

DevOps Insights for the Executable Digital Twin

Roland Pastorino, Bernhard Sputh
Model-based System Testing

May 16-18, 2023
DevOps Enterprise Summit, Amsterdam



The Executable Digital Twin

Definition

Executable form of a Digital Twin

Deployed alongside or as part of a physical system-under-test

Creates a live connection between the virtual world and the real world.

Enable digital twins to predict and optimize the performance and behavior of real systems.



The Executable Digital Twin Examples

Accelerate product R&D with real-time testing following a system-in-the-loop approach

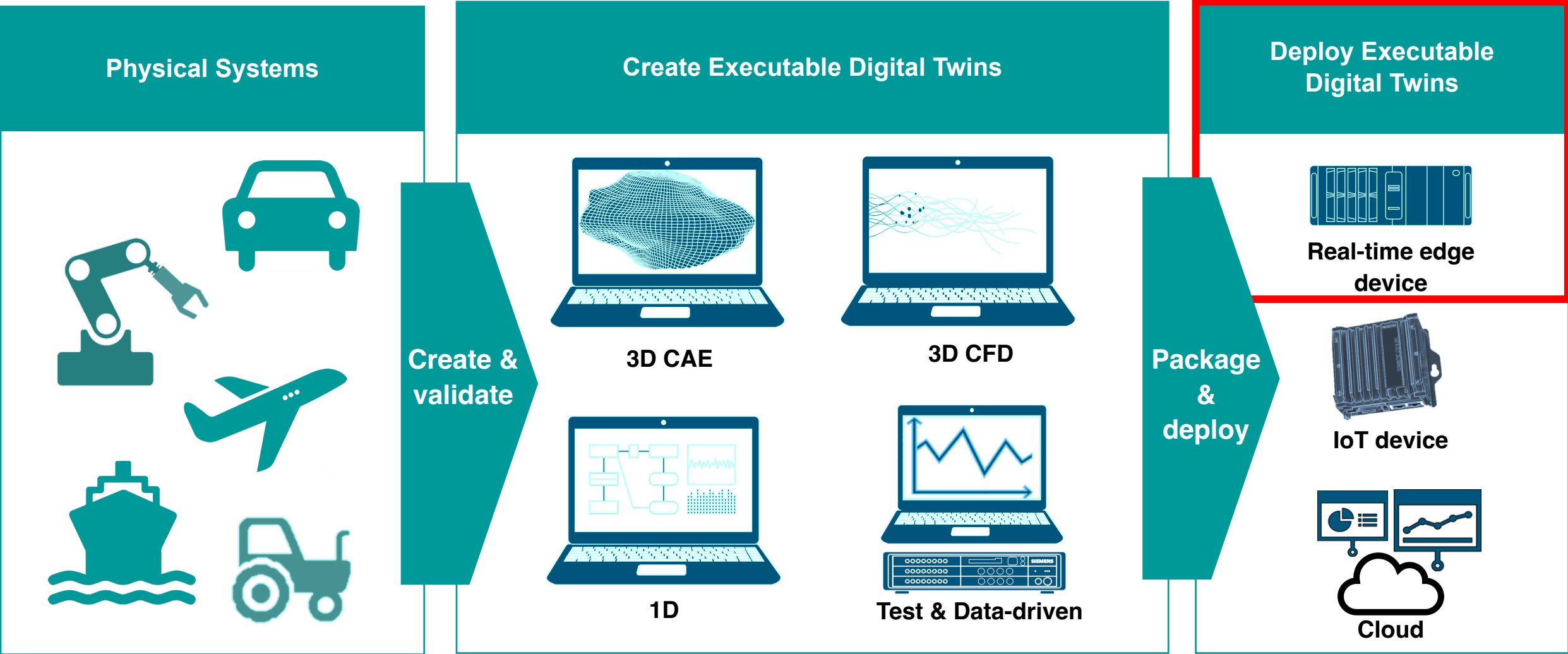


New engineering insights with virtual sensors to obtain otherwise unmeasurable quantities



The Executable Digital Twin

Creation and deployment



The DevOps starting point for one of the Digital Twin Executable platforms



**Real-time edge
device**

New concepts, new difficulties

- The concept of “Executable Digital Twin” was not yet established but was crystallizing.
- The variety in Digital Twins was a big challenge to the ideation of their execution needs and execution platforms.

Industrial solutions

- The product incubation phase had reached its limits in terms of quality & expandability.
- All aspects of industrial solutions (quality, safety, traceability...) had to be taken into account to further progress.



