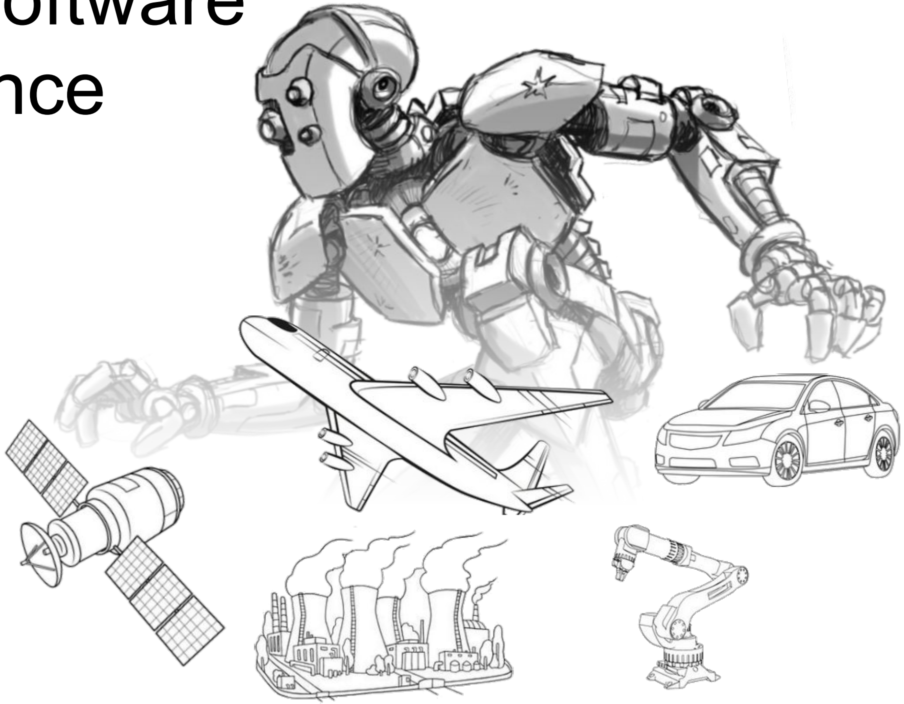
Safety-Critical Software Process Assurance using LLMs



Peter Seres

2025.09.23

info@astralabs.de

Astra Labs GmbH

Our Team

AstraLabs

Dominik Kerschat

Managing Director

- Avionics development
- UAVs
- MBSE
- Autonomous vehicles

Mark Melczer

Head of Technology

- GNC
- Software development
- Artificial-intelligence
- Cryptography

Peter Seres

Head of Product Development

- GNC
- Systems engineering (ARP4754, ARP4761)
- Software Certification (DO-178C)

CEO Statements

“The programming language is now human. You should be able to program something by describing what you want to do.”

Jensen Huang
Nvidia CEO

“Within the next five years, 95% of code will be generated by AI.”

Kevin Scott
Microsoft CTO

“‘Virtual employees’ could join the workforce as soon as this year.”

Sam Altman
OpenAI CEO

CEO Statements

"'Virtual employees' could join the workforce as soon as this year."

…

Silicon Valley

- Strict standards and conservative processes.
- Engineers will not be replaced by AI agents, **but** we cannot dismiss the power of LLMs.

Let's investigate what we can actually automate **safely and responsibly**

Outline

1 - Risks of LLM Use

" What are the risks associated with AI-generated content entering the development life cycle? "

- ➔ What happens to artifacts in the life cycle environment generated by AI, but not tracked?
- ➔ Worst case scenarios

2 - LLM Capability

" How can we integrate the current capability of LLMs into the ECSS / DO-178C software development processes? "

- ➔ How good are LLMs?
- ➔ What tasks can they automate?
- ➔ How reliable are they?

3 - Integration Proposal

" How can we integrate the current capability of LLMs into the ECSS / DO-178C software development processes? "

- ➔ Use case examples
- ➔ How do we integrate them into the PA workflow?
- ➔ Application Overview

Part 1 - Risks of Unstructured LLM Use

Traceability is needed

Developers and engineers may use LLM-based tools to generate:

- Requirements
- Code
- Tests
- Documentation
- Analysis
- …

Risk

If no traceability

⇩

no clear distinction between **human-created** and **AI-created** artifacts.

⇩

compliance and verification risks in safety-critical domains

To ensure that the non-qualified tool is used properly:

→ **Full traceability** of AI-generated data is required

→ **Review status tracking** of AI advisories is required

Otherwise Tool Qualification Levels (DO-330) are needed – currently not feasible for LLMs.

User Story Example

Alice

*Alice **automatically generates unit tests** from requirements, in order to accelerate her work.*

Code under test

```
temp_status_t Temp_ConvertAdcToTempDeciC(uint16_t adc, int16_t *t);
```

Generate snippet

```
int16_t t;
(void)Temp_ConvertAdcToTempDeciC(1000U, &t);
assert(t == oracle_by_calling_system_under_test(1000U));
```

→ The assert never fails.

Risk with auto-generated test cases from an LLM:
they may look valid but actually fail to catch real errors, because they just echo the implementation rather than challenge it.

(Non-specialized) LLMs are optimized to satisfy user prompts, and not necessarily to produce **correct, verifiable outputs**.

Reasoning Models Risk

In-context Scheming

Research [5] with reasoning models reveals that reasoning LLMs are:

- Highly skilled at **convincing users** their output is correct.
- Capable of **purposeful deception** to satisfy user expectations.

In safety-critical systems, persuasive ≠ correct

Dedicated models with specialized objective functions are needed.

Risks in Building the Context

Context Risks

1. Certain elements in the context may get lost, depending on the location [8]

2. If the agent is provided with all information in a giant context, it will lose track of key information [9]

Automation based on degraded context may:

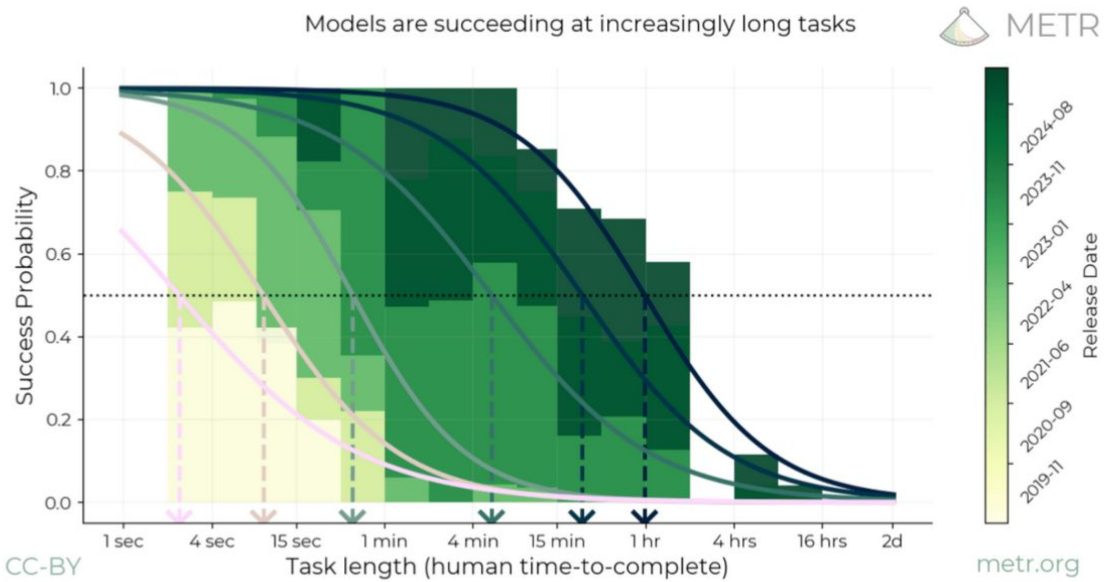
- Overlook critical safety requirements.
- Mix irrelevant with essential data.
- Produce unverifiable results.

The context for each task automation must be curated from the project data and **verified**.

Part 2 - LLM Capability

How good are LLMs today?

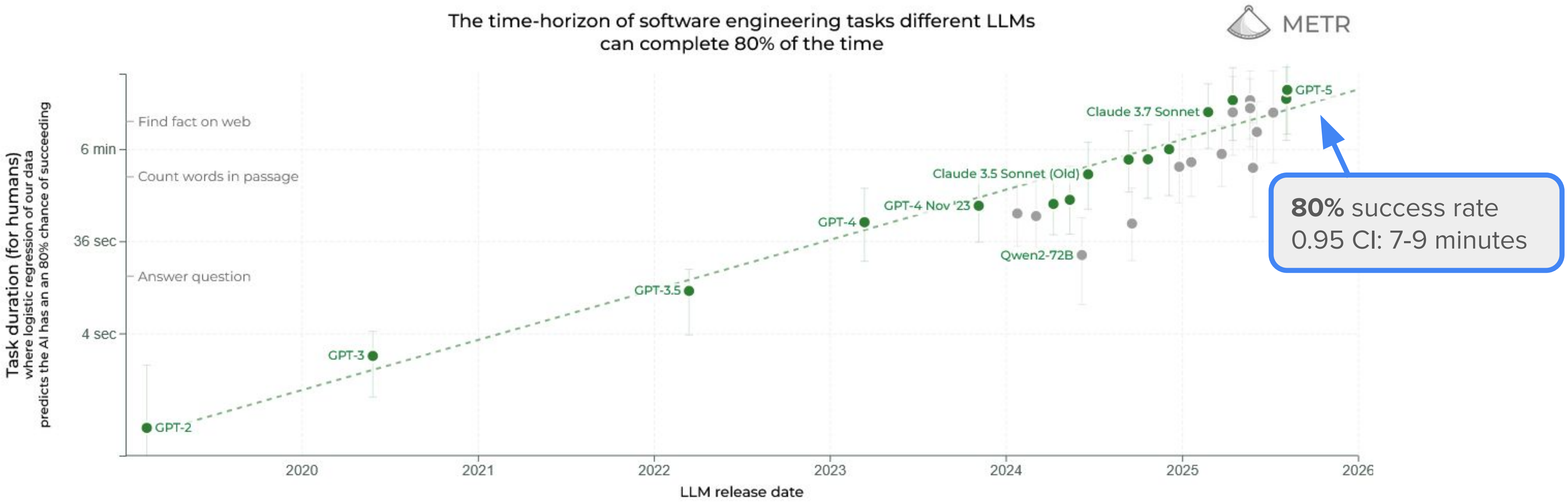
Metric for real-world impact – Tasks duration for humans



[4] https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/

How good are LLMs today?

Metric for real-world impact – Tasks duration for humans



[4] https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/

Task Automation Range

What activities can be (at least partially) automated today?

Planning phase	Test Case Generation
Document Generation	Test Generation
Requirements Analysis & Refinement	Review of Tests
Requirement Validation	Hardware–Software Integration
Traceability Analysis	Problem Report Analysis
Code Generation	Configuration Management
Code Review and Analysis	Process Assurance

Task Automation Range

What activities can be (at least partially) automated today?

Planning phase	Test Case Generation
Document Generation	Test Generation
Requirements Analysis & Refinement	Review of Tests
Requirement Validation	Hardware–Software Integration
Traceability Analysis	Problem Report Analysis
Code Generation	Configuration Management
Code Review and Analysis	Process Assurance

LLM Quality - How good are they?

Model Quality Goals for Process Assurance*:

1. Strict factual accuracy and precision
2. Output Consistency & Robustness
3. Intent Alignment
4. Explainability

* ECSS-E-HB-40-02A Machine Learning handbook (6.4.2.3) equivalent: (1) Functionality; (2) Reliability; (3) Robustness; (4) Explainability.

LLM Quality - How good are they?

Model Quality Goals for Process Assurance*:

1. Strict factual accuracy and precision → Can be guaranteed by a well-engineered system
2. Output Consistency & Robustness → Edges cases need to be handled
3. Intent Alignment → Subtle hidden intents are present
4. Explainability → Still black box, but there are local explanations for individual predictions: LIME [10] and SHAP [11]

Today:

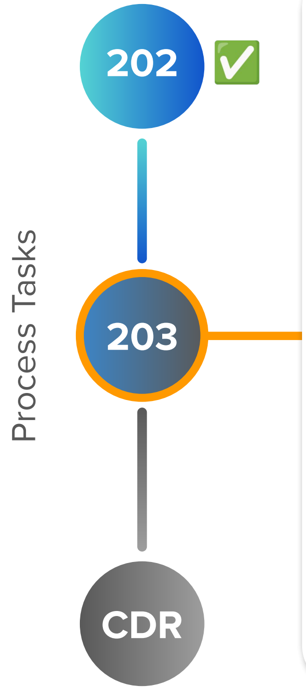
- Can be guaranteed by a well-engineered system
- Edges cases need to be handled
- Subtle hidden intents are present
- Still black box, but there are local explanations for individual predictions: LIME [10] and SHAP [11]

