



Webinar:

AI-supported requirements management

Development of Requirements Engineering (1/2)

1968 – The „Software Crisis“

From Software Crisis to Formal Specification:

1968/1969 NATO Conferences on Software Engineering **introduced the term *Software Engineering*** as a response to large-scale project failures.
Recognition that unclear or missing requirements were a root cause of failure.

1993 – Requirements Engineering becomes a discipline

Requirements Engineering as an Independent Discipline:

1993 – First IEEE International **Symposium on Requirements Engineering (RE'93)** establishes RE as its own research and practice domain.
1997 – UML Standardization (OMG) Model-based representation of requirements and system behavior.

1998 – Standardization of SRS quality (IEEE 830)

1998 – IEEE 830-1998 (Software Requirements Specification):

Defined: Structure of SRS documents, Quality attributes (correct, unambiguous, complete, consistent, verifiable, traceable, etc.)
→ Became a global reference for “good requirements”.

2001 – Agile Requirements Engineering

2001 – Agile Manifesto

- Shift from heavy documentation to working software
- Emphasis on collaboration and adaptability

2011 – ISO/IEC/IEEE 29148

Systems Engineering and Compliance-Driven RE:

ISO/IEC/IEEE 29148:2011 defines **consolidated requirements engineering processes** and characteristics of high-quality requirements.

Introduction of Functional Safety & Regulated Domains

Development of Requirements Engineering (2/2)



2018 – Systems perspective strengthened

2018 – ISO/IEC/IEEE 29148 (2nd Edition):

- Strengthened **integration into systems engineering lifecycle**
- Key Shift: Requirements Engineering became central to safety, compliance, and system integration.

2023–2025 – AI-supported and Continuous RE

Continuous and AI-Supported Requirements Engineering:

Continuous Requirements Engineering – integration with DevOPS, Continuous Integration, MBSE

AI and LLMs: Research and early adoption in Requirements elicitation, automation, traceability support, check for completeness

2026- 2030: (Forecast):
Autonomous & Knowledge Centric Requirements Engineering

Autonomous & Knowledge Centric Requirements Engineering

Based on current research: AI Agents as Requirements Co-Engineers; Semantic knowledge graphs replace static documents; MBSE integration; Continuous certification & Compliance automation



Researches and current studies indicates not just better text generation, but **machine-reasonable requirements ecosystems!**

Typical Challenges within Requirements Management

- **Increasing System Complexity:**

- Systems are interdisciplinary (software, hardware, mechanics)
- Multiple abstraction levels (stakeholder → system → subsystem → software)



Understanding the **impact of changes** across the systems becomes **difficult**.

- **Change Volatility**

- Agile iterations
- Customer feedback loops
- Regulatory updates



Requirements are frequently updated → **risk of inconsistency** and **outdated artifacts**.

- **Traceability Overhead**

- Bidirectional traceability required in regulated environments



Manual maintenance is time-consuming and error-prone.

- **Inconsistency & Ambiguity**

- Contradictory requirements



Errors propagate into design and test phases.

- **Siloed Toolchains**

- Requirements tools separate from: Architecture tools; Test management systems; DevOps pipelines



Governance, alignment, and visibility become **difficult**.

- **Scalability in Large Organization**

Characteristics of Modern Requirements Management



- **Lifecycle Integration**

- Integrates requirements across: Engineering, Architecture, Testing, Operations
- Supports continuous refinement



**Complexity
Tool silos**

- **Continuous Traceability**

- Impact analysis capabilities
- Bidirectional traceability across abstraction levels



**Traceability
Overhead**

- **Versioned & Incremental Requirements**

- Baselines
- Variant management



**Change
Volatility**

- **Quality Governance & Review Mechanisms**

- Standardized requirement templates
- Formal review workflows



**Ambiguity
Inconsistency**

- **Model-Based & Structured Representation**

- Integration with MBSE
- Structured attributes instead of pure text



**Complexity
Scalability**

AI-Augmented Requirements Management (Emerging)



From Manual Control to Intelligent Assistance

AI Use Cases:

▪ Automatic Ambiguity Detection

- Detects vague terms (“fast”, “user-friendly”, “sufficient”)
- Identifies missing measurable criteria
- Suggests reformulations based on predefined requirement template



- **Higher** requirement **quality** before review
- **Reduced rework** in later phases

▪ Semantic Similarity Checks

- Embedding-based similarity detection across large requirement sets
- Identifies:
 - Duplicate requirements
 - Overlapping statements
 - Potential contradictions



- **Prevents redundancy**
- Reveals **hidden inconsistencies**
- **Supports harmonization** across teams

▪ Consistency Verification

- Cross-checks requirements across abstraction levels (Stakeholder → System → Software → Test)
- Detects logical contradictions
- Identifies missing refinements or broken trace links
- Supports compliance validation (e.g., safety constraints)



- Reduced **hidden inconsistencies**
- Improved **traceability integrity**

AI-Augmented Requirements Management (Emerging)



From Manual Control to Intelligent Assistance

AI Use Cases:

▪ Requirement Derivation Support

- Automatically derives:
 - System requirements from stakeholder requirements
 - Software requirements from system requirements
 - Initial test cases from requirements
- Highlights missing decomposition steps



- **Reduced manual effort**
- **Accelerated** engineering workflows
- **More complete requirement** hierarchies

▪ Smart Impact Analysis

- Analyzes change requests semantically
- Identifies potentially affected:
 - Requirements
 - Architecture elements
 - Test cases
 - Risk items
- Prioritizes impact probability



- **Faster change evaluation**
- Reduced **review bottlenecks**
- **Improved decision support**



AI-supported requirements management

© 2026 A. Schmidt • Innovigation

Organizational Topics

- Participants shall be muted during presentation
- We use “F&A” (Q&A) functionality to collect questions. Questions will be discussed at the end of the presentation.



- Please note that the webinar will be recorded!



Innovigate GmbH

Innovative product development and process optimization

As a management consultancy, we accompany product developments and processes through project support, workshops and training courses in the direction of innovation, quality and cost reduction.

In doing so, we rely on the combination of specialist knowledge and proven methods, which you can learn about in our practice-oriented courses and workshops and apply directly in your projects.

We also offer internationally recognized certificate training courses and are members of expert networks.



Our formats:

- Trainings
- Consulting & Workshops
- Project support
- Coaching



R&D Consulting & Training

Innovation & Business Excellence

Consulting Support

Advanced Development and Series Development

- Design for Six Sigma (DfSS)
- TRIZ Innovation Methods
- Design Thinking
- Lean Development
- System Architecture
- Product verification & validation

Series Ramp-Ups

- Six Sigma (DMAIC)
- TRIZ Innovation Methods
- Lean Manufacturing
- Risk Management
- Problem Solving & 8D Reporting

Systematic Technical Problem Solving

- Methodical & operational management of task forces
- Problem solving workshops
- Workshops for proactive failure prevention

AI-Powered Innovation and Design for Excellence



Team



Achim Schmidt

Managing Director Innovigate GmbH

Dipl.-Ing. Electrical Engineering

Six Sigma Master Black Belt

TRIZ-Level 3 Expert

- Innovation & Methodology Consulting
- AI-Driven Innovation & Product Development
- Lean Six Sigma & Design for Six Sigma
- Training, Workshops & Change Management



Manuel Schmidt

Softwareengineer

B.Sc. Business informatics

M.Sc. Artificial Intelligence

- Full-Stack Software Engineering
- Data-Driven Applications & Systems
- Autonomous & Intelligent Systems
- DevSecOps & Secure Infrastructure



Tracelane motivation + history

- Many smaller companies still manage requirements using Excel spreadsheets
- Existing tools are often too complex and too expensive
- Clear need for a simple and affordable requirements management solution
- Vision: Combine a modern affordable, easy-to-use requirements engineering tool with AI

- 2025-02: Start of development
- 2025-03: First MVP
- 2025-04: Integration of first AI features
- 2025-07: AI Agent mode
- 2025-08: Version Control and Baselining
- 2025-10: Start of cooperation with Process Fellows
- 2026-01: First public on-prem release
- 2026-02: Ollama Integration



Tracelane principles

Cloud-Native

Access Tracelane instantly in your browser. No installation, no setup headaches, fully containerized.

AI integration

Leverage AI to accelerate requirement creation, identify logical conflicts and analyze consistency across system levels

On-Prem Focus

Deploy on your own infrastructure for full data control, compliance, and security, without sacrificing usability.

Affordable

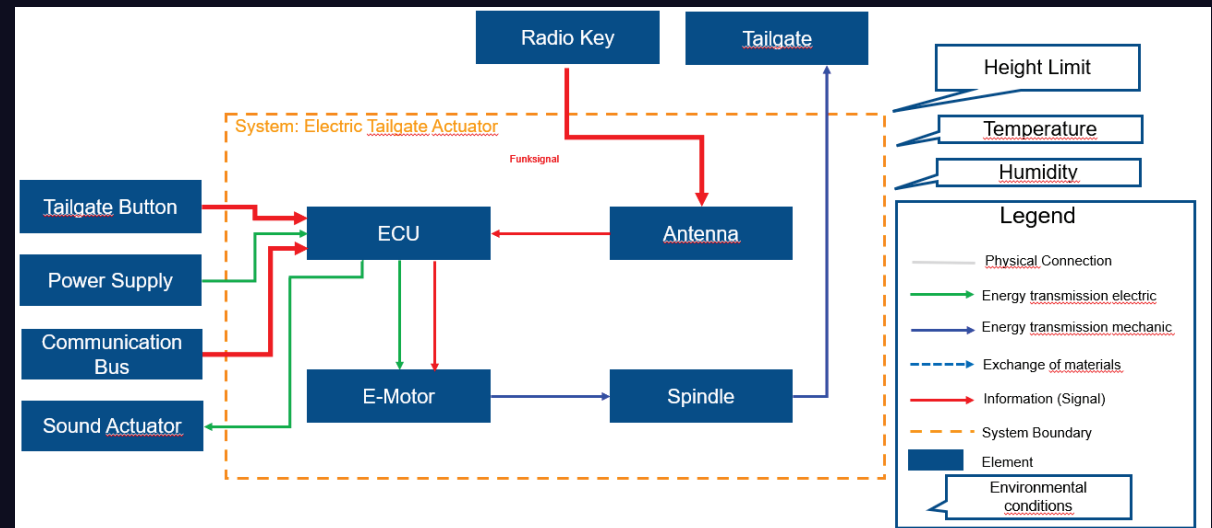
Requirements management at a fraction of the cost of traditional tools. Save small companies from being stuck with Excel!



ETG Example

Electronic Tailgate

- Our dummy project is about developing a electronic tailgate for a car
- The system consists of multiple system elements
- We create use cases and requirements on (sub-)system and domain level



UC1 – Ambiguity Detection and Correction

- The AI rewrites a poorly written requirement, makes it measurable and specific.

SYRS-1033 ✕

Headline: Behavior in operating mode "Opening" Type: Functional Req

Content

Normal text **B** *I* Editor Raw

should open continuously

Improve

SYRS-1033 ✕

Headline: Behavior in Operating Mode "Opening" Type: Functional Req

Content

Normal text **B** *I* Editor Raw

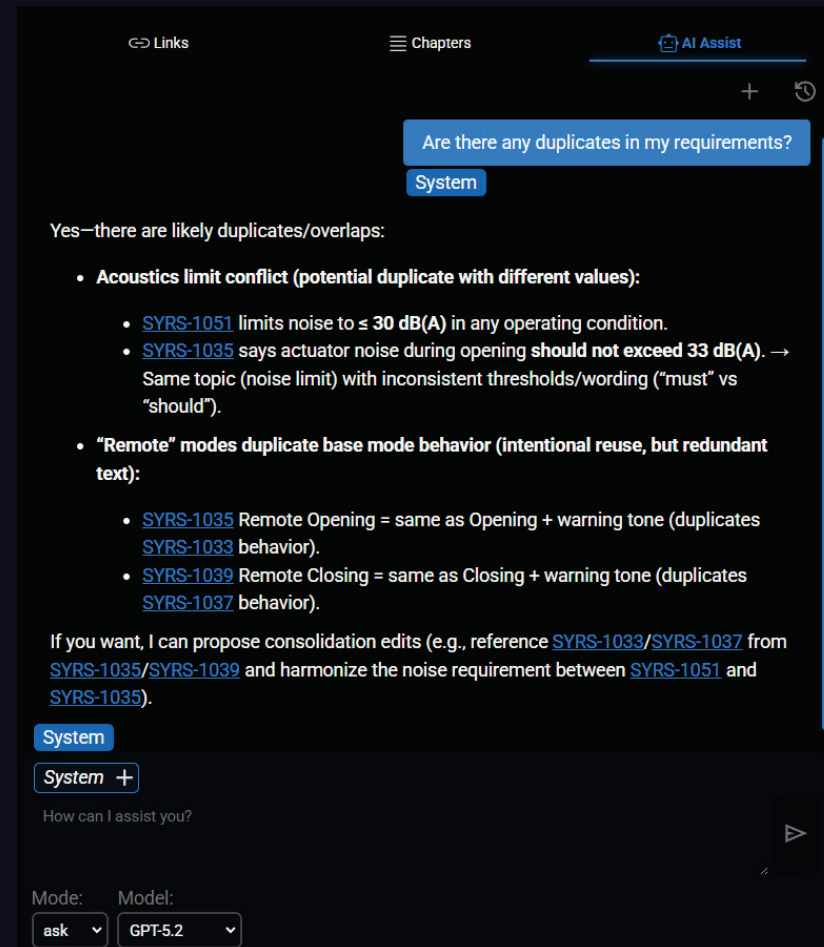
The system shall open the tailgate at a constant speed, without pauses or interruptions, until the fully open position is reached, provided that the operating mode remains set to Opening.

Revert Accept



UC2 – Semantic Similarity Checks

- The AI can identify duplicate requirements, overlaps and ambiguities
- It successfully detects that acoustic limits for the ETG actuators are defined twice in this example



The screenshot shows a dark-themed interface for an AI assistant. At the top, there are navigation links for "Links", "Chapters", and "AI Assist". A search bar contains the question "Are there any duplicates in my requirements?" and a "System" button. Below the search bar, the AI response is displayed, starting with "Yes—there are likely duplicates/overlaps:". The response lists two categories of issues:

- **Acoustics limit conflict (potential duplicate with different values):**
 - [SYRS-1051](#) limits noise to ≤ 30 dB(A) in any operating condition.
 - [SYRS-1035](#) says actuator noise during opening **should not exceed 33 dB(A)**. → Same topic (noise limit) with inconsistent thresholds/wording ("must" vs "should").
- **"Remote" modes duplicate base mode behavior (intentional reuse, but redundant text):**
 - [SYRS-1035](#) Remote Opening = same as Opening + warning tone (duplicates [SYRS-1033](#) behavior).
 - [SYRS-1039](#) Remote Closing = same as Closing + warning tone (duplicates [SYRS-1037](#) behavior).

Below the list, the AI suggests: "If you want, I can propose consolidation edits (e.g., reference [SYRS-1033/SYRS-1037](#) from [SYRS-1035/SYRS-1039](#) and harmonize the noise requirement between [SYRS-1051](#) and [SYRS-1035](#))." At the bottom, there is a "System" button, a "System +" button, and a "How can I assist you?" prompt. The interface also shows "Mode: ask" and "Model: GPT-5.2" dropdown menus.



UC3 – Consistency Verification across System levels

- Tracelane can analyze existing links between requirements semantically and point out links which might be incorrectly set
- It successfully detects, that environmental system requirement was linked to a software requirement about blockage detection

The screenshot displays the Tracelane AI Assist interface. At the top, there are navigation tabs for 'Links', 'Chapters', and 'AI Assist'. A blue question box asks: 'Are there any incorrect Links between the System and Software modules?'. Below this, there are two buttons: 'System' and 'Software'. The response indicates that one link is likely incorrect: 'SYRS-1053 (temperature range) → SWRS-1019 (blockage detection)'. It explains that this link is likely incorrect because temperature range does not directly specify blockage detection behavior. Below this, it states 'The existing links look plausible:' and lists three consistent links: 'SYRS-1046 → SWRS-1015', 'SYRS-1018 → SWRS-1014', and 'SYRS-1018 → SWRS-1021'. At the bottom, there is a 'System +' button, a text input field 'How can I assist you?', and dropdown menus for 'Mode: ask' and 'Model: GPT-5.2'.



UC4 – Smart Impact Analysis

- Tracelane can analyze semantically which lower level requirements would be affected by a requirements change.
- Traditional systems in contrast would just flag all downstream requirements which are actively linked
- In this case, it successfully detects that the Software Requirement regarding the “Standby” mode would be affected by the change on system level.

The screenshot displays the AI Assist interface with the following content:

- Navigation: Links, Chapters, AI Assist (active), +, ↻
- Query (highlighted in blue): "When we change 'SYRS-1023', that the system state shall be changed to 'Standby' within 0.5s instead of 0.3s which downstream software requirements would be affected?"
- Context: System Software
- Analysis: "Changing [SYRS-1023](#) ('Remote Opening' → 'Standby' within X s) from **0.3 s** to **0.5 s** would impact downstream SW requirements that specify the same **300 ms** timing for stopping actions when entering **STANDBY**, namely:"
- Result (bullet point):
 - [SWRS-1011](#) (Standby): "all movements are stopped after at latest **300 msec**"
- Summary: "No other linked SW requirements in the provided context explicitly reference the 0.3 s / 300 ms 'Remote Opening leave' timing."
- Context: System Software
- Input field: System +
- Placeholder: "How can I assist you?"
- Mode: ask (dropdown)
- Model: GPT-5.2 (dropdown)



UC5 – Requirements Derivation support

- When the user enables “Agent” Mode, the AI is able to derive Sub-System requirements from higher level requirements which can be used as starting point to accelerate the requirements breakdown

The screenshot displays a requirements management interface. On the left, a table lists requirements with columns for ID, Type, and Content. Each requirement has 'Reject' and 'Accept' buttons. The requirements listed are:

ID	Type	Content
AI-d2b05b86	FR	End-position detection interface The hardware shall provide sensing (direct or indirect) to determine at least: Fully closed, Fully opened, and intermediate position to support reaching a configured target opening angle. (Derived from [SYRS-1023], [SYRS-1015], [SYRS-1018], [SYRS-1021])
AI-89c518da	NFR	Position sensing plausibility The hardware architecture shall support plausibility checking of position/speed signals (e.g., redundant sensing or independent estimation) to detect sensor faults that could cause unintended motion.
AI-0d9ab174	TXT	Obstacle / anti-pinch support (HW enablers)
AI-439489c0	FR	Motor current measurement The motor drive hardware shall measure motor current with sufficient range and bandwidth to support obstacle detection and current limiting during opening/closing.

On the right, the 'AI Assist' panel is active. It contains a text box with the instruction: 'Please have a look at the System requirements (SYRS-*) and derive Hardware requirements. Also have a look at the current Mechanical requirements, so that we do not have an overlap in between the two.' Below this are buttons for 'System', 'Mechanical', and 'Hardware'. A paragraph of text explains: 'Derived HW requirements from SYRS-* with focus on electrical/electronic HW (ECU, motor drive, sensing, power, comms). Mechanical constraints already covered in MERS (e.g., max extension, manual force, weight, acoustics, temp) are not duplicated; instead HW requirements focus on enabling/monitoring/control to meet those system-level requirements.' Below this is a 'Suggestions: 30 pending' indicator and 'Accept all' / 'Reject all' buttons. At the bottom, there is a 'Hardware +' button, a 'How can I assist you?' prompt, and dropdown menus for 'Mode: agent' and 'Model: GPT-5.2'. A yellow arrow points to the 'Mode' dropdown.



Tracelane outlook

- ReqIF import/export for full compatibility with DOORS, Polarion, Jama Connect
- Integration with Jira
- Integration with Git
- Proactive AI suggestions and analysis
- Review and approval workflows



Thank you for your Attention!

Innovation

Let's navigate towards innovation

<https://innovigation.com/>



<https://processfellows.de/>