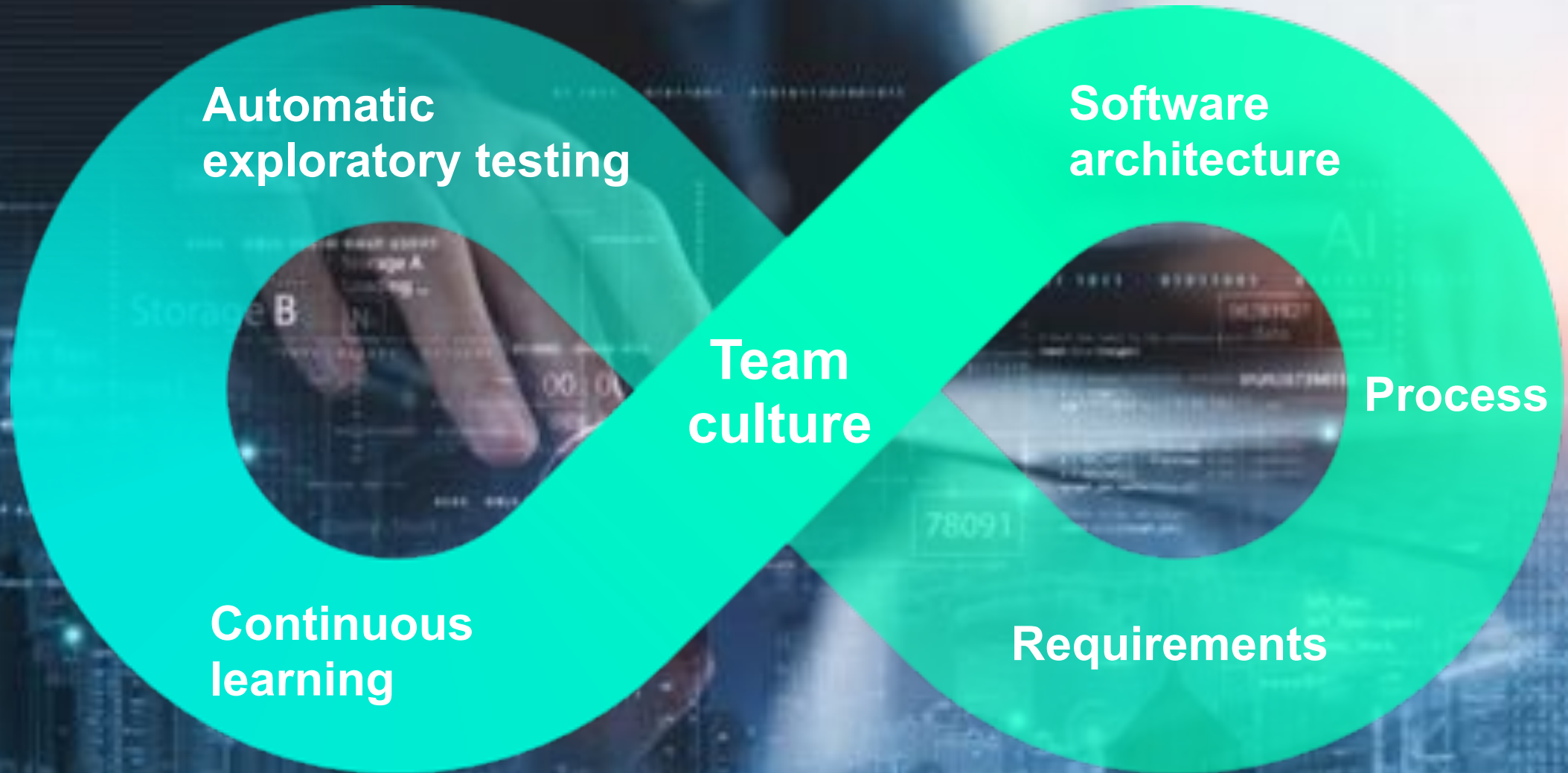
New code base & stakeholders

- The new code base was quickly growing despite the development processes not being fully in place.
- The number of stakeholders (developers, testers, users, Digital Twin providers...) was also becoming larger thus introducing new organizational dynamics.

Need to scale

- The product solutions had to scale with the variety in Digital Twins and in their connection to the physical world.
- The product development (software architecture, processes...) had to sustainably scale for the coming years to be successful.

6 DevOps Insights



DevOps Insights #1

Team culture



Executable Digital Twins for vehicle R&D on test benches

5 essential values with impact for us

	Knowledge	... is paramount. It needs to be carefully managed at team & individual levels for years.
	Team goals	...always comes first. They prevail over individual goals or interests.
	Value mindset	...must be applied systematically to maximize the impact of everyday tasks.
	Responsibility	...for the team's work and one's work, inside and outside the organization.
	Openness	...is at the core of team work and is embedded in all processes.

DevOps Insights #1






Team culture

5 essential values

	Knowledge
	Team goals
	Value mindset
	Responsibility
	Openness



5 positive impacts we experienced

	Expertise	...is constantly revised and adapted. It spreads in the team across seniority levels ensuring necessary knowledge redundancy. It prevails over seniority.
	Long-lasting trust	...is established between individuals, facilitating mutual support.
	Getting the right thing done	...on time, at team & individual levels. This ensure delivery speed and quality for new product features.
	Technical debt	...accumulation is minimized since the start. It is later reduced or eliminated to improve code base health.
	Psