# Scaled Agile for Safety-Critical Systems

**Jan-Philipp Steghöfer**, Eric Knauss, Jennifer Horkoff,
Rebekka Wohlrab

SOFTWARE CENTER
CHALMERS | GÖTEBORGS UNIVERSITET

Originally presented at PROFES 2019 (Best Paper Award)

November 23, 2021

# Motivation

- R-Scrum and SafeScrum help organisations combine documentation needs and rigour with an agile approach
- Provide no support for scaling
- SAFe and LESS are all about scaling but have no support for safety-critical systems

# Research Questions

RQ1: Which common principles and practices can be derived from existing approaches for agile development of safety-critical systems?

RQ2: Which practical challenges exist when applying these principles and practices in a large-scale industrial setting?

# Methodology

1. Prepare overview of SafeScrum and R-Scrum
2. Focus group with three industrial experts
   - Present overview
   - Brainstorming of challenges
   - Topical sorting
3. Member checking of summarised results

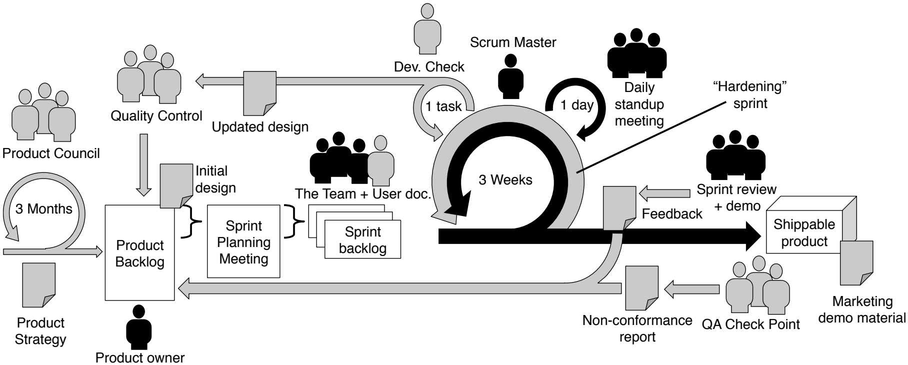
**Context of industry experts:**

- Domains: automotive and medical devices
- Highly-configurable systems ($>$10000 features)
- Large organisations ($>$10000 employees)

# Outline

1. Regulated Scrum and SafeScrum
2. Open Challenges According to Industry
3. Outlook

# Regulated Scrum [1]




## Main Approaches

- Continuous Compliance
- Hardening Sprints
- Living Traceability

# Regulated Scrum [1] (cont.)

## Continuous Compliance: each sprint audited by QA

- Audit completed within three days after sprint end
- Allows potential delivery after every sprint

## Hardening Sprints

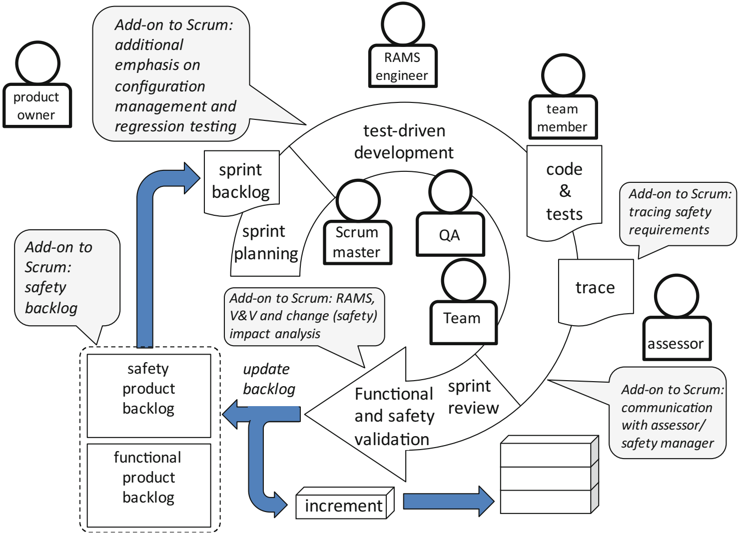
- Run directly before a product release
- Close all open issues
- Finalise user documentation, deployment infrastructure, marketing material, etc.
- DoD includes regulatory compliance

# Regulated Scrum [1] (cont.)

## Living Traceability

- Printed spreadsheets continuously updated
- Tool-chain ensures traceability between requirements and code
- Update of documentation part of code reviews
- "Initial requirements can be traced to stories, and in turn to tasks and sub-tasks, to design documentation, to source code, to code reviews, to builds, to unit tests, to rework and bug- fixes, to function and system testing, to production code."
- Transparency greatly simplifies process audits

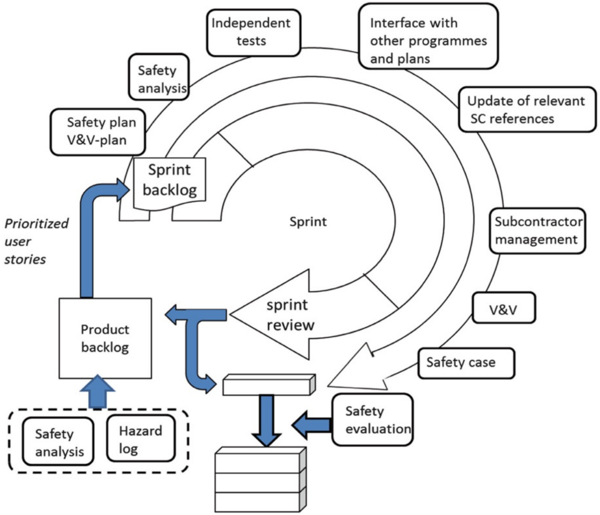
# SafeScrum [2]




## Main Approaches

- Separate Safety Backlog
- Traceability
- Include assessor in work
- Include safety CIA in each sprint

# SafeScrum [2] – Additional Activities




## Main Approaches

- In parallel with dev cycle
- Performed by RAMS engineer
- Update safety and V&V plans
- Run safety and risk analysis
- Maintain agile safety case
- Perform safety validation in each sprint

# Commonalities

Regulated Scrum and SafeScrum share some principles:

- focus on traceability
- safety as an ongoing set of activities
- shared responsibility of the team
- involvement of assessors or auditors in ongoing development

# Open Issues

**Mixed criticality:** safety-critical parts of products need to be developed with more ceremony than parts that are not safety-critical

**Automation:** automate generation of "proof of compliance" documentation within complex CI/CD tool-chain

**Scaling safe Scrum:** combining the scalability of SAFe with the safety features of Regulated Scrum or SafeScrum for multi-team projects

# Outline

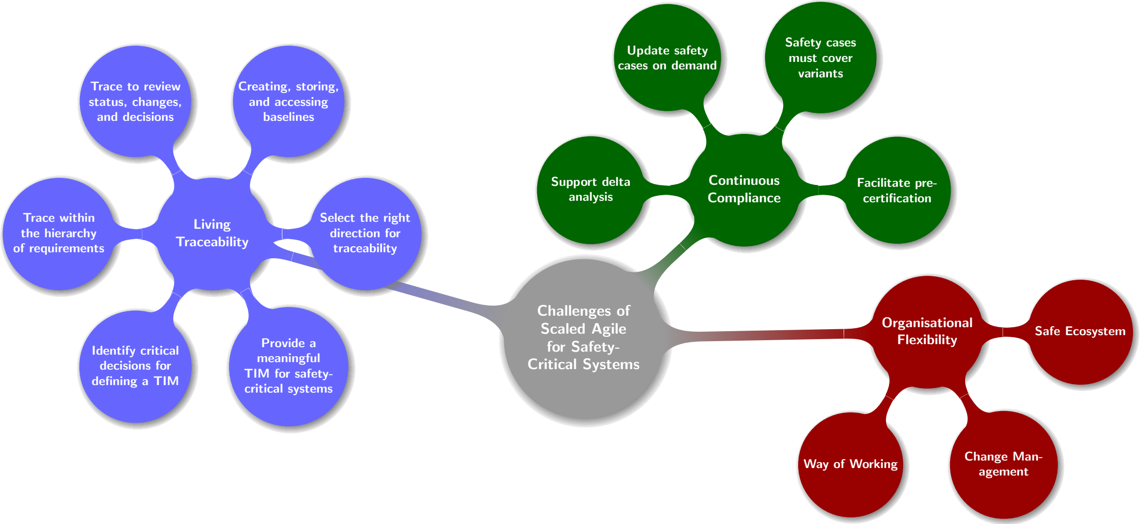
# Areas of Interest

**The foundation: living traceability.** Continuous creation, maintenance, and deletion of trace links to enable construction of safety cases on demand.

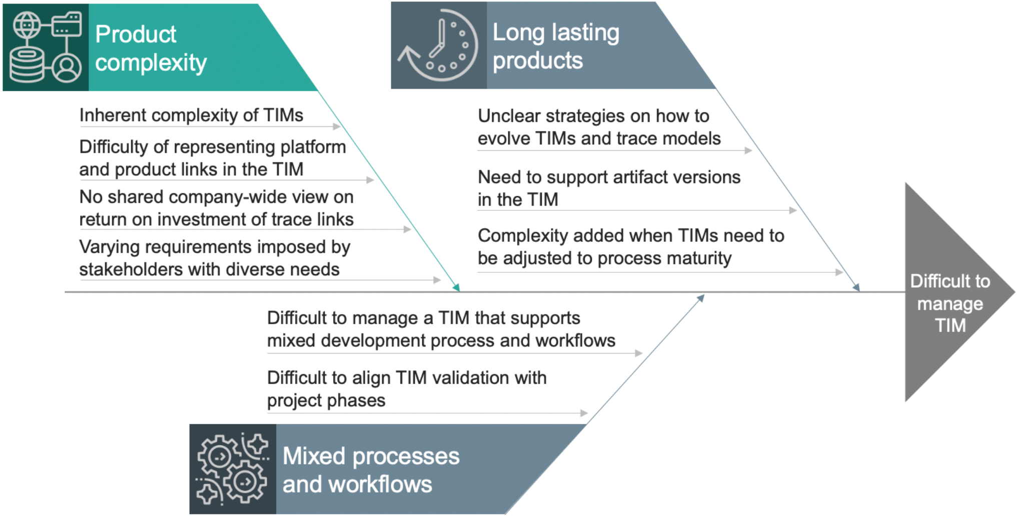
**The goal: continuous compliance.** Continuous production and maintenance of required safety arguments to ensure compliance can be proven at any point in the development process.

**The next step: organisational flexibility.** Establish an *ecosystem* of components for exchange with suppliers, enable *change management* and a *way of working* with safety artifacts.

# Overview of Challenges



# Living Traceability – Challenges of TIM construction [3]



# Living Traceability – Design Decisions in TIM construction [4]

## Critical design decisions and their drivers

| Decision | TIM | Driver |
|---|---|---|
| Coverage of artifacts | TIM I | *Purpose, minimal traceability information* |
| Concreteness of artifact types | | *Applicability, inclusion in existing processes* |
| Rationale and storage | | *Adherence to a safety standard* |
| Coverage of artifacts | TIM II | *Adherence to a safety standard, purpose* |
| Internal trace links | | *Light-weight documentation of traceability information, purpose* |
| Trace direction | | *Parallel evolution of artifacts and traceability information* |
| Granularity | | *Purpose, minimal traceability information, ongoing development* |
| Tooling | | *Applicability, intended reuse in timing/safety analyses* |
| Design approach | TIM I<br>TIM II | *Process-driven*<br>*Work product-driven* |
| Artifact focus | TIM I<br>TIM II | *Role-focused*<br>*Type-focused* |

## Clear and objective criteria for the evaluation of design alternatives

| Criterion | Defining Questions and *Possible Values* |
|---|---|
| **Stated Purpose** | Is the purpose of the TIM clearly stated (*defined*)? Are the different stakeholders and their respective needs identified in that purpose (*fully defined*)? |
| **Coverage** | Does the TIM provide *partial* or *full* coverage of the artifacts required to fulfill its purpose? |
| **Specificity** | Is the TIM *general purpose*, *specific to a purpose*, or even *highly specific* to a certain team, organization, or system? |
| **Design Approach** | Is the starting point of TIM design the *process* or the *work products*? |
| **Artifact Focus** | Are the *roles* of the artifacts or their *type* reflected in the TIM? |
| **Mapping** | Does the TIM map to the work products in a *direct* way or is an *indirect* mapping necessary, e.g., because not all concepts in the TIMs map to artifact types and it is not unambiguous which elements of the TIM represent trace link types? |
| **Typing** | Are the traceable artifact types identified by generic types (*weak*)? Or are traceable artifact types more concretely identified via the meta-classes of the respective domain-specific languages (*strong*)? Are these levels *mixed*? |

# Continuous Compliance

*Challenge*: Ensure that safety can be proven at any given point in the development process.

- Update the *relevant part* of the safety case when changes in the system necessitate it.
- Invest the (potentially manual) work of updating a safety case only when required.
- Cover all variants that are relevant in production and systematically show safety for them.

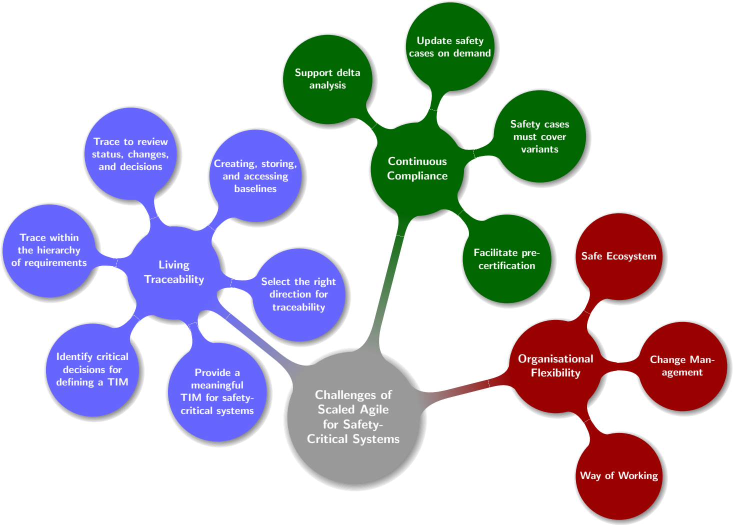
# Organisational Flexibility – Change Management

*Challenge*: react to changes quickly and adapt what is being built within a short period of time

- Individual teams should be able to make design decisions and update the safety case locally.
- Provide automated decision support for escalating changes to a higher level if safety case is affected.

# Outline

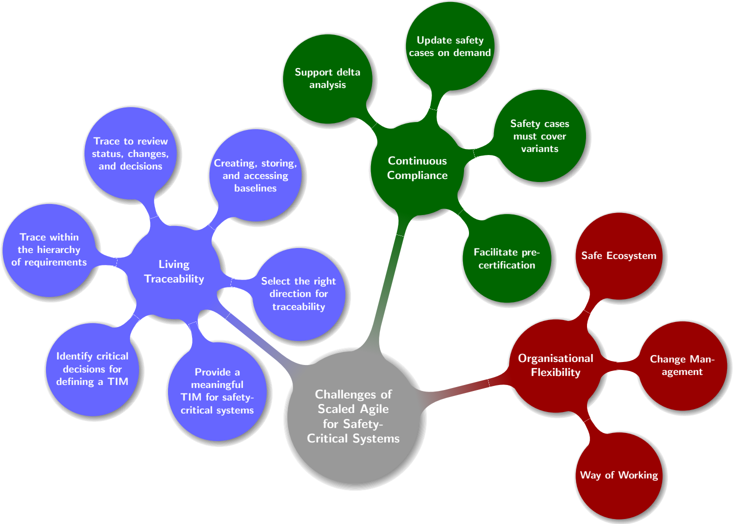
# Outlook




## Next Steps

- Constructive method to define specific WoW for SCS per project
- Develop (best/suitable) practices, e.g., in relation to SAFe
- TIM for SCS, connecting requirements, safety cases, tests and guiding their evolution
- Knowledge management and safety-related boundary objects
- Best practices to define SOPs to harmonize with SAFe / SafeScrum / R-Scrum

# Get in touch!




## Contact Information

Jan-Philipp Steghöfer
jan-philipp.steghofer@gu.se
jpsteghofer.net
072 974 6321

# References I


Brian Fitzgerald, Klaas-Jan Stol, Ryan O'Sullivan, and Donal O'Brien.
Scaling agile methods to regulated environments: An industry case study.
In *Int. Conf. on Software Engineering*, ICSE '13, pages 863–872, Piscataway, NJ, USA, 2013. IEEE Press.

Geir Kjetil Hanssen, Tor Stålhane, and Thor Myklebust.
*SafeScrum®-Agile Development of Safety-Critical Software*.
Springer, 2018.

Salome Maro, Jan-Philipp Steghöfer, Eric Knauss, Jennifer Horkoff, Rashidah Kasauli, Rebekka Wohlrab, Jesper Lysemose Korsgaard, Florian Wartenberg, Niels Jørgen Strøm, and Ruben Alexandersson.
Managing traceability information models: Not such a simple task after all?
*IEEE Software*, 28(5), 2021.

Jan-Philipp Steghöfer, Björn Koopmann, Jan Steffen Becker, Mikaela Törnlund, Yulla Ibrahim, and Mazen Mohamad.
Design decisions in the construction of traceability information models for safe automotive systems.
In *Proceedings of the 30th IEEE International Conference on Requirements Engineering (RE'21)*. IEEE, 2021.