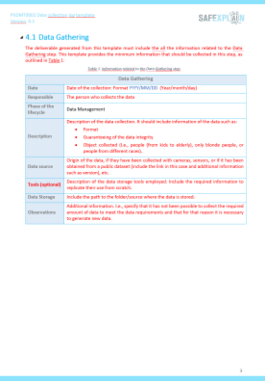
6 Data Requirements Specifications

Table 1 collects the attributes required for the definition of each data requirement:

Table 1. Attributes necessary to the definition of each data requirement

<identifier>	<Title>
Description	A brief description clearly and unambiguously defining the requirement
Source	The person, department, or source of relevant information associated with the description of the requirement.
Phase of the lifecycle	Data Management
Reference	References relevant to the requirement, i.e. documents, files,
Type	Mandatory/Desirable/Optional
Validation criteria	Inspection/Analysis/Test
Date	Date of the definition of the requirements: Format YYYY/MM/DD
Version	The version has to follow a consecutive order

Data Collection



PhDMT0002 Data collection log template
Version 0.1

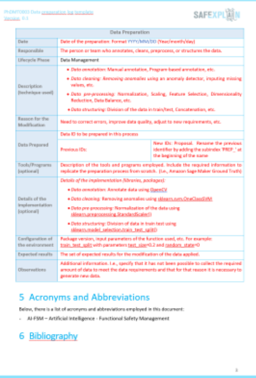
4.1 Data Gathering

The deliverable generated from this template must include the all the information related to the Data Gathering step. This template provides the minimum information that should be collected in this step, as outlined in Table 1:

Table 1. Information related to the Data Gathering step

Data Gathering	
Date	Date of the collection: Format YYYY/MM/DD (Year/month/day)
Responsible	The person who collects the data
Phase of the lifecycle	Data Management
Description	Description of the data collection. It should include information of the data such as: ▪ Format ▪ Guaranteeing of the data integrity ▪ Object collected (i.e., people (from kids to elderly), only blonde people, or people from different races).
Data source	Origin of the data, if they have been collected with cameras, sensors, or if it has been obtained from a public dataset (include the link in this case and additional information such as version), etc.
Tools (optional)	Description of the data storage tools employed. Include the required information to replicate their use from scratch.
Data Storage	Include the path to the folder/source where the data is stored.
Observations	Additional information. I.e., specify that it has not been possible to collect the required amount of data to meet the data requirements and that for that reason it is necessary to generate new data.

Data Preparation



PhDMT0003 Data preparation log template
Version 0.1

Data Preparation		
Date	Date of the preparation: Format YYYY/MM/DD (Year/month/day)	
Responsible	The person or team who annotates, cleans, preprocess, or structures the data.	
Lifecycle Phase	Data Management	
Description (technique used)	• *Data annotation: Manual annotation, Program based annotation, etc.* • *Data cleaning: Removing anomalies using an anomaly detector, imputing missing values, etc.* • *Data pre-processing: Normalization, Scaling, Feature Selection, Dimensionality Reduction, Data Balance, etc.* • *Data structuring: Division of the data in train/test, Concatenation, etc.*	
Reason for the Modification	Need to correct errors, improve data quality, adjust to new requirements, etc.	
Data Prepared	Data ID to be prepared in this process	
	Previous IDs:	New IDs: Proposal. Rename the previous identifier by adding the subindex 'PREP_' at the beginning of the name
Tools/Programs (optional)	Description of the tools and programs employed. Include the required information to replicate the preparation process from scratch. (I.e., Amazon Sage Maker Ground Truth)	
Details of the implementation (optional)	*Details of the implementation (libraries, packages):* • *Data annotation:* Annotate data using OpenCV • *Data cleaning:* Removing anomalies using sklearn.svm.OneClassSVM • *Data pre-processing:* Normalization of the data using sklearn.preprocessing.StandardScaler() • *Data structuring:* Division of data in train test using sklearn.model_selection.train_test_split()	
Configuration of the environment	Package version, input parameters of the function used, etc. For example: train_test_split with parameters test_size=0.2 and random_state=0	
Expected results	The set of expected results for the modification of the data applied.	
Observations	Additional information. I.e., specify that it has not been possible to collect the required amount of data to meet the data requirements and that for that reason it is necessary to generate new data.	

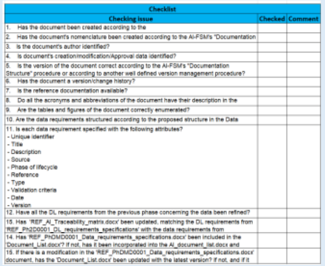
5 Acronyms and Abbreviations

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

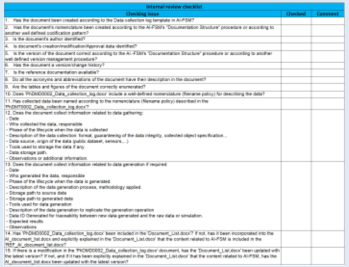
- AI-FSM – Artificial Intelligence - Functional Safety Management

6 Bibliography

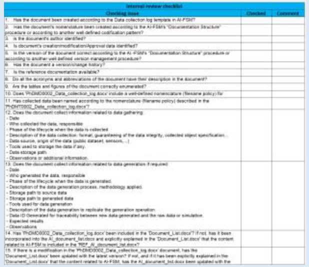
IRs



Checklist		
Checking issue	Checked	Comment
1. Has the document been created according to the		
2. Has the document's nomenclature been created according to the AI-FSM's "Documentation		
3. Is the document's author identified?		
4. Is document's creation/modification/Approval data identified?		
5. Is the version of the document correct according to the AI-FSM's "Documentation Structure" procedure or according to another well defined version management procedure?		
6. Has the document a version/change history?		
7. Is the reference documentation available?		
8. Do all the acronyms and abbreviations of the document have their description in the		
9. Are the tables and figures of the document correctly enumerated?		
10. Are the data requirements structured according to the proposed structure in the Data		
11. Is each data requirement specified with the following attributes? - Unique identifier - Title - Description - Source - Phase of lifecycle - Reference - Type - Validation criteria - Date - Version		
12. Have all the DL requirements from the previous phase concerning the data been refined?		
13. Has 'REF_AI_Traceability_matrix.docx' been updated, matching the DL requirements from 'REF_Ph2D0001_DL_requirements_specifications' with the data requirements from		
14. Has 'REF_PhDMD0001_Data_requirements_specifications.docx' been included in the 'Document_List.docx'? If not, has it been incorporated into the AI_document_list.docx and		
15. If there is a modification in the 'REF_PhDMD0001_Data_requirements_specifications.docx' document, has the 'Document_List.docx' been updated with the latest version? If not, and if it		



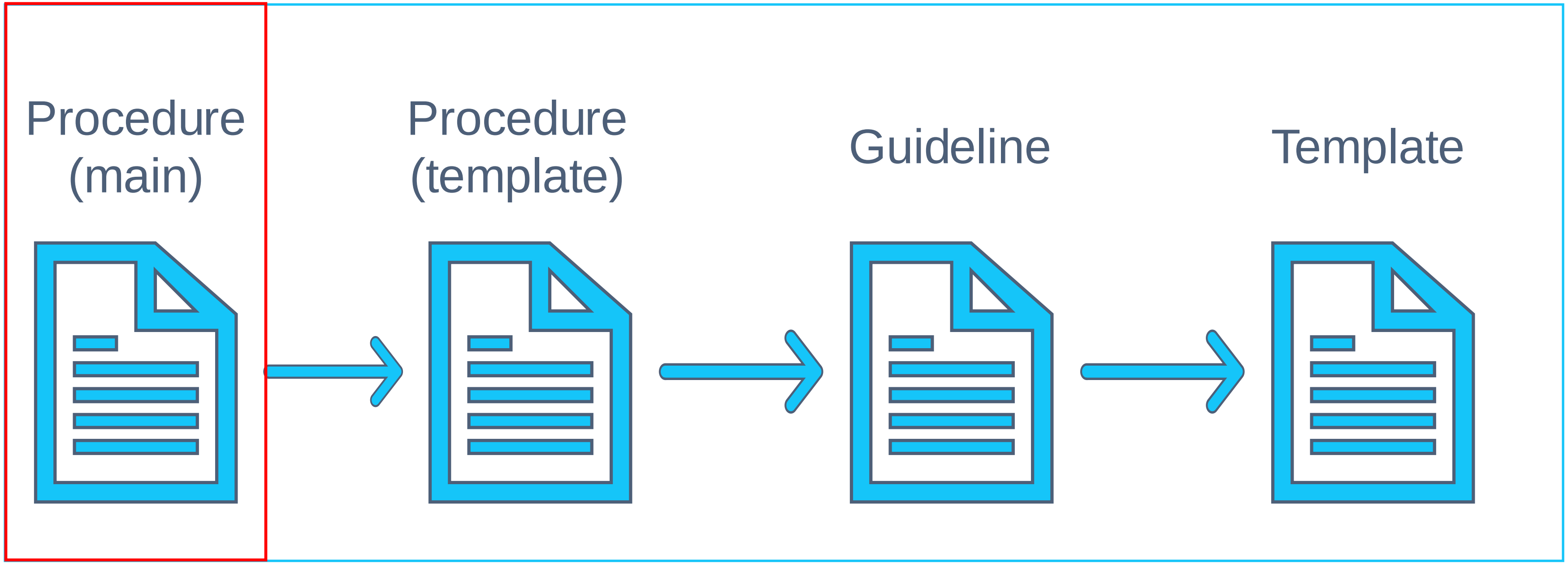
Internal review checklist		
Checking issue	Checked	Comment
1. Has the document been created according to the Data collection log template in AI-FSM?		
2. Has the document's nomenclature been created according to the AI-FSM's "Documentation Structure" procedure or according to another well defined codification pattern?		
3. Is the document's author identified?		
4. Is document's creation/modification/Approval data identified?		
5. Is the version of the document correct according to the AI.-FSM's "Documentation Structure" procedure or according to another well defined version management procedure?		
6. Has the document a version/change history?		
7. Is the reference documentation available?		
8. Do all the acronyms and abbreviations of the document have their description in the document?		
9. Are the tables and figures of the document correctly enumerated?		
10. Does 'PhDMD0002_Data_collection_log.docx' include a well-defined nomenclature (filename policy) for describing the data?		
11. Has collected data been named according to the nomenclature (filename policy) described in the 'PhDMT0002_Data_collection_log.docx'?		
12. Does the document collect information related to data gathering: - Date - Who collected the data, responsible - Phase of the lifecycle when the data is collected - Description of the data collection: format, guaranteeing of the data integrity, collected object specification… - Data source, origin of the data (public dataset, sensors,…) - Tools used to storage the data if any. - Data storage path. - Observations or additional information.		
13. Does the document collect information related to data generation if required: - Date - Who generated the data, responsible - Phase of the lifecycle when the data is generated. - Description of the data generation process, methodology applied. - Storage path to source data - Storage path to generated data - Tools used for data generation - Description of the data generation to replicate the generation operation - Data ID Generated for traceability between new data generated and the raw data or simulation. - Expected results - Observations		
14. Has 'PhDMD0002_Data_collection_log.docx' been included in the 'Document_List.docx'? If not, has it been incorporated into the AI_document_list.docx and explicitly explained in the 'Document_List.docx' that the content related to AI-FSM is included in the 'REF_AI_document_list.docx'?		
15. If there is a modification in the 'PhDMD0002_Data_collection_log.docx' document, has the 'Document_List.docx' been updated with the latest version? If not, and if it has been explicitly explained in the 'Document_List.docx' that the content related to AI-FSM, has the AI_document_list.docx been updated with the latest version?		



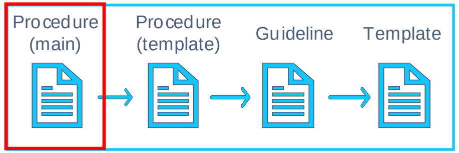
Internal review checklist		
Checking issue	Checked	Comment
1. Has the document been created according to the Data collection log template in AI-FSM?		
2. Has the document's nomenclature been created according to the AI-FSM's "Documentation Structure" procedure or according to another well defined codification pattern?		
3. Is the document's author identified?		
4. Is document's creation/modification/Approval data identified?		
5. Is the version of the document correct according to the AI.-FSM's "Documentation Structure" procedure or according to another well defined version management procedure?		
6. Has the document a version/change history?		
7. Is the reference documentation available?		
8. Do all the acronyms and abbreviations of the document have their description in the document?		
9. Are the tables and figures of the document correctly enumerated?		
10. Does 'PhDMD0002_Data_collection_log.docx' include a well-defined nomenclature (filename policy) for		
11. Has collected data been named according to the nomenclature (filename policy) described in the 'PhDMT0002_Data_collection_log.docx'?		
12. Does the document collect information related to data gathering: - Date - Who collected the data, responsible - Phase of the lifecycle when the data is collected - Description of the data collection: format, guaranteeing of the data integrity, collected object specification… - Data source, origin of the data (public dataset, sensors,…) - Tools used to storage the data if any. - Data storage path. - Observations or additional information.		
13. Does the document collect information related to data generation if required: - Date - Who generated the data, responsible - Phase of the lifecycle when the data is generated. - Description of the data generation process, methodology applied. - Storage path to source data - Storage path to generated data - Tools used for data generation - Description of the data generation to replicate the generation operation - Data ID Generated for traceability between new data generated and the raw data or simulation. - Expected results - Observations		
14. Has 'PhDMD0002_Data_collection_log.docx' been included in the 'Document_List.docx'? If not, has it been incorporated into the AI_document_list.docx and explicitly explained in the 'Document_List.docx' that the content related to AI-FSM is included in the 'REF_AI_document_list.docx'?		
15. If there is a modification in the 'PhDMD0002_Data_collection_log.docx' document, has the 'Document_List.docx' been updated with the latest version? If not, and if it has been explicitly explained in the 'Document_List.docx' that the content related to AI-FSM, has the AI_document_list.docx been updated with the		

AI-FSM in-depth

AI-FSM in-depth



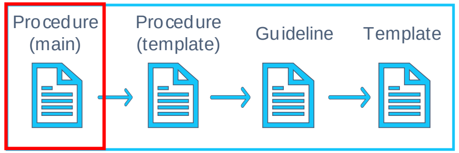
AI-FSM in-depth: Procedure (main)



Procedure (main) → Procedure (template) → Guideline → Template

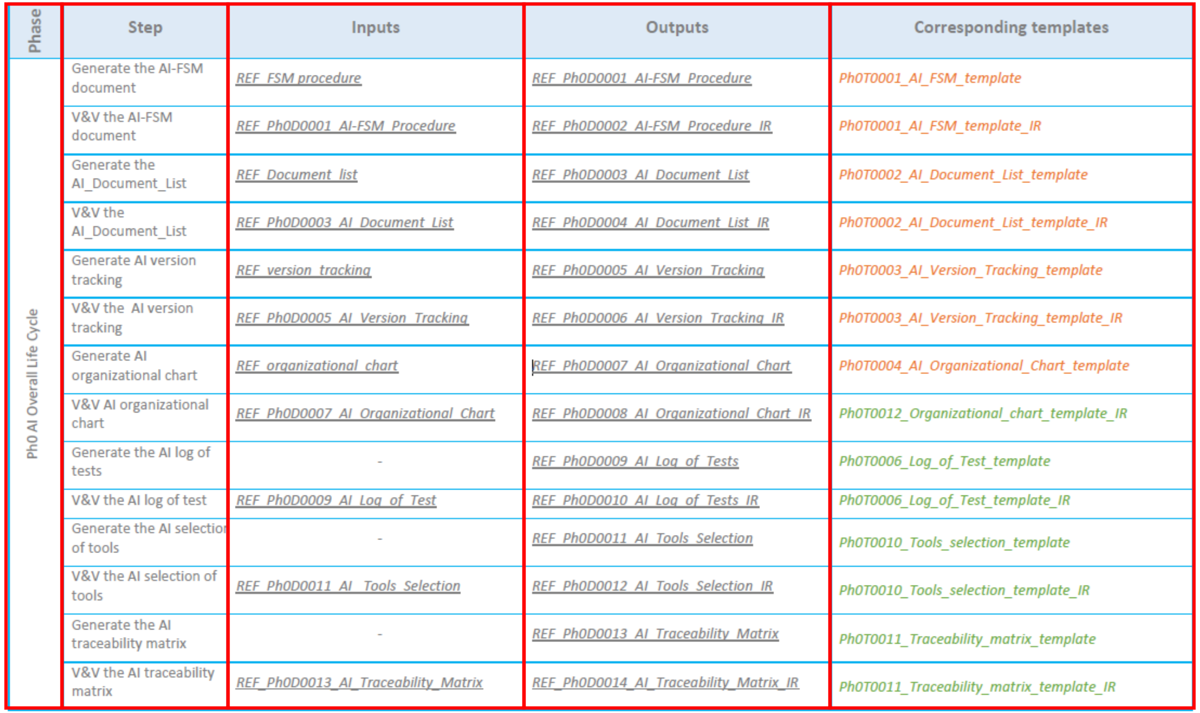
- Defines the context:
  - AI definitions
  - Limitations of the current AI-FSM version
- Defines the traditional FSM lifecycle and the AI lifecycle.
- Expands the traditional FSM lifecycle, mapping it with the AI lifecycle.
- Proposes a folder structure for storing the documents and artifacts for each phase.
- Describes the inputs and outputs of each phase, identifying the corresponding template for their generation.
- Describes how these templates shall be generated and stored for each phase.

