Agile development of safety-critical products

Guest lecture at FAU Erlangen-Nürnberg

08.07.2020

Engineering und Software-Entwicklung

Agenda

Engineering und Software-Entwicklung

- About ESE & me
- Safety, Security & Agility – What‘s the problem?
- Agile development in compliance with Standards & Laws
  - A look into the agile manifesto
  - Can we apply Scrum?
  - The opinion of the AAMI
  - Opinions from the ESE
- Testing safety relevant products in agile processes
  - What can we learn from ISTQB?
  - Static code analysis
  - HIL Tests
- Time to talk

About ESE & me

Engineering und Software-Entwicklung

About ESE & Me

- My name is Ralf Spengler
- Studied at FAU
  - in the beginning Molecular Science for 4 semesters
  - switched to computer science in 2010 (one of the best decisions of my life)
  - Finished my master thesis in 10/2015
- First worked at HEITEC AG while studying and afterwards
- Since 2018: ESE – Engineering und Software-Entwicklung GmbH in Erlangen
- I‘m
  - Software Engineer with high experience in agile development of safety critical products
  - Software Tester (ISTQB Certified Tester Advanced Level – Test Manager)

Why am I here?

- I hate reading bad news (some examples from heise.de):
  - 03/2019: **Boeing 737 Max**:
    - After two plane crashes with 346 deaths flying ban on the whole world.
    - The software problems that caused it were already known in 2017.
    - The problems are not fully solved yet.
  - 12/2019: **Hospital of Fürth infected with malware**
    - Emergency care had to be stopped.
    - Not the first time: The hospital has already been infected in 2016.
- I‘ve learned myself much about security on university, but not about safety, although it‘s important and everywhere in our daily live (starting with the electric kettle when brewing tea early in the morning).

Data and Facts

Engineering und Software-Entwicklung

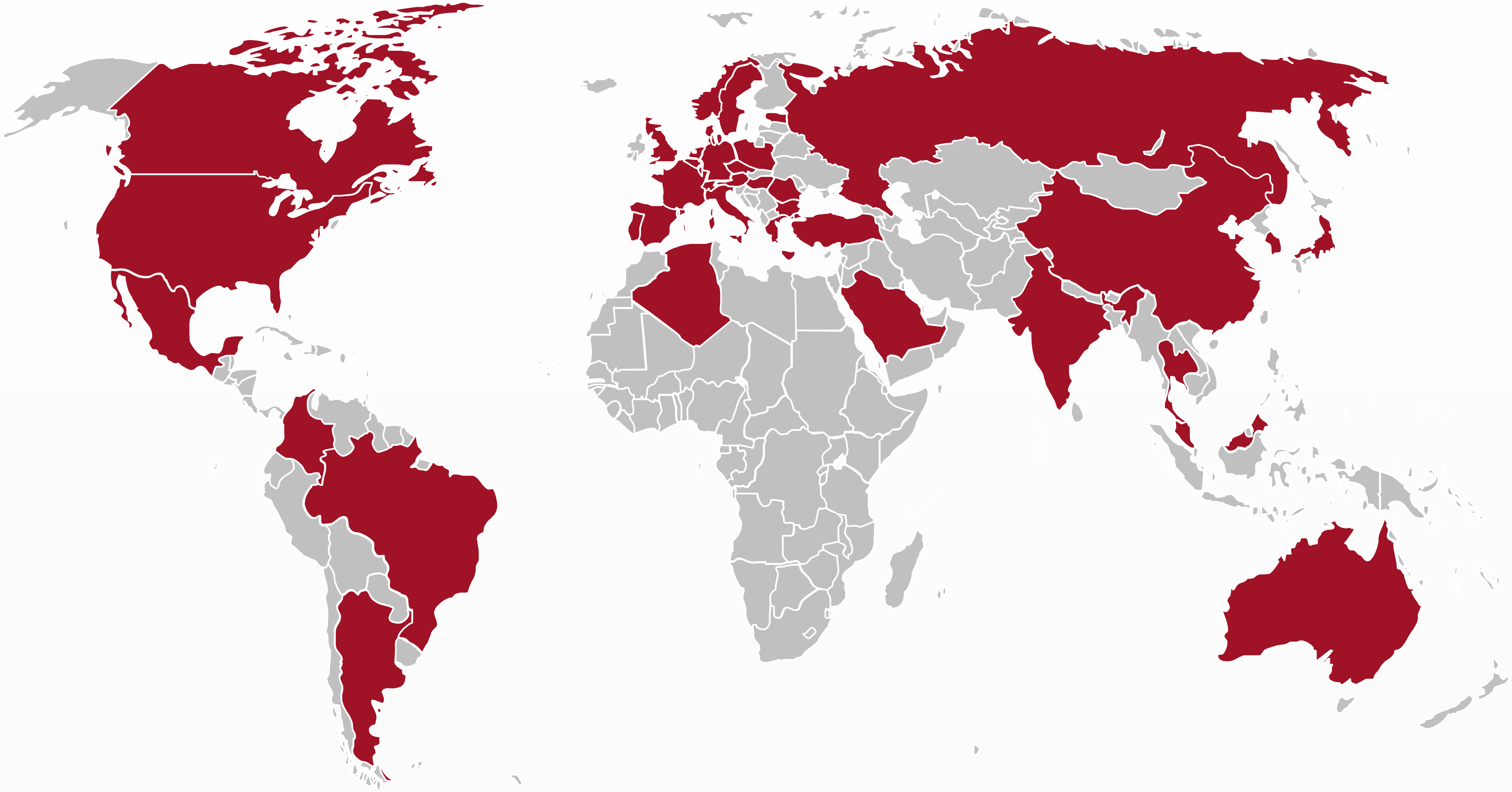
founded	Headquarter	Locations	Employees	Revenue	Certificates
1997	Braunschweig	Berlin / Hennigsdorf Braunschweig Erlangen Frankfurt am Main Hannover Hildesheim München Wolfsburg	2019: 320	2019: 29,5 Mio. Euro 2018: 27 Mio. Euro 2017: 24 Mio. Euro	CPPM according to iSQI®/ PMI® DIN EN ISO/IEC 17020 FRA-Consultants ISO 9001:2015 IRIS (ISO TS 22163:2017) iSTQB® -Tester TISAX Certificate

Our Competences

Services	Sectors	Projects
Software-Engineering	Rail	Project realization
Testing & Verification	Automotive	Consulting
Assessment Services	IT /Industry	Project support

ESE worldwide

Our projects around the globe



- ADIT GmbH
- Audi AG
- Bosch SoftTec
- BMW Group AG
- Denso
- Robert Bosch GmbH
- Siemens AG
- Valeo Siemens eAutomotive GmbH
- Volkswagen AG
- WABCO Development
- ...

Customers

Cleantech Solutions.
Anywhere.

Evac

ACTIA®

BMW

BOSCH

ALSTOM

Bosch SoftTec
Advanced Driver
Information
Technology

BVG

SIEMENS
Ingenuity for life

Valeo
SMART TECHNOLOGY
FOR SMARTER CARS

SAMSUNG

BOMBARDIER

Audi

WABCO

Safety, Security & Agility

What‘s the problem?

Engineering und Software-Entwicklung

Definition of terms

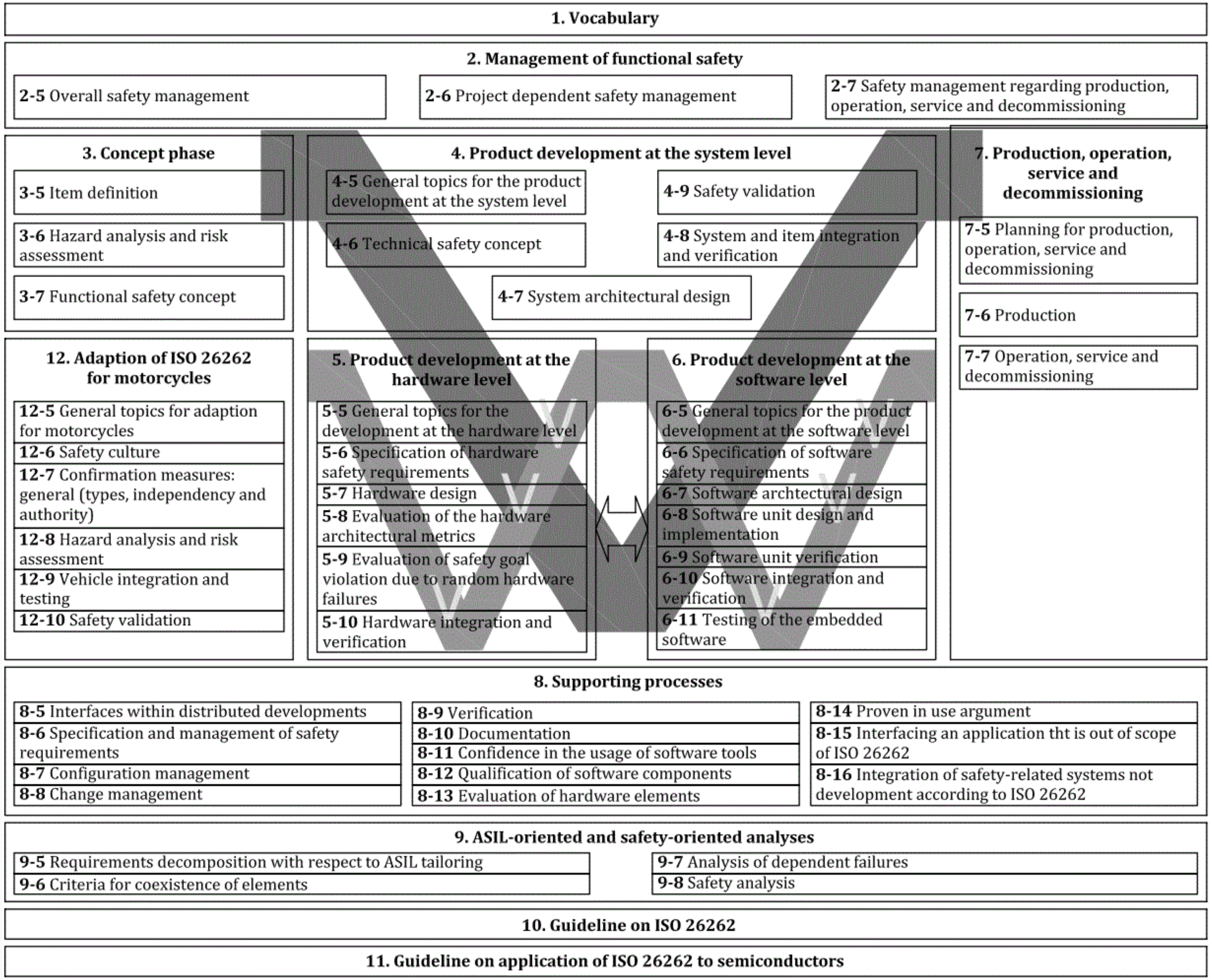
Based on the ISTQB glossary

- **Safety Critical System:**
  A system whose failure or malfunction may result in death or serious injury to people, or loss or severe damage to equipment, or environmental harm.

- **(Functional) Safety:**
  The absence of unreasonable risk (e.g. due to hazards caused by malfunctioning behavior of Electric/Electronic(E/E) – Systems)

- **Security**
  The degree to which a component or system protects information and data so that persons or other components or systems have the degree of access appropriate to their types and levels of authorization.
  (CIA goals: confidentiality, integrity, availability)

An example from ISO 26262 „Road vehicles – Functional safety“



1. Vocabulary

2. Management of functional safety

2-5 Overall safety management

2-6 Project dependent safety management

2-7 Safety management regarding production, operation, service and decommissioning

3. Concept phase

3-5 Item definition

3-6 Hazard analysis and risk assessment

3-7 Functional safety concept

4. Product development at the system level

4-5 General topics for the product development at the system level

4-6 Technical safety concept

4-7 System architectural design

4-8 System and item integration and verification

4-9 Safety validation

7. Production, operation, service and decommissioning

7-5 Planning for production, operation, service and decommissioning

7-6 Production

7-7 Operation, service and decommissioning

12. Adaption of ISO 26262 for motorcycles

12-5 General topics for adaption for motorcycles

12-6 Safety culture

12-7 Confirmation measures: general (types, independency and authority)

12-8 Hazard analysis and risk assessment

12-9 Vehicle integration and testing

12-10 Safety validation

5. Product development at the hardware level

5-5 General topics for the development at the hardware level

5-6 Specification of hardware safety requirements

5-7 Hardware design

5-8 Evaluation of the hardware architectural metrics

5-9 Evaluation of safety goal violation due to random hardware failures

5-10 Hardware integration and verification

6. Product development at the software level

6-5 General topics for the product development at the software level

6-6 Specification of software safety requirements

6-7 Software archtectural design

6-8 Software unit design and implementation

6-9 Software unit verification

6-10 Software integration and verification

6-11 Testing of the embedded software

8. Supporting processes

8-5 Interfaces within distributed developments

8-6 Specification and management of safety requirements

8-7 Configuration management

8-8 Change management

8-9 Verification

8-10 Documentation

8-11 Confidence in the usage of software tools

8-12 Qualification of software components

8-13 Evaluation of hardware elements

8-14 Proven in use argument

8-15 Interfacing an application tht is out of scope of ISO 26262

8-16 Integration of safety-related systems not development according to ISO 26262

9. ASIL-oriented and safety-oriented analyses

9-5 Requirements decomposition with respect to ASIL tailoring

9-6 Criteria for coexistence of elements

9-7 Analysis of dependent failures

9-8 Safety analysis

10. Guideline on ISO 26262

11. Guideline on application of ISO 26262 to semiconductors

- ISO 26262 is „just“ an extension to the **quality management process** for safety critical electrical systems within motor vehicles
- Heavy **V-model** approach with nested V-models for hardware and software.
- Nearly all safety relevant standards (and even some laws) regulate such **processes**.

Important during development

- **Risk analysis**: The overall process of
  - **Risk identification**: The process of finding, recognizing and describing risks.
  - **Risk assessment**: The process to examine identified risks and determine the risk level.

- **Risk mitigation**: The process through which decisions are reached and protective measures are implemented for reducing or maintaining risks to specified levels. Examples:
  - Architectural risk mitigation
  - Documentation (e.g. warning in user manual)
  - Testing

Important during development

- **Verification**: Confirmation by examination and through provision of objective evidence that specified requirements have been fulfilled. (Are we doing things right?)
- **Validation**: Confirmation by examination and through provision of objective evidence that the requirements for a specific intended use or application have been fulfilled. (Are we doing the right things?)
- Special workflow for **defects**
- And much more …

Traceability

- **Traceability**: The degree to which a relationship can be established between two or more work products.

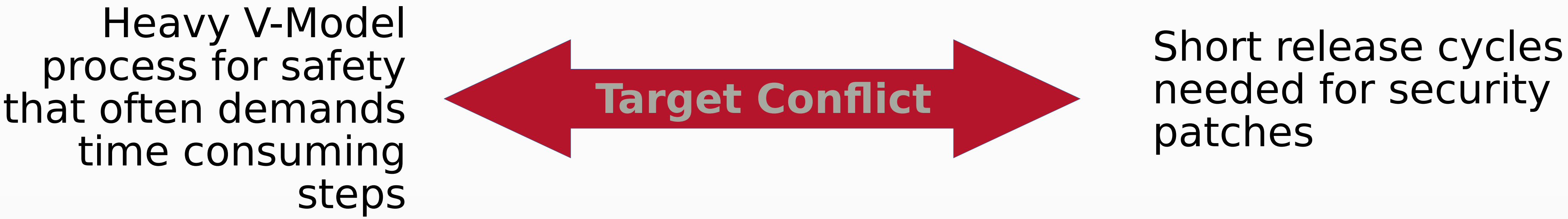
- Making all important artifacts of development cycle (specifications, risks, software, test cases, test results, defects, …) traceable through the whole development and life cycle of a product is important for
  - Auditors to prove that you did everything in compliance with standards and laws
  - Yourself so you can remember what you did and why
  - You and your company if you get sued after a failure of your system which caused harm to someone

Some standards and laws

Area	Standard & Laws
Rail	DIN EN 50126, 50128 & 50129 ESiV (Eisenbahn-Sicherheitsverordnung) RL (EU) 2018/762 (common safety methods on safety management system requirements)
Automotive	ISO 26262 StVG (Straßenverkehrsgesetz)
Aviation	DO-178B VO (EU) 2018/1139 (includes aircraft certification by EASA inside european union)
Industry	IEC 61508 Directive (EU) 2001/95/EC (general product safety)
Health care	EN 62304 FDA-535

No safety without security today

- Many devices are getting connected with the upcoming IoT, e.g.
  - Medical devices to servers storing patient data
  - Car intercommunication for autonomous driving
  - Industrial machines with business software to support batch-1
  - …
- Security
  - Must be implemented by design right from the start
  - Always needs the possibility of fast security patches



Heavy V-Model process for safety that often demands time consuming steps

Target Conflict

Short release cycles needed for security patches

Agile Development

in compliance with Standards & Laws

Engineering und Software-Entwicklung

A look into the agile manifesto

On the first glance it looks not so good

But we need processes following standards and laws.

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan.

Safety critical products are often complex and expensive.
We need solid contracts.

The process includes planning!

Documentation is important to prove our product is safe.

A look into the agile manifesto

But taking a look closer there is no conflict

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.

A look into the agile manifesto

But taking a look closer there is no conflict

The values on the left side are important, ...

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan.

… but it is not said what‘s on the right side is unimportant!

A look into the agile manifesto

But taking a look closer there is no conflict

A process can describe who will work together in which way.

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan.

Building complex systems need collaboration with the customer (e.g. clarifying requirements). Agile pricing models can be part of contracts.

Change is a natural thing. A good process contains rules how to handle change requests.

It‘s not written „documentation is unimportant“. It‘s more „documentation has no value if your product does not work“.

Can we apply Scrum?

- From „The Scrum Guide™“:
  - “Scrum is a framework for developing, delivering, and sustaining complex products.“
  - “Scrum is a process framework …“
- But we need a process!
- Does this mean Scrum cannot be used?
  - NO!
  - You can make Scrum or ScrumBut part of your process.
  - You should consider the SafeScrum® methodology:
    - Two backlogs: Safety product backlog & functional product backlog
    - Focus on traceability
    - Functional and RAMS (Reliability, Availability, Maintainability, Safety) validation

The opinion of the AAMI

TIR45

- AAMI: „Association for the Advancement of Medical Instrumentation“

- TIR45 (Technical Information Report 45 from 2012):
  Guidance on the use of agile practices in the development of medical device software
  - **States that agile software development compliant with FDA requirements and IEC 62304 is possible.**
  - Demands: quality management system, development life cycle, change management process, documentation according to standards and laws, very sharp and strong definition of done
  - Suggests: 4 different nested iteration layers (from project down to a single story)

    => Allows parallel development of requirements, architecture and code

Assessment Service Center

The independent expert organization of ESE GmbH

Engineering und Software-Entwicklung

The ASC‘s opinion

- **Biggest challenges for combining agility & safety**:
  - Agile development is strongly shaped by software development but safety critical products often contain much more.
  - Agile frameworks (e.g. Scrum) often have one development team “doing everything“ but some standards require independent testing (and if they would not: it‘s best practice)
- **Most common mistake when combining agility & safety:**
  - Believing Scrum is a process
- **Advice:**
  - Use a hybrid approach combining agile development for non safety relevant functionality and a plan based development for safety relevant functionality.
  - Incremental V-Model or SafeScrum® can be an approach

Mandy‘s opinion

Project Leader at ESE GmbH

Engineering und Software-Entwicklung

Mandy‘s opinion

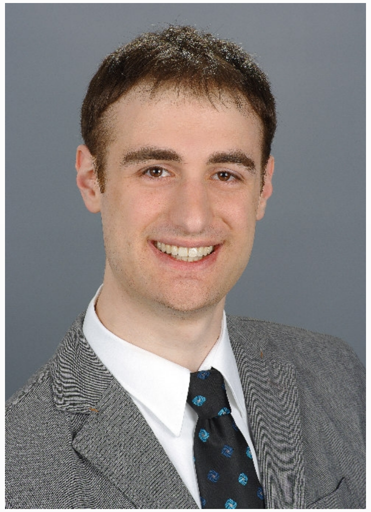
- **Biggest challenges for combining agility & safety**:
  - Changing mindset of people used to work in waterfall or V-model
  - Establishing hybrid way of working
- **Most common mistake when combining agility & safety:**
  - Forcing employees to work agile
  - No common understanding of agile development
  - No retrospectives or actions resulting from them
- **Advice:**
  - Don‘t be agile to be agile
  - Select methods & tools with respect to the project & team.
  - Evaluate new things by experimentation.
  - Do agile transformation step by step.

My personal opinion

Engineering und Software-Entwicklung

My personal opinion

- **Biggest challenges for combining agility & safety**:
  - Building a cross-functional dev-team that covers all knowledge needed
  - Matrix-organization: Many high level experts at one discipline, few allrounders
- **Most common mistake when combining agility & safety:**
  - No common understanding of agile
  - Not valuing the independence of testing
  - Doing no retrospectives
- **Advice:**
  - Don‘t be agile to be agile.
  - Don‘t forget negative tests (as developer I also do that sometimes).
  - Distinguish between software only and projects involving hardware.
  - **There‘s more than Scrum!**



Testing safety relevant products

in agile processes

Engineering und Software-Entwicklung

What can we learn from ISTQB?

