Verification, Validation and Qualification of AI Systems

ECSS-E-HB-40-02A Machine Learning Qualification Handbook

ECSS Working Group

28 September 2023

Purpose of the ECSS-E-HB-40-02A handbook:

- Provide guidelines on how:
  - to create reliable AI functions
  - data can be selected and qualified,
  - training of the ML models can be performed,
  - a “safety cage architecture” can be applied
- Current major trends of AI (supervised and unsupervised learning approaches) are covered
- Guideline are provided for the following specific processes:
  1. Data Qualification and ML Model Development Process
  2. Machine Learning Model Testing
  3. System Testing and Qualification

This presentation will provide an overview of the current status of the handbook

Public review is still open until 6 October 2023 - ECSS-E-HB-40-02A DIR1 Public Review

Machine Learning Qualification Handbook

Qualification of ML demands a multi-disciplinary focus.

The Handbook limited to software criticality categories B/C/D software (excluding life critical Cat. A functions).

When to use ML solutions and when not to

Business consideration

Just because we have data, does not mean we have to build an AI/ML model.

Just because the problem can be solved with AI/ML, does not mean that it is the best solution.

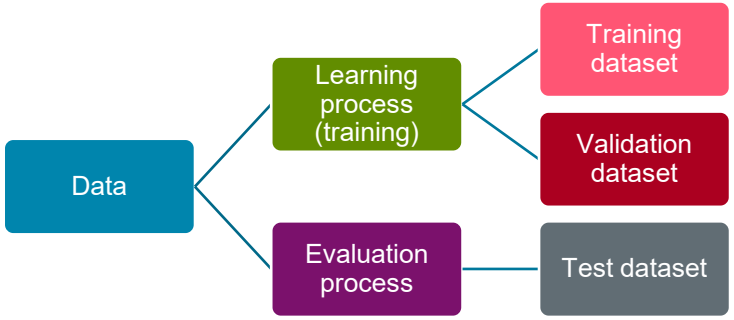
As an initial consideration, consider if you have the right:

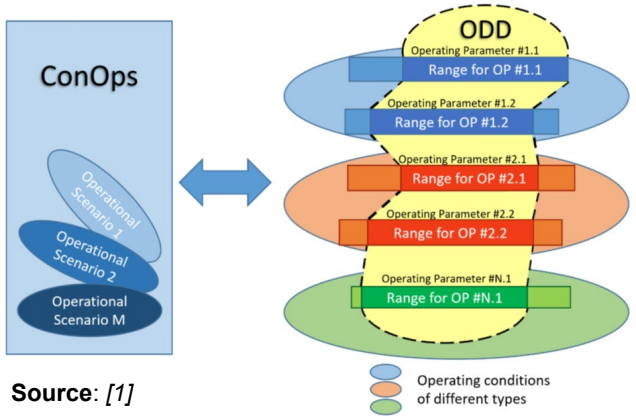
1. Strategy for data acquisitions, to assure capturing maximum value.
2. The correct infrastructure to manage the data and the development process.
3. Some form of application deployment environment, allowing streaming of data to applications, and monitoring of performance.
4. Dedicated personal to work with the data, e.g. data scientist, data manager, AI product manager, machine learning engineer, etc.

See Section 6.2

1. Data Qualification

- DATA LIFE CYCLE: Quality assessment and data splitting
  - ML development iterative process => Data Quality assessment is a continuous process
  - Once initial data is gathered first assessment on data quality should be performed
  - Evaluation for data quality and splitting of data for training/validation/testing. With respect to:
    - **Dataset representativeness**
    - **Hold-out dataset**
    - **Operational scenarios and operational design domain**[1]





Source: *[1]*

[1] EASA Concept Paper: First usable guidance for Level 1 machine learning applications.