AI-FSM in-depth: Procedure (main)



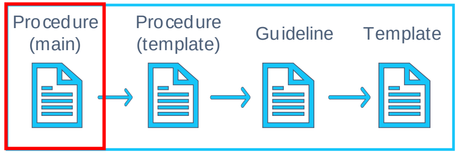
Ph0 Overall lifecycle

Table 1. Inputs and outputs of the overall lifecycle phase (Ph0)



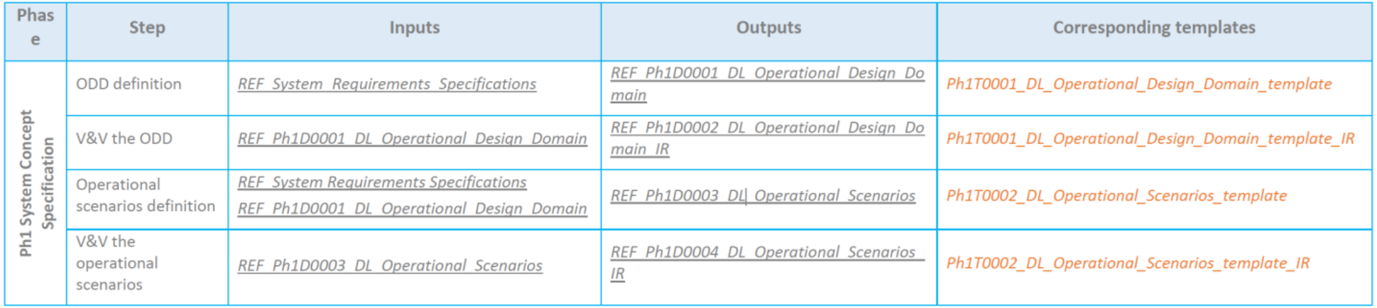
Phase	Step	Inputs	Outputs	Corresponding templates
Ph0 AI Overall Life Cycle	Generate the AI-FSM document	*REF_FSM procedure*	*REF_Ph0D0001_AI-FSM_Procedure*	*Ph0T0001_AI_FSM_template*
	V&V the AI-FSM document	*REF_Ph0D0001_AI-FSM_Procedure*	*REF_Ph0D0002_AI-FSM_Procedure_IR*	*Ph0T0001_AI_FSM_template_IR*
	Generate the AI_Document_List	*REF_Document_list*	*REF_Ph0D0003_AI_Document_List*	*Ph0T0002_AI_Document_List_template*
	V&V the AI_Document_List	*REF_Ph0D0003_AI_Document_List*	*REF_Ph0D0004_AI_Document_List_IR*	*Ph0T0002_AI_Document_List_template_IR*
	Generate AI version tracking	*REF_version_tracking*	*REF_Ph0D0005_AI_Version_Tracking*	*Ph0T0003_AI_Version_Tracking_template*
	V&V the AI version tracking	*REF_Ph0D0005_AI_Version_Tracking*	*REF_Ph0D0006_AI_Version_Tracking_IR*	*Ph0T0003_AI_Version_Tracking_template_IR*
	Generate AI organizational chart	*REF_organizational_chart*	*REF_Ph0D0007_AI_Organizational_Chart*	*Ph0T0004_AI_Organizational_Chart_template*
	V&V AI organizational chart	*REF_Ph0D0007_AI_Organizational_Chart*	*REF_Ph0D0008_AI_Organizational_Chart_IR*	*Ph0T0012_Organizational_chart_template_IR*
	Generate the AI log of tests	-	*REF_Ph0D0009_AI_Log_of_Tests*	*Ph0T0006_Log_of_Test_template*
	V&V the AI log of test	*REF_Ph0D0009_AI_Log_of_Test*	*REF_Ph0D0010_AI_Log_of_Tests_IR*	*Ph0T0006_Log_of_Test_template_IR*
	Generate the AI selection of tools	-	*REF_Ph0D0011_AI_Tools_Selection*	*Ph0T0010_Tools_selection_template*
	V&V the AI selection of tools	*REF_Ph0D0011_AI_ Tools_Selection*	*REF_Ph0D0012_AI_Tools_Selection_IR*	*Ph0T0010_Tools_selection_template_IR*
	Generate the AI traceability matrix	-	*REF_Ph0D0013_AI_Traceability_Matrix*	*Ph0T0011_Traceability_matrix_template*
	V&V the AI traceability matrix	*REF_Ph0D0013_AI_Traceability_Matrix*	*REF_Ph0D0014_AI_Traceability_Matrix_IR*	*Ph0T0011_Traceability_matrix_template_IR*

AI-FSM in-depth: Procedure (main)



Ph1 DL-Related Concept Specification

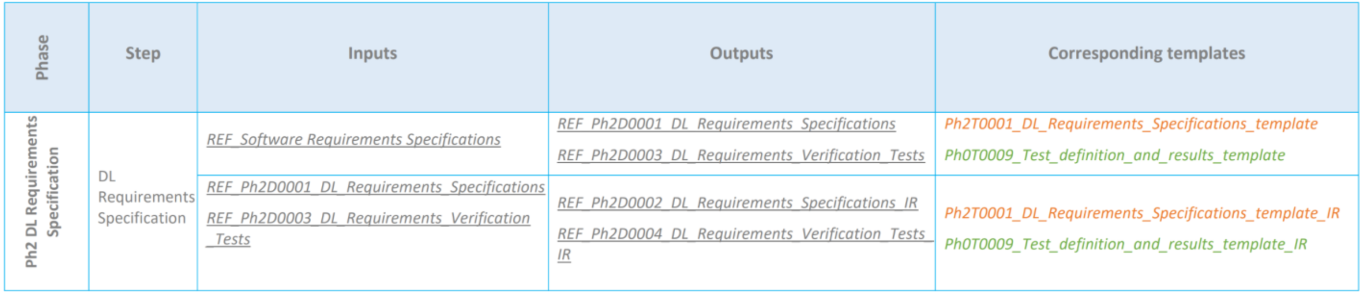
Table 2. Inputs and outputs of the System Concept Specification phase (Ph1)



Phase	Step	Inputs	Outputs	Corresponding templates
Ph1 System Concept Specification	ODD definition	*REF_System_Requirements_Specifications*	*REF_Ph1D0001_DL_Operational_Design_Domain*	*Ph1T0001_DL_Operational_Design_Domain_template*
	V&V the ODD	*REF_Ph1D0001_DL_Operational_Design_Domain*	*REF_Ph1D0002_DL_Operational_Design_Domain_IR*	*Ph1T0001_DL_Operational_Design_Domain_template_IR*
	Operational scenarios definition	*REF_System Requirements Specifications* *REF_Ph1D0001_DL_Operational_Design_Domain*	*REF_Ph1D0003_DL_Operational_Scenarios*	*Ph1T0002_DL_Operational_Scenarios_template*
	V&V the operational scenarios	*REF_Ph1D0003_DL_Operational_Scenarios*	*REF_Ph1D0004_DL_Operational_Scenarios_IR*	*Ph1T0002_DL_Operational_Scenarios_template_IR*

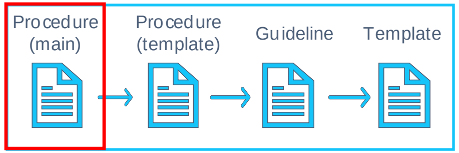
Ph2 DL Requirements Specification

Table 3. Inputs and outputs of the definition of the DL requirements (Ph2)



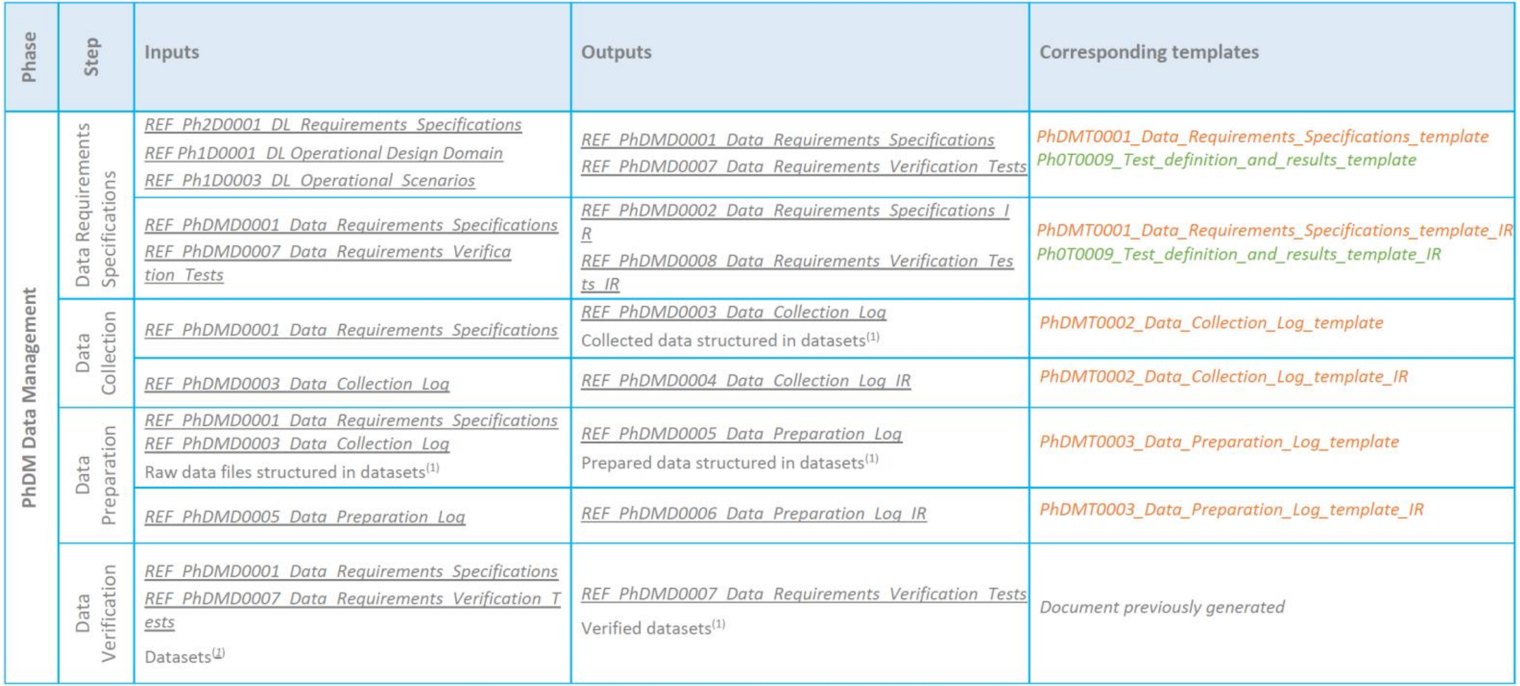
Phase	Step	Inputs	Outputs	Corresponding templates
Ph2 DL Requirements Specification	DL Requirements Specification	*REF_Software Requirements Specifications*	*REF_Ph2D0001_DL_Requirements_Specifications* *REF_Ph2D0003_DL_Requirements_Verification_Tests*	*Ph2T0001_DL_Requirements_Specifications_template* *Ph0T0009_Test_definition_and_results_template*
		REF_Ph2D0001_DL_Requirements_Specifications *REF_Ph2D0003_DL_Requirements_Verification_Tests*	*REF_Ph2D0002_DL_Requirements_Specifications_IR* *REF_Ph2D0004_DL_Requirements_Verification_Tests_IR*	*Ph2T0001_DL_Requirements_Specifications_template_IR* *Ph0T0009_Test_definition_and_results_template_IR*

AI-FSM in-depth: Procedure (main)



PhDM Data Management

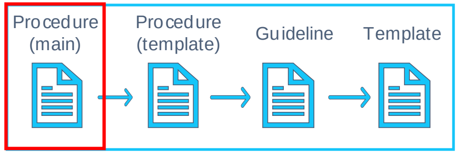
Table 4. Inputs and outputs of each step of the Data Management phase (related to Ph3, Ph4 and Ph5 of the traditional lifecycle)



Phase	Step	Inputs	Outputs	Corresponding templates
PhDM Data Management	Data Requirements Specifications	*REF Ph2D0001 DL Requirements Specifications* *REF Ph1D0001 DL Operational Design Domain* *REF Ph1D0003 DL Operational Scenarios*	*REF PhDMD0001 Data Requirements Specifications* *REF PhDMD0007 Data Requirements Verification Tests*	*PhDMT0001_Data_Requirements_Specifications_template* *Ph0T0009_Test_definition_and_results_template*
		REF PhDMD0001 Data Requirements Specifications *REF PhDMD0007 Data Requirements Verification Tests*	*REF PhDMD0002 Data Requirements Specifications IR* *REF PhDMD0008 Data Requirements Verification Tests IR*	*PhDMT0001_Data_Requirements_Specifications_template_IR* *Ph0T0009_Test_definition_and_results_template_IR*
	Data Collection	*REF PhDMD0001 Data Requirements Specifications*	*REF PhDMD0003 Data Collection Log* Collected data structured in datasets(1)	*PhDMT0002_Data_Collection_Log_template*
		REF PhDMD0003 Data Collection Log	*REF PhDMD0004 Data Collection Log IR*	*PhDMT0002_Data_Collection_Log_template_IR*
	Data Preparation	*REF PhDMD0001 Data Requirements Specifications* *REF PhDMD0003 Data Collection Log* Raw data files structured in datasets(1)	*REF PhDMD0005 Data Preparation Log* Prepared data structured in datasets(1)	*PhDMT0003_Data_Preparation_Log_template*
		REF PhDMD0005 Data Preparation Log	*REF PhDMD0006 Data Preparation Log IR*	*PhDMT0003_Data_Preparation_Log_template_IR*
	Data Verification	*REF PhDMD0001 Data Requirements Specifications* *REF PhDMD0007 Data Requirements Verification Tests* Datasets(1)	*REF PhDMD0007 Data Requirements Verification Tests* Verified datasets(1)	*Document previously generated*

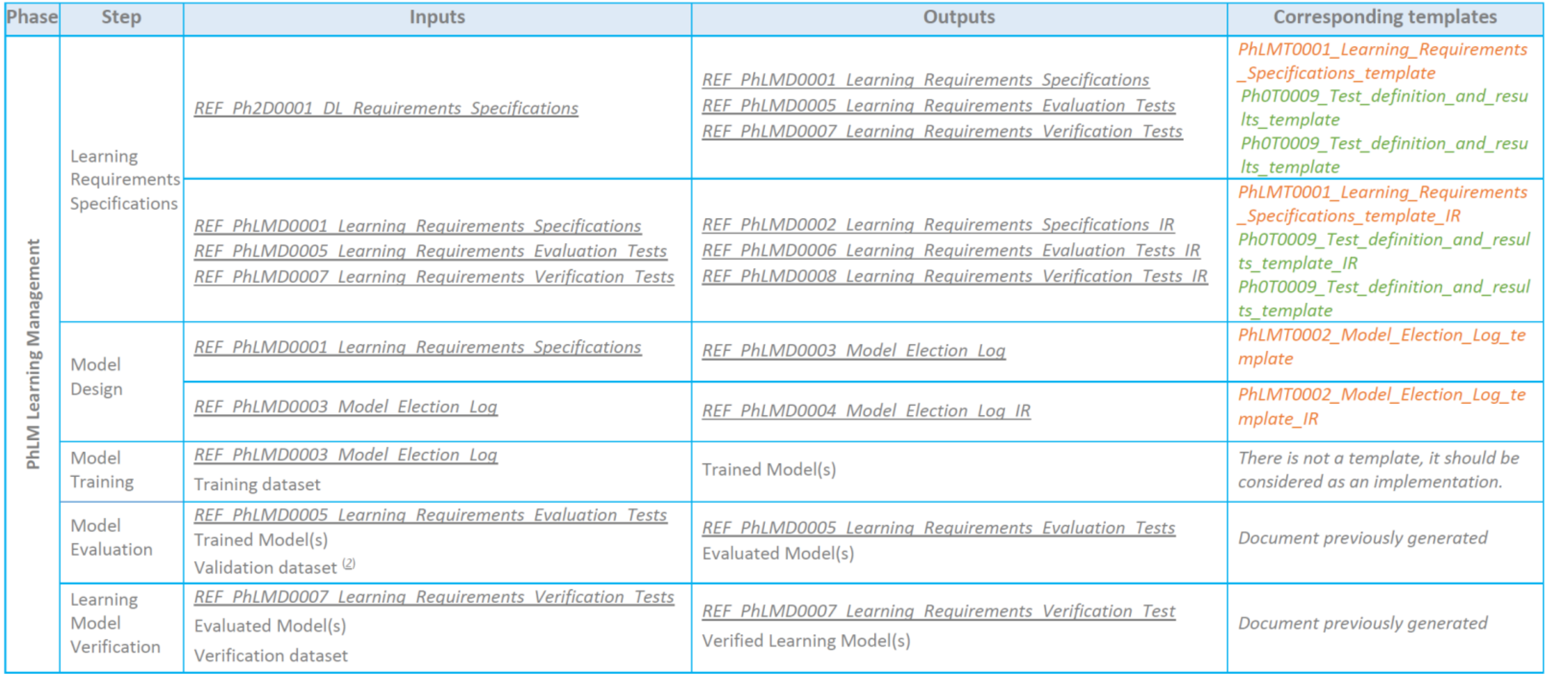
(*) Datasets include: i) Development (training and validation), ii) verification datasets.

AI-FSM in-depth: Procedure (main)



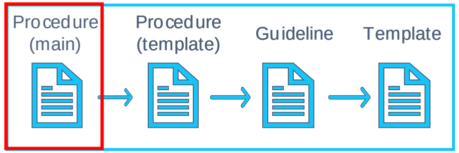
PhLM Learning Management

Table 5. Inputs and outputs of each step of the Learning Management phase (related to Ph3, Ph4 and Ph5 of the traditional lifecycle)



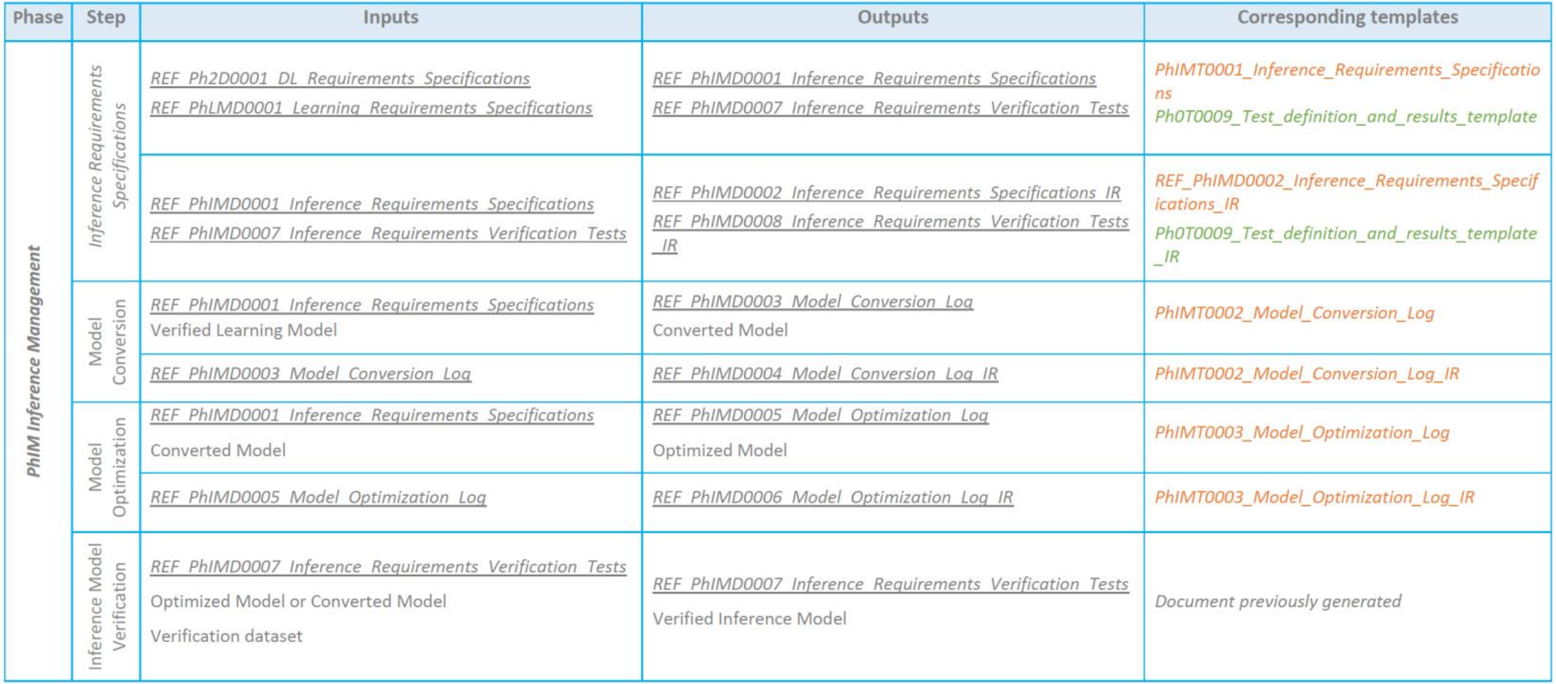
Phase	Step	Inputs	Outputs	Corresponding templates
PhLM Learning Management	Learning Requirements Specifications	REF_Ph2D0001_DL_Requirements_Specifications	REF_PhLMD0001_Learning_Requirements_Specifications REF_PhLMD0005_Learning_Requirements_Evaluation_Tests REF_PhLMD0007_Learning_Requirements_Verification_Tests	*PhLMT0001_Learning_Requirements_Specifications_template* *Ph0T0009_Test_definition_and_results_template* *Ph0T0009_Test_definition_and_results_template*
		REF_PhLMD0001_Learning_Requirements_Specifications REF_PhLMD0005_Learning_Requirements_Evaluation_Tests REF_PhLMD0007_Learning_Requirements_Verification_Tests	REF_PhLMD0002_Learning_Requirements_Specifications_IR REF_PhLMD0006_Learning_Requirements_Evaluation_Tests_IR REF_PhLMD0008_Learning_Requirements_Verification_Tests_IR	*PhLMT0001_Learning_Requirements_Specifications_template_IR* *Ph0T0009_Test_definition_and_results_template_IR* *Ph0T0009_Test_definition_and_results_template*
	Model Design	REF_PhLMD0001_Learning_Requirements_Specifications	REF_PhLMD0003_Model_Election_Log	*PhLMT0002_Model_Election_Log_template*
		REF_PhLMD0003_Model_Election_Log	REF_PhLMD0004_Model_Election_Log_IR	*PhLMT0002_Model_Election_Log_template_IR*
	Model Training	REF_PhLMD0003_Model_Election_Log Training dataset	Trained Model(s)	*There is not a template, it should be considered as an implementation.*
	Model Evaluation	REF_PhLMD0005_Learning_Requirements_Evaluation_Tests Trained Model(s) Validation dataset (2)	REF_PhLMD0005_Learning_Requirements_Evaluation_Tests Evaluated Model(s)	*Document previously generated*
	Learning Model Verification	REF_PhLMD0007_Learning_Requirements_Verification_Tests Evaluated Model(s) Verification dataset	REF_PhLMD0007_Learning_Requirements_Verification_Test Verified Learning Model(s)	*Document previously generated*

AI-FSM in-depth: Procedure (main)



PhIM Inference Management

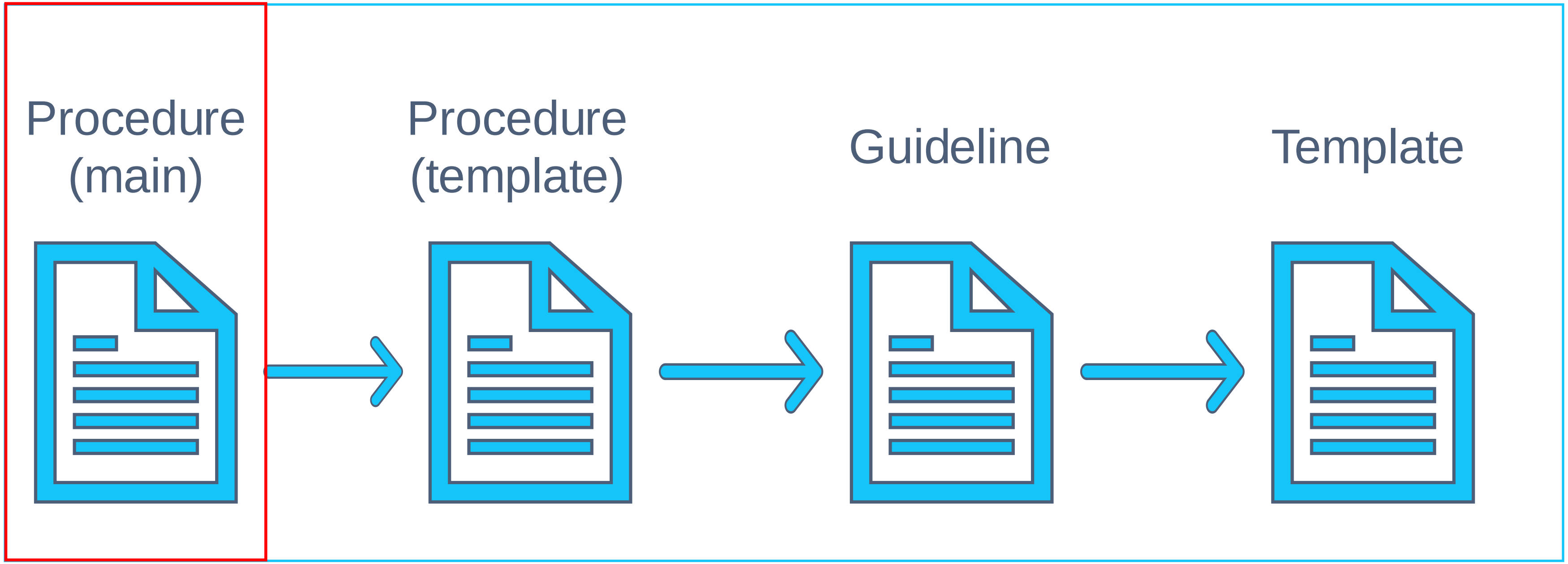
Table 6. Inputs and outputs of each step of the inference stage (related to Ph3, Ph4 and Ph5 of the traditional lifecycle)



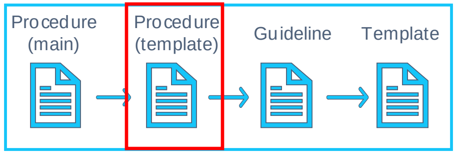
Phase	Step	Inputs	Outputs	Corresponding templates
PhIM Inference Management	Inference Requirements Specifications	REF_Ph2D0001_DL_Requirements_Specifications REF_PhLMD0001_Learning_Requirements_Specifications	REF_PhIMD0001_Inference_Requirements_Specifications REF_PhIMD0007_Inference_Requirements_Verification_Tests	PhIMT0001_Inference_Requirements_Specifications Ph0T0009_Test_definition_and_results_template
		REF_PhIMD0001_Inference_Requirements_Specifications REF_PhIMD0007_Inference_Requirements_Verification_Tests	REF_PhIMD0002_Inference_Requirements_Specifications_IR REF_PhIMD0008_Inference_Requirements_Verification_Tests_IR	REF_PhIMD0002_Inference_Requirements_Specifications_IR Ph0T0009_Test_definition_and_results_template_IR
	Model Conversion	REF_PhIMD0001_Inference_Requirements_Specifications Verified Learning Model	REF_PhIMD0003_Model_Conversion_Log Converted Model	PhIMT0002_Model_Conversion_Log
		REF_PhIMD0003_Model_Conversion_Log	REF_PhIMD0004_Model_Conversion_Log_IR	PhIMT0002_Model_Conversion_Log_IR
	Model Optimization	REF_PhIMD0001_Inference_Requirements_Specifications Converted Model	REF_PhIMD0005_Model_Optimization_Log Optimized Model	PhIMT0003_Model_Optimization_Log
		REF_PhIMD0005_Model_Optimization_Log	REF_PhIMD0006_Model_Optimization_Log_IR	PhIMT0003_Model_Optimization_Log_IR
	Inference Model Verification	REF_PhIMD0007_Inference_Requirements_Verification_Tests Optimized Model or Converted Model Verification dataset	REF_PhIMD0007_Inference_Requirements_Verification_Tests Verified Inference Model	Document previously generated

SAFEXPLAIN

AI-FSM in-depth: Procedure (main)



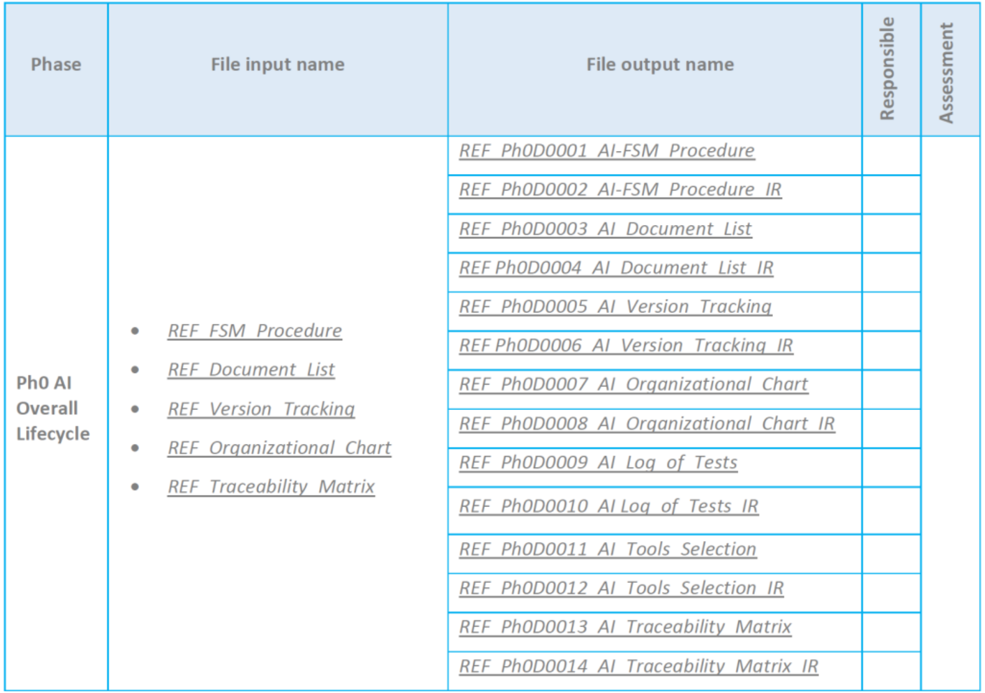
AI-FSM in-depth: Procedure (templ)



Overall Lifecycle – Phase 0 (Ph0)

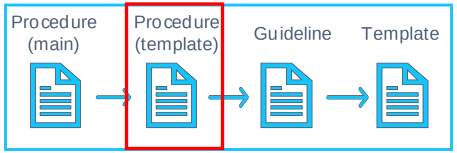
- Definition activities:
  - Update the AI_Document_List
  - Complete the AI_Version_Tracking
  - Fulfill the AI_Organizational_Chart
  - Fulfill the AI_Tools_selection
  - Complete the AI_Traceability_Matrix
- Verification and validation activities:
  - Conduct the IRs

Table 1: Overall lifecycle - Phase 0 summary



Phase	File input name	File output name	Responsible	Assessment
Ph0 AI Overall Lifecycle	• REF_FSM_Procedure • REF_Document_List • REF_Version_Tracking • REF_Organizational_Chart • REF_Traceability_Matrix	REF_Ph0D0001_AI-FSM_Procedure		
		REF_Ph0D0002_AI-FSM_Procedure_IR		
		REF_Ph0D0003_AI_Document_List		
		REF Ph0D0004_AI_Document_List_IR		
		REF_Ph0D0005_AI_Version_Tracking		
		REF Ph0D0006_AI_Version_Tracking_IR		
		REF_Ph0D0007_AI_Organizational_Chart		
		REF_Ph0D0008_AI_Organizational_Chart_IR		
		REF_Ph0D0009_AI_Log_of_Tests		
		REF_Ph0D0010_AI Log_of_Tests_IR		
		REF_Ph0D0011_AI_Tools_Selection		
		REF_Ph0D0012_AI_Tools_Selection_IR		
		REF_Ph0D0013_AI_Traceability_Matrix		
		REF_Ph0D0014_AI_Traceability_Matrix_IR		

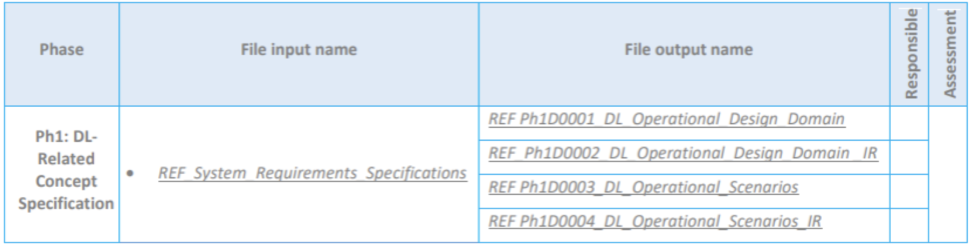
AI-FSM in-depth: Procedure (templ)



DL-Related Concept Specification– Phase 1 (Ph1)

- Definition activities:
  - Complete the DL_Operational_Design_Domain
  - Complete the DL_Operational_Scenarios
- Verification and validation activities:
  - Conduct the IRs

Table 2: DL-Related Concept Specification - Phase 1 summary

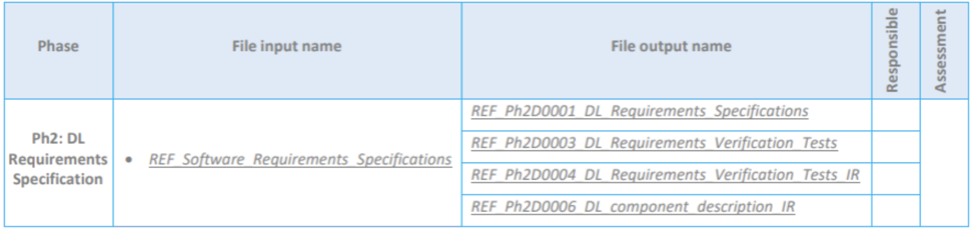


Phase	File input name	File output name	Responsible	Assessment
Ph1: DL-Related Concept Specification	• *REF_System_Requirements_Specifications*	*REF_Ph1D0001_DL_Operational_Design_Domain*		
		REF_Ph1D0002_DL_Operational_Design_Domain_IR		
		REF_Ph1D0003_DL_Operational_Scenarios		
		REF_Ph1D0004_DL_Operational_Scenarios_IR		

DL Requirements Specification– Phase 2 (Ph2)

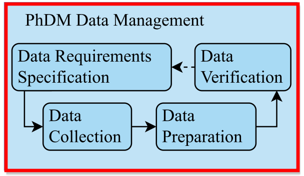
- Definition activities:
  - Complete the DL_Requirements_Specification
- Verification and validation activities:
  - Conduct the IRs

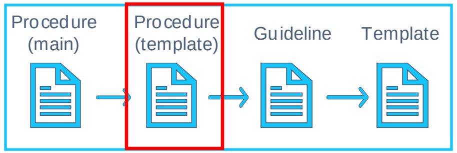
Table 3: DL Requirements Specification - Phase 2 summary



Phase	File input name	File output name	Responsible	Assessment
Ph2: DL Requirements Specification	• *REF_Software_Requirements_Specifications*	*REF_Ph2D0001_DL_Requirements_Specifications*		
		REF_Ph2D0003_DL_Requirements_Verification_Tests		
		REF_Ph2D0004_DL_Requirements_Verification_Tests_IR		
		REF_Ph2D0006_DL_component_description_IR		

AI-FSM procedure template

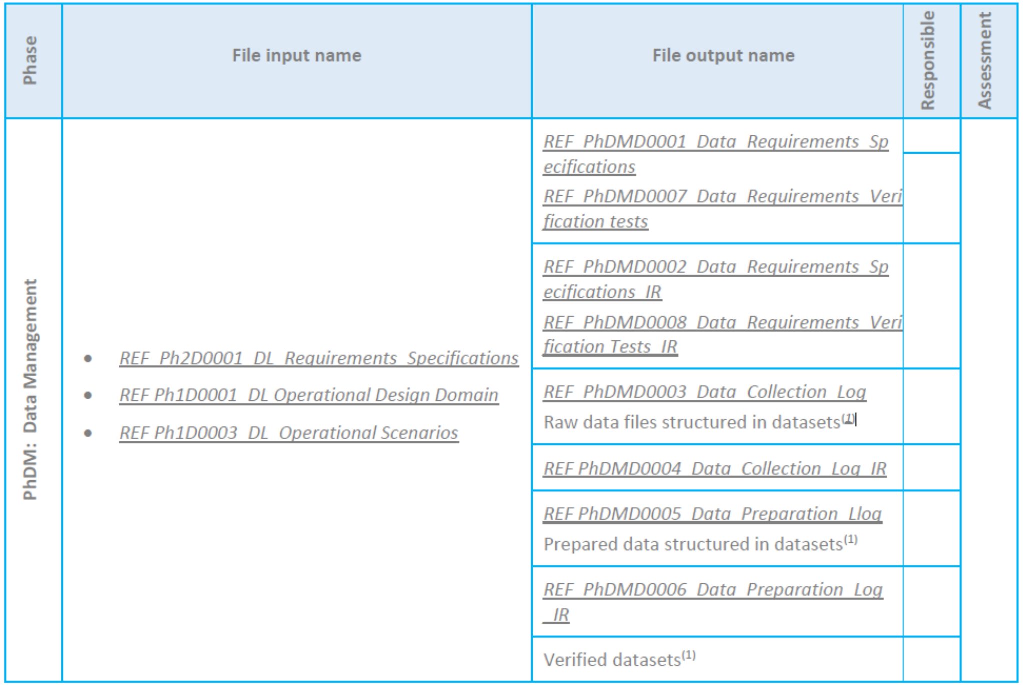




Data Management – Phase DM (PhDM)

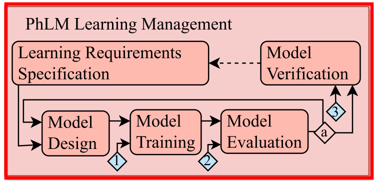
- Definition activities:
  - Collect data requirements
  - Define data req. verification tests
  - Data Collection
  - Data Preparation
- Verification & validation:
  - Implement data req. verification tests
  - Conduct the IRs
- Collect the tests in AI Log Test file
- Update the state of AI Document List

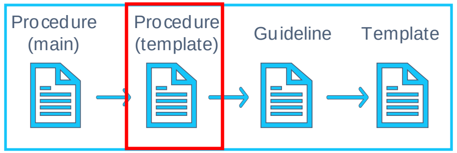
Table 4: Data Management - PhDM summary (related to Ph3, Ph4 and Ph5 of the traditional lifecycle)



Phase	File input name	File output name	Responsible	Assessment
PhDM: Data Management	• REF_Ph2D0001_DL_Requirements_Specifications • REF Ph1D0001_DL Operational Design Domain • REF Ph1D0003_DL_Operational Scenarios	REF_PhDMD0001_Data_Requirements_Specifications REF_PhDMD0007_Data_Requirements_Verification tests		
		REF_PhDMD0002_Data_Requirements_Specifications_IR REF_PhDMD0008_Data_Requirements_Verification Tests_IR		
		REF_PhDMD0003_Data_Collection_Log Raw data files structured in datasets(1)		
		REF PhDMD0004_Data_Collection_Log_IR		
		REF PhDMD0005_Data_Preparation_Llog Prepared data structured in datasets(1)		
		REF_PhDMD0006_Data_Preparation_Log_IR		
		Verified datasets(1)		

AI-FSM procedure template

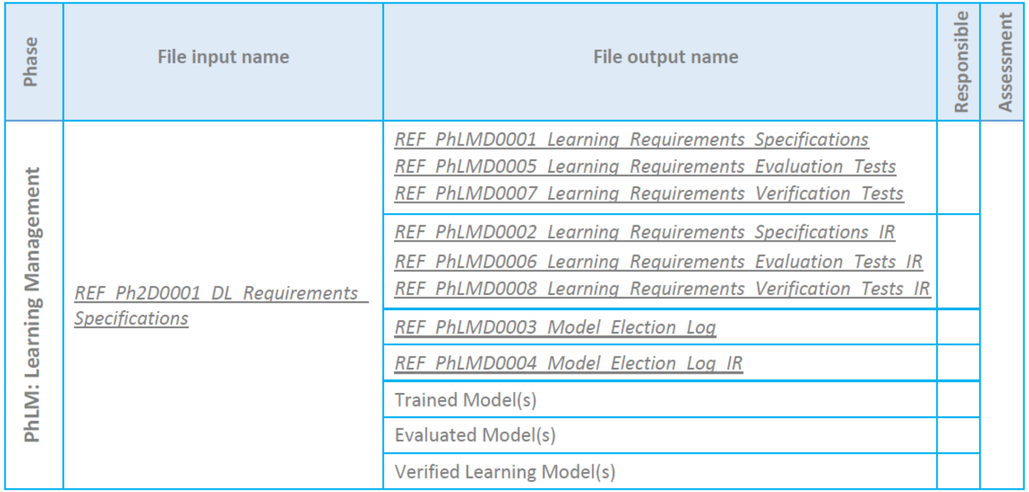




Learning Management – Phase LM (PhLM)

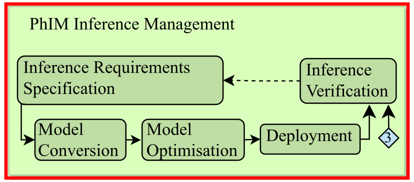
- Definition activities:
  - Collect learning requirements
  - Define learning req. evaluation tests & Learning req. verification tests
  - Design, train and evaluate the model
- Verification & validation:
  - Implement:
    - Learning req. evaluation tests
    - Learning req. verification tests
  - Conduct the IRs
  - Collect the tests in AI Log Test file
  - Update the state of AI Document List

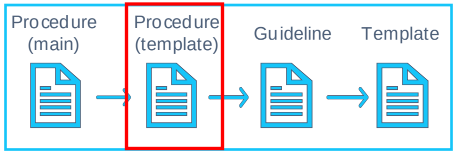
Table 5: Learning Management - PhLM summary (related to Ph3, Ph4 and Ph5 of the traditional lifecycle)



Phase	File input name	File output name	Responsible	Assessment
PhLM: Learning Management	*REF_Ph2D0001_DL_Requirements_Specifications*	*REF_PhLMD0001_Learning_Requirements_Specifications* *REF_PhLMD0005_Learning_Requirements_Evaluation_Tests* *REF_PhLMD0007_Learning_Requirements_Verification_Tests*		
		REF_PhLMD0002_Learning_Requirements_Specifications_IR *REF_PhLMD0006_Learning_Requirements_Evaluation_Tests_IR* *REF_PhLMD0008_Learning_Requirements_Verification_Tests_IR*		
		REF_PhLMD0003_Model_Election_Log		
		REF_PhLMD0004_Model_Election_Log_IR		
		Trained Model(s)		
		Evaluated Model(s)		
		Verified Learning Model(s)		

AI-FSM procedure template

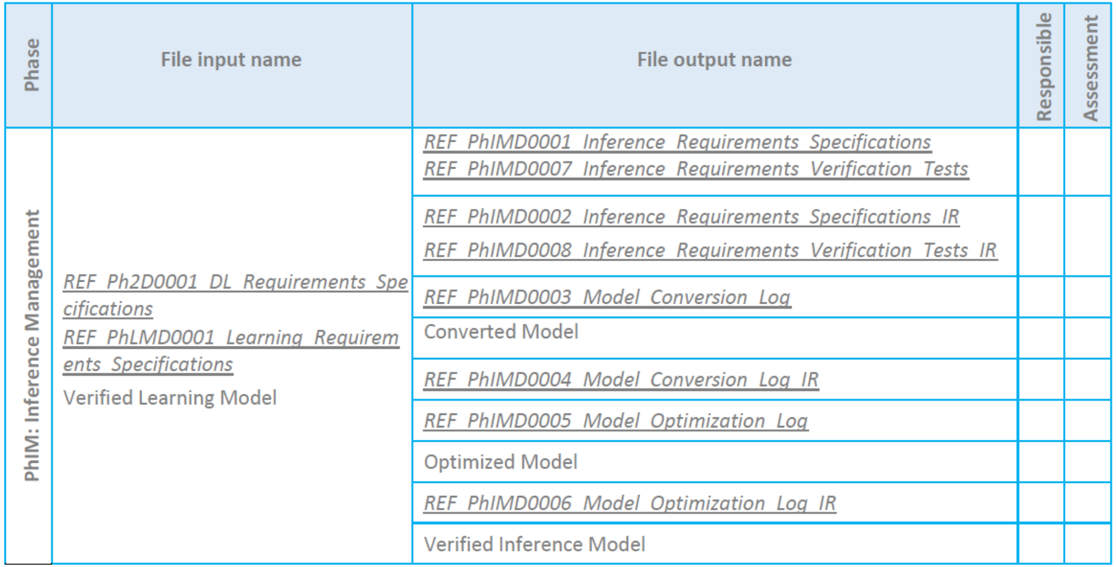




Inference Management – Phase IM (PhIM)

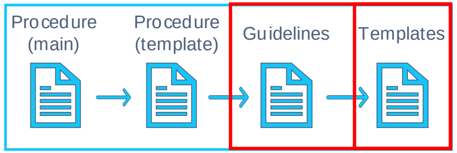
- Definition activities:
  - Collect inf. requirements
  - Define inf. req. verification tests
  - Convert the model
  - Optimise the model
- Verification & validation:
  - Implement inf. req. verification tests
  - Conduct the IRs
  - Collect the tests in AI Log Test file
  - Update the state of AI Document List

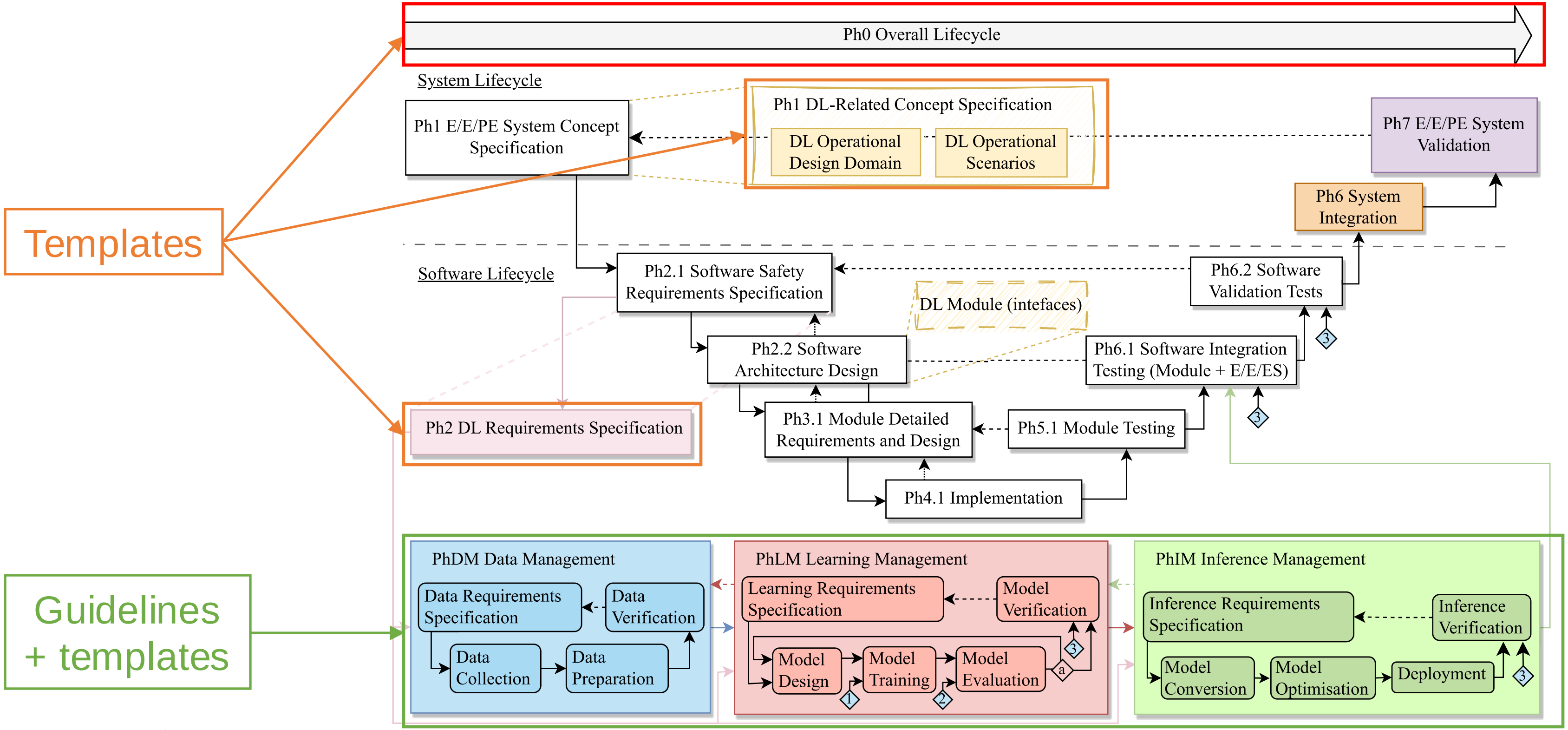
Table 6: Inference Management – PhIM summary (related to Ph3, Ph4 and Ph5 of the traditional lifecycle)



Phase	File input name	File output name	Responsible	Assessment
PhIM: Inference Management	*REF_Ph2D0001_DL_Requirements_Specifications* *REF_PhLMD0001_Learning_Requirements_Specifications* Verified Learning Model	*REF_PhIMD0001_Inference_Requirements_Specifications* *REF_PhIMD0007_Inference_Requirements_Verification_Tests*		
		REF_PhIMD0002_Inference_Requirements_Specifications_IR *REF_PhIMD0008_Inference_Requirements_Verification_Tests_IR*		
		REF_PhIMD0003_Model_Conversion_Log		
		Converted Model		
		REF_PhIMD0004_Model_Conversion_Log_IR		
		REF_PhIMD0005_Model_Optimization_Log		
		Optimized Model		
		REF_PhIMD0006_Model_Optimization_Log_IR		
		Verified Inference Model		

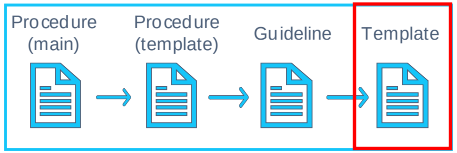
AI-FSM in-depth

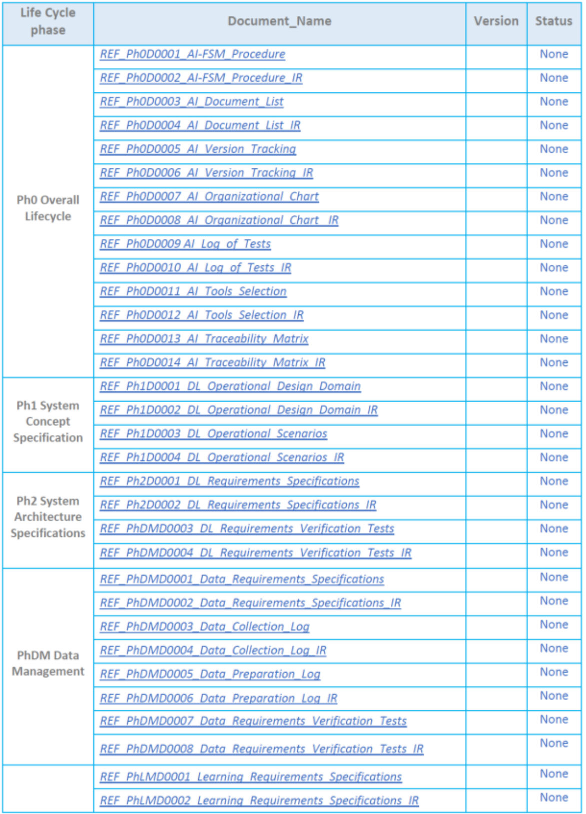


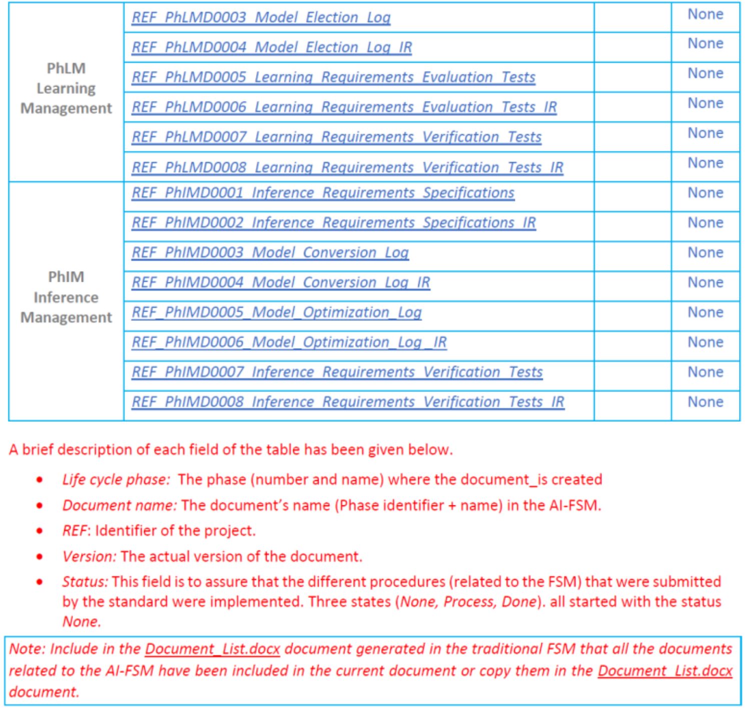


Templates
Guidelines + templates
Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
3
3
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
3
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
3

AI-FSM in-depth: AI Document List







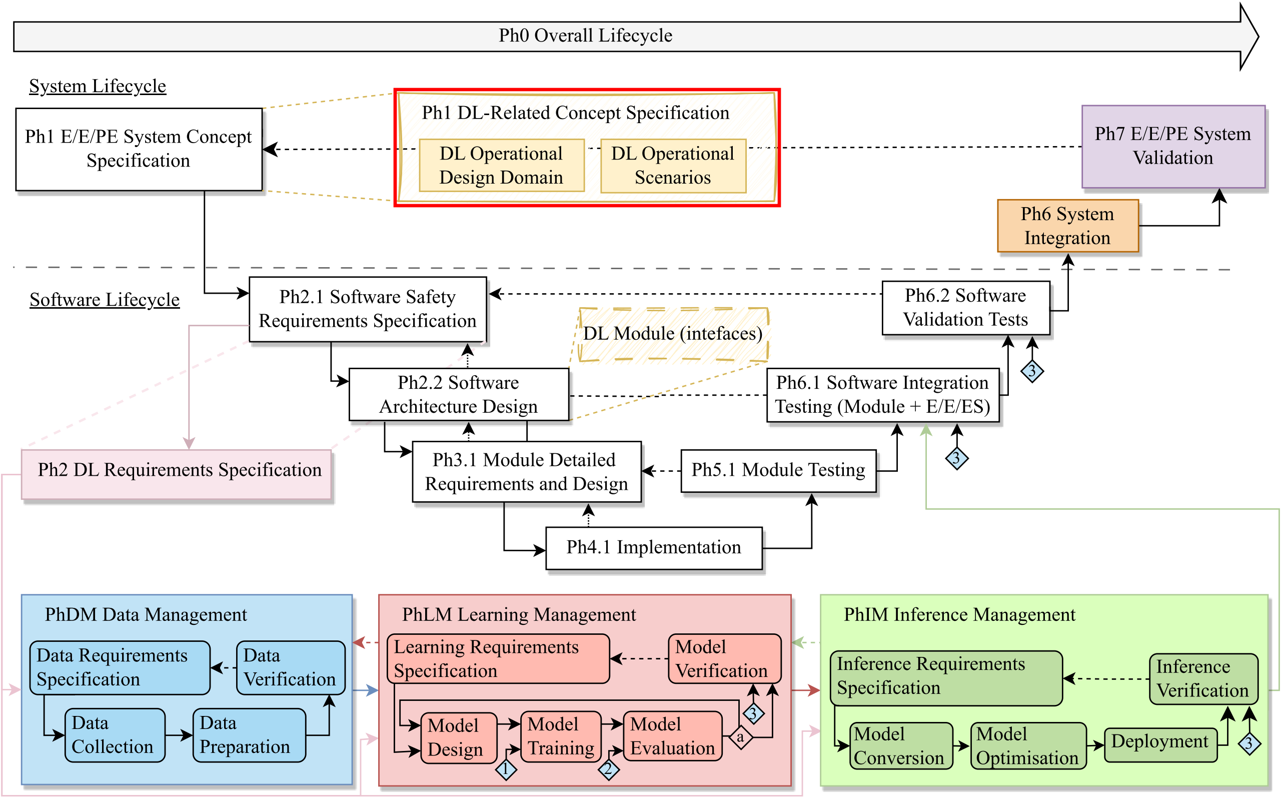
Life Cycle phase	Document_Name	Version	Status
Ph0 Overall Lifecycle	REF_Ph0D0001_AI-FSM_Procedure		None
	REF_Ph0D0002_AI-FSM_Procedure_IR		None
	REF_Ph0D0003_AI_Document_List		None
	REF_Ph0D0004_AI_Document_List_IR		None
	REF_Ph0D0005_AI_Version_Tracking		None
	REF_Ph0D0006_AI_Version_Tracking_IR		None
	REF_Ph0D0007_AI_Organizational_Chart		None
	REF_Ph0D0008_AI_Organizational_Chart_IR		None
	REF_Ph0D0009 AI_Log_of_Tests		None
	REF_Ph0D0010_AI_Log_of_Tests_IR		None
	REF_Ph0D0011_AI_Tools_Selection		None
	REF_Ph0D0012_AI_Tools_Selection_IR		None
	REF_Ph0D0013_AI_Traceability_Matrix		None
	REF_Ph0D0014_AI_Traceability_Matrix_IR		None
Ph1 System Concept Specification	REF_Ph1D0001_DL_Operational_Design_Domain		None
	REF_Ph1D0002_DL_Operational_Design_Domain_IR		None
	REF_Ph1D0003_DL_Operational_Scenarios		None
	REF_Ph1D0004_DL_Operational_Scenarios_IR		None
Ph2 System Architecture Specifications	REF_Ph2D0001_DL_Requirements_Specifications		None
	REF_Ph2D0002_DL_Requirements_Specifications_IR		None
	REF_PhDMD0003_DL_Requirements_Verification_Tests		None
	REF_PhDMD0004_DL_Requirements_Verification_Tests_IR		None
PhDM Data Management	REF_PhDMD0001_Data_Requirements_Specifications		None
	REF_PhDMD0002_Data_Requirements_Specifications_IR		None
	REF_PhDMD0003_Data_Collection_Log		None
	REF_PhDMD0004_Data_Collection_Log_IR		None
	REF_PhDMD0005_Data_Preparation_Log		None
	REF_PhDMD0006_Data_Preparation_Log_IR		None
	REF_PhDMD0007_Data_Requirements_Verification_Tests		None
	REF_PhDMD0008_Data_Requirements_Verification_Tests_IR		None
PhLM Learning Management	REF_PhLMD0001_Learning_Requirements_Specifications		None
	REF_PhLMD0002_Learning_Requirements_Specifications_IR		None
	REF_PhLMD0003_Model_Election_Log		None
	REF_PhLMD0004_Model_Election_Log_IR		None
	REF_PhLMD0005_Learning_Requirements_Evaluation_Tests		None
	REF_PhLMD0006_Learning_Requirements_Evaluation_Tests_IR		None
	REF_PhLMD0007_Learning_Requirements_Verification_Tests		None
	REF_PhLMD0008_Learning_Requirements_Verification_Tests_IR		None
PhIM Inference Management	REF_PhIMD0001_Inference_Requirements_Specifications		None
	REF_PhIMD0002_Inference_Requirements_Specifications_IR		None
	REF_PhIMD0003_Model_Conversion_Log		None
	REF_PhIMD0004_Model_Conversion_Log_IR		None
	REF_PhIMD0005_Model_Optimization_Log		None
	REF_PhIMD0006_Model_Optimization_Log_IR		None
	REF_PhIMD0007_Inference_Requirements_Verification_Tests		None
	REF_PhIMD0008_Inference_Requirements_Verification_Tests_IR		None

A brief description of each field of the table has been given below.

- *Life cycle phase:* The phase (number and name) where the document_is created
- *Document name:* The document's name (Phase identifier + name) in the AI-FSM.
- *REF*: Identifier of the project.
- *Version:* The actual version of the document.
- *Status:* This field is to assure that the different procedures (related to the FSM) that were submitted by the standard were implemented. Three states (*None, Process, Done*). all started with the status *None.*

Note: Include in the Document_List.docx document generated in the traditional FSM that all the documents related to the AI-FSM have been included in the current document or copy them in the Document_List.docx document.

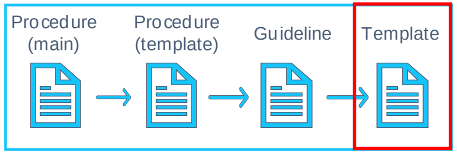
AI-FSM in-depth


Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
3
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
3
Ph4.1 Implementation
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
3
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
3

SAFEXPLAIN

Virtual – 2024/07/03

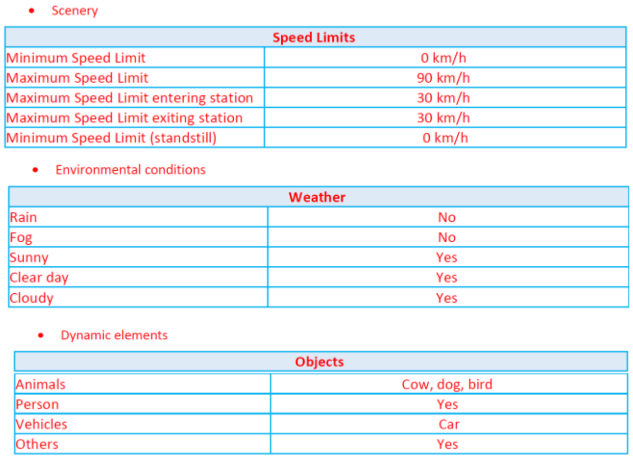
Ph1 DL-related Concept Specifications



REF_Ph1D0001_DL_Operational_Design_Domain.docx

- Purpose: Operating conditions under which a given overall system or feature is specifically designed to function (e.g., environmental restrictions, certain scenery characteristics, and dynamic elements surrounding the system).
  - Ph1T0001_DL_Operational_Design_Domain_template.docx
    - Categorization to describe the ODD, but customizable.

1) Scenery
   a) Physical infrastructure
   b) Operational constraints
   c) Zones
2) Environmental conditions
   a) Weather
   b) Particulate
   c) Illumination
   d) Connectivity
3) Dynamic elements
   a) Object types
   b) Object characteristics



- Scenery

Speed Limits	
Minimum Speed Limit	0 km/h
Maximum Speed Limit	90 km/h
Maximum Speed Limit entering station	30 km/h
Maximum Speed Limit exiting station	30 km/h
Minimum Speed Limit (standstill)	0 km/h

- Environmental conditions

Weather	
Rain	No
Fog	No
Sunny	Yes
Clear day	Yes
Cloudy	Yes

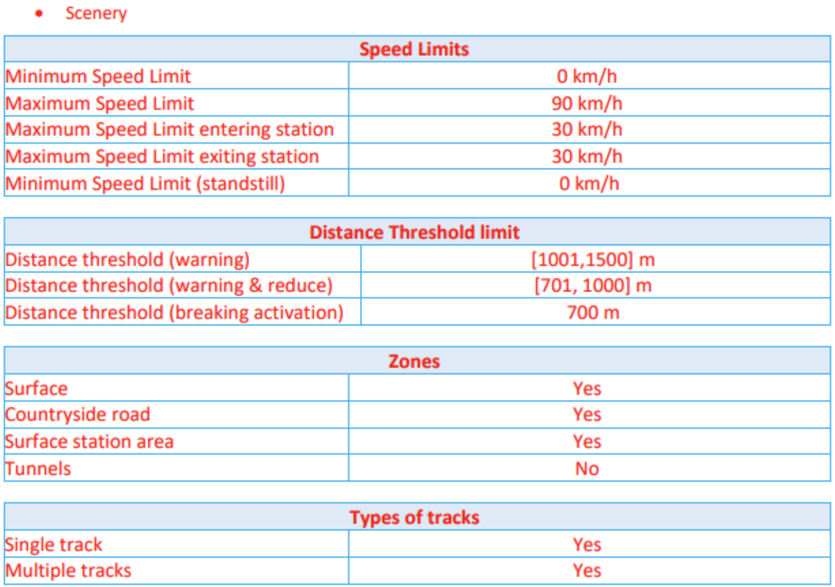
- Dynamic elements

Objects	
Animals	Cow, dog, bird
Person	Yes
Vehicles	Car
Others	Yes

Ph1 DL-related Concept Specifications

Railway application domain example

REF_Ph1D0001_DL_Operational_Design_Domain.docx



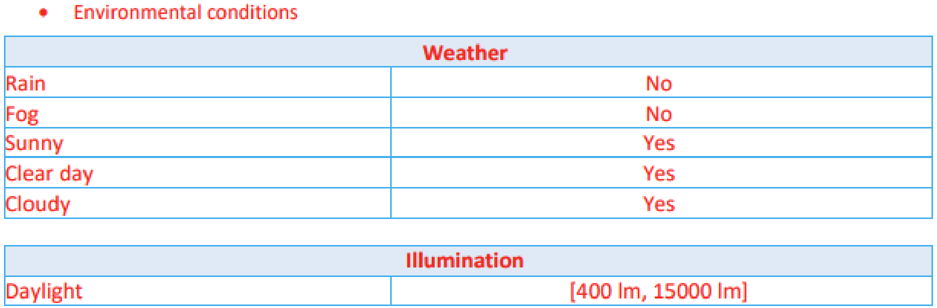
- Scenery

Speed Limits	
Minimum Speed Limit	0 km/h
Maximum Speed Limit	90 km/h
Maximum Speed Limit entering station	30 km/h
Maximum Speed Limit exiting station	30 km/h
Minimum Speed Limit (standstill)	0 km/h

Distance Threshold limit	
Distance threshold (warning)	[1001,1500] m
Distance threshold (warning & reduce)	[701, 1000] m
Distance threshold (breaking activation)	700 m

Zones	
Surface	Yes
Countryside road	Yes
Surface station area	Yes
Tunnels	No

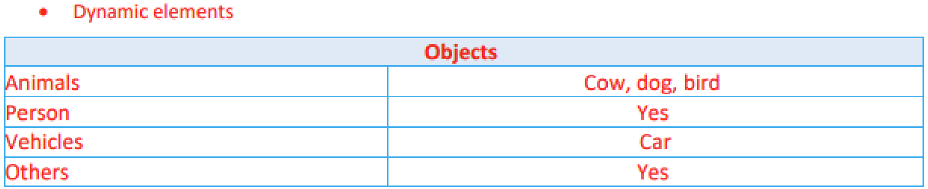
Types of tracks	
Single track	Yes
Multiple tracks	Yes



- Environmental conditions

Weather	
Rain	No
Fog	No
Sunny	Yes
Clear day	Yes
Cloudy	Yes

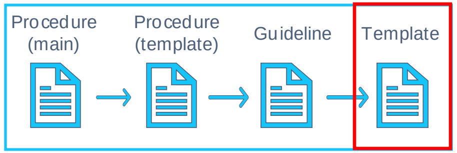
Illumination	
Daylight	[400 lm, 15000 lm]



- Dynamic elements

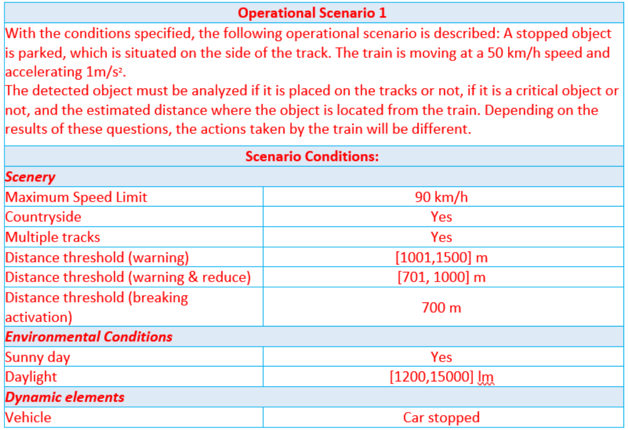
Objects	
Animals	Cow, dog, bird
Person	Yes
Vehicles	Car
Others	Yes

Ph1 DL-related Concept Specifications



REF_Ph1D0002_DL_Operational_Scenarios.docx

- Objective: Specify operations, scenarios and environmental conditions for the system in which it has to function according to the specification within ODD. It must include standard situations but also challenging environments and cornerstone situations.
- Ph1T0002_DL_Operational_Scenarios_Template.docx
  - Gathers information of the specific scenario conditions

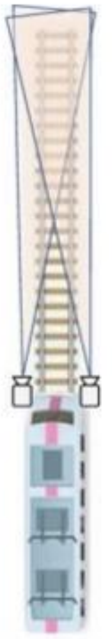


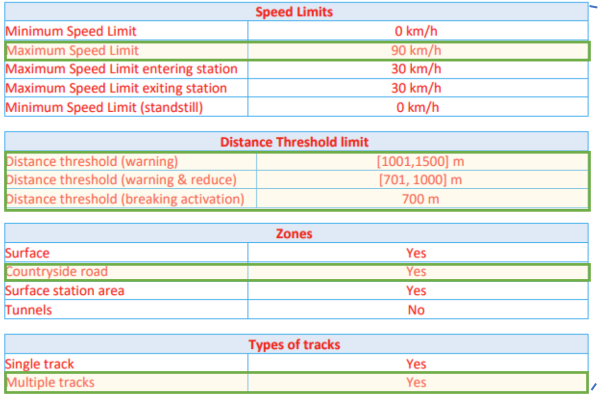
Operational Scenario 1	
With the conditions specified, the following operational scenario is described: A stopped object is parked, which is situated on the side of the track. The train is moving at a 50 km/h speed and accelerating $1m/s^2$. The detected object must be analyzed if it is placed on the tracks or not, if it is a critical object or not, and the estimated distance where the object is located from the train. Depending on the results of these questions, the actions taken by the train will be different.	
Scenario Conditions:	
Scenery	
Maximum Speed Limit	90 km/h
Countryside	Yes
Multiple tracks	Yes
Distance threshold (warning)	[1001,1500] m
Distance threshold (warning & reduce)	[701, 1000] m
Distance threshold (breaking activation)	700 m
Environmental Conditions	
Sunny day	Yes
Daylight	[1200,15000] lm
Dynamic elements	
Vehicle	Car stopped

Ph1 DL-related Concept Specifications

Railway application domain example

REF_Ph1D0002_DL_Operational_Scenarios.docx



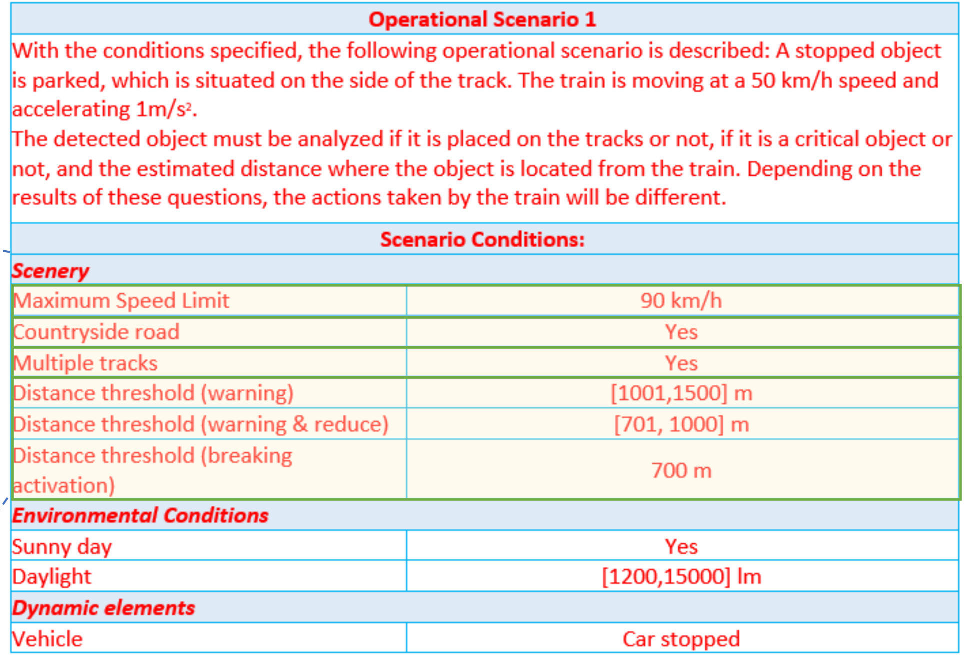


Speed Limits	
Minimum Speed Limit	0 km/h
Maximum Speed Limit	90 km/h
Maximum Speed Limit entering station	30 km/h
Maximum Speed Limit exiting station	30 km/h
Minimum Speed Limit (standstill)	0 km/h

Distance Threshold limit	
Distance threshold (warning)	[1001,1500] m
Distance threshold (warning & reduce)	[701, 1000] m
Distance threshold (breaking activation)	700 m

Zones	
Surface	Yes
Countryside road	Yes
Surface station area	Yes
Tunnels	No

Types of tracks	
Single track	Yes
Multiple tracks	Yes



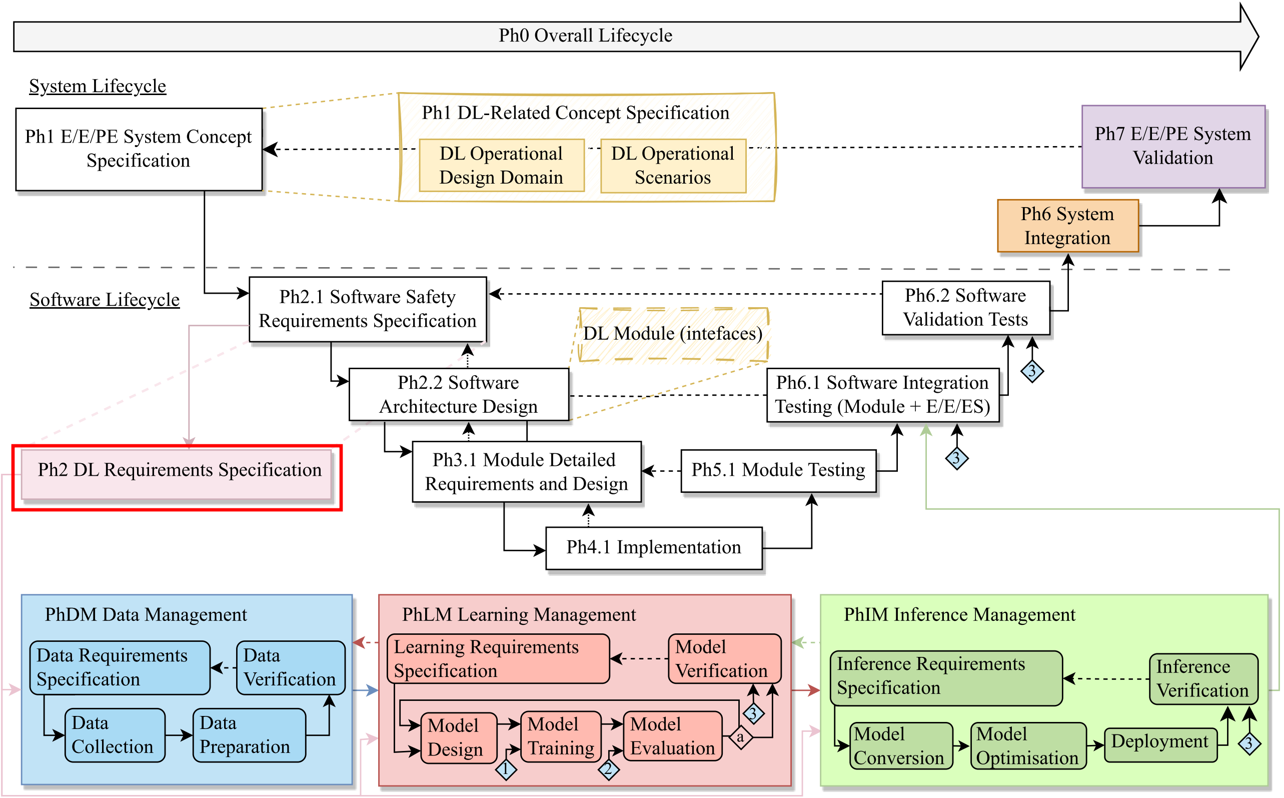
Operational Scenario 1

With the conditions specified, the following operational scenario is described: A stopped object is parked, which is situated on the side of the track. The train is moving at a 50 km/h speed and accelerating 1m/s².

The detected object must be analyzed if it is placed on the tracks or not, if it is a critical object or not, and the estimated distance where the object is located from the train. Depending on the results of these questions, the actions taken by the train will be different.

Scenario Conditions:	
Scenery	
Maximum Speed Limit	90 km/h
Countryside road	Yes
Multiple tracks	Yes
Distance threshold (warning)	[1001,1500] m
Distance threshold (warning & reduce)	[701, 1000] m
Distance threshold (breaking activation)	700 m
Environmental Conditions	
Sunny day	Yes
Daylight	[1200,15000] lm
Dynamic elements	
Vehicle	Car stopped

AI-FSM in-depth



Ph0 Overall Lifecycle

System Lifecycle

Ph1 E/E/PE System Concept Specification

Ph1 DL-Related Concept Specification

DL Operational Design Domain

DL Operational Scenarios

Ph7 E/E/PE System Validation

Ph6 System Integration

Software Lifecycle

Ph2.1 Software Safety Requirements Specification

Ph6.2 Software Validation Tests

DL Module (intefaces)

Ph2.2 Software Architecture Design

Ph6.1 Software Integration Testing (Module + E/E/ES)

Ph2 DL Requirements Specification

Ph3.1 Module Detailed Requirements and Design

Ph5.1 Module Testing

Ph4.1 Implementation

PhDM Data Management

Data Requirements Specification

Data Verification

Data Collection

Data Preparation

PhLM Learning Management

Learning Requirements Specification

Model Verification

Model Design

Model Training

Model Evaluation

PhIM Inference Management

Inference Requirements Specification

Inference Verification

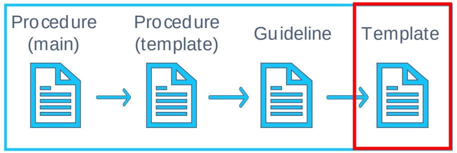
Model Conversion

Model Optimisation

Deployment

Virtual – 2024/07/03

Ph2 DL Requirements Specification

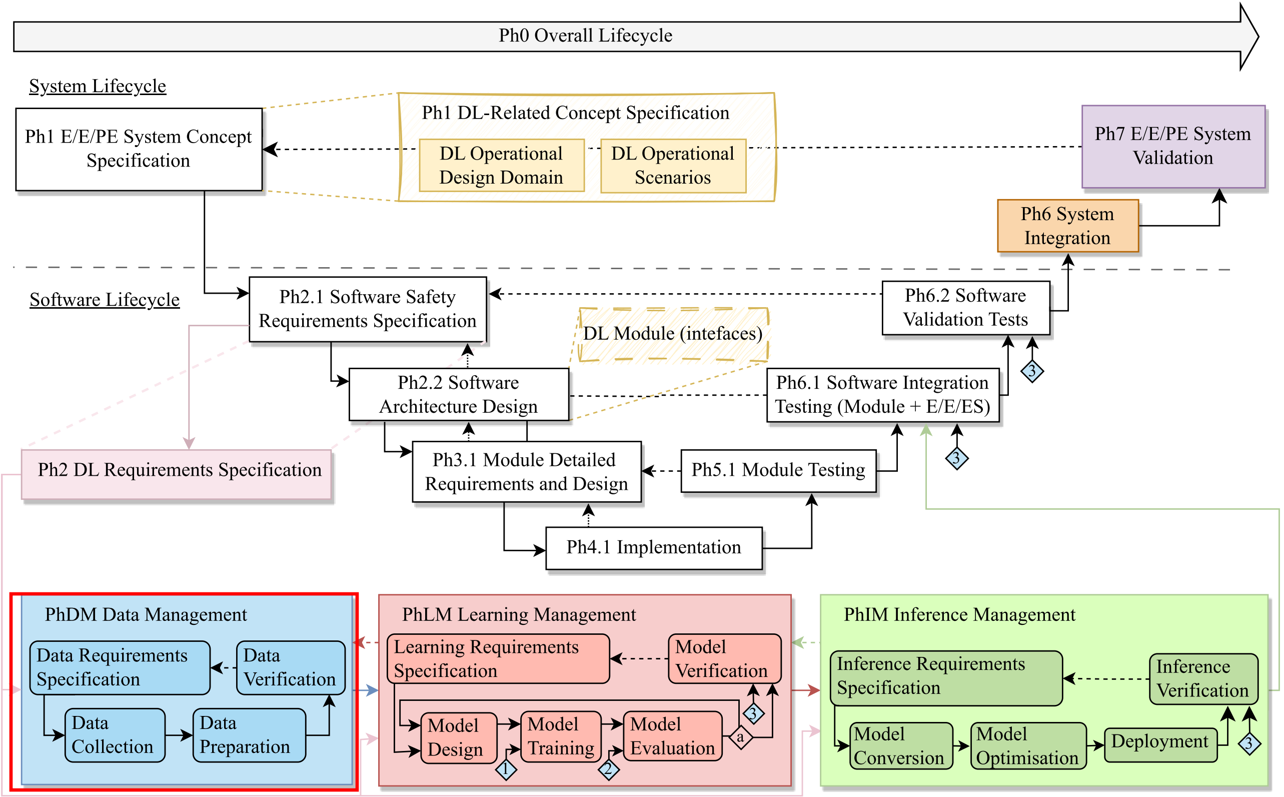

Procedure (main)
Procedure (template)
Guideline
Template

REF_Ph3D0001_DL_requirements_specification.docx

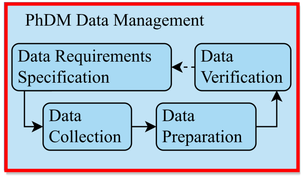
Objective: Allocate the SW Reqs. Specification to the DL constituent and refine them.

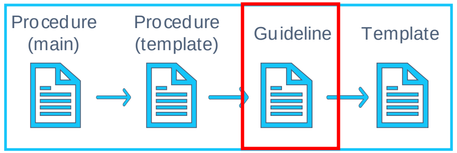
- They shall be: **Unambiguous, clear, concise, verifiable, traceable, complete and feasible.**
- The following listed items shall be considered during the definition:
  - Functional:
    - Safety functions
    - Non-safety functions
  - Non-Functional – Characterizing properties
  - Software systematic capabilities
  - Operation Modes
  - Interfaces
  - Diagnostics

AI-FSM in-depth


Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
Ph4.1 Implementation
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
1
2
3
a

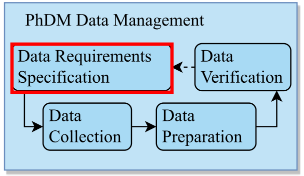
Data Management guideline

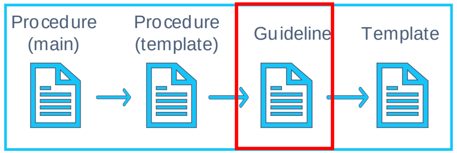




- The objective of this phase is the generation of:
  - Development dataset:
    - Training dataset.
    - Validation datasets.
  - Verification dataset.
- As previously mentioned, the following document should be generated:
  - REF_PhDMD0001_Data_Requirements_Specifications.docx. (+IR)
  - REF_PhDMD0003_Data_Collection_Log.docx. (+IR)
  - REF_PhDMD0005_Data_Preparation_Log.docx. (+IR)
  - REF_PhDMD0007_Data_Requirements_Verification_Tests. (+IR)
- All the documents should be stored in the “PhDM Data Management” folder.

Data Management guideline

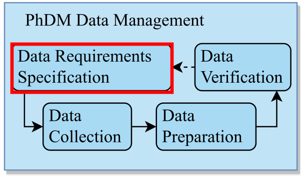


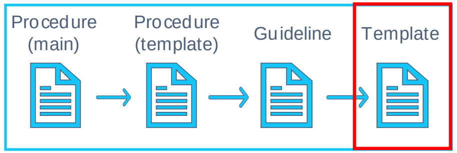


Data Requirements Specification step

- Define the data requirements:
  - Allocate DL requirements specification associated with the data requirement specification.
  - Refine those requirements and define additional ones.
  - Define the data notation policy.
  - This guideline proposes to decompose the requirements into two subcategories:
    - Dataset requirements specification.
    - Data requirements specification.
- Define the mechanisms or tests that must be carried out to check that the data meets the associated data requirements specification.
- Conduct the IRs

Data Requirements Specification template



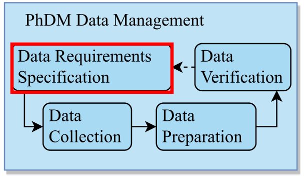


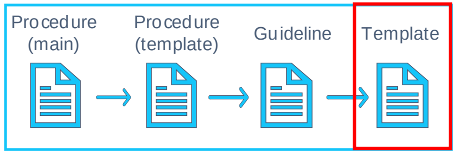
REF_PhDMD0001_Data_Requirements_Specification.docx

It proposes to decompose these reqs. to the following subgroups:

- Data reqs. specification (format, data characteristics)
- Dataset reqs. Specification
  - Completeness
  - Representativeness
  - Volume
  - Data origin
  - Degree of differentiation between the datasets.

Data Requirements Specification template

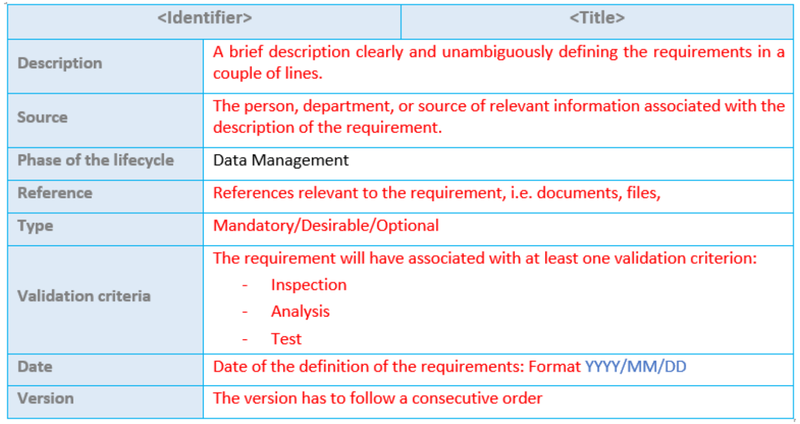




REF_PhDMD0001_Data_Requirements_Specification.docx

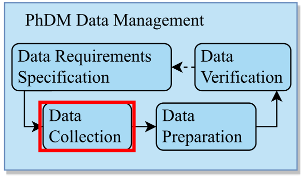
It includes:

- Example of definition of the filename policy: <Data_Procedence>_<ID_number>.<Data_Format>
  - <Data_Procedence>: Sensors (SENS), Synthetically generated data (SYNT), normalized data (NORM) …
  - <ID_number>: Identifier starting from 0 to N. Each <Data_Procedence> group starts at 0.
  - <Data_format>: I.e., resolution (1920x1080)
- Requirement Specification Table (common to all the phases)



<Identifier>	<Title>
Description	A brief description clearly and unambiguously defining the requirements in a couple of lines.
Source	The person, department, or source of relevant information associated with the description of the requirement.
Phase of the lifecycle	Data Management
Reference	References relevant to the requirement, i.e. documents, files,
Type	Mandatory/Desirable/Optional
Validation criteria	The requirement will have associated with at least one validation criterion: - Inspection - Analysis - Test
Date	Date of the definition of the requirements: Format YYYY/MM/DD
Version	The version has to follow a consecutive order

Data Management guideline



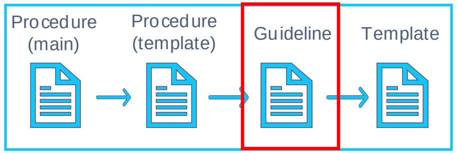
PhDM Data Management

Data Requirements Specification

Data Verification

Data Collection

Data Preparation



Procedure (main)

Procedure (template)

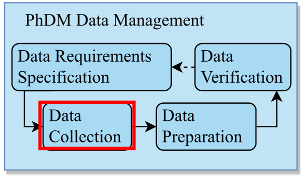
Guideline

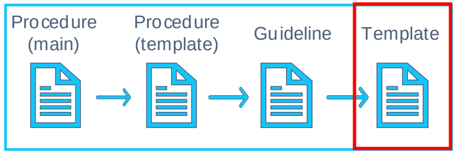
Template

Data Collection step

- It can be decomposed into two substeps:
  - Data gathering: Referring to data directly obtained from sensors and datasets (before being prepared)
  - Data generation. New data that is synthetically generated, employing for example data augmentation techniques.
- Raw data files collected in each iteration of Data collection shall be stored in the “PhDM Data Management/Collected data” folder.
- Conduct the IR

Data Collection template

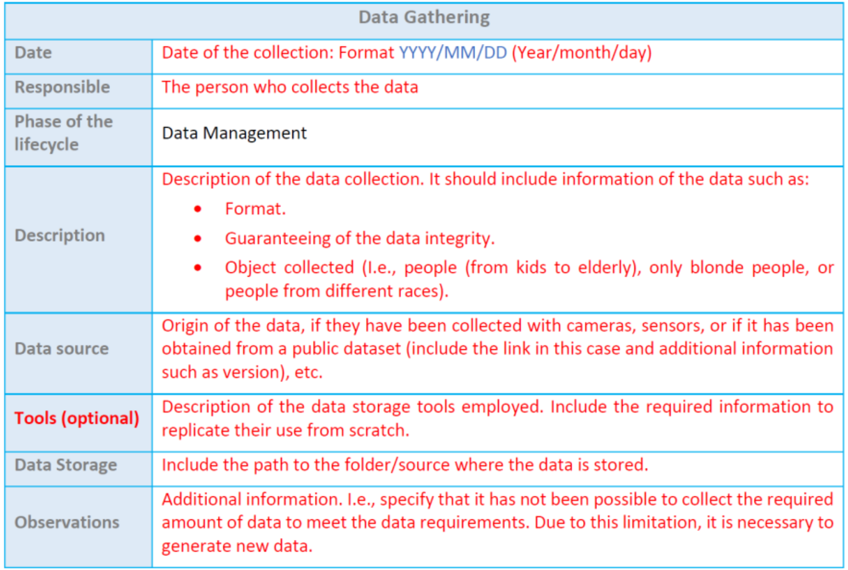




REF_PhDMD0002_Data_Collection.docx

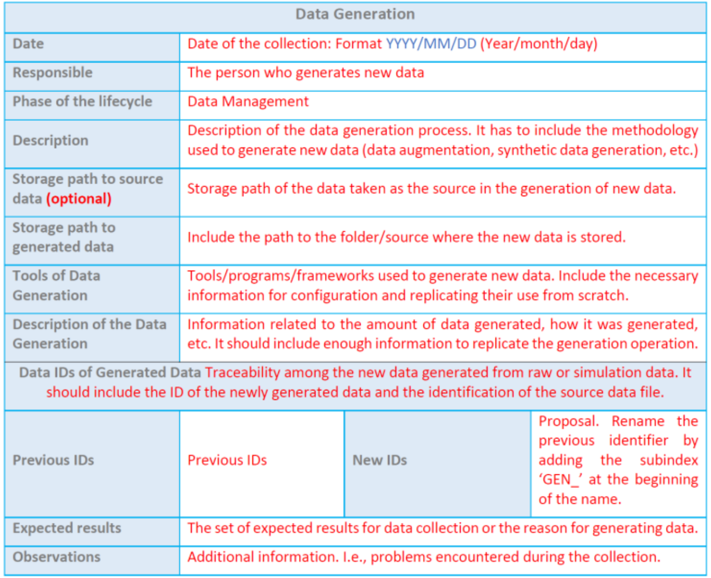
It includes

Table 1. Information related to the Data Gathering step



Data Gathering	
Date	Date of the collection: Format YYYY/MM/DD (Year/month/day)
Responsible	The person who collects the data
Phase of the lifecycle	Data Management
Description	Description of the data collection. It should include information of the data such as: - Format. - Guaranteeing of the data integrity. - Object collected (I.e., people (from kids to elderly), only blonde people, or people from different races).
Data source	Origin of the data, if they have been collected with cameras, sensors, or if it has been obtained from a public dataset (include the link in this case and additional information such as version), etc.
Tools (optional)	Description of the data storage tools employed. Include the required information to replicate their use from scratch.
Data Storage	Include the path to the folder/source where the data is stored.
Observations	Additional information. I.e., specify that it has not been possible to collect the required amount of data to meet the data requirements. Due to this limitation, it is necessary to generate new data.

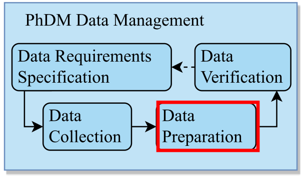
Table 2. Information related to the Data Generation step

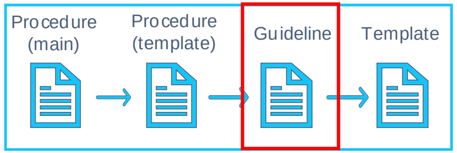


Data Generation			
Date	Date of the collection: Format YYYY/MM/DD (Year/month/day)		
Responsible	The person who generates new data		
Phase of the lifecycle	Data Management		
Description	Description of the data generation process. It has to include the methodology used to generate new data (data augmentation, synthetic data generation, etc.)		
Storage path to source data **(optional)**	Storage path of the data taken as the source in the generation of new data.		
Storage path to generated data	Include the path to the folder/source where the new data is stored.		
Tools of Data Generation	Tools/programs/frameworks used to generate new data. Include the necessary information for configuration and replicating their use from scratch.		
Description of the Data Generation	Information related to the amount of data generated, how it was generated, etc. It should include enough information to replicate the generation operation.		
Data IDs of Generated Data Traceability among the new data generated from raw or simulation data. It should include the ID of the newly generated data and the identification of the source data file.			
Previous IDs	Previous IDs	New IDs	Proposal. Rename the previous identifier by adding the subindex 'GEN_' at the beginning of the name.
Expected results	The set of expected results for data collection or the reason for generating data.		
Observations	Additional information. I.e., problems encountered during the collection.		

SAFEXPLAIN

Data Management guideline

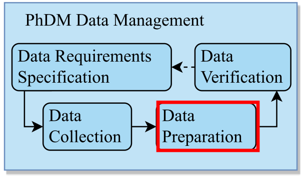


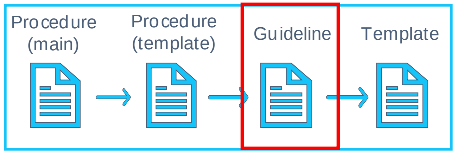


Data Preparation step

- Summarize the objective and the cases in which this step is necessary:
  - When the data need to be cleaned, processed or annotated.
- All the documents should be stored in the “PhDM Data Management/Preparation” folder.
- Conduct the IRs

Data Preparation template

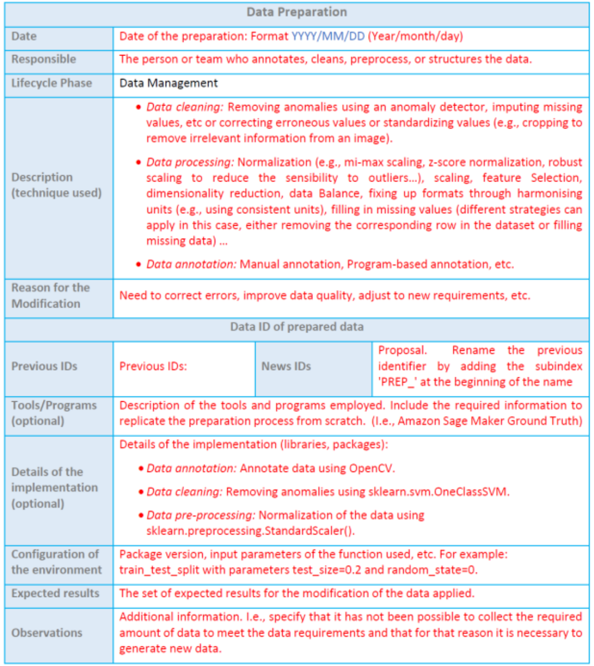




REF_PhDMD0003_Data_Preparation.docx

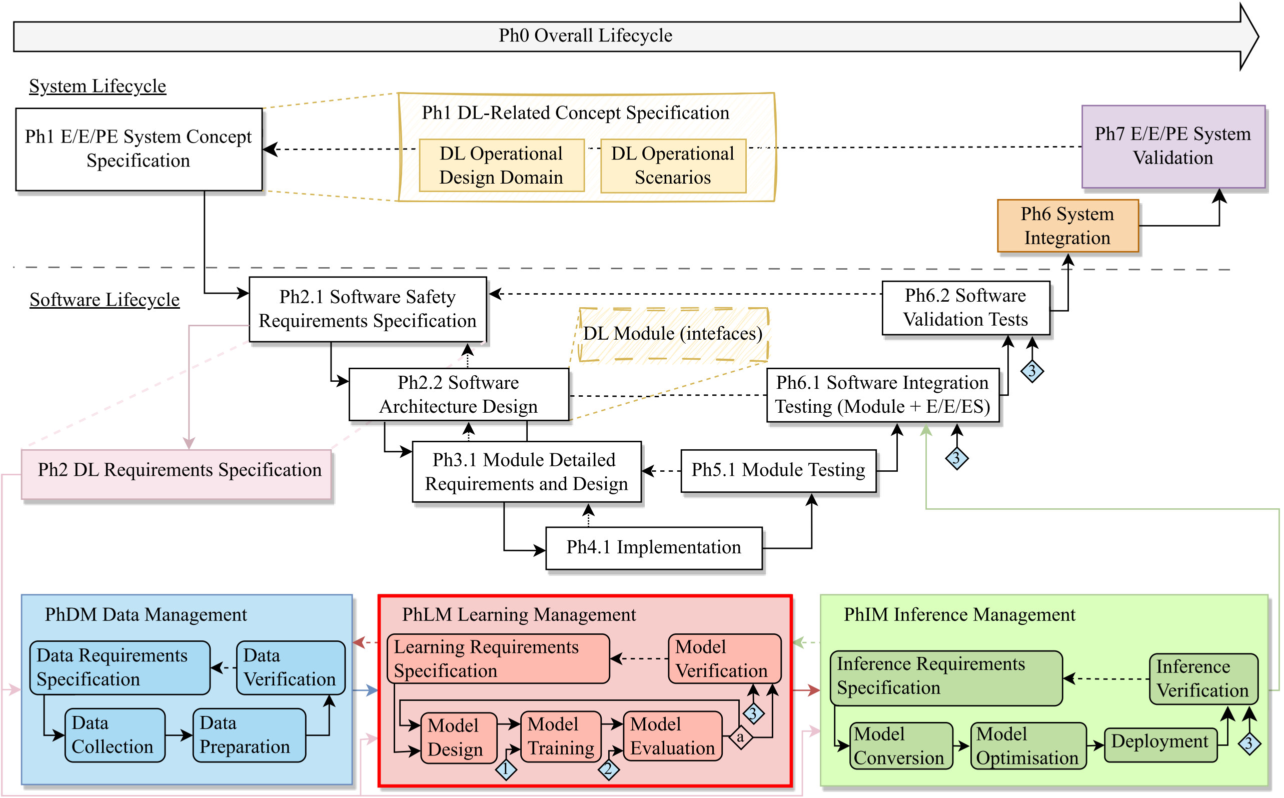
It includes:

Table 1: Information related to the Data Preparation step



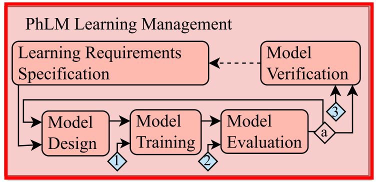
Data Preparation			
Date	Date of the preparation: Format YYYY/MM/DD (Year/month/day)		
Responsible	The person or team who annotates, cleans, preprocess, or structures the data.		
Lifecycle Phase	Data Management		
Description (technique used)	• *Data cleaning:* Removing anomalies using an anomaly detector, imputing missing values, etc or correcting erroneous values or standardizing values (e.g., cropping to remove irrelevant information from an image). • *Data processing:* Normalization (e.g., mi-max scaling, z-score normalization, robust scaling to reduce the sensibility to outliers…), scaling, feature Selection, dimensionality reduction, data Balance, fixing up formats through harmonising units (e.g., using consistent units), filling in missing values (different strategies can apply in this case, either removing the corresponding row in the dataset or filling missing data) … • *Data annotation:* Manual annotation, Program-based annotation, etc.		
Reason for the Modification	Need to correct errors, improve data quality, adjust to new requirements, etc.		
Data ID of prepared data			
Previous IDs	Previous IDs:	News IDs	Proposal. Rename the previous identifier by adding the subindex 'PREP_' at the beginning of the name
Tools/Programs (optional)	Description of the tools and programs employed. Include the required information to replicate the preparation process from scratch. (I.e., Amazon Sage Maker Ground Truth)		
Details of the implementation (optional)	Details of the implementation (libraries, packages): • *Data annotation:* Annotate data using OpenCV. • *Data cleaning:* Removing anomalies using sklearn.svm.OneClassSVM. • *Data pre-processing:* Normalization of the data using sklearn.preprocessing.StandardScaler().		
Configuration of the environment	Package version, input parameters of the function used, etc. For example: train_test_split with parameters test_size=0.2 and random_state=0.		
Expected results	The set of expected results for the modification of the data applied.		
Observations	Additional information. I.e., specify that it has not been possible to collect the required amount of data to meet the data requirements and that for that reason it is necessary to generate new data.		

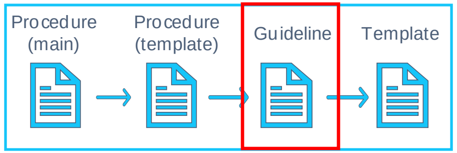
AI-FSM in-depth


Ph0 Overall Lifecycle
System Lifecycle
Ph1 E/E/PE System Concept Specification
Ph1 DL-Related Concept Specification
DL Operational Design Domain
DL Operational Scenarios
Ph7 E/E/PE System Validation
Ph6 System Integration
Software Lifecycle
Ph2.1 Software Safety Requirements Specification
Ph6.2 Software Validation Tests
DL Module (intefaces)
Ph2.2 Software Architecture Design
Ph6.1 Software Integration Testing (Module + E/E/ES)
3
Ph2 DL Requirements Specification
Ph3.1 Module Detailed Requirements and Design
Ph5.1 Module Testing
3
Ph4.1 Implementation
PhDM Data Management
Data Requirements Specification
Data Verification
Data Collection
Data Preparation
PhLM Learning Management
Learning Requirements Specification
Model Verification
Model Design
Model Training
Model Evaluation
1
2
3
a
PhIM Inference Management
Inference Requirements Specification
Inference Verification
Model Conversion
Model Optimisation
Deployment
3

Virtual – 2024/07/03

Learning Management guideline

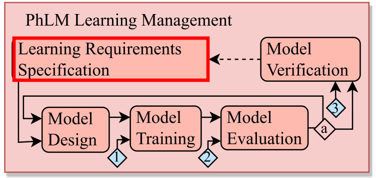


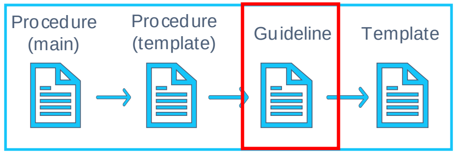


PhLM Learning Management

- The objective of this phase is the generation of:
  - Model Trained
  - Model Evaluated
  - Learning Model verified
- As previously mentioned, the following document should be generated:
  - REF_PhLMD0001_Learning_Requirements_Specifications.docx. (+IR)
  - REF_PhLMD0003_Model_Election_Log.docx. (+IR)
  - REF_PhLMD0005_Learning_Requirements_Evaluation_Tests.docx. (+IR)
  - REF_PhLMD0007_Learning_Requirements_Verification_Tests (+IR)
- All the documents should be stored in the “PhLM Learning Management” folder.

Learning Management guideline





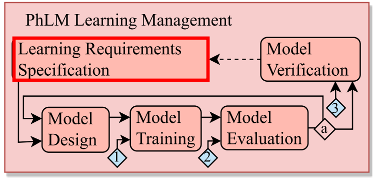
Learning requirements specification

- It directly addresses the safety designer to the learning reqs. specification template.
- Define the mechanisms or tests that must be carried out to check that the learning model meets the associated learning requirements specification:
  - Learning reqs. evaluation tests
  - Learning reqs. verification tests

IMP: These tests are not verification or validation tasks according to functional safety standards.

- Conduct the IRs

Learning Requirements Specification template


PhLM Learning Management

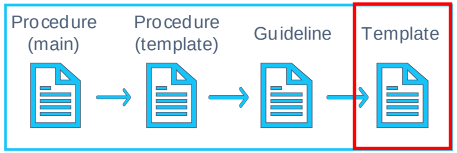
Learning Requirements Specification

Model Verification

Model Design

Model Training

Model Evaluation


Procedure (main)

Procedure (template)

Guideline

Template

REF_PhLMD0001_Learning_Requirement_Specification.docx

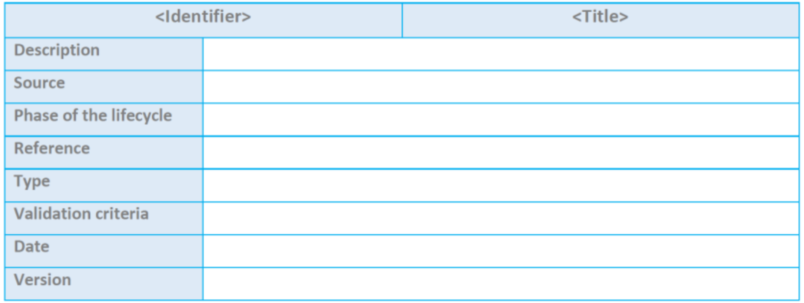
It proposes decomposing the Learning reqs. into:

- Quantitative:
    - Model bias and variance boundaries → focusing on avoiding underfitting and overfitting
    - Performance and robustness reqs. → For ex: recall, precision, accuracy or F1 score.
- Qualitative:
    - Methodology for searching the hyperparamenters

Define a Model Election criteria. For example:

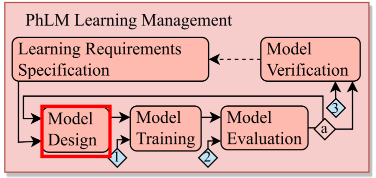
- Prioritizing classes accuracy
- Robustness regarding especific environments
- Emphasis on explainability

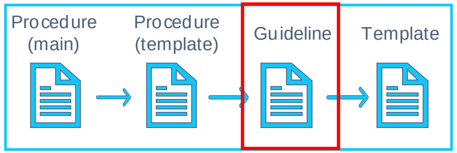
Table 1. Table of attributes for each requirement



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

Learning Management guideline

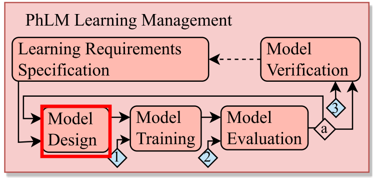


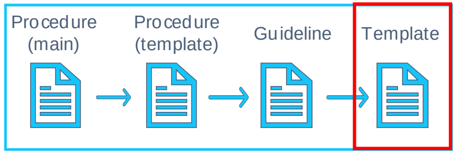


Model Design

- The objective of this step is to specificate a set of DL models that suits the application
- It explains aspects to be considered in the election of the DL such as:
  - Model Architecture
  - Pretrained Models
  - Hyperparameter tunning
  - …
- It finally addresses the user to the REF_PhLMD0003_Model_Election_Log.docx template.

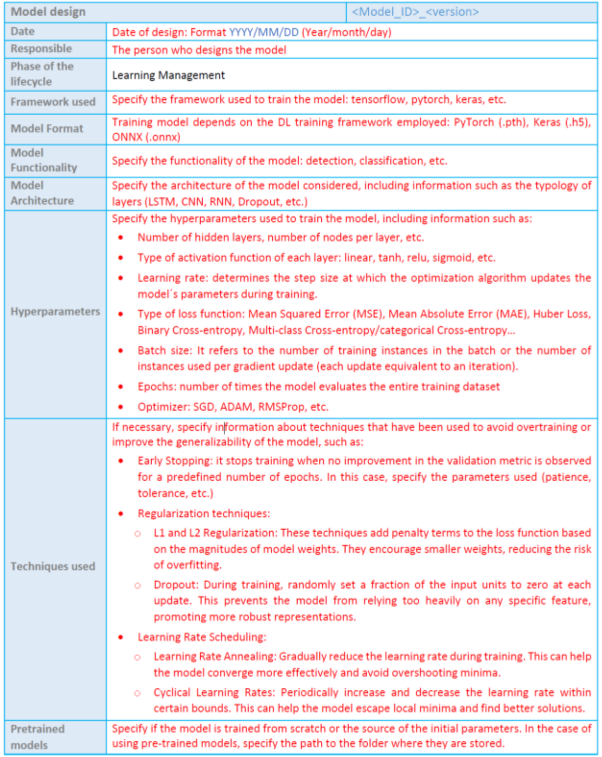
Model Election Log





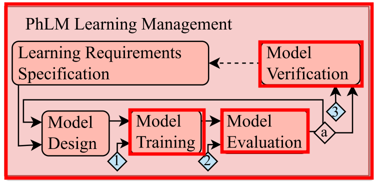
REF_PhLMD0003_Model_Election.docx

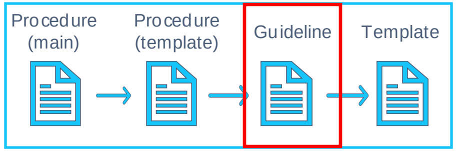
- It includes:



Model design	<Model_ID>_<version>
Date	Date of design: Format YYYY/MM/DD (Year/month/day)
Responsible	The person who designs the model
Phase of the lifecycle	Learning Management
Framework used	Specify the framework used to train the model: tensorflow, pytorch, keras, etc.
Model Format	Training model depends on the DL training framework employed: PyTorch (.pth), Keras (.h5), ONNX (.onnx)
Model Functionality	Specify the functionality of the model: detection, classification, etc.
Model Architecture	Specify the architecture of the model considered, including information such as the typology of layers (LSTM, CNN, RNN, Dropout, etc.)
Hyperparameters	Specify the hyperparameters used to train the model, including information such as: • Number of hidden layers, number of nodes per layer, etc. • Type of activation function of each layer: linear, tanh, relu, sigmoid, etc. • Learning rate: determines the step size at which the optimization algorithm updates the model´s parameters during training. • Type of loss function: Mean Squared Error (MSE), Mean Absolute Error (MAE), Huber Loss, Binary Cross-entropy, Multi-class Cross-entropy/categorical Cross-entropy… • Batch size: It refers to the number of training instances in the batch or the number of instances used per gradient update (each update equivalent to an iteration). • Epochs: number of times the model evaluates the entire training dataset • Optimizer: SGD, ADAM, RMSProp, etc.
Techniques used	If necessary, specify information about techniques that have been used to avoid overtraining or improve the generalizability of the model, such as: • Early Stopping: it stops training when no improvement in the validation metric is observed for a predefined number of epochs. In this case, specify the parameters used (patience, tolerance, etc.) • Regularization techniques: ○ L1 and L2 Regularization: These techniques add penalty terms to the loss function based on the magnitudes of model weights. They encourage smaller weights, reducing the risk of overfitting. ○ Dropout: During training, randomly set a fraction of the input units to zero at each update. This prevents the model from relying too heavily on any specific feature, promoting more robust representations. • Learning Rate Scheduling: ○ Learning Rate Annealing: Gradually reduce the learning rate during training. This can help the model converge more effectively and avoid overshooting minima. ○ Cyclical Learning Rates: Periodically increase and decrease the learning rate within certain bounds. This can help the model escape local minima and find better solutions.
Pretrained models	Specify if the model is trained from scratch or the source of the initial parameters. In the case of using pre-trained models, specify the path to the folder where they are stored.

Learning Management guideline





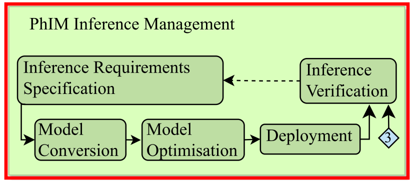
Model Training: In this step, the specified models are generated employing the training dataset

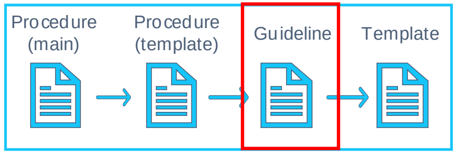
Model Evaluation: Once the model(s) are trained, they are evaluated employing the validation dataset:

- Situations that can arise:
  - None of the candidate models achieve the expected performance:
    1. Iterative repeat the design, training, and evaluation steps until meeting them
    2. If they are not met → new iteration of the Data Management phase
  - Multiple candidates demonstrate the expected performance → All will be verified in the next step

Model Verification: This phase not only evaluates the generalization capabilities and identifies potential issues using the verification dataset but also checks if the reqs. are met.

Inference Management guideline

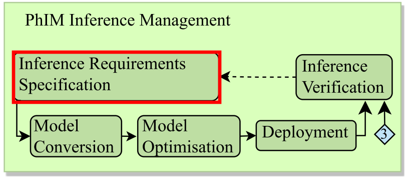


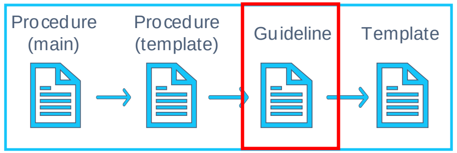


PhIM Inference Management

- The objective of this phase is the generation of:
  - Model converted
  - Model optimised
  - Inference model verified
- As previously mentioned, the following document should be generated:
  - REF_PhIMD0001_Inference_Requirements_Specifications.docx. (+IR)
  - REF_PhIMD0003_Model_Conversion_Log.docx. (+IR)
  - REF_PhIMD0005_Model_Optimization_Log.docx. (+IR)
  - REF_PhIMD0007_Inference_Requirements_Verification_Tests. (+IR)
- All the documents should be stored in the “PhIM Inference Management” folder

Inference Management guideline

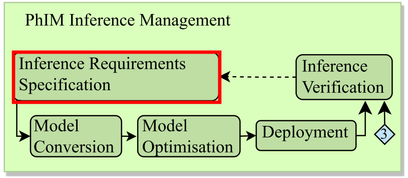


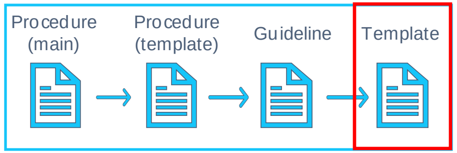


PhIM Inference Reqs. Specification

- Inference management guideline directly addresses the user to the template.
- Inference Management guideline indicates that in this step:
  - The requirements and verification tests shall be defined
  - The IRs shall be conducted

Inference Requirements Specification template



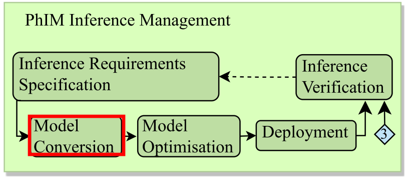


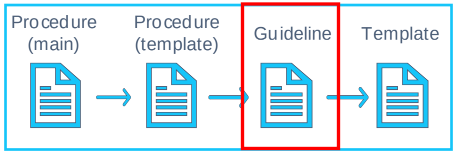
REF_PhIMD0001_Inference_Requirements_Specifications.docx

It proposes decomposing the Inference reqs. into:

- Reqs. associated with model conversion
  - Computer arithmetic
  - Software dependencies
- Rqs. associated with model optimization
  - Model quantization
  - Model pruning
- Reqs. associated with model deployment
  - Memory limitations
  - Execution time restrictions

Inference Management guideline





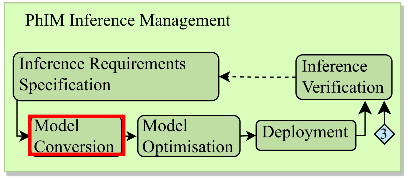
Model Conversion

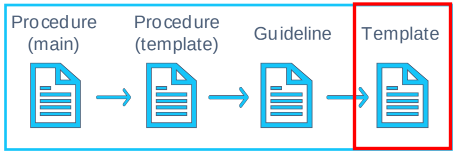
- Inference Management Guideline includes:
  - Definition of the model conversión
  - Specifies that all the information of this step shall be documented in the associated template.Ex:
    - Training-specific operations removed
    - Loading and converting operations performed.
- Conduct the IR

SAFEXPLAIN

Virtual – 2024/07/03

Model Conversion template

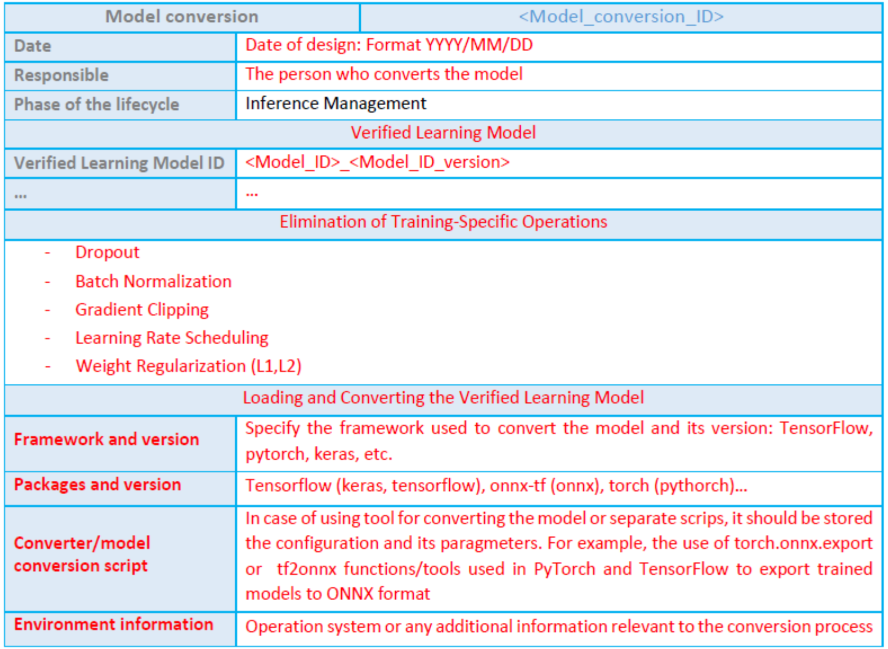




REF_PhIMD0003_Model_Conversion_Log.docx

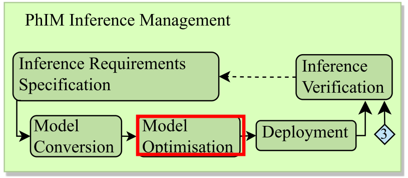
- It includes:

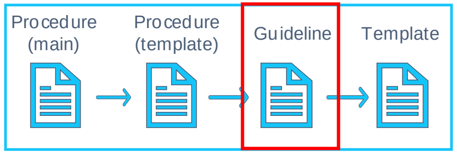
Table 1. Model conversion information



Model conversion	<Model_conversion_ID>
Date	Date of design: Format YYYY/MM/DD
Responsible	The person who converts the model
Phase of the lifecycle	Inference Management
Verified Learning Model	
Verified Learning Model ID	<Model_ID>_<Model_ID_version>
…	…
Elimination of Training-Specific Operations	
- Dropout - Batch Normalization - Gradient Clipping - Learning Rate Scheduling - Weight Regularization (L1,L2)	
Loading and Converting the Verified Learning Model	
Framework and version	Specify the framework used to convert the model and its version: TensorFlow, pytorch, keras, etc.
Packages and version	Tensorflow (keras, tensorflow), onnx-tf (onnx), torch (pythorch)…
Converter/model conversion script	In case of using tool for converting the model or separate scrips, it should be stored the configuration and its paragmeters. For example, the use of torch.onnx.export or tf2onnx functions/tools used in PyTorch and TensorFlow to export trained models to ONNX format
Environment information	Operation system or any additional information relevant to the conversion process

Inference Management guideline





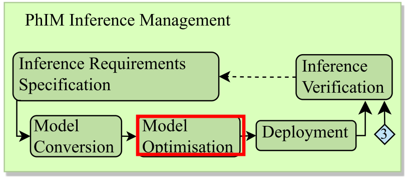
Model Optimisation:

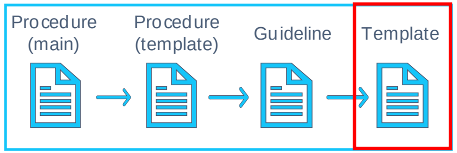
The guideline proposes completing the template with the information related to model optimization and outlines some information that shall be included in it:

- Calibration fundamental operations
- Post-training quantization specifications
- Pruning specifications
- Techniques to recover accuracy.

- Conduct the IR

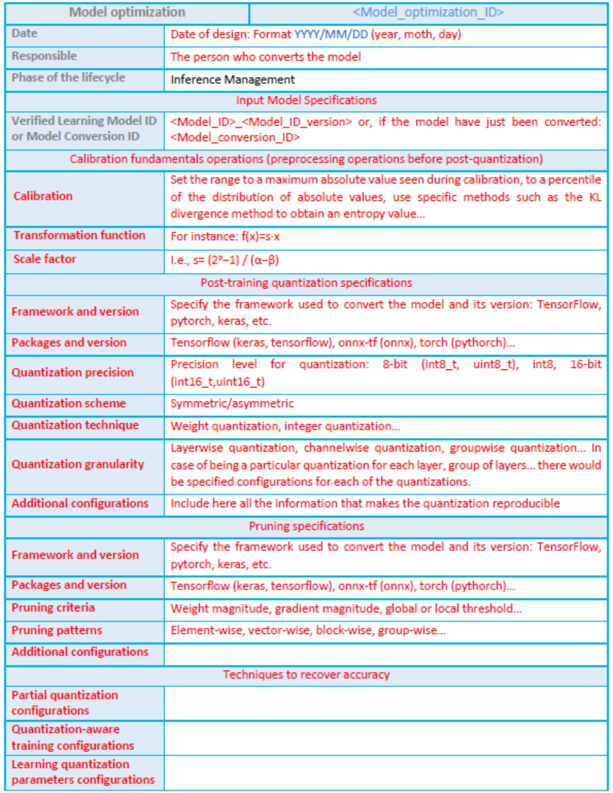
Model Optimisation template





REF_PhIMD0005_Model_Optimization_Log.docx

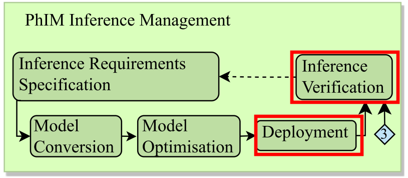
- It includes:

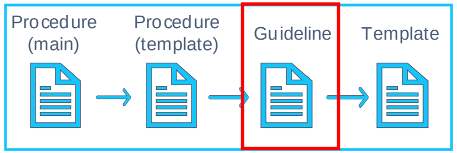


Model optimization	<Model_optimization_ID>
Date	Date of design: Format YYYY/MM/DD (year, moth, day)
Responsible	The person who converts the model
Phase of the lifecycle	Inference Management
Input Model Specifications	
Verified Learning Model ID or Model Conversion ID	<Model_ID>_<Model_ID_version> or, if the model have just been converted: <Model_conversion_ID>
Calibration fundamentals operations (preprocessing operations before post-quantization)	
Calibration	Set the range to a maximum absolute value seen during calibration, to a percentile of the distribution of absolute values, use specific methods such as the KL divergence method to obtain an entropy value…
Transformation function	For instance: f(x)=s·x
Scale factor	I.e., s= (2^b-1) / (α−β)
Post-training quantization specifications	
Framework and version	Specify the framework used to convert the model and its version: TensorFlow, pytorch, keras, etc.
Packages and version	Tensorflow (keras, tensorflow), onnx-tf (onnx), torch (pythorch)…
Quantization precision	Precision level for quantization: 8-bit (int8_t, uint8_t), int8, 16-bit (int16_t,uint16_t)
Quantization scheme	Symmetric/asymmetric
Quantization technique	Weight quantization, integer quantization…
Quantization granularity	Layerwise quantization, channelwise quantization, groupwise quantization… In case of being a particular quantization for each layer, group of layers… there would be specified configurations for each of the quantizations.
Additional configurations	Include here all the information that makes the quantization reproducible
Pruning specifications	
Framework and version	Specify the framework used to convert the model and its version: TensorFlow, pytorch, keras, etc.
Packages and version	Tensorflow (keras, tensorflow), onnx-tf (onnx), torch (pythorch)…
Pruning criteria	Weight magnitude, gradient magnitude, global or local threshold…
Pruning patterns	Element-wise, vector-wise, block-wise, group-wise…
Additional configurations	
Techniques to recover accuracy	
Partial quantization configurations	
Quantization-aware training configurations	
Learning quantization parameters configurations	

SAFEXPLAIN

Inference Management guideline





Deployment:

- This step entails the implementation of the model in the target platform.

Inference verification.

- This step not only evaluates the generalization capabilities and identifies potential issues using the verification dataset but also checks if the reqs. are met.
    - If they are not meet, the inference model process shall be reiterated. If the inference model still does not meet the inference requirements specifications, further corrective actions or adjustments in the Data Management and the Learning Management may be required.
- Conduct the IR

Safety technical assesment

Safety technical assesment

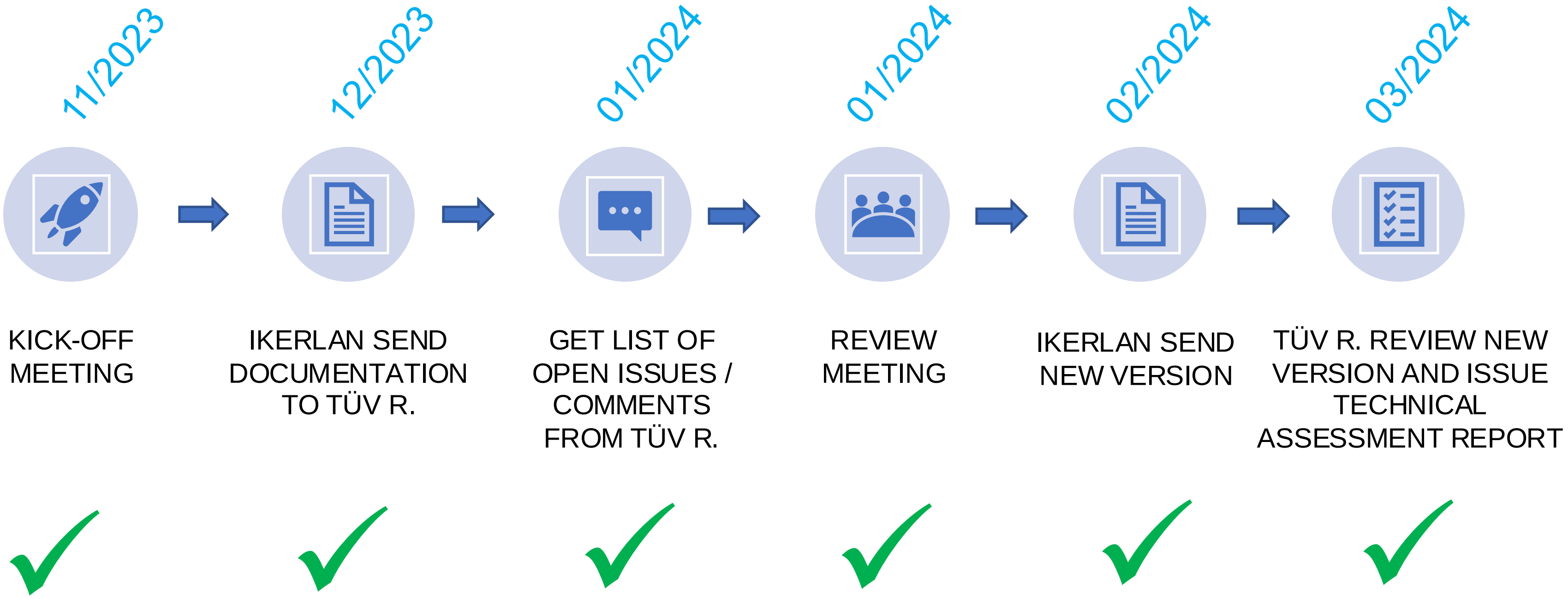
Project-internal evaluation:

- Exida partner

Project-external evaluation:

- TÜV Rheinland entity

TÜV Rheinland collaboration



11/2023
KICK-OFF MEETING
✓

12/2023
IKERLAN SEND DOCUMENTATION TO TÜV R.
✓

01/2024
GET LIST OF OPEN ISSUES / COMMENTS FROM TÜV R.
✓

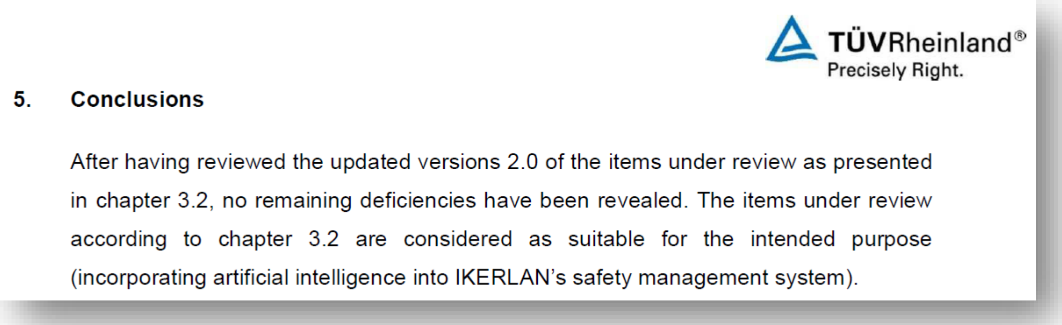
01/2024
REVIEW MEETING
✓

02/2024
IKERLAN SEND NEW VERSION
✓

03/2024
TÜV R. REVIEW NEW VERSION AND ISSUE TECHNICAL ASSESSMENT REPORT
✓

TÜV Rheinland collaboration

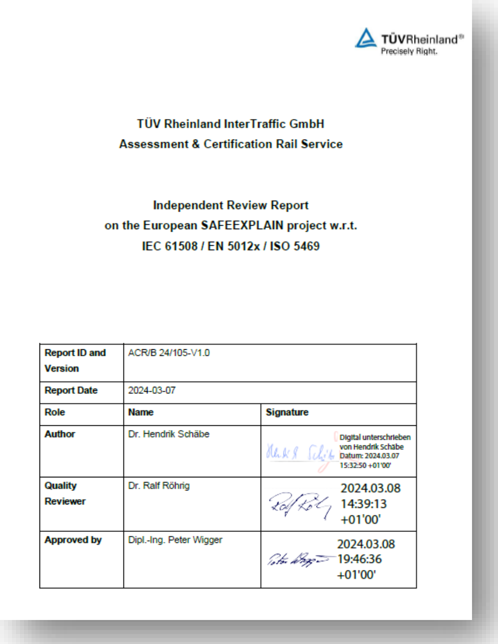
- Positive assessment received from TÜV for the AI-FSM



TÜVRheinland®
Precisely Right.

5. Conclusions

After having reviewed the updated versions 2.0 of the items under review as presented in chapter 3.2, no remaining deficiencies have been revealed. The items under review according to chapter 3.2 are considered as suitable for the intended purpose (incorporating artificial intelligence into IKERLAN's safety management system).



TÜVRheinland®
Precisely Right.

TÜV Rheinland InterTraffic GmbH
Assessment & Certification Rail Service

Independent Review Report
on the European SAFEXPLAIN project w.r.t.
IEC 61508 / EN 5012x / ISO 5469

Report ID and Version	ACR/B 24/105-V1.0	
Report Date	2024-03-07	
Role	**Name**	**Signature**
Author	Dr. Hendrik Schäbe	Digital unterschrieben von Hendrik Schäbe Datum: 2024.03.07 15:32:50 +01'00'
Quality Reviewer	Dr. Ralf Röhrig	2024.03.08 14:39:13 +01'00'
Approved by	Dipl.-Ing. Peter Wigger	2024.03.08 19:46:36 +01'00'

Questions?

THANKS FOR YOUR ATTENTION!!

SAFEXPLAIN

Safe and Explainable
Critical Embedded Systems based on AI

Follow us on social media:

www.safexplain.eu

Funded by
the European Union

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