- **Risk-based testing**: Testing in which the management, selection, prioritization, and use of testing activities and resources are based on corresponding risk types and risk levels.
  - Risks often influence a systems architecture, including error handling
  - Software architecture becomes code
  - Tests must cover **functionality** and **error handling** (negative tests!)
  - **Provoking errors** (especially automatically) can be an art
- **Not everything can be tested within an iteration, sprint, ...:**
  - e.g. hardware-software-integration tests often need finished increments of multiple teams
  - If something does not fit easily into an iteration, do not force it in. Find another way (e.g. separate test-iterations with a different duration).

Static code analysis

- **Use static code analysis to**
  - **Teach and assess yourself**
  - **Let human reviewers focus on functionality instead of semantics**
  - Find potential security vulnerabilities
  - Find potential bugs

- Do not only run linters on the command line or in an IDE but **make the results visible** (e.g. with SonarQube™)
  - Show management that quality is good and even is improving
  - Supports pareto analysis: Most of your problems arise from few causes (e.g. 80% of the errors typically can be lead back to 20% of the code). To detect these 20% often a visual tool is enough.

HIL Tests

- HIL: Hardware in the loop
- HIL test: Dynamic testing using real hardware with integrated software in a simulated environment.
- Very powerful to detect errors that else often would be found at system integration level for the first time:
  - You can pull the emergency break a thousand times without the need of a single train.
  - Reduces the gap between hardware- and software-developers
    → Increased system knowledge can help to detect uncovered errors.
  - Start testing software before the system’s hardware does physically exist.

Summary

Summary

- Agile development of safety critical products is not impossible.
- Agile transformation that create processes that combine agility while fulfilling standards and laws is a challenge.
- Hybrid approaches are a good way to close the gap between pure agile development and standards and laws.
- Never forget the importance and benefits of testing.

Get in contact with us!

Recruiting
https://www.ese.de
Phone: +49 531 23880-30
jobs@ese.de

Dr. rer. nat. Ralf Goldstein
Business Manager
Phone: +49 9131 6102-984
Mobile: +49 178 148 66 78
Ralf.Goldstein@ese.de

Ralf Spengler
Software Engineer
Phone: +49 9131 6102-984
Ralf.Spengler@ese.de

Engineering und Software-Entwicklung

Thank you for listening!

Time to talk

Engineering und Software-Entwicklung

References

- ISO 26262 - „Road vehicles – Functional safety“
- Spillner/Roßner/Winter/Linz: „Praxiswissen Softwaretest – Testmanagement“ (ISBN: 978-3-86490-052-5)
- ISTQB® Glossary: https://glossary.istqb.org
- DIRECTIVE 2001/95/EC: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32001L0095&from=DE
- StVG: https://www.gesetze-im-internet.de/stvg/
- EASA: https://www.easa.europa.eu/
- Agile Manifesto: https://agilemanifesto.org/iso/en/manifesto.html
- The Scrum Guide™: https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf
- SafeScrum® – Agile Development of Safety-Critical Software (ISBN: 978-3-319-99333-1)
- AAMI: https://www.aami.org/

References

- AAMI TIR45:
  - https://my.aami.org/aamiresources/previewfiles/TIR45_1208_PREVIEW.PDF
  - https://www.johner-institut.de/blog/iec-62304-medizinische-software/agile-softwareentwicklung-fuer-medizinprodukte/
- SonarQube™:
  - https://www.sonarqube.org/
  - https://commons.wikimedia.org/wiki/File:Sonarqube-48x200.png
- Cucumber: https://cucumber.io/docs/guides/overview/
- Gherkin: https://cucumber.io/docs/gherkin/reference/

References

- heise.de:
  - Malware at hospital Fürth: https://www.heise.de/newsticker/meldung/Computervirus-Klinikum-Fuerth-offline-und-mit-eingeschraenktem-Betrieb-4615427.html
  - Boeing 737 Max 8:
    - https://www.heise.de/newsticker/meldung/Absturz-von-Ethiopian-Airlines-Flug-Erste-Startverbote-fuer-Boeing-737-Max-8-4330748.html
    - https://www.heise.de/newsticker/meldung/Fast-ein-Jahr-Flugverbot-fuer-737-Max-Wie-geht-es-weiter-mit-Boeing-4676570.html
    - https://www.heise.de/newsticker/meldung/Boeing-wusste-seit-2017-von-Problem-mit-Ungluecksflieger-737-Max-4413595.html