Data Qualification

See Section 6.4.1.2

1. Data Qualification

- **DATA SOURCES: Types and main considerations**
  - Real data
  - Simulation and Synthetic data
    - Generated by highly representative simulation methods
    - Difference between simulated data and real data expected to be limited, ***however,*** *=> It is* ***still vital*** *to ensure their representativeness*
  - Augmentation data
    - Synthetic data based on real data
    - Less representativeness problems
  - Surrogate data
    - Ensure post-processed data still allows learning

- **SPECIFIC APPLICATIONS:**
  - Supervised learning
    - Quality aspect related to the representativeness of the data w.r.t. ConOps
    - Quality aspect relate to the labeling
  - Unsupervised learning
    - No labeling involved
    - Quality aspect related to the representativeness of the data w.r.t. ConOps
  - Reinforcement learning

Data Qualification

See Section 6.4.1.3 - 6.4.1.4

ECSS

1. Model Development Process

OVERALL WORKFLOW

1. Data gathering => D*ata quality assessment*
2. Data splitting => *Data quality assessment*
3. Training process => *Fit of the model*
   - Hyper-parameter optimization
   - Initial performance against validation data
4. Evaluation against hold-out data
5. Model Testing
6. [Model optimization]

FRAMEWORK

- No development framework is identified as preferred for the development process and choosing one over other answers to different factors such as: previous experience, functionality provided or compatibility with relevant software stack

It is an iterative process; any step can be revisited (i.e. due to underperforming)

After applying optimization techniques steps 4-5 need to be revisited

ML Development process

See Section 6.4.2

1. Model Development Process

- **MODEL QUALITY CHARACTERISTICS**
  - **Functionality**
    The capability of the ML model to provide functions which meet stated and implied needs
  - **Reliability**
    The capability of an ML-based component to maintain a specified level of performance when used under specified conditions
  - **Robustness**
    Local Robustness
    Global Robustness
  - **Resilience**
    The ability for a system to continue to operate while an error or a fault has occurred
  - **Explainability**
    The ease with which a human can comprehend an ML model, its data, and its results and outputs

Characteristics	Model selection	Model testing
Functionality	✓	✓
Reliability		✓
Robustness	✓	✓
Resilience:		✓
Interpretability/ Explainability	✓	✓

ML Development process

See Section 6.4.2.2

2. Machine Learning Model Testing

- Performance metrics should be used as the main performance indicators during the model development phase
- Performance metrics against the training and validation datasets help us to identify the best candidate model
- The same performance metrics against the test dataset allow us to evaluate the behaviour of the model under unseen data.

Model testing aims at improving the model trustworthiness, by applying different methods, to complement the model evaluation based on performance metrics.

