Systems Engineering and the Sins of Complex Software

OSADL
Nicholas Mc Guire
<*safety@osadl.org*>

April 13, 2018

Why do we need regulation ?

- To establish a common problem space definition
- To gain a common understanding of context and scope
- To ensure peer-review of solution space (industry, academia, public)

Why do we need regulation ?

- To establish a common problem space definition
- To gain a common understanding of context and scope
- To ensure peer-review of solution space (industry, academia, public)

Goal:

- Wide acceptance of technology in the technical community and society
- Defined limits and accepted responsibilities

Acceptance

Jumping orders of magnitude in acceptance ?

- 1.2M people killed in car accidents per year worldwide
- 0-1k people killed in civil aircraft per year

What gets more attention in the media ?

Acceptance

Jumping orders of magnitude in acceptance ?

- 1.2M people killed in car accidents per year worldwide
- 0-1k people killed in civil aircraft per year

What gets more attention in the media ?

Controllability is a subjective issue - never the less it determines perceived responsibility and risk acceptance.

What‘s in a standard ?

- Agreed upon **consolidated state-of-the-art**: encoded as processes, measures and techniques fit to the problem space

What‘s in a standard ?

- Agreed upon **consolidated state-of-the-art**: encoded as processes, measures and techniques fit to the problem space

What‘s the state-of-the-art for Autonomous Vehicles ?

What‘s in a standard ?

- Agreed upon **consolidated state-of-the-art**: encoded as processes, measures and techniques fit to the problem space

What‘s the state-of-the-art for Autonomous Vehicles ?

What‘s the state-of-the-art for verifying AI/ML ?

What‘s wrong with ISO 26262

- Table driven safety
- Only covers low complexity systems
- Assumes software correctness implies behavioral correctness
- Software is considered to be deterministic
- Assumes a driver is in control
- ISO 26262 was classical micro-controllers and automotive applications consolidated into the then state-of-the-art for low complexity systems.

What‘s wrong with ISO 26262

- Table driven safety
- Only covers low complexity systems
- Assumes software correctness implies behavioral correctness
- Software is considered to be deterministic
- Assumes a driver is in control
- ISO 26262 was classical micro-controllers and automotive applications consolidated into the then state-of-the-art for low complexity systems.

No standard — no (verifiable) safety properties

What‘s wrong with ISO 26262

- Table driven safety
- Only covers low complexity systems
- Assumes software correctness implies behavioral correctness
- Software is considered to be deterministic
- Assumes a driver is in control
- ISO 26262 was classical micro-controllers and automotive applications consolidated into the then state-of-the-art for low complexity systems.

No standard — no (verifiable) safety properties

Wrong standard — no safety either

What is the state-of-the-art ?

- Thats really an open issue for autonomous vehicles - nobody knows !
- Fundamental issues are unresolved (nondeterministic algorithms, FP-usage, reproducibility,...)
- Applicable domain standards do not exist yet
- Accepted procedures for establishing tolerable safety not agreed (if they exist at all)
- Expected behavior of systems - not agreed - not even understood
- V&V of AI/ML ? indicators and methods - not known

Classes of Safe Systems

- Type A System (low complexity)
  1. The failure modes are well-defined; and
  2. The behavior under fault conditions can be completely defined
- Type B Systems (complex)
  1. The failure modes are not well-defined; or
  2. The behavior under fault conditions cannot be completely defined

Classes of Safe Systems

- Type A System (low complexity)
  1. The failure modes are well-defined; and
  2. The behavior under fault conditions can be completely defined
- Type B Systems (complex)
  1. The failure modes are not well-defined; or
  2. The behavior under fault conditions cannot be completely defined
- Type C Systems ? (high complexity ?)
  1. What constitutes a failure is not well-understood; or
  2. The behavior under absence of (SW/HW) faults cannot be completely defined

In autonomous systems the correlation between software correctness and system behavior is essentially lost.

Common fallacies

- Using wrong standards and Checklist safety
- Functional focus - safety as post-processing event
- Hazard mitigation before hazard elimination
- Keep it simple at the code rather than the design level
- Correct code does not imply correctness of behavior in AI
- Focus on local mitigations rather than system scope
- Focus on confirmation of acceptability rather than risk assessment
- Building compliant rather than safe system
- Separation of safety competence and decision authority
- Lack of communication with respect to safety issues
- Lack of responsibility for and awareness of safety issues
- Time/market and management pressure: functional focus
- ...

Why this - What changed ?

- Totally different solution space - data & behavior -> evolution
- Fundamental change of software capabilities needed
- Change of legal environment
- Novel, untested, unexplored, not understood technologies
- It seems nobody bothered to build a solid foundation

And then there is this complexity problem - nobody has a clue how to manage this level of complexity for safety - absolutely nobody.

What are some mitigations ?

- (finally) Introduce system-safety engineering in automotive industry - there is no such things as SEooC for novel systems.
- Establish and **then** agree on the state-of-the-art
- Develop a set of suitable standards
- Jointly define a legal environment: Vienna Convention on Road Traffic++ ?, security ?, data ownership ?
- Build up a safety awareness/culture around autonomy related technologies in academia and industry
- Educate the public (including politicians) on benefits **and** safety risks
- ...

Conclusion

- Safe processes depends on an established state-of-the-art
- Coordination and standardization is mandatory for safety
- The existing measures and techniques/metrics will not do
- A set of domain standards for autonomous vehicles needed
- Establishing new methods is research and has no guranteed time-line - it‘s done when it‘s done - product time-lines are not relevant for safety

Safety in autonomous vehicles is not going to be achieved as an add-on to a conglomerate of unverifiable assist-systems.

Rant

- Rail industry worked on autonomy for 50 years and brought it into service in safe step-by-step mode - did you hear much about this in the news ?
- If business interests decide safety it will be a (continued) bloody mess - with nobody taking responsibility.
- Lets iterartively build a solid foundations first — **understood** Technology, **agreed** Standards and **sound** Regulations.

Rant

- Rail industry worked on autonomy for 50 years and brought it into service in safe step-by-step mode - did you hear much about this in the news ?
- If business interests decide safety it will be a (continued) bloody mess - with nobody taking responsibility.
- Lets iterartively build a solid foundations first — **understood** Technology, **agreed** Standards and **sound** Regulations.

Expect the first **safe autonomous vehicles** on the road in **2040+**

Thanks !