seerene
The Digital Boardroom

Analytics as “Navigation System” for Steering towards highest Excellence in Software Production

Conference:
The Ideal Software Factory for Finance & Insurance
October 15th, 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, … →

Core IT applications enable efficient business operations.

Customer-facing software creates new business opportunities.

← InsureTech, FinTech

Digital Service Ecosystems

← Manufacturing, Mechanical Engineering

Smart Hardware (IoT)

→ **Tailor-made software is the basis for competitive advantages**

→ **User-centricity requires high-frequent delivery → Agility/Speed**

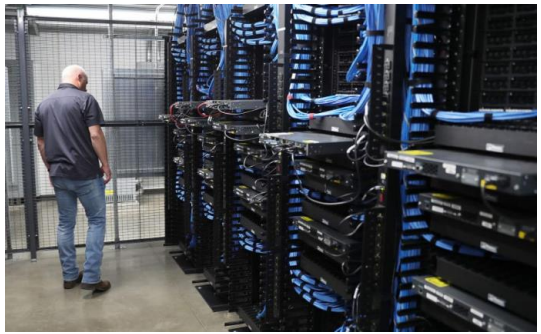
Multiple Strategies are necessary to Tackle the Challenges

Legacy Migration

- Replacing with “standard” solutions

Minimum Maintenance

- In-house or 3rd-party teams deliver releases with low pace



Core Backend Systems

Agile Methodology

- MVPs & Prototypes
- Long-term software product development with agile methodology
- Scaled agile organization

Customer-facing Software

Collaboration

- Startups as partners
- Integration into ecosystems

IoT & Ecosystems

Software Development Expertise is Key

Taking responsibility &
Active management of software development
is essential for success.

Legacy Migration

- Replacing with “standard” solutions

Minimum Maintenance

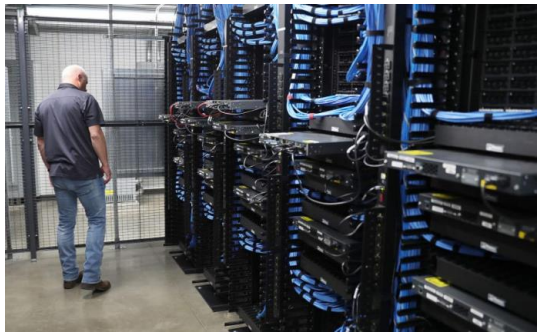
- In-house or 3rd-party teams deliver releases with low pace

Agile Methodology

- MVPs & Prototypes
- Long-term software product development with agile methodology
- Scaled agile organization

Collaboration

- Startups as partners
- Integration into ecosystems



Core Backend Systems

Customer-facing Software

IoT & Ecosystems

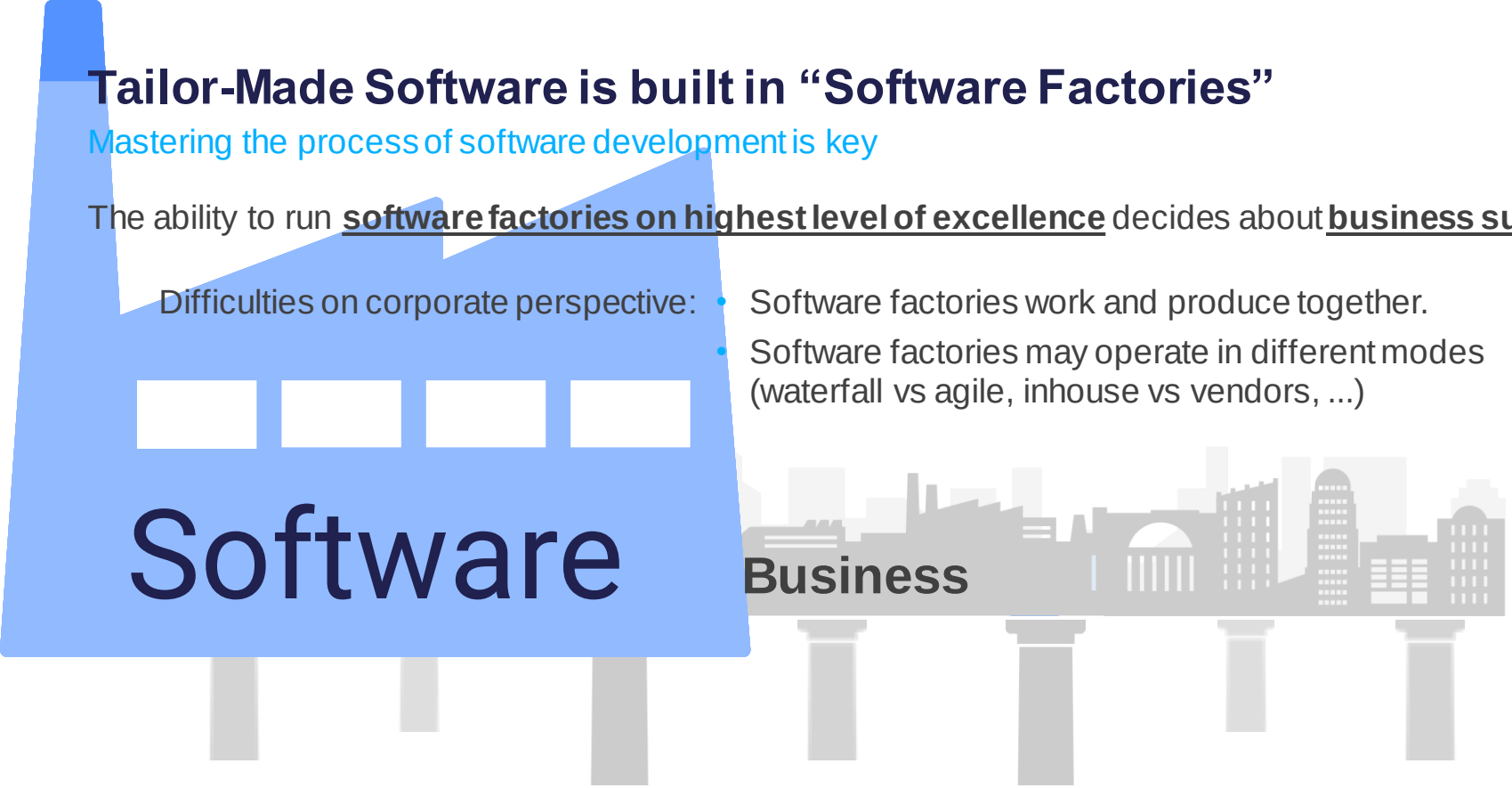
Tailor-Made Software is built in “Software Factories”

Mastering the process of software development is key

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

Difficulties on corporate perspective:

- Software factories work and produce together.
- Software factories may operate in different modes (waterfall vs agile, inhouse vs vendors, ...)



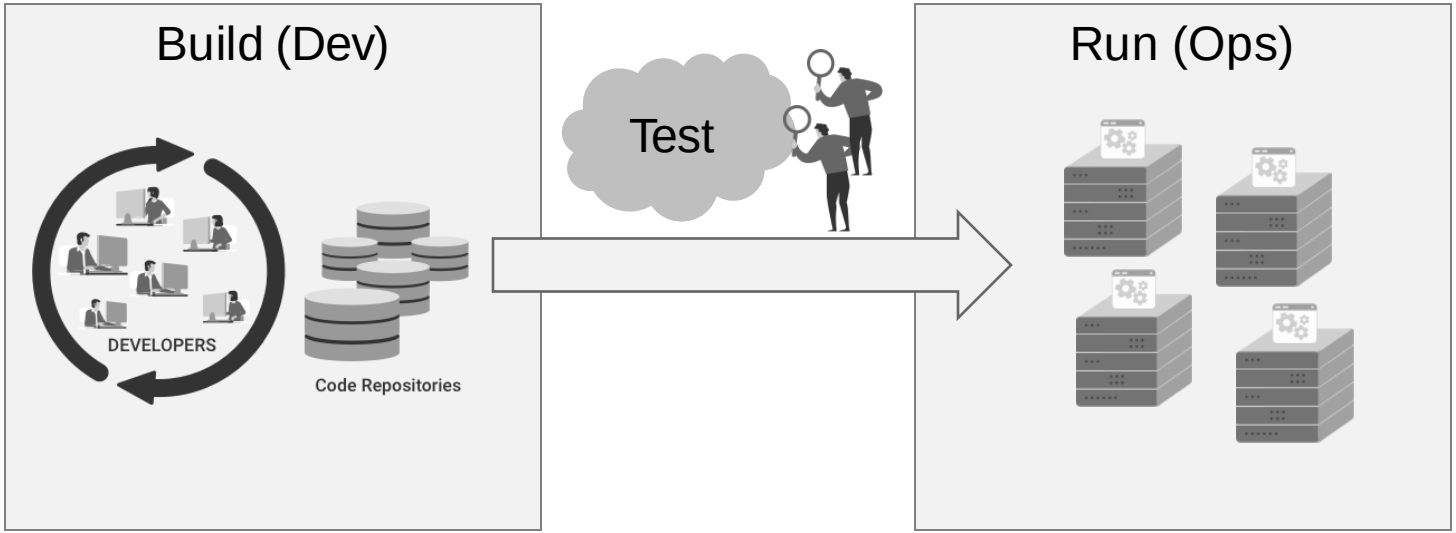
A brief history of a typical

Journey towards Agile

...and the typical problems that arise

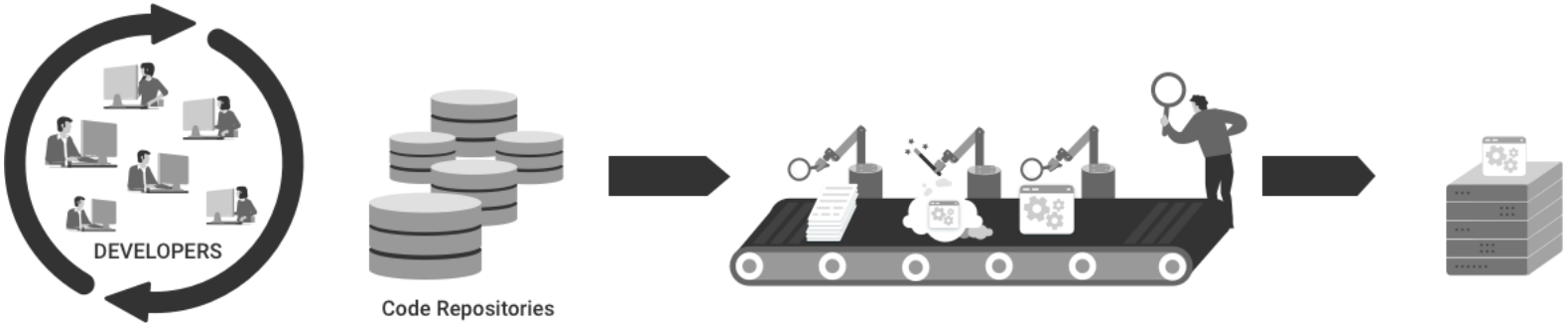
Tearing down the wall between “Build” and “Run”

Slow innovation cycles due to separated organizations



Dev + Ops = DevOps

Seamless process from coding, over testing to deploying production



Advantages:

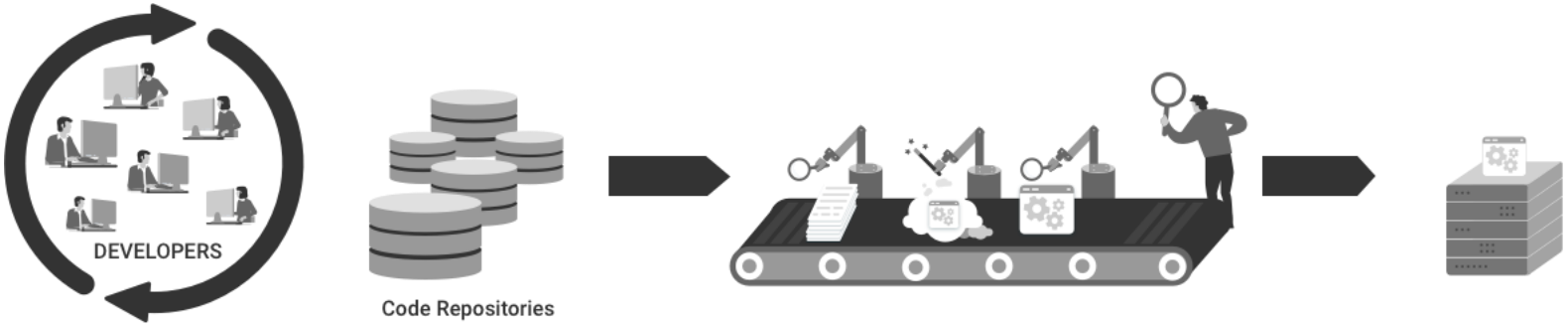
- Faster innovation cycles

Technical prerequisites for DevOps approach:

- Automated continuous integration & deployment processes (CI/CD)
- High degree of test automation
- Flexible, virtualized hardware setups
- Decoupled (runtime) architecture

DevOps creates follow-up Challenges

Seamless process from coding, over testing to deploying production



DevOps approach requires:

- Automated continuous integration & deployment processes (CI/CD)
- High degree of test automation
- Flexible, virtualized hardware setups
- Decoupled (runtime) architecture

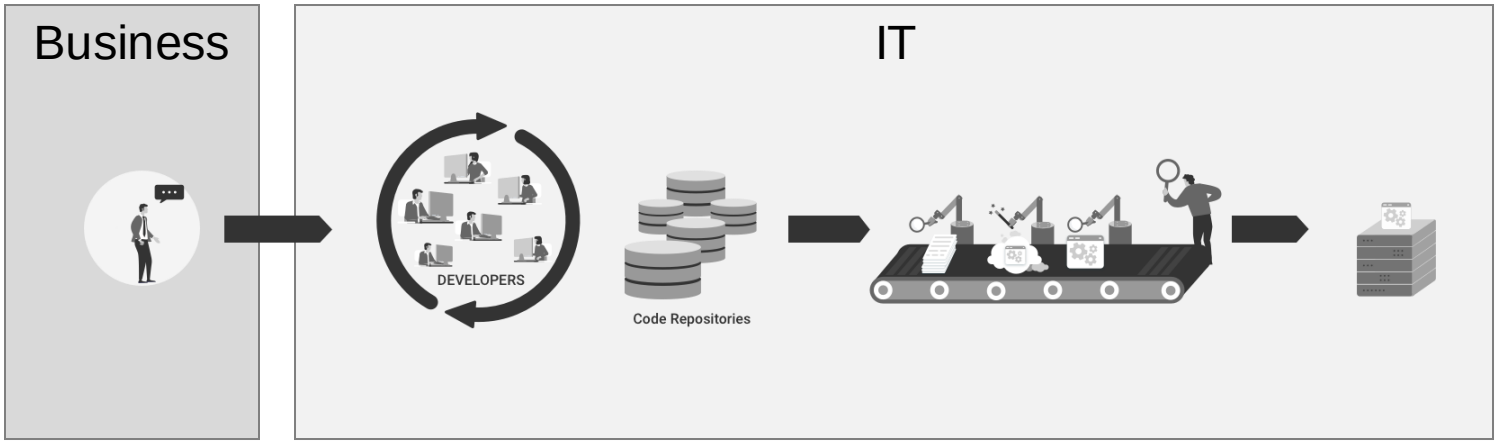
Major follow-up challenges:

- Cloud Infrastructures
- [Micro]Service Architectures (breaking monoliths)
- Heavy coding investment into automated tests

The next wall: Tearing down the wall between Business and IT

Agile methodology includes the business players into the equation: BizDevOps

Mind shift: From **pre-planned projects** to **trust-based time-and-material journeys**



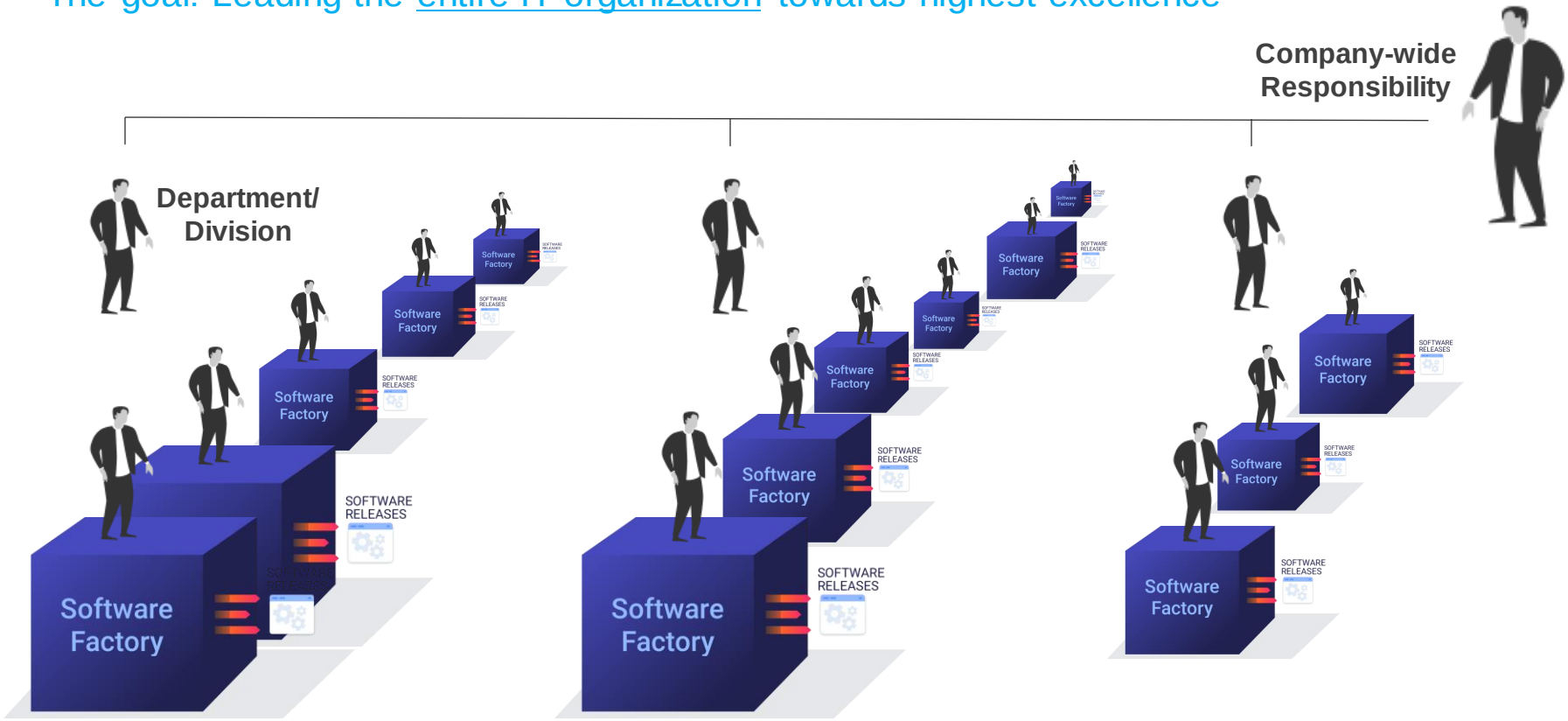
Better: Business & IT are sitting in the same boat → more trustful negotiations

Major challenge remains:

- How to prove the business that the new way of working is beneficial for the business?
  - Before: "know upfront what to get for the money".
    Now: "don't know what to get, don't know the costs".
- How to convince business to vote for technical housekeeping work instead of features?

Scaled Agile (on company level) is much more than Agile

The goal: Leading the entire IT organization towards highest excellence



Management prior to Agile Transformation: Culture of Control

"Big ship projects", planned and equipped in advanced before put to sea

Advantages:
(supposedly)

- Control over costs
- Clearly defined requirements / features
- Plan vs. actual comparison during the journey

Disadvantages:

The manifold reasons why your organization started the IT-transformation:

- Innovation cycles are too slow for your company's business in a fast-paced world.
- Planned requirements are oftentimes outdated when the software release is delivered.
- …

Agile Methodology lacks the Birds-Eye Perspective

How to lead the entire “fleet” to highest excellence?

Agile methodology is team-focused:

- Measurements for optimization are only valid within each individual team (e.g., story points, burn down rates, self assessments, …)
- Only loosely sharing of knowledge and practices across teams via “guilds” or “communities of practice”.
- Even large-scale agile approaches (e.g., SAFe) steer the “fleet” only via the scope (features) into the teams.

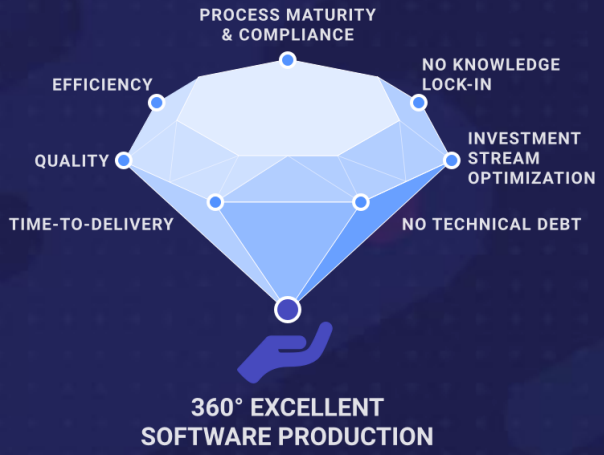
Major challenge:

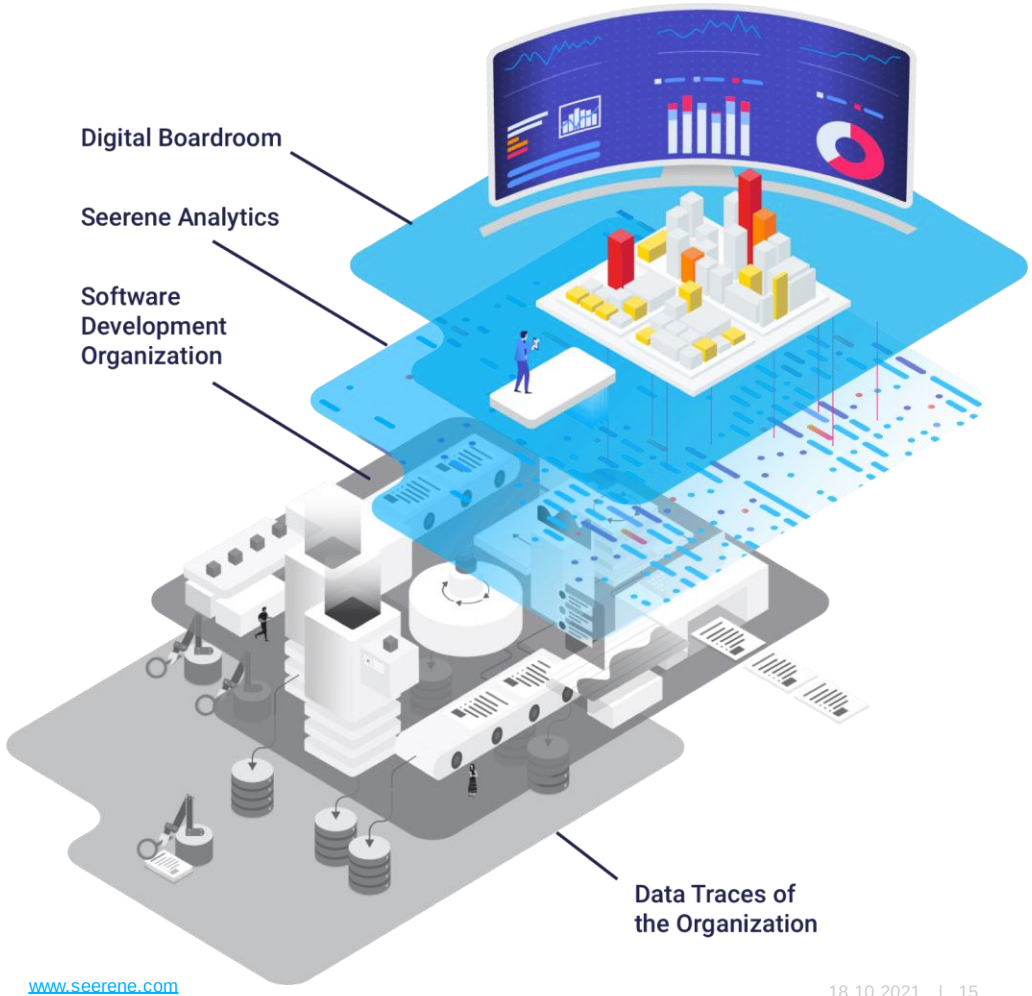
- How to objectively measure and compare across teams without reducing the teams’ empowerment and autonomy?
- How to prove to the business that agile is better than waterfall methodology?


Full Transparency about the
Software Production Process...
...across all software factories
in the entire company

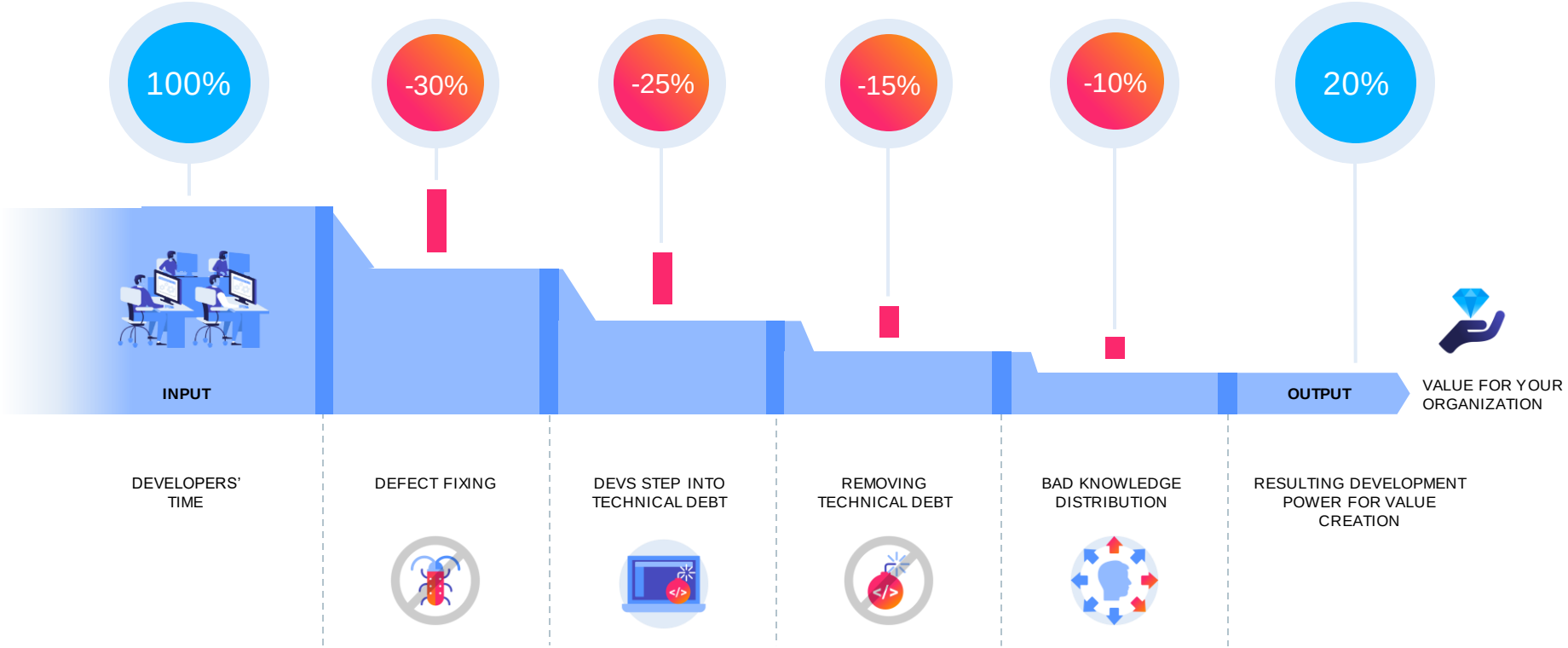
Establish a Digital Boardroom with Analytics

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



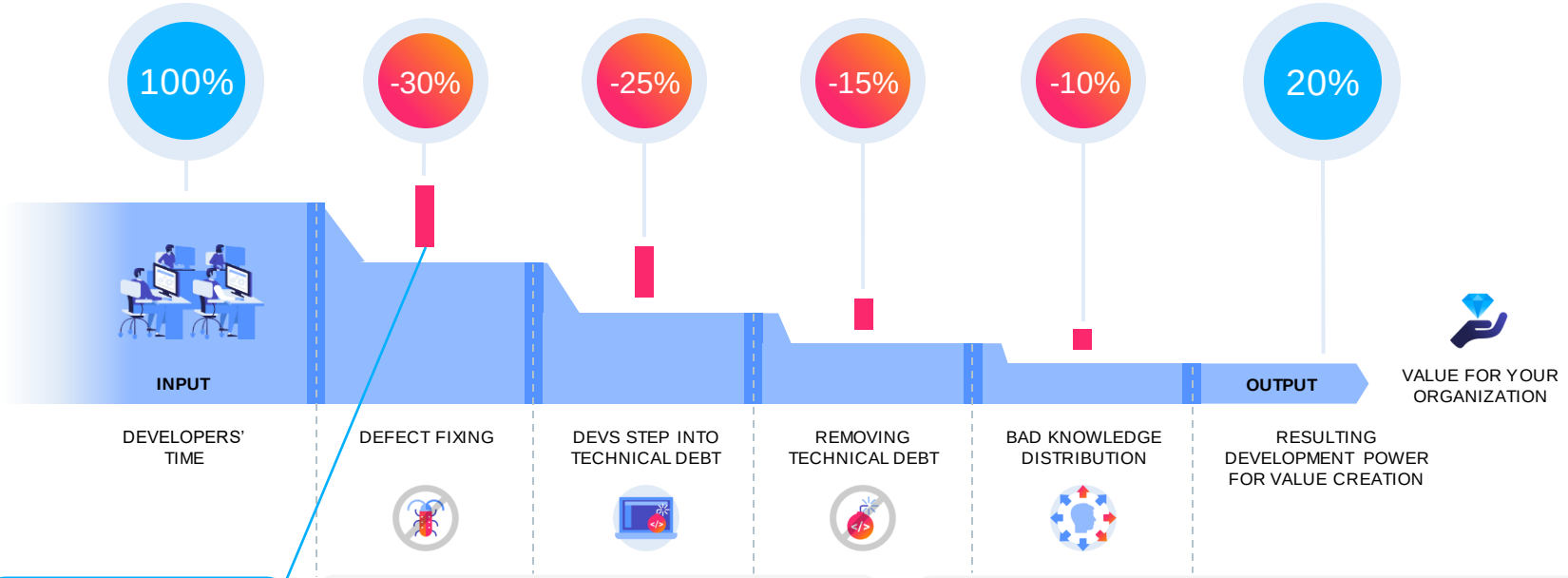


Quantifying Inefficiencies in the Software Production Process



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**

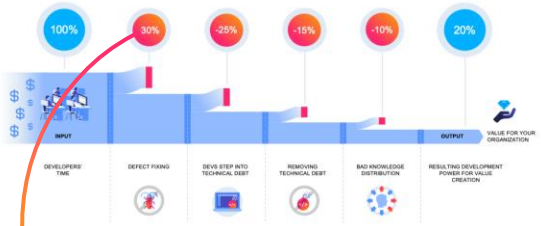
- **Seerene KPIs** enable you to quantify and manage the problem

Expert Level: **Actionable Insights**

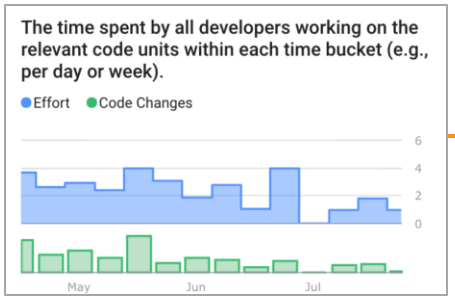
- **Seerene drill-downs (e.g., Code Map Visualization)** enable you to precisely locate the root-cause for the problem and fix it with minimal investment

Actionable Insights for Gaining Extra Power

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



Coding effort spent for defect-fixing



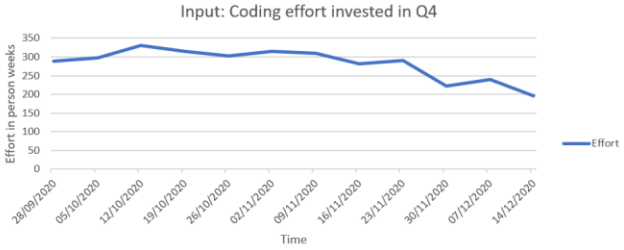
- Reveal how much effort is spent for defect-fixing
- Locate code units that constantly need to be fixed

→ Be able to effectively eliminate the problem

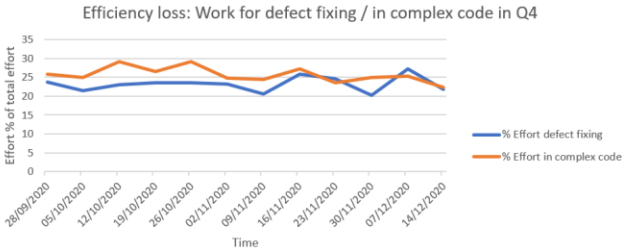
Transparency across all Software Factories in the Company

Inhouse vs. 3rd-party, programming languages and tech stacks, legacy vs. new, ...

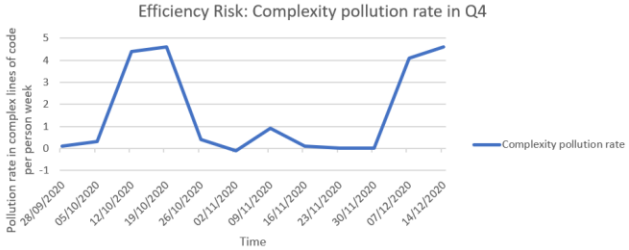
Investment

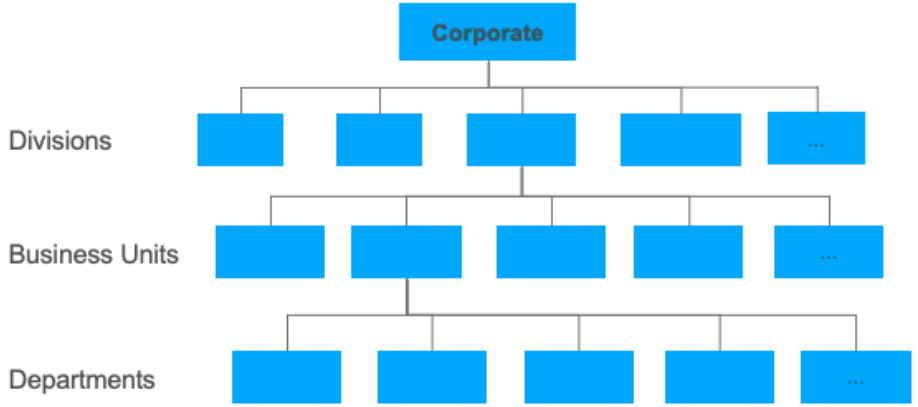


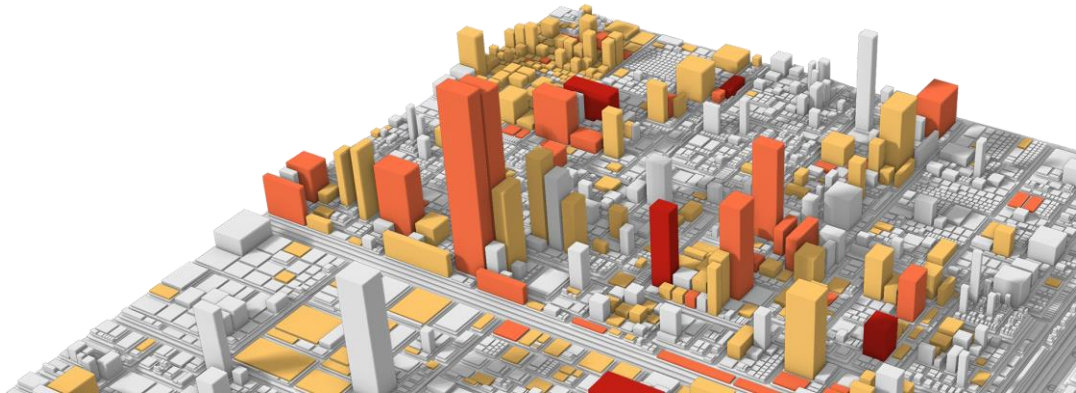
Efficiency



Technical Debt
(Future Efficiency Risk)







Breakdown of Performance KPIs According to Sub-Org-Structure

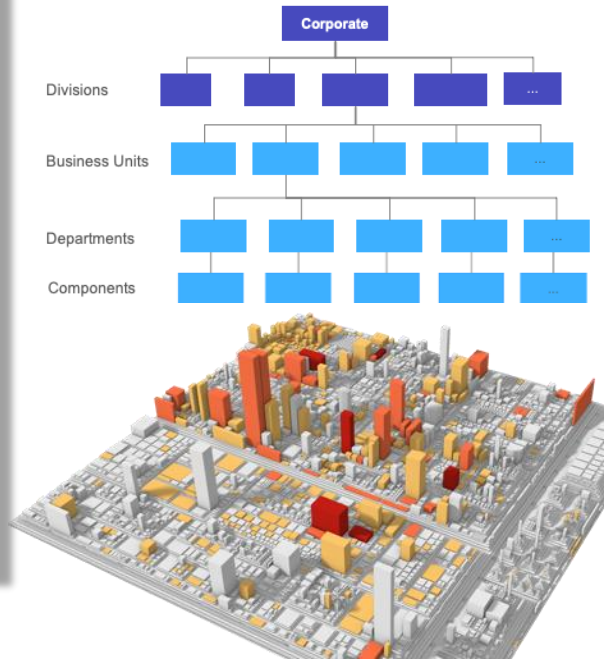
Where do we invest in code production?

Where do the teams experience productivity losses?

Where do the teams leave future problems behind?

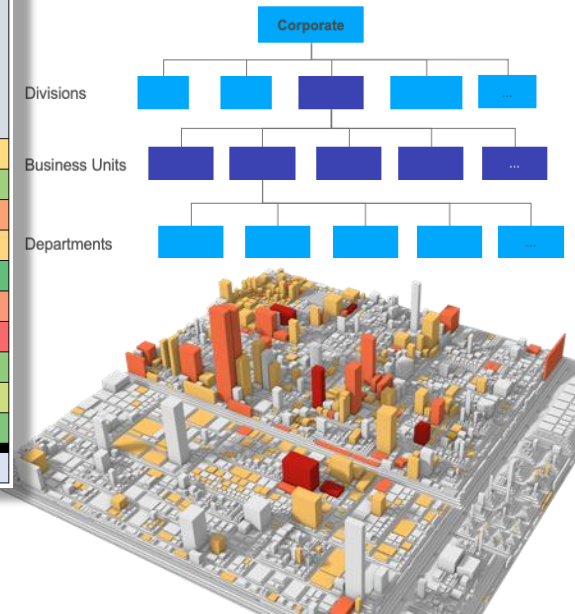
Where is new code being produced? (delta logic statements per developer w eek)

	Coding Invest		Efficiency				Technical Debt		Output
Divisions	Effort total	Resources	% Effort Defect Fixing	% Effort in Complex Code	Cognitive Load by Complexity	Cognitive Load by Undoc Logic	Complexity Pollution Rate	Undoc Logic Pollution Rate	Throughput
	762,8	216	22%	34%	39,0	12,2	11	1	45,8
	639,8	172	28%	22%	9,8	5,8	0	1	6,6
	448,4	138	22%	33%	13,3	78,8	4	13	27,9
	297,2	127	20%	20%	9,7	0,8	2	0	6,3
	280,8	213	5%	0%	0,2	0,5	0	0	0,9
	198,5	88	27%	34%	32,1	8,3	-3	0	2,3
	179,3	72	35%	38%	68,0	17,6	11	3	11,9
	127,8	56	32%	35%	42,5	10,0	0	0	2,4
	68,2	39	19%	11%	6,9	9,8	9	1	31,2
	40,0	36	37%	46%	50,7	4,6	0	0	2,0
	11,3	9	29%	5%	1,2	2,0	4	1	20,3



Further Drill-Down into the Organization

Governance		Input		Efficiency				Technical debt		Output
Divisions	BUs	Effort total	Resources	% Effort Defect Fixing	% Effort in Complex Code	Cognitive Load by Complexity	Cognitive Load by Undoc Logic	Complexity Pollution Rate	Undocumented Logic Pollution Rate	Throughput
		125,7	48	24%	24%	15	6	0,2	0,0	6,1
		104,7	32	15%	15%	8	4	-0,8	0,0	1,1
		104,6	38	31%	11%	3	3	0,4	0,6	11,9
		79,8	23	41%	29%	5	3	0,5	0,2	6,6
		75,6	24	35%	22%	4	2	-1,3	0,0	-1,1
		61,2	24	24%	25%	11	8	0,3	1,3	12,9
		45,1	13	31%	36%	33	27	6,4	3,7	17,7
		31,9	14	30%	25%	3	1	0,0	0,8	0,5
		10,8	7	39%	32%	25	6	0,4	0,6	2,8
		0,4	2	0%	0%	0	0	0,0	0,0	0,0
	Total	639,8	172	28%	22%	10	5,8	0	1	6,6



Drill-Down into Root Causes on Source Code Level

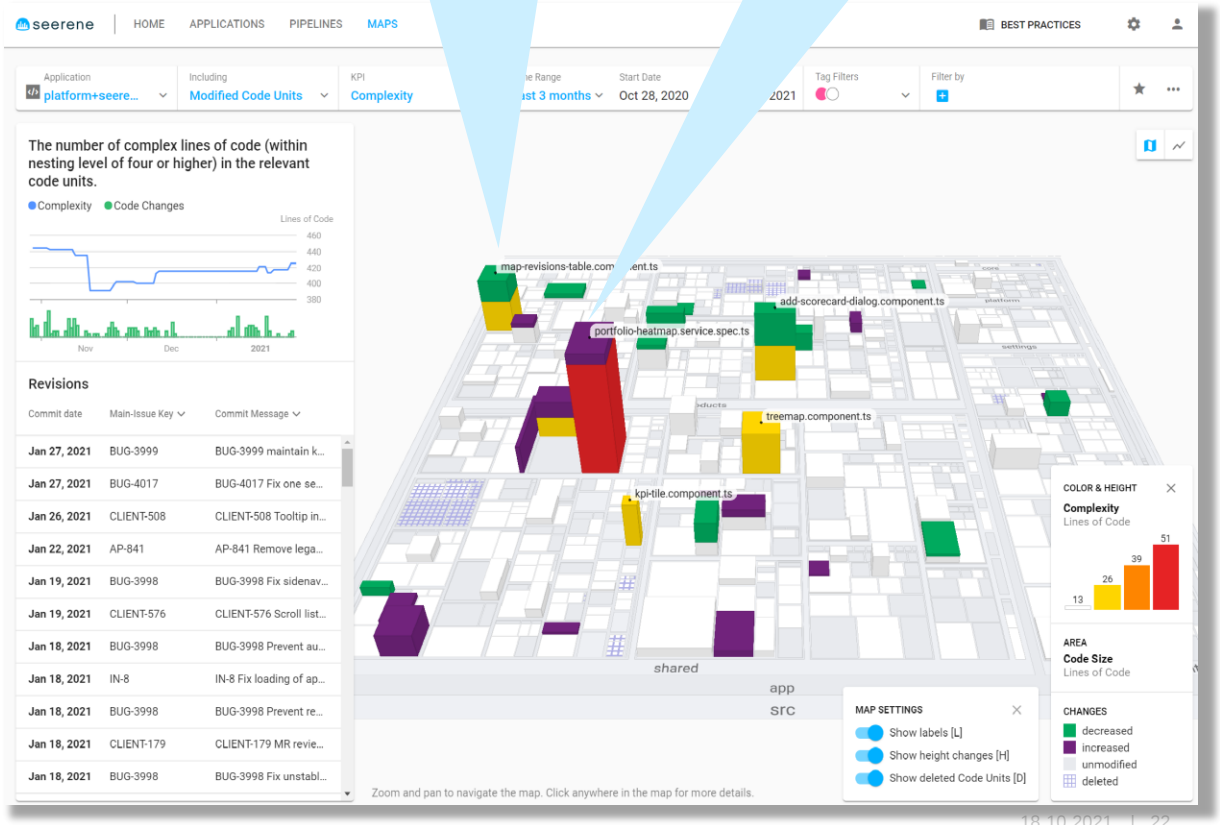
green “roofs” = good = KPI decrease

purple “roofs” = bad = KPI increase

Performance KPI for Business Unit X

Governance		Input		Efficiency				Technical debt		Output
		Effort total	Resources	% Effort Defect Fixing	% Effort in Complex Code	Cognitive Load by Complexity	Cognitive Load by Undoc Logic	Complexity Pollution Rate	Undocumented Logic Pollution Rate	Throughput
		125,7	48	24%	24%	15	6	0,2	7,0	6,1
		104,7	32	15%	15%	8	4	-0,8	0,0	1,1
		104,6	38	31%	11%	3	3	0,4	0,6	11,9
		79,8	23	41%	29%	5	3	0,5	0,2	6,6
		75,6	24	35%	22%	4	2	-1,3	0,0	-1,1
		61,2	24	24%	25%	11	8	0,3	1,3	12,9
		45,1	13	31%	36%	33	27	6,4	8,7	17,7
		31,9	14	30%	25%	3	1	0,0	0,8	0,5
		10,8	7	39%	32%	25	6	0,4	0,6	2,8
		0,4	2	0%	0%	0	0	0,0	0,0	0,0
	Total	639,8	172	28%	22%	10	5,7	0	1	6,6

KPI ComplexityPollutionRate is high for department Y



Summary

Manifold Software Challenges in IT

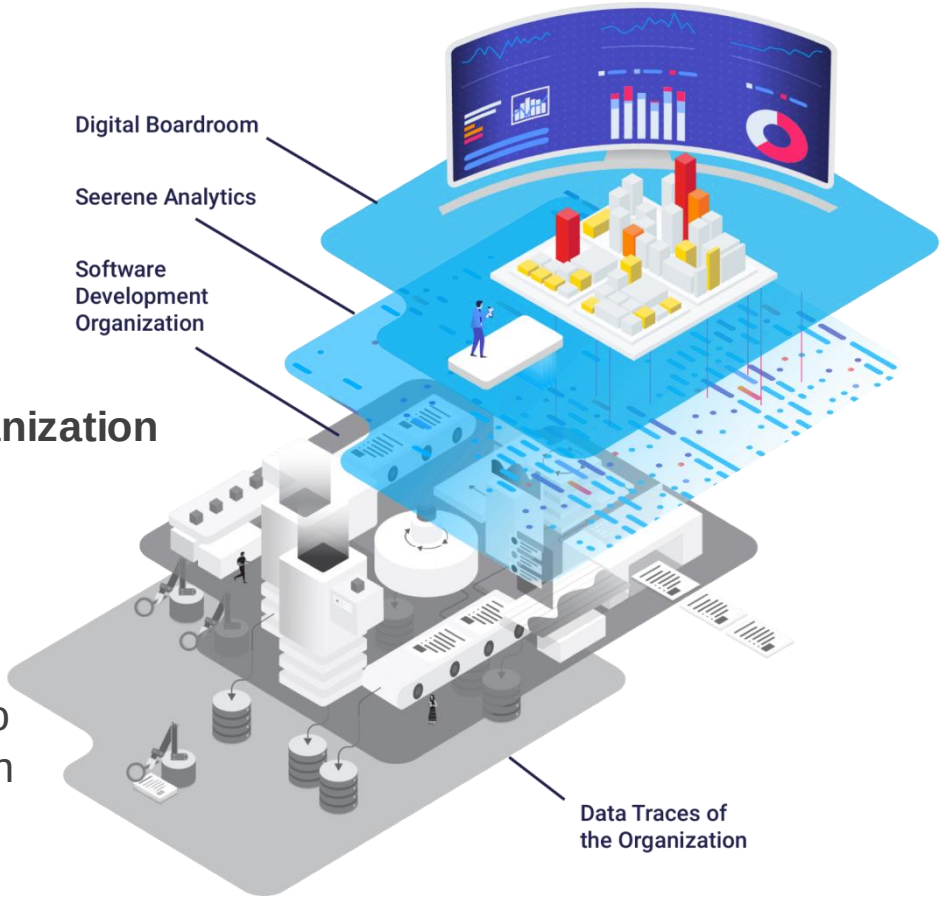
- Legacy systems
-

nt Organization

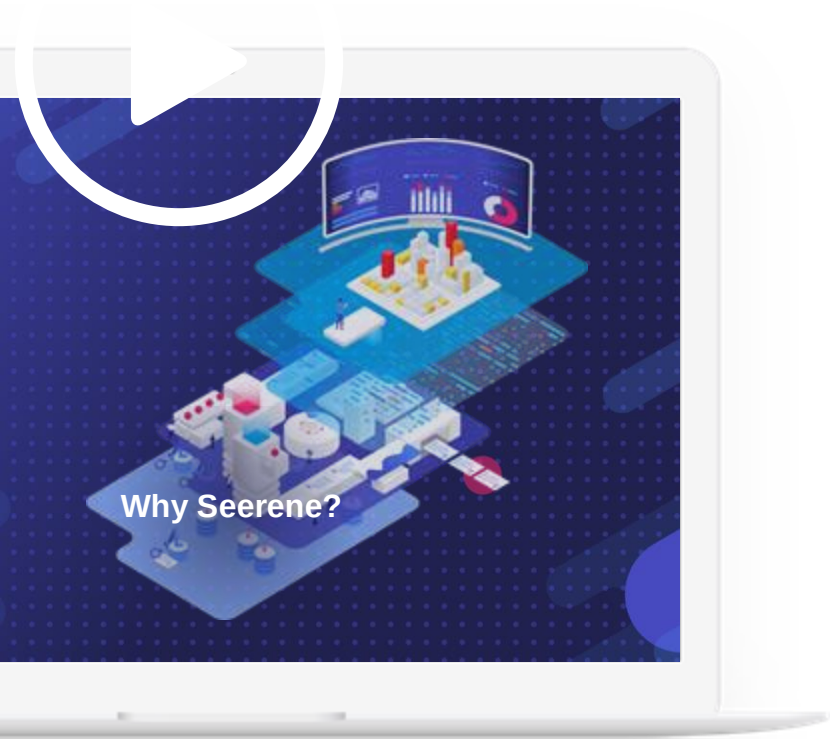
es

Analytics-based Transparency

as fundamental approach to be able to actively steer the software organization towards success and excellence



Software



Thank You!

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