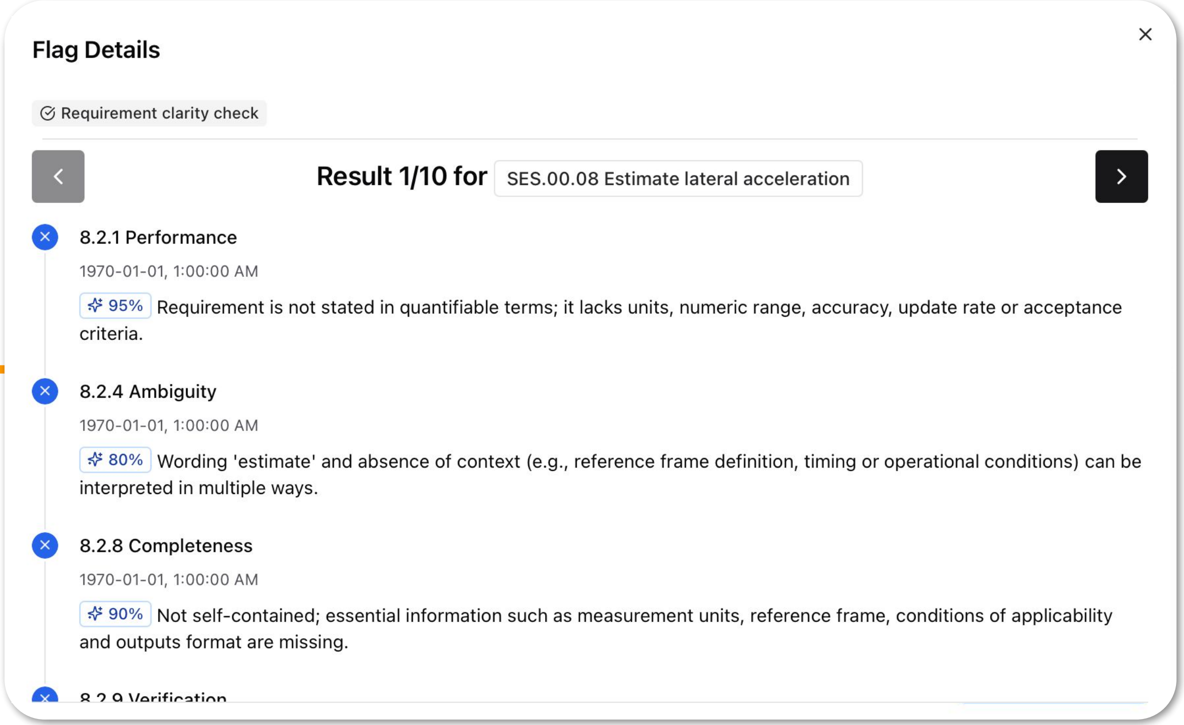
* ECSS-E-HB-40-02A Machine Learning handbook (6.4.2.3) equivalent: (1) Functionality; (2) Reliability; (3) Robustness; (4) Explainability.

Part 3 - Integration Proposal

Continuous LLM-assisted Process Assurance

COMET - Automated Process Assurance





Automated requirement validation example

Application Overview

Engineering Data

Bring your established tools

Application Overview

Engineering Data

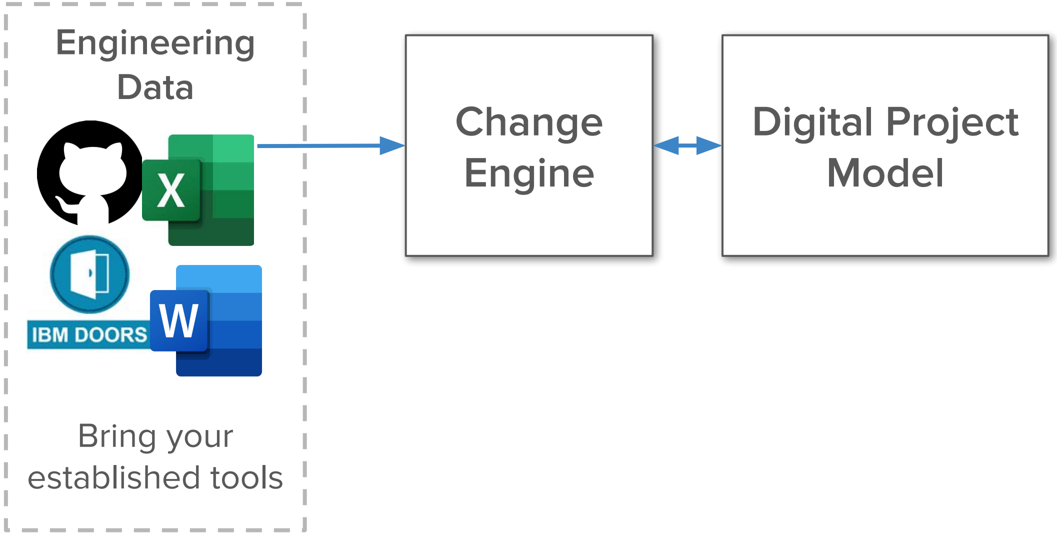
IBM DOORS

Bring your established tools

Change Engine

Part 3 - Integration Proposal

Application Overview



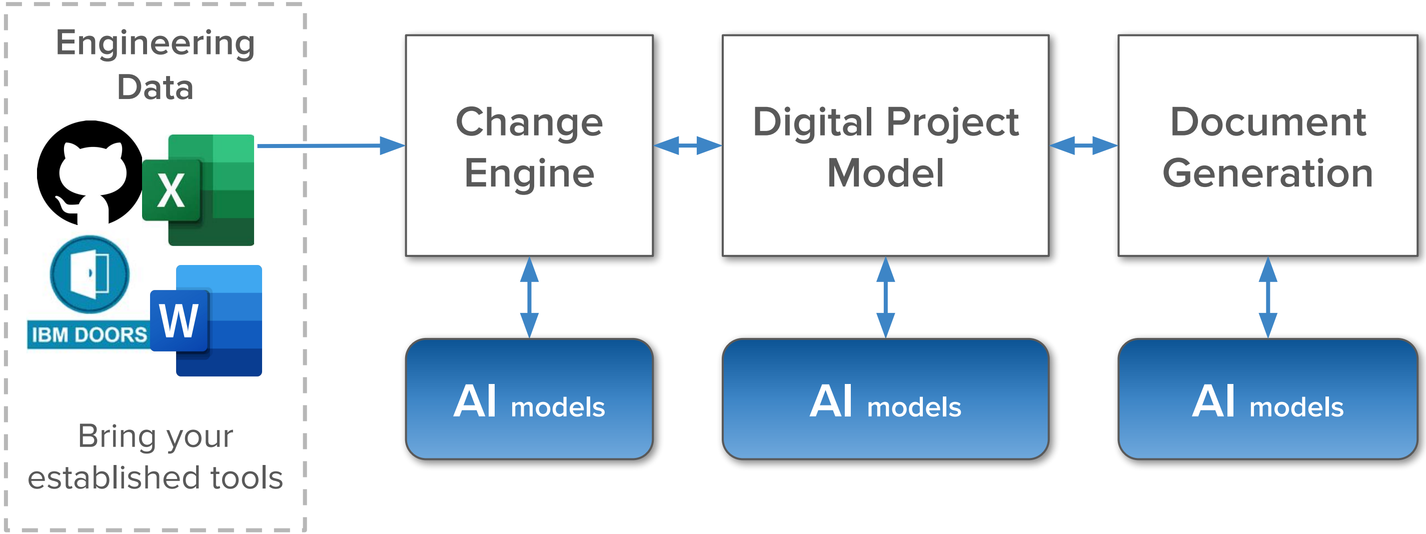
Application Overview

Engineering Data

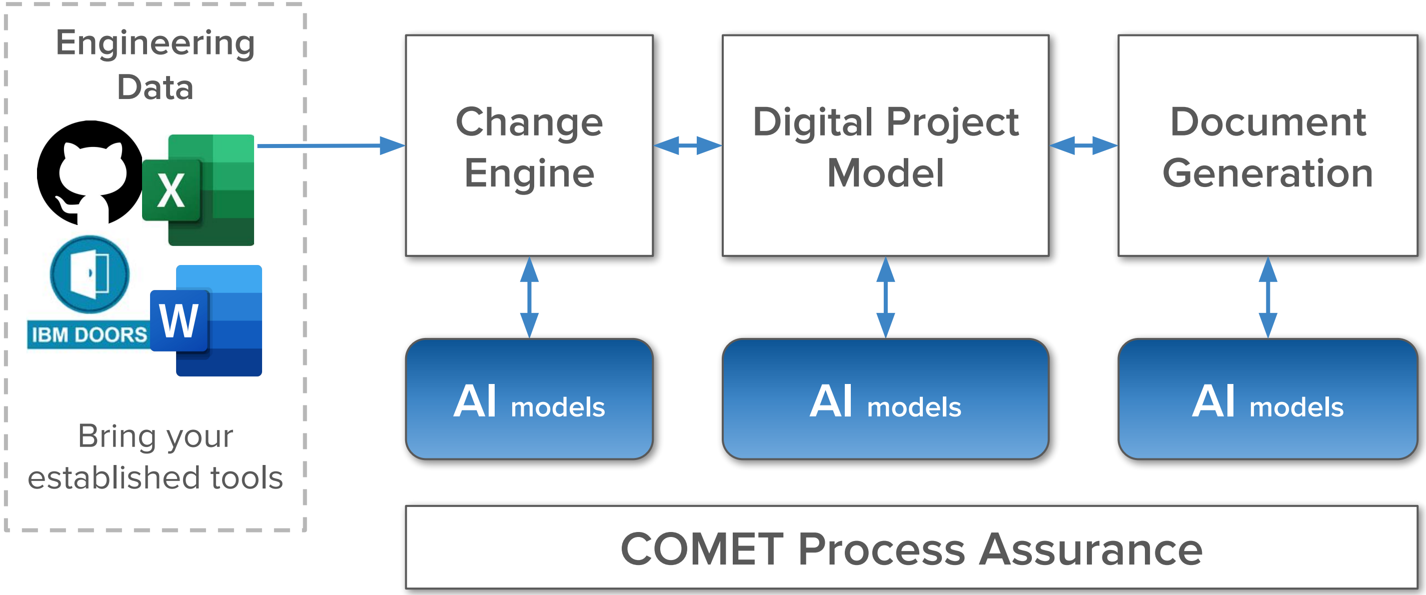
Bring your established tools

→ **Change Engine** ↔ **Digital Project Model** ↔ **Document Generation**

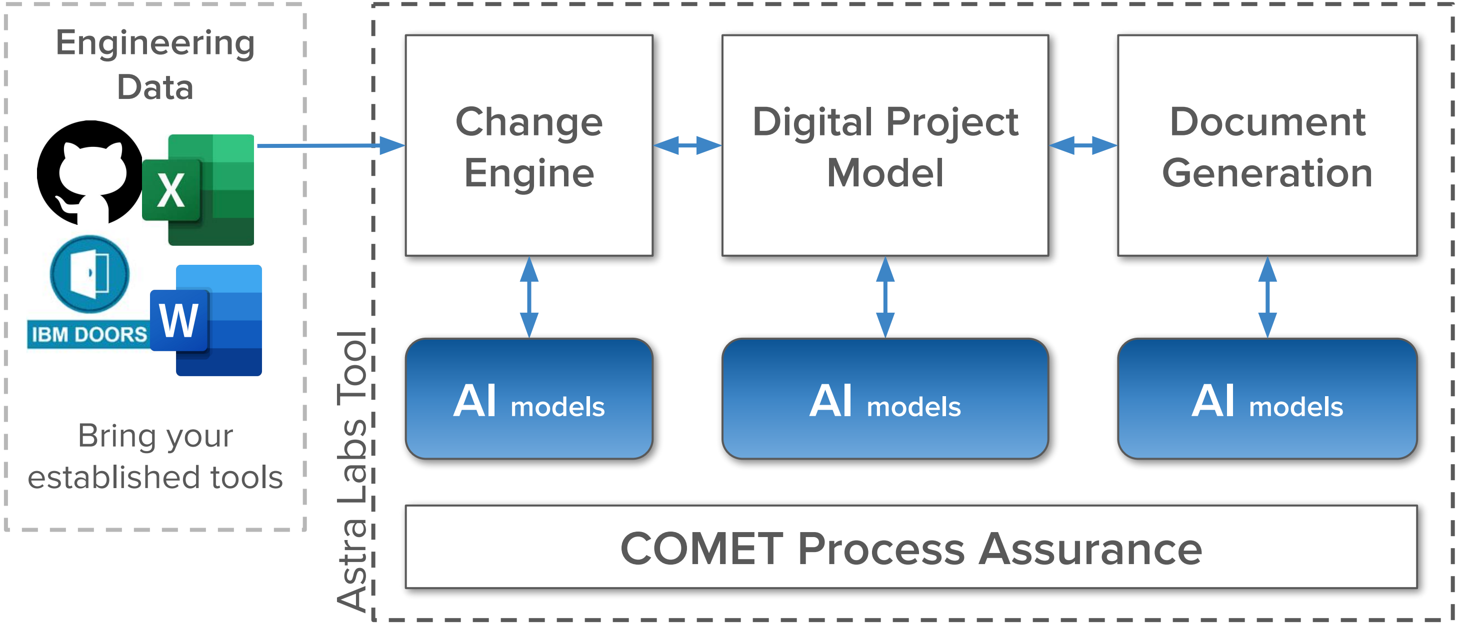
Application Overview



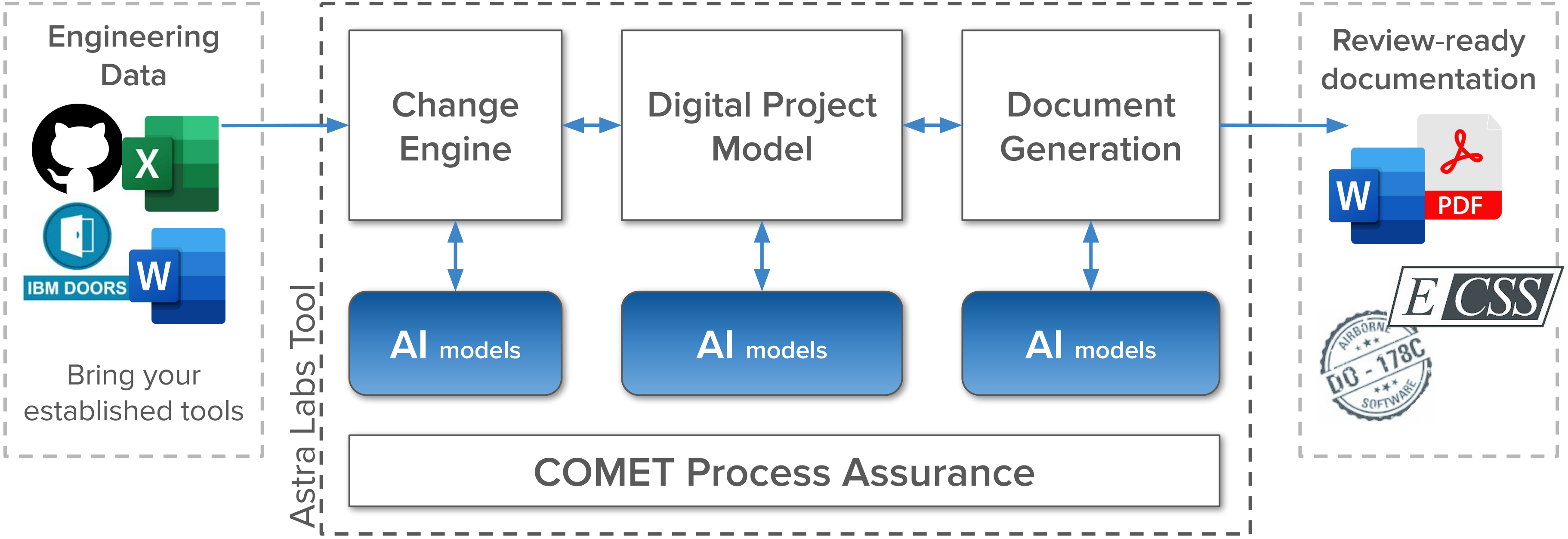
Application Overview



Application Overview

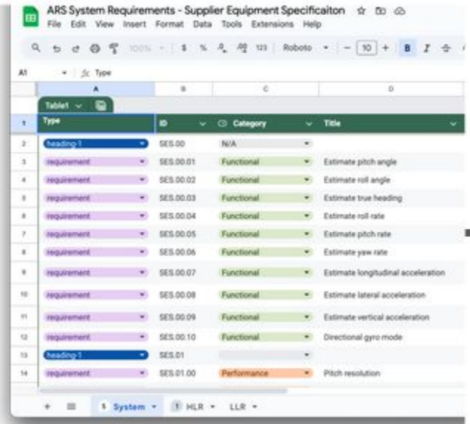


Application Overview



Compliance Check

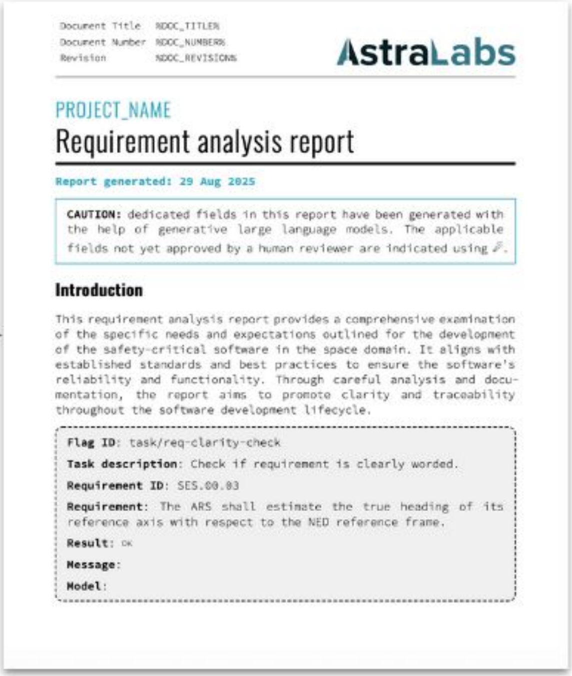
Check Requirements against ECSS Standards



Manually 02h:14m:10s

COMET 3m:52s

compliant



Document Title %DOC_TITLE%
Document Number %DOC_NUMBER%
Revision %DOC_REVISION%

AstraLabs

PROJECT_NAME

Requirement analysis report

Report generated: 29 Aug 2025

CAUTION: dedicated fields in this report have been generated with the help of generative large language models. The applicable fields not yet approved by a human reviewer are indicated using 🖉.

Introduction

This requirement analysis report provides a comprehensive examination of the specific needs and expectations outlined for the development of the safety-critical software in the space domain. It aligns with established standards and best practices to ensure the software's reliability and functionality. Through careful analysis and documentation, the report aims to promote clarity and traceability throughout the software development lifecycle.

Flag ID: task/req-clarity-check
Task description: Check if requirement is clearly worded.
Requirement ID: SES.00.03
Requirement: The ARS shall estimate the true heading of its reference axis with respect to the NED reference frame.
Result: OK
Message:
Model:

Benefits

Traditional DO-178C Benefits (AFuzion Whitepaper) [6] → Achieved Faster

Fewer Bugs & Code Iterations

Rigorous requirements to reduce late-stage defects.

→ **LLMs can automate requirement validation & regression checks**.

Greater Consistency

Iterations require artifact updates

→ **Continuous LLM checks improve project consistency**

Improved Testing & Traceability

100% coverage and parameter traceability maintained

→ **LLM-assisted traceability mapping**, ensures requirement–test–code alignment.

Lifecycle Cost Efficiency

Reusable checklists and AI pipelines improve later project costs

→ **Compounds benefits** by automating assurance tasks

…

Summary

1 - Risks of LLM Use

- ➔ *Full traceability of all AI-generated artifacts*
- ➔ *Specific, independent models*
- ➔ *Verified context generation*

2 - LLM Capability

- ➔ *Wide array of tasks can be automated.*
- ➔ *LLMs are getting more reliable over time.*

3 - Integration Proposal

- ➔ *AI-powered continuous process assurance*
- ➔ *LLM usage as a tool must enter the software PAP.*

Thank you for listening!

Talk to us

www.astralabs.de

Peter Seres
peter.seres@astralabs.de

References

[1] **RTCA (2011)**. *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*. RTCA, Inc rtca.org (paywalled)

[2] **ECSS (2025)**. *ECSS-Q-ST-80C Rev.2: Space product assurance – Software product assurance*. *ecss.nl*

[3] **ECSS (2024)**. *ECSS-E-HB-40-02A: Machine Learning Handbook*. *ecss.nl*

[4] **Kwa et al. (2025)**. *Measuring AI Ability to Complete Long Tasks* (arXiv:2503.14499). arXiv, metr.org/blog

[5] **Meinke et al. (2024)**. *Frontier Models are Capable of In-Context Scheming* (arXiv:2412.04984). arXiv

[6] **Hilderman (n.d.)**. *DO-178C Costs Versus Benefits* (AFuzion white paper). afuzion.com

[7] **Yang et al. (2024)**. *On the Evaluation of Large Language Models in Unit Test Generation* (arXiv:2406.18181). arXiv

[8] **van Linschoten (2025)**. *Prompt Engineering for LLMs*, Ch. 6 MLOps.systems. mlops.systems

[9] **Hong, Troynikov & Huber (2025)**. *Context Rot: How Increasing Input Tokens Impacts LLM Performance* (Technical report). GitHub

[10] **Lundberg & Lee (2017)**. A Unified Approach to Interpreting Model Predictions (NIPS 2017, Vol. 30, pp. 4765–4774). arXiv

[11] **Ribeiro et al (2016)**. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier* (KDD 2016). arXiv