Machine Learning Model Testing

See Section 6.4.3.1

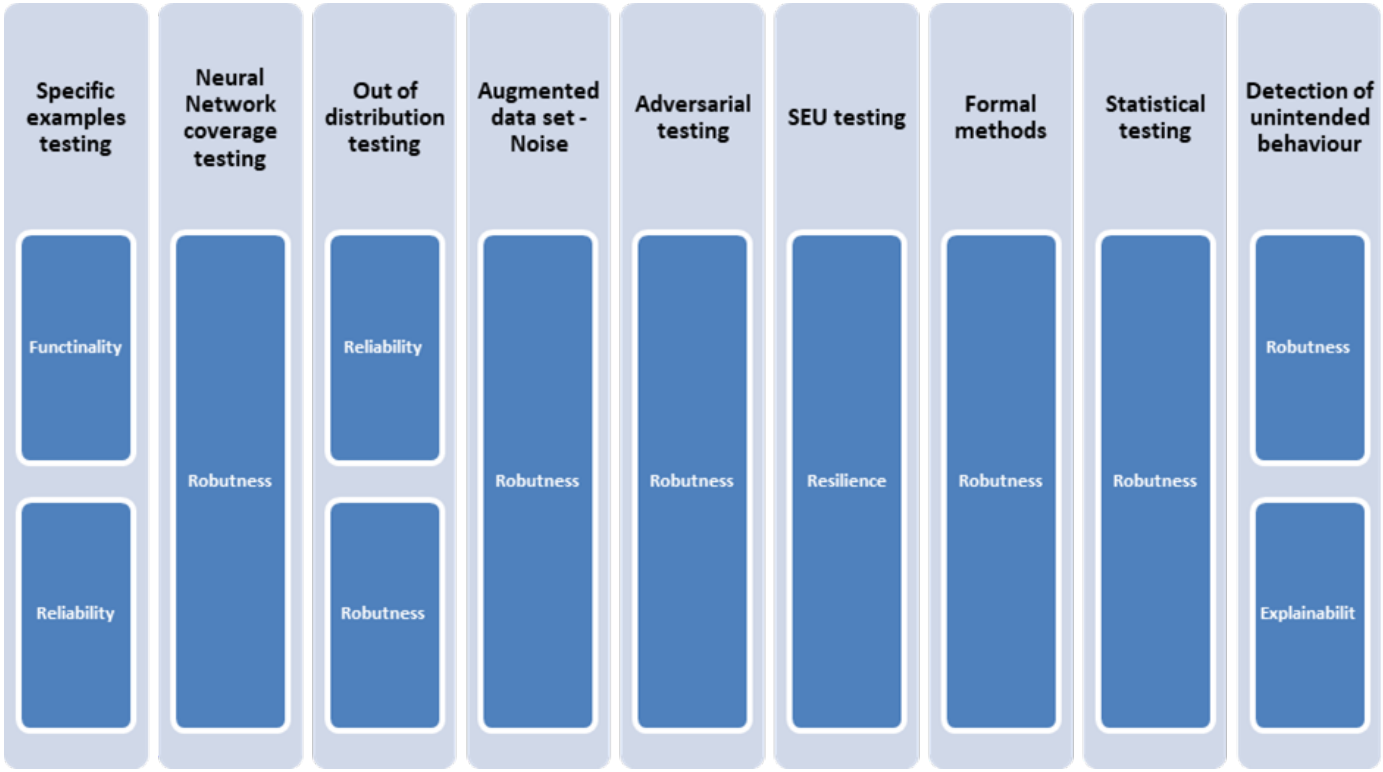
2. Machine Learning Model Testing

- **TESTING METHODS**
  - **Specific example testing**
    - Inputs importance
    - Specific range of ODD (Operational Design Domain)
  - **Neural Network coverage testing**
  - **Out of distribution testing**
    - When done at system level allows to see if any implemented back-up system can deal with OOD data
  - **Augmented data set – Noise**
    - Noise expected during operation
    - Aging
    - Unknown noise
  - **Adversarial testing**
  - **Formal methods and mathematical verification**
  - **Statical Testing**

Machine Learning Model Testing

See Section 6.4.3.1.2

2. Machine Learning Model Testing



Machine Learning Model Testing

See Section 6.4.3.1.2

3. System Testing and Qualification

Reliability, safety, and performance of AI systems

Assessing the model's performance on **unseen data** and ensuring that it generalizes well to new situations.

Confirmation of correctly implementation of **models** and that the training and testing data are representative of the intended problem domain, and training/testing procedure are accurately followed.

Considering not only their performance but also their **safety, reliability, and ethical implications,** compared to general usage (w.r.t regulations, etc.).

Comprehensive evaluation of performance, safety, and reliability in the specific context in which it will be used. May involve stress-testing the models under a range of conditions, validating their performance in real-world scenarios, and ensuring that they are robust to changes in input data, environmental factors, and other potential sources of uncertainty. **Additionally, qualification may require addressing the interpretability, explainability, and transparency of AI/ML systems to ensure they can be trusted by users and stakeholders**.

Reliability, safety, and performance

Verification, Validation, certification and qualification as means of assurance.

