seerene
The Digital Boardroom

How the Automotive Industry can learn from digitally-born Software Companies

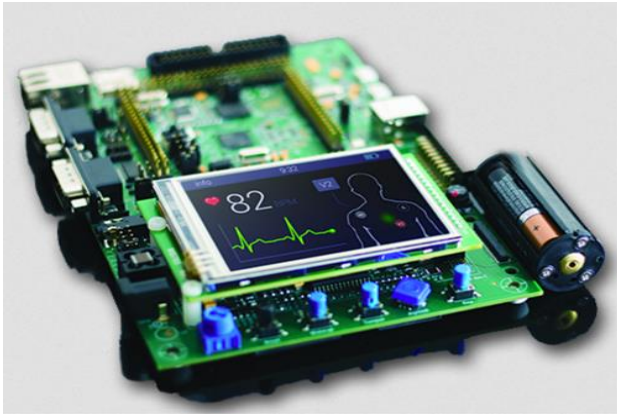
Conference: Automotive Software Factory
September 22nd, 2021
Hasso-Plattner-Institute, Potsdam

Dr. Johannes Bohnet
Founder & Co-CEO
Seerene GmbH
johannes.bohnet@seerene.com

Software is a Key Business-Driver for every Industry

Corporate IT of Banking, Insurance, Telco, Retail, ... →

← Manufacturing, Mechanical Engineering



Core IT applications enable efficient business operations.

Customer-facing software creates new business opportunities.

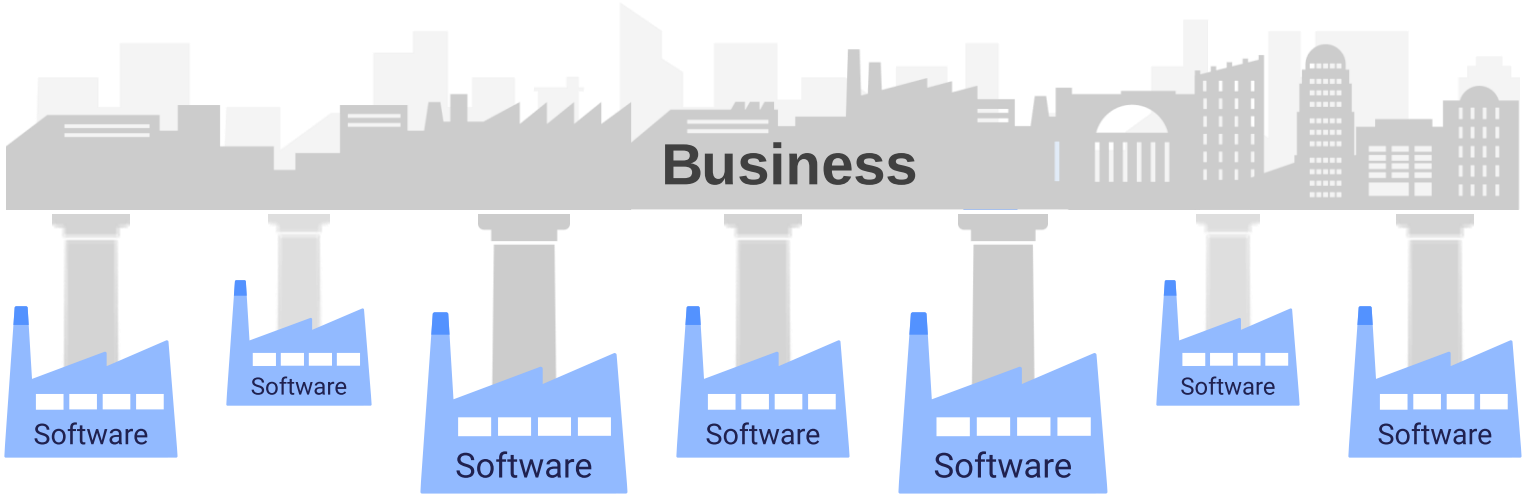
Hardware becomes smarter & builds IoT networks.

Tailor-made software is the basis for competitive advantages

Tailor-Made Software is built in “Software Factories”

Mastering the process of software development is key

The ability to run a **software factory on highest level of excellence** decides about **business success**.



Software Factories

Can Automotive learn from digitally-born Software Companies?

Isn't the "Netflix software" totally different to the hardware-constrained software in car?

Runtime Contexts of Software

Embedded Systems

- Hardware with limited computing-power
- Same software with variants for multiple hardware

Smart IoT Devices & Infotainment & …

- Central computers with significant computing power

→ Comparable to traditional desktop/mobile software

IoT Ecosystem & Services

→ Comparable to "Netflix software"

Two Key Ingredients for Excellent Software Production

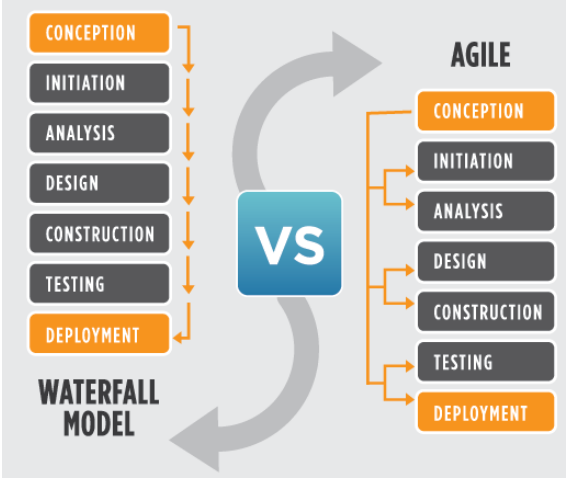
Learning from hardware factories:

- Full transparency about the production process.
- Continuously: Measure, analyze & optimize the process.

Remark: Also digitally-born software companies lack management transparency.

Mind shift from “projects” to “products”:

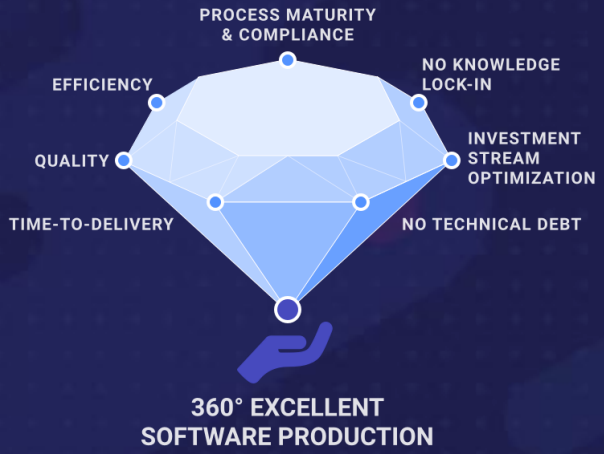
- You might deliver/sell via projects, but…
- Software systems are assets that are never finished.
- Creating business innovation implies a continuously evolving code base.

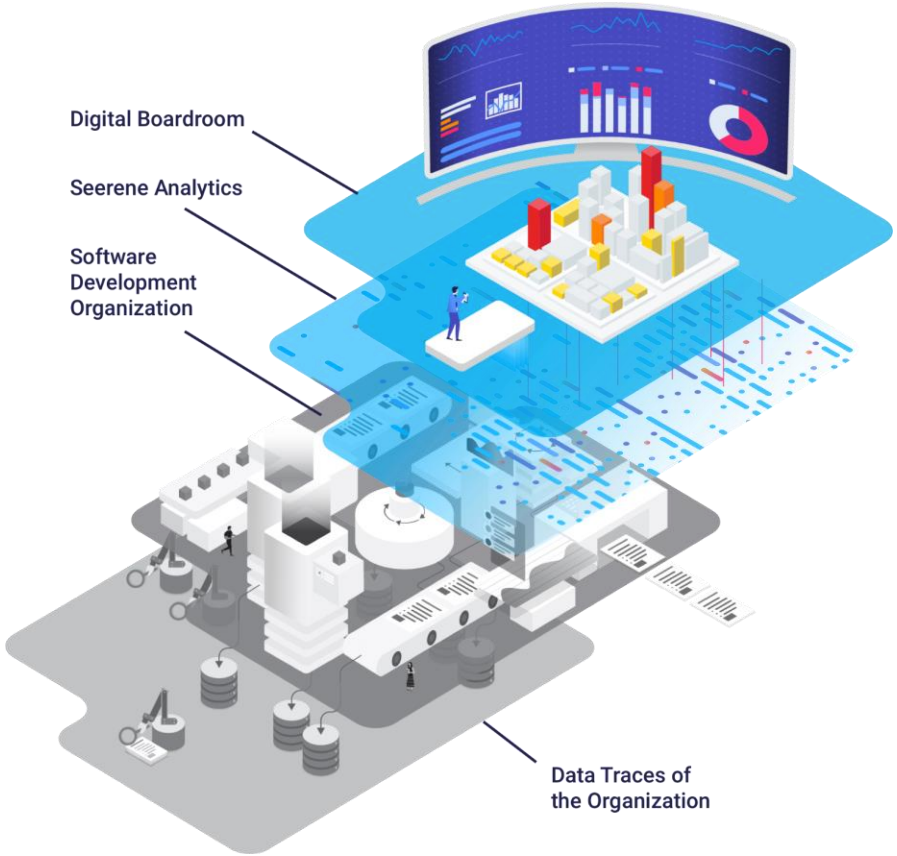



Full Transparency about the
Software Production Process

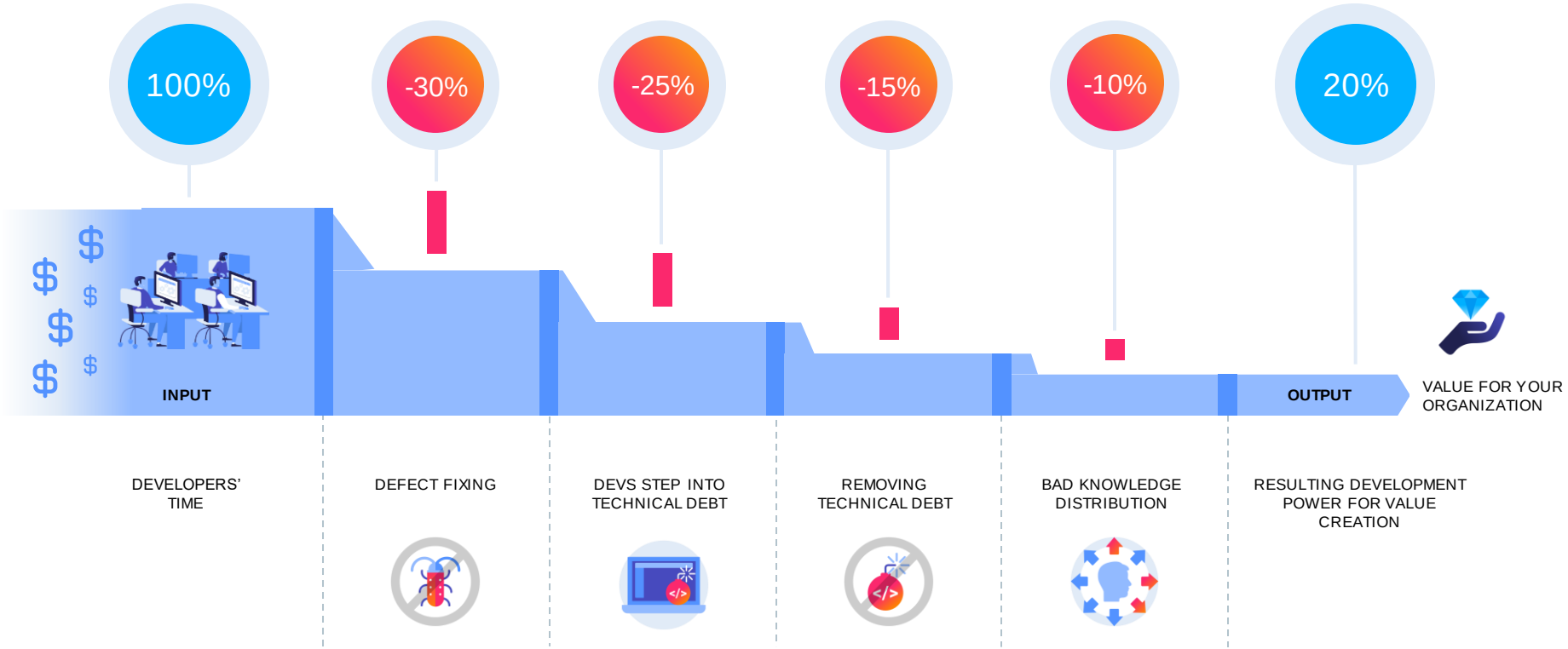
Establish a Digital Boardroom with Analytics

KPIs & drill-downs for
balancing and optimizing the challenges of
software development organizations



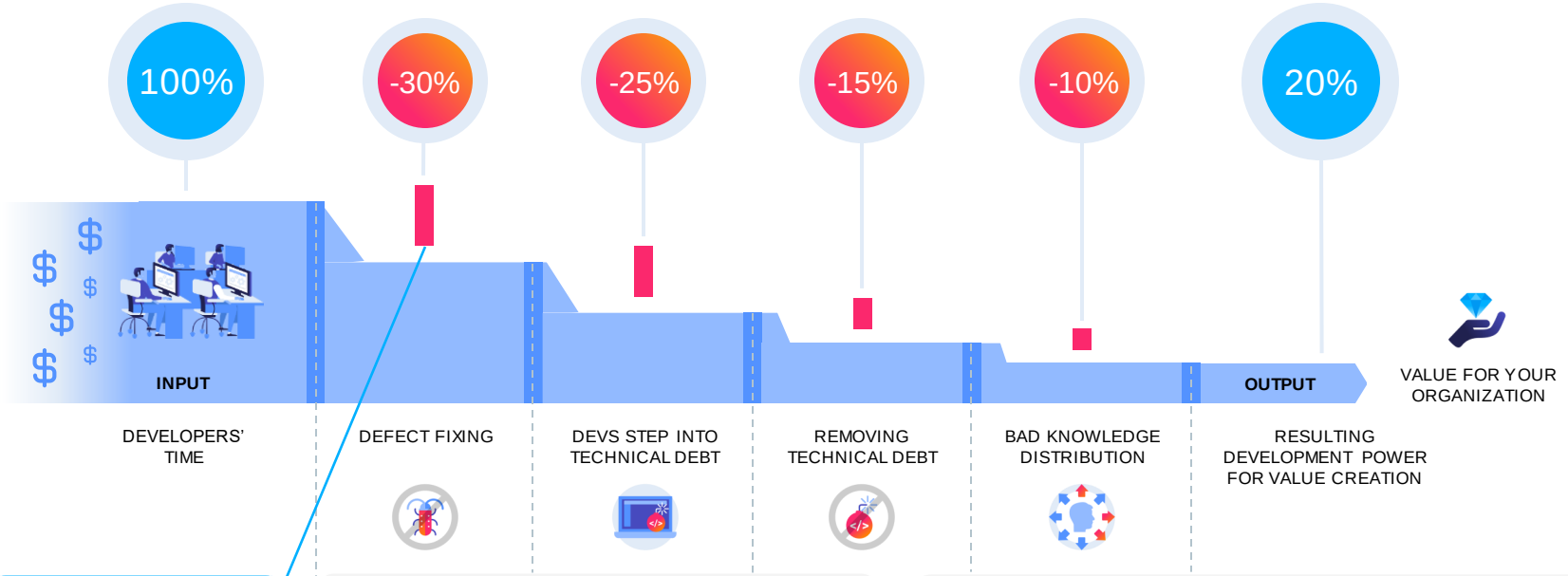


Quantifying Inefficiencies in the Software Production Process



Locating and Eliminating Root-Causes of the Inefficiencies

…and create extra power – literally out of nothing



Seerene KPIs

"Effort for Defect Fixing"

Management Level: **Monitoring & Steering**

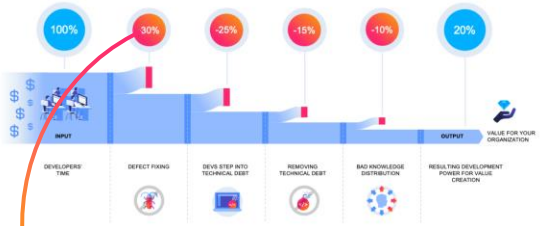
- **Seerene KPIs** enable you to quantify and manage the problem

Expert Level: **Actionable Insights**

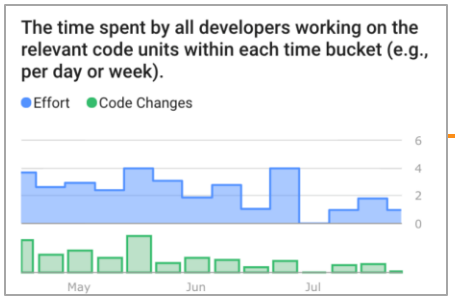
- **Seerene drill-downs (e.g., Code Map Visualization)** enable you to precisely locate the root-cause for the problem and fix it with minimal investment

Actionable Insights for Gaining Extra Power

Example: Precisely identifying and fixing the code that frequently contains defects



Coding effort spent for defect-fixing



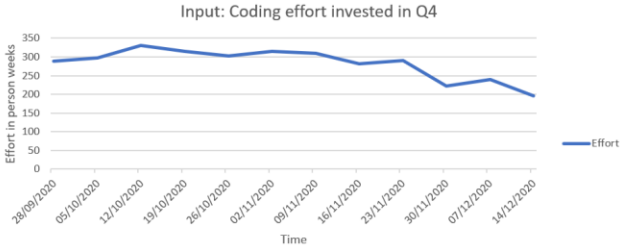
- Reveal how much effort is spent for defect-fixing
- Locate code units that constantly need to be fixed

→ Be able to effectively eliminate the problem

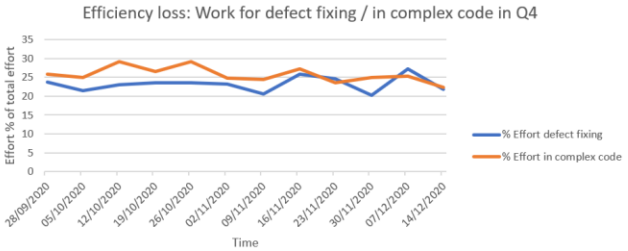
Transparency across the entire Company

Comparable KPIs that can be applied across heterogenous processes/tools/methods

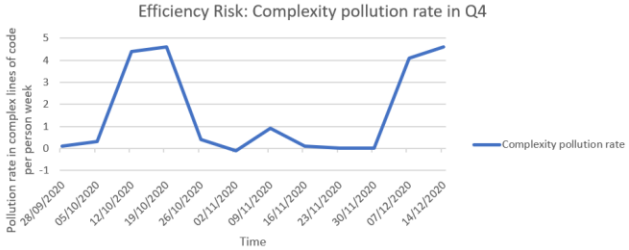
Investment

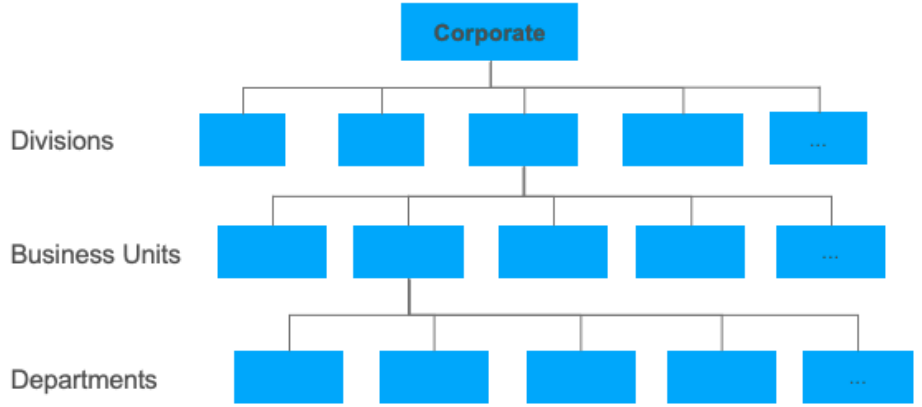


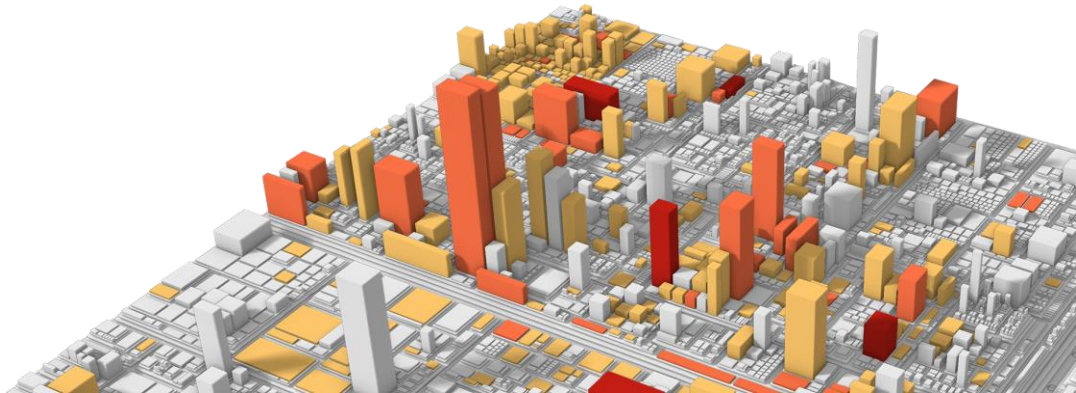
Efficiency



Technical Debt
(Future Efficiency Risk)



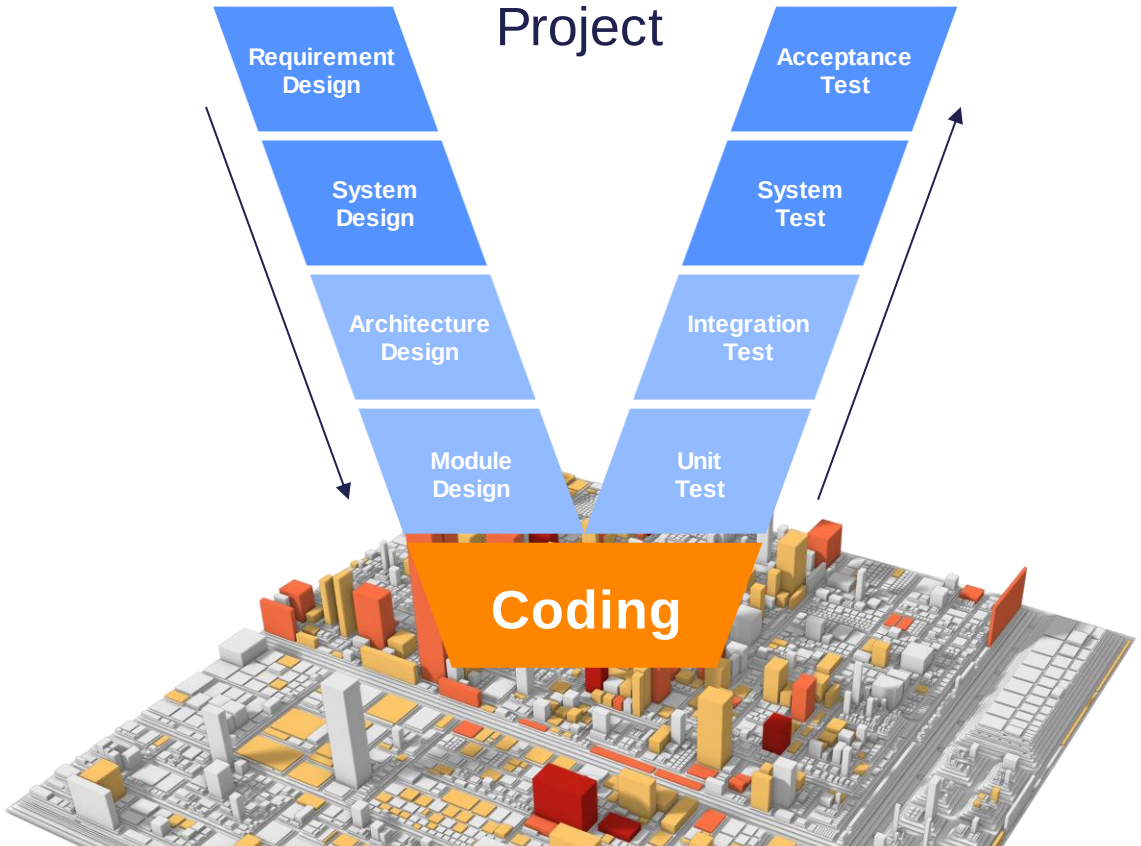





Mind Shift:
From „Projects“ to „Products“

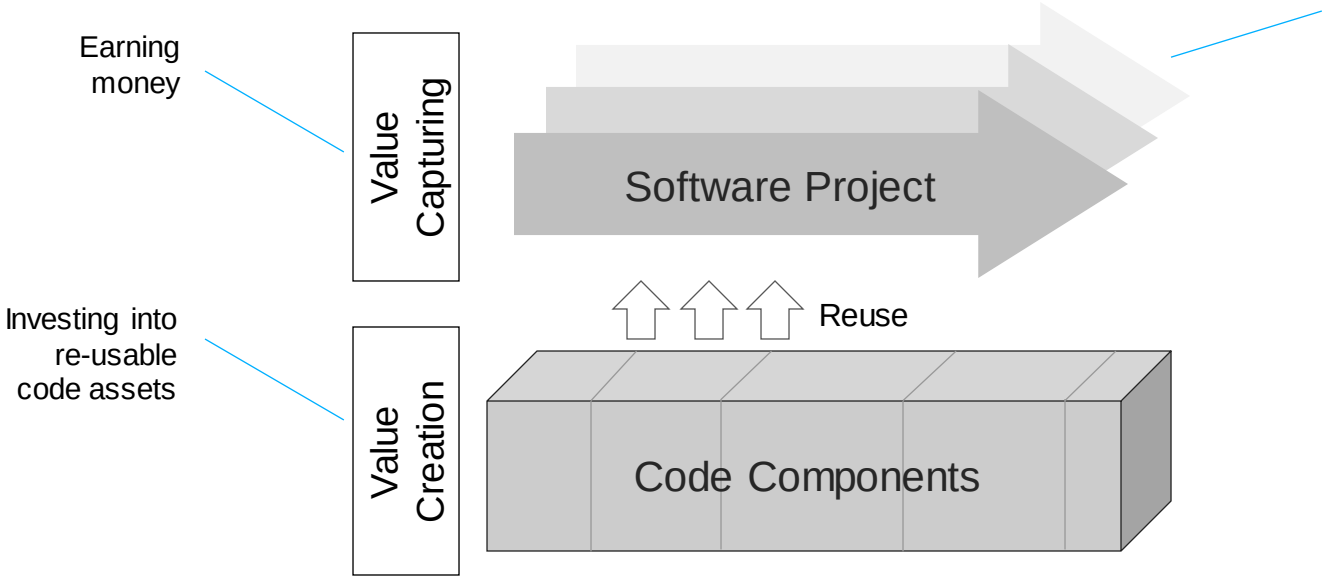
Mind Shift from “Projects” to “Products”

Embedded software is delivered via projects – but is built on top of an existing code base.



Reusable Code Components as key for Efficiency

A successful software factory distinguishes between value creation & value capturing



The Perfect Project

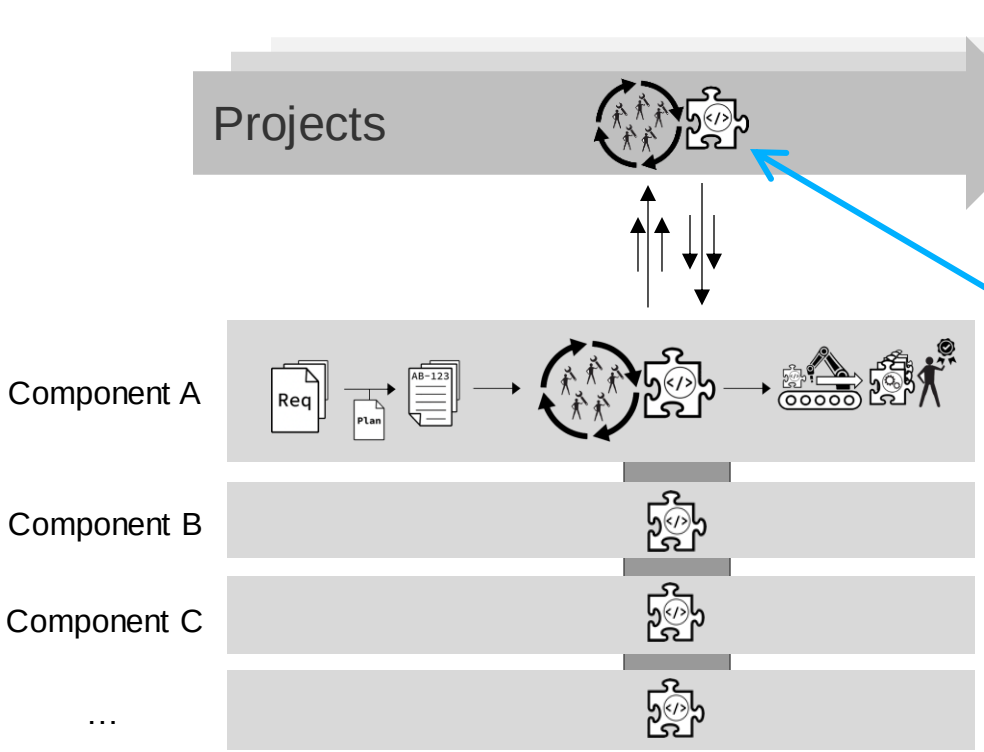
- would take pre-manufactured code components
- would not require any coding effort.

Project effort would shrink to activities for

- requirements management
- configuration/calibration
- testing

Measuring Code Reuse with Analytics

Systematically moving recurring project-specific coding effort towards reusable components



Projects (as of today)

1. Take “re-usable” code component as starting point.
2. Rework / extend / adapt the code to make it work for the project.

Measure
KPI Coding Effort in project context

→ locate code areas that are always modified in project context

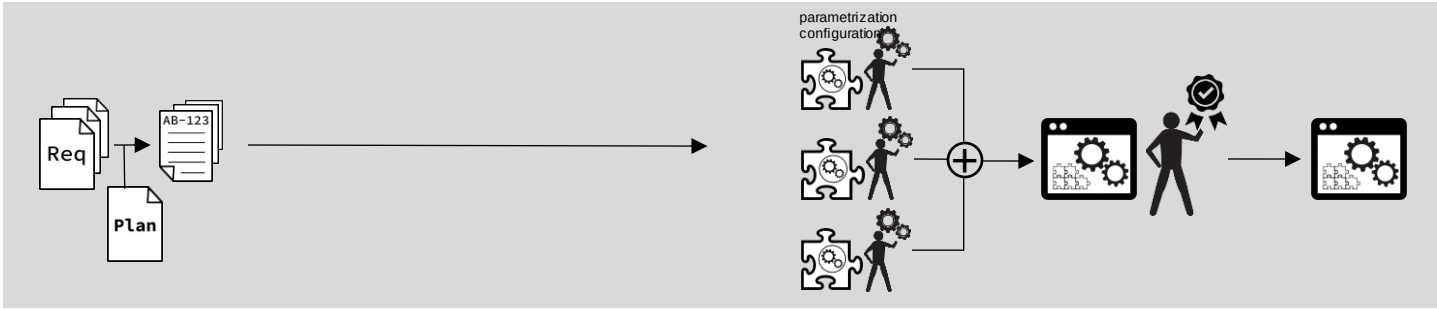
→ extend generic, re-usable code component accordingly

→ save a lot of time and money in all follow-up projects

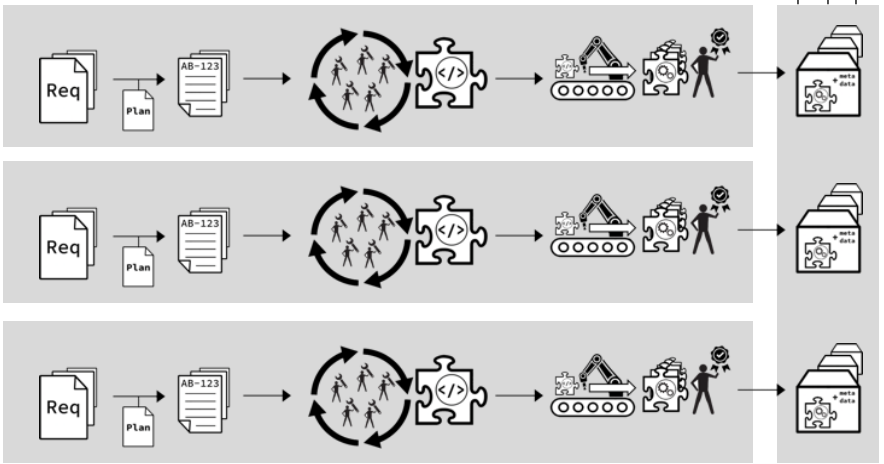
The Ideal Software Factory

"Project Mindset"

Harvesting Value in Projects



Building Reusable Code Components (Driving Innovation)

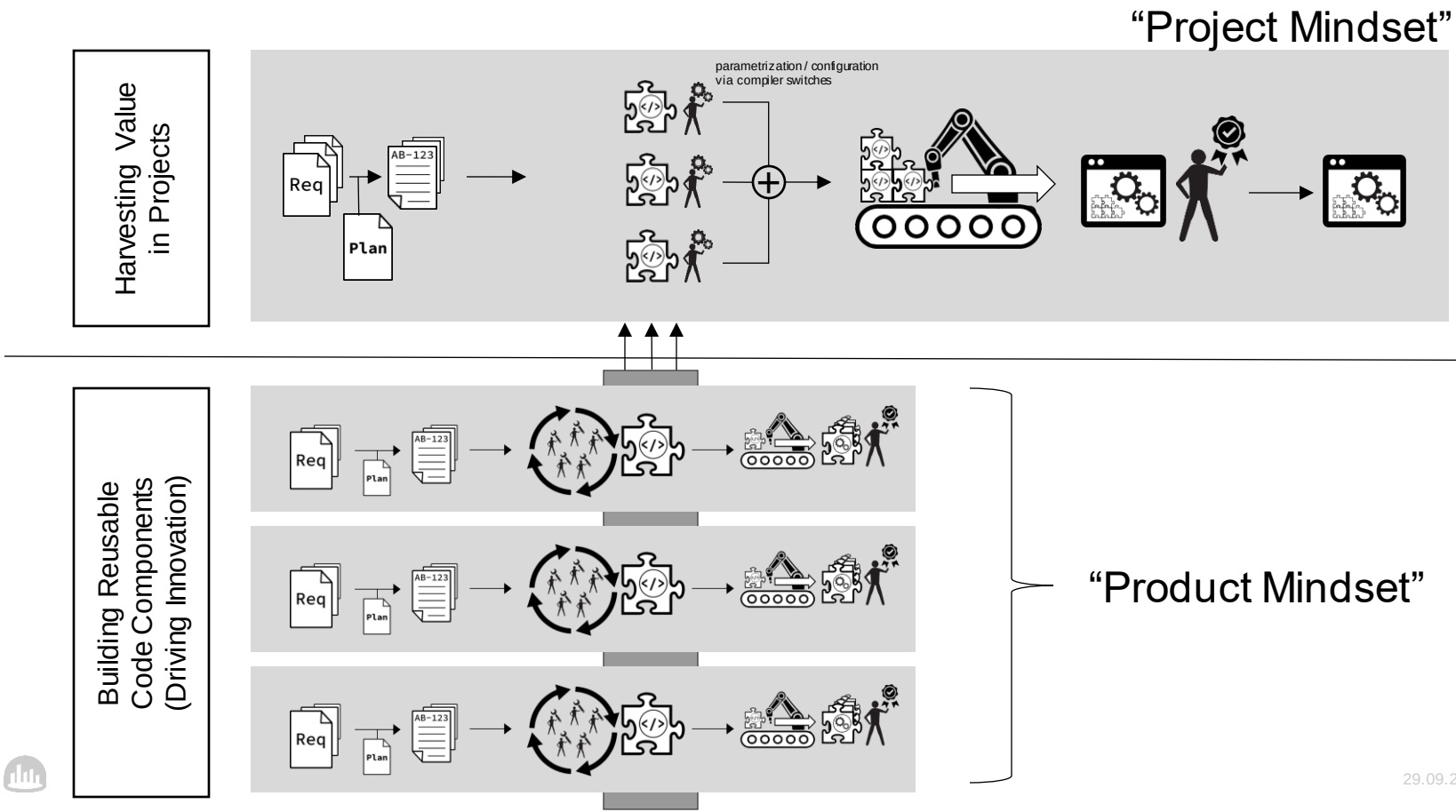


"Product Mindset"

each component is managed as a product

- **Long-term responsibility for code evolution**
  → dedicated code repository
- **Tested standalone without project context**
  → fully-fledged CI/CD for the component
- **Strategic innovation roadmap**
  → farsighted planning across projects
- **"Selling" the component to the projects**
  → maximize (re)use in projects
  → maximize ease-of-applicability in projects

The Ideal Software Factory (if compile-time optimization is necessary)

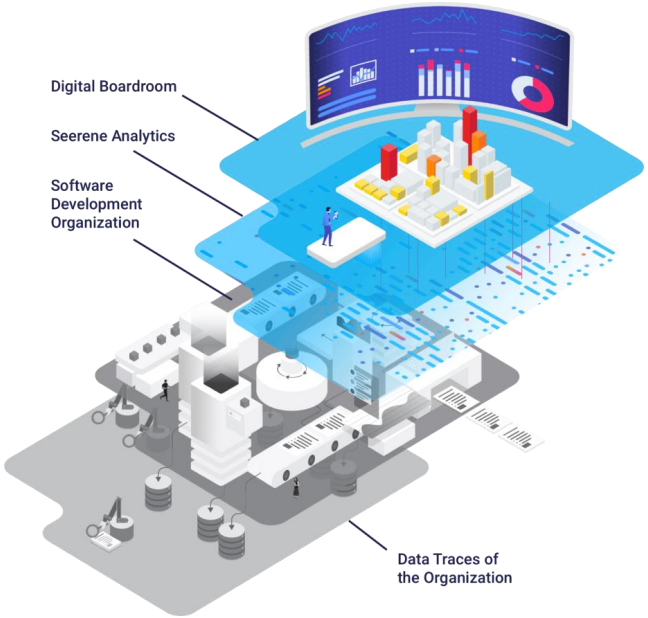


Summary

Can Automotive learn from digitally-born Software Companies? Yes – but also vice versa.

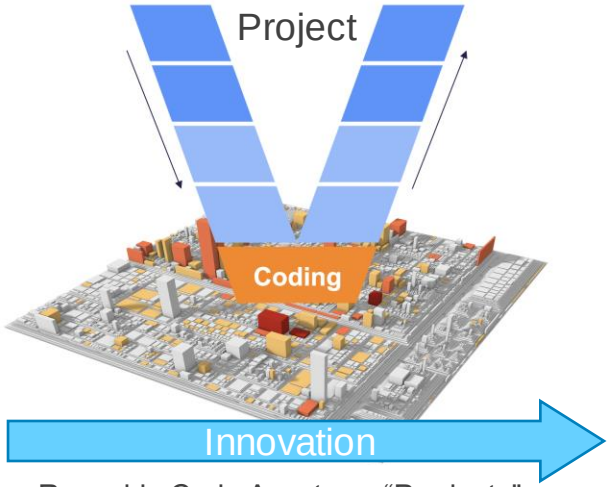
Establish data-driven transparency in your software factory.

Optimize process efficiency, time-to-delivery, resource allocation, ... like in hardware factories.

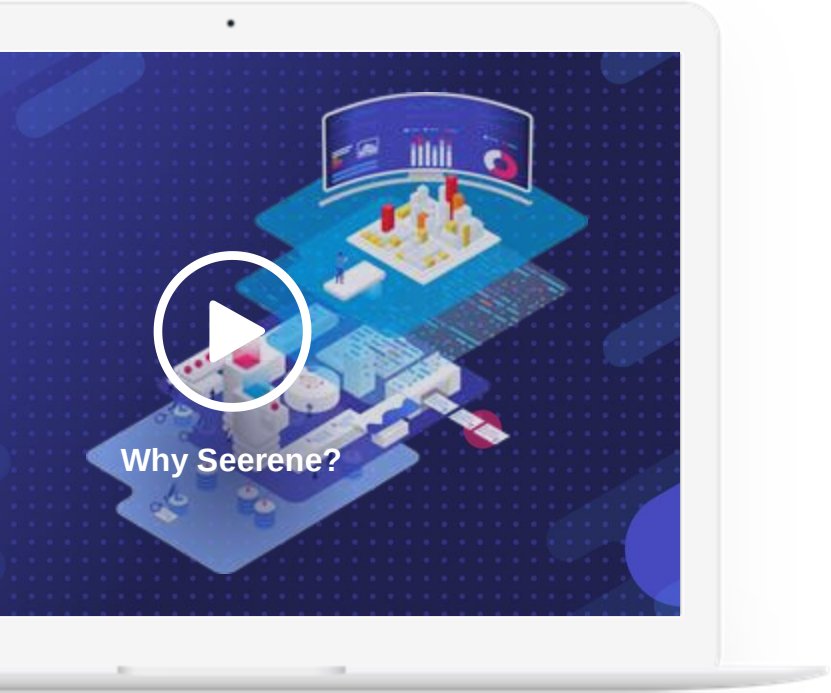


Shift thinking from "Projects" to "Product".

Decouple software from hardware.
Manage code bases as assets that are continuously equipped with innovation.



Reusable Code Assets as "Products"



Thank You!

Contact
+49 331 7062340
johannes.bohnet@seerene.com