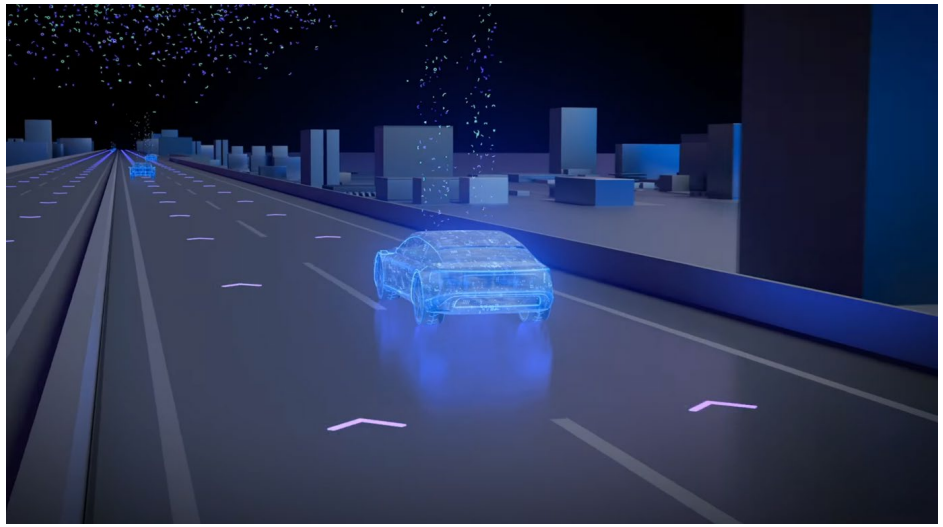
Safety Factories

- a Manifesto -

Carmen Carlan (TÜV SÜD)
Daniel Ratiu (CARIAD)
Michael Wagner (Edge Case)

Software-Defined-X

Continuous Software Engineering - Speed!



Software Factories tools and methods to build software, focus on:

avoiding manual repetitive tasks - pipelines

high degree of content formalization – X-as-Code

having no islands of content – mono-repositories

everything being saved exactly once - single source of truth

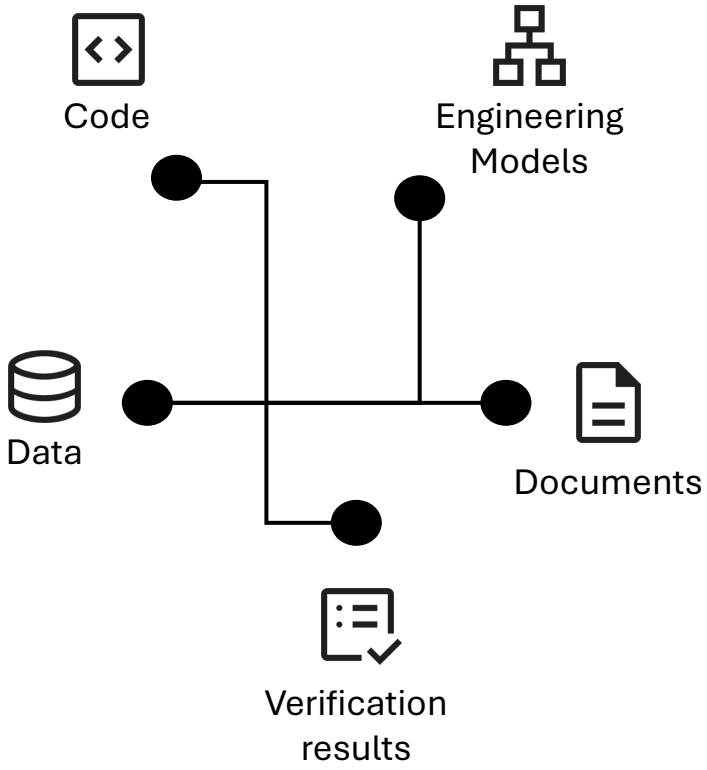
enforcing reviews and quality checks - discipline and rigor

traceability to tickets, tasks, history - ownership and accountability

Software Factories enable continuous software engineering – early MVPs, continuous delivery

Today‘s Safety Engineering Practice

Safety Tools Isolated from each other and from System Development Tools – Silos!



Weakly formalized content – spreadsheets, plain text, pictures

- difficult to automate

Different persistency mechanisms – e.g. file formats, databases, …

- islands of overlapping content

System / Software / Safety engineers are working in tooling silos

- lack of transparency between working teams

Assessors are given text documents instead of access to live data

- long cycles

Isolated tools produce isolated content leads to silos and friction.

Safety Factory Manifesto

Adopt Successfully Applied Software Engineering Receipes

- **Safety Artefacts as Code** over informal, unstructured information.
- **Single Source of Truth and Integration** over overlapping content islands.
- **Automated Impact Analysis** over manual analysis of implicit dependencies.
- **Continuous Safety Builds** over sparse safety validation milestones.
- **Documentation Close to Content** over separated pictures and plain text.
- **Company-wide Safety Mindset** over silos and compartimentalisation.
- **Accountability and Ownership** over hierarchies and approval committees.

Processes, methods and tools closer to the continuous delivery pipelines specific to software intensive systems