For “traditional” systems see ECSS-E-ST-40C

ECSS

3. System Testing and Qualification

(Some) New Challenges (from PA perspective) (1/2)

- **MAIN CHALLENGE #1: PROBABILISTIC ASSESSMENT**
  - For safety critical systems, quantitative safety analysis is used to assess properties such as “the probability of a catastrophic event of an aircraft shall be lower than 10-9 per flight hour”.
  - Question: How to estimate/define probability?
- **MAIN CHALLENGE #2: RESILIENCE**
  - With Machine Learning, resilience is made more complex because of the usually wider range of possible inputs (e.g. images), the difficulties to adopt classical strategies (e.g., redundancy with dissimilarity), and the ML-specific vulnerabilities.
  - Question: How to increate resilience?
- **MAIN CHALLENGE #5: EXPLAINABILITY**
  - The opacity of ML models is seen as a major limitation for their development and deployment, especially for systems delivering high stake decisions
  - Question: Can we understand, trust and rely on ML results?

New Challenges

According to DEEL Paper

See Section 4.3, page 21ff

3. System Testing and Qualification

(Some) New Challenges (from PA perspective) (2/2)

- **MAIN CHALLENGE #6: ROBUSTNESS**
  - Robustness raises many challenges, from the definition of metrics for assessing robust-ness or similarity, to out-of-domain detection, and obviously adversarial attacks and defence
  - Question: What could have impacts on the robustness of the system?

Can we anticipate all the perimeters to ensure the safe operation of the system?

New Challenges

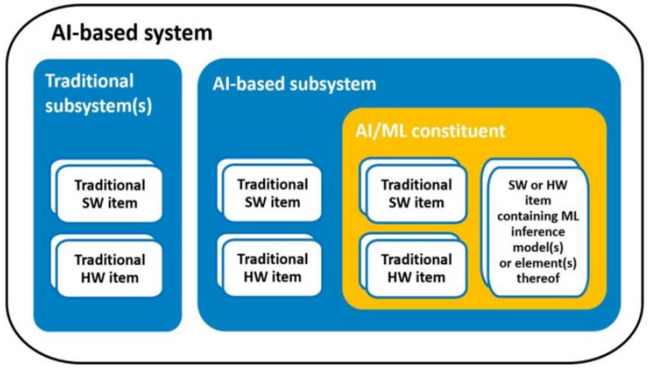
According to DEEL Paper

See Section 4.3, page 21ff

3. System Testing and Qualification

Failure analysis approach

- Based on evaluation of the ML part, hard to predict all types of failures.
- However, it is possible to treat the part **as a system part using FMEA/FMECA** process for assessing the Severity and Probability levels for potential failures due to the ML/AI parts of the system:



- Estimation of Severity is simple, but what about Probability?
- In according to ISO guide on medical software (DS/IEC/TR 80002-1):

  "…. No consensus exists for a method of estimating the probability of occurrence of a software failure. This is even more true for a ML based applications. When software is present in a sequence of events leading to a hazardous situation, the probability of the software failure occurring cannot be considered in estimating the risk for the hazardous situation. In such cases, considering a worse case probability is appropriate, and the probability for the software failure occurring should be set to 1."

Classical FMEA and FMECA

FMEA/FMECA for assessing Risk mitigation needs

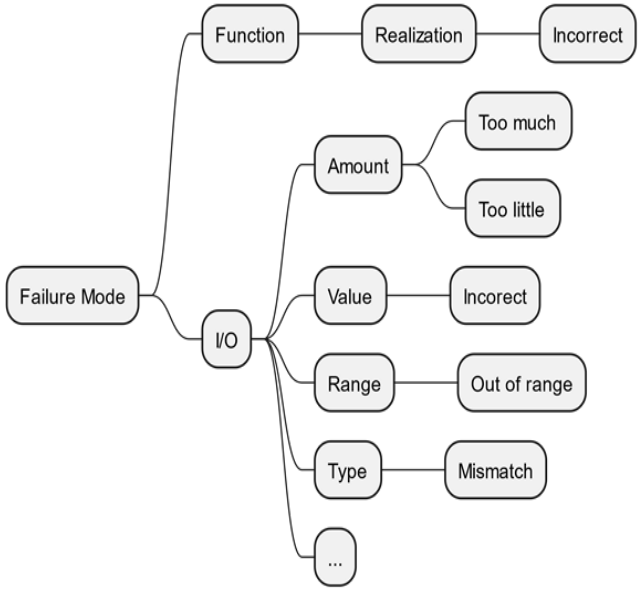
See
Section 6.4.4.2 page 61 and
Section 6.4.4.3.2.4, page 71

3. System Testing and Qualification

Performing a FME(C)A for AI components

- Handbook contains a detailed example on how to create a FMECA
- Below an example on how to define FMEA and FMECA items

- Define a failure taxonomy, e.g



- Implement a (functional) FMEA

FMEA						
Product:		**System:**		**Subsystem:**		**Equipment:**
No.	**Item**	**Function**	**Failure mode**	**Failure Cause**	**Failure effects**	**…**
		E.g. Calculate pressure	Function.Realization.Incorrect			
			I/O.Amount.Too_Much			
			I/O.Amount.Too_Little			
			I/O.Value.Incorrect			
			I/O.Range.Out_Of_Range			
			I/O.Type.Missmatch			
			…			

- Create criticality matrix

Severity level	Severity category	SN
1	Catastrophic	4
2	Critical	3
3	Major	2
4	Negligible	1

X

Level	Limits	PN
Probable	P=1	4
Occasional	$10^{-3}<P<=10^{-1}$	3
Remote	$10^{-5}<P<=10^{-3}$	2
Extremely remote	$P<=10^{-5}$	1

Severity category	SNs	Probability level			
		10^{-5}	10^{-3}	10^{-1}	1
		PNs			
		1	2	3	4
catastrophic	4	4	8	12	16
critical	3	3	6	9	12
major	2	2	4	6	8
negligible	1	1	2	3	4

- Remember, changed limits of PN 4 because of DS/IEC/TR 80002-1!

See Section 6.4.4.3.2, page 66ff

3. System Testing and Qualification

Interfaces and interactions to be considered

- In creating the FMEA/FMECA, the following interfaces and interactions should be considered, to describe and analyze the product (i.e. function or hardware):
  1. Software/Software
  2. Software/Hardware
  3. Hardware/Software
  4. (Hardware/Hardware)
- In case analysis including hardware is needed, the interface between hardware and software will be subject to the Hardware-Software Interaction Analysis (HSIA), which then provides the input to FMEA/FMECA.
- A ML SW, a suitable approach can be:
  1. Start with the functional descriptions, which help creating the function FMEA/FMECA. For this the FMEA/FMECA worksheet shall contain a concise statement of the function performed by the item.
  2. Then consider the interfaces. If hardware is involved write HSIA, but also consider here HW/SW as well as SW/HW interaction effects.
  3. Then consider the interrelationships and interdependencies of the items which constitute the product.

Interfaces and HW?

Use HSIA as needed

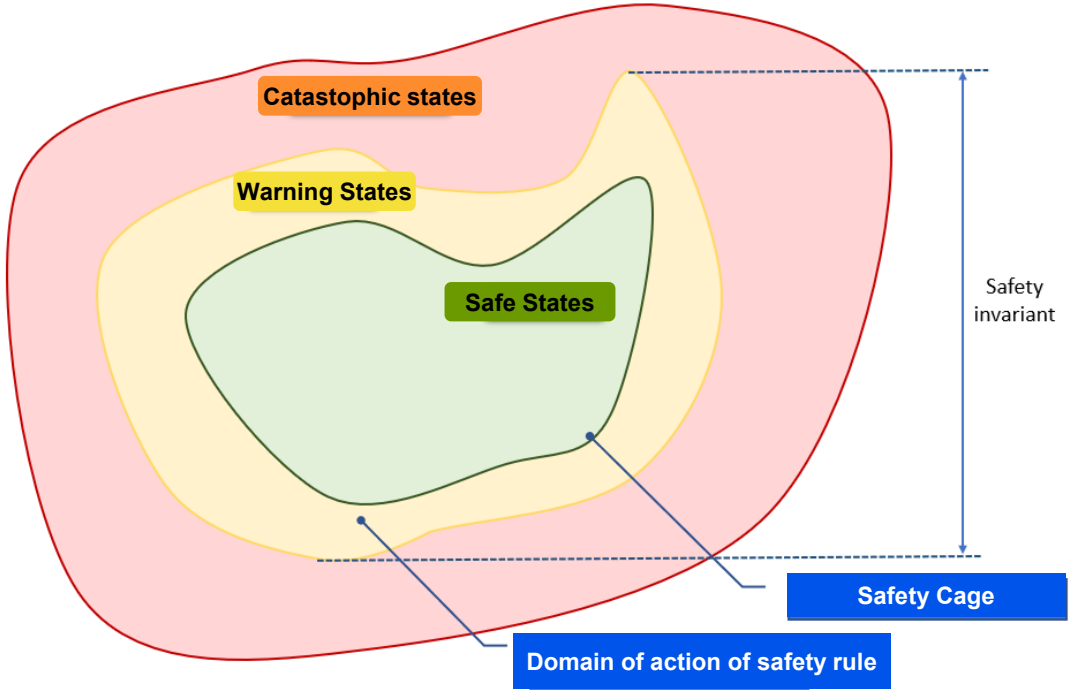
See Section 6.4.4.3.2.4, page 72ff

3. System Testing and Qualification

Risk Mitigation

- In addition to the risk mitigations that can be done at data and model level, anything identified at system level via the FMEA/FMECA can be mitigated via a well designed safety cage.

- The safety cage, also known as a safety strategy, is a set of safety rules to ensure a set of safety invariants, designed to abort all paths to the catastrophic states.



Simplified representation of the safety cage, warning states and catastrophic states within the solution space

SAFETY CAGE ARCHITECTURE

Designed for the specific risk mitigations / scenarios

See Section 6.4.4.3.2.5, page 74ff

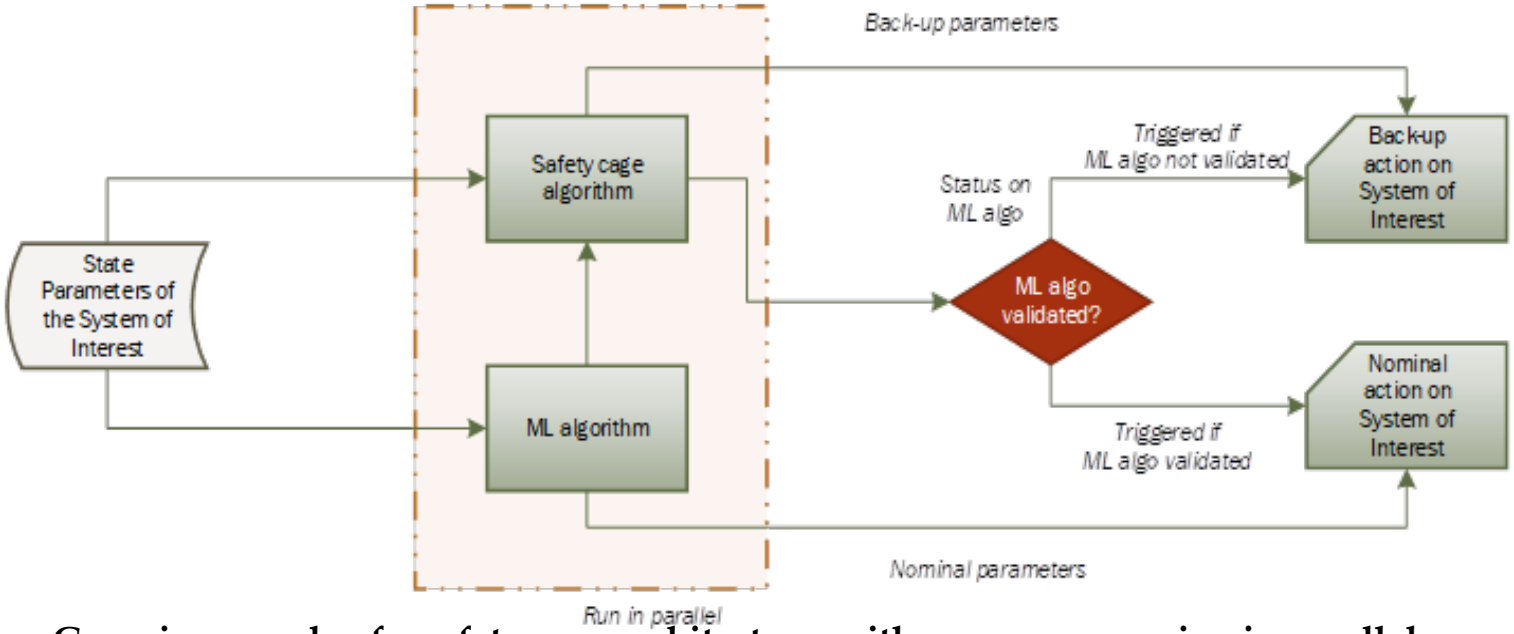
3. System Testing and Qualification

Building blocks and tools

A safety cage is an architecture designed to the specific risk w.r.t. requirements, acceptable mitigation and/or mission scenario

Building blocks and tools:

- Symbolic AI/ Implementation of rules/logic gates
- Reference Hardware
- Physical gate/filter
- Data Distribution check
- Monitoring schemes
- Reference software.
- Redundant Software system
- Backup System “Man-in-the-Loop”



Generic example of a safety cage architecture, with processes running in parallel.

SAFETY CAGE ARCHITECTURE

Designed for the specific risk mitigations / scenarios

Examples

See Section 6.4.4.3.2.5, page 75ff and 80ff

We‘d like to thank you!

Active Member	Affiliation
Uwe Brauer Mads Hedegard	Airbus
Luis Manzilla Evridiki Ntagiou Cora Janse Jan Reerink Christophe Honvault Manrico Fedi Casas Alessandro Donati Jonathan Woodburn	ESA
Stephan van Beek Konstantin Dmitrie	Mathworks
Lionel Brayeur	Spacebel
Bruno Ferrard Serge Le Gonidec	Ariane Group
Benoit Garcon	CNES
Michael Bädorf	DLR

Active Members of the ML/AI Working Group

3. System Testing and Qualification

Reliability, safety, and performance of systems/AI systems

- The qualification of space systems (independent of the use of AI and Machine Learning) is essential to ensure their safety, reliability, and effectiveness for a given mission.

	Software or Software System	AI System
Verification	<software> process to confirm that adequate specifications and inputs exist for any activity, and that the outputs of the activities are correct and consistent with the specifications and input (ECSS-E-ST-40C)	Assessing the model's performance on **unseen data** and ensuring that it generalizes well to new situations.
Validation	<software> process to confirm that the requirements baseline functions and performances are correctly and completely implemented in the final product (ECSS-E-ST-40C)	Confirmation of correctly implementation of **models** and that the training and testing data are representative of the intended problem domain, and training/testing procedure are accurately followed.
Certification	Evaluation and attesting that a software system, component, or process meets certain predefined standards, regulatory requirements, or industry best practices.	Considering not only their performance but also their **safety, reliability, and ethical implications,** compared to general usage (w.r.t regulations, etc.).
Qualification	The process of demonstrating that a software system, component, or process is fit for its intended use, often within a specific environment or context. (This process often contains verification and validation processes).	Comprehensive evaluation of performance, safety, and reliability in the specific context in which it will be used. May involve stress-testing the models under a range of conditions, validating their performance in real-world scenarios, and ensuring that they are robust to changes in input data, environmental factors, and other potential sources of uncertainty. **Additionally, qualification may require addressing the interpretability, explainability, and transparency of AI/ML systems to ensure they can be trusted by users and stakeholders**.

- For space systems, due to changing nature from mission to mission, emphasis is often placed on qualification over certification w.r.t. mission requirements.

Reliability, safety, and performance

Verification, Validation, certification and qualification as means of assurance.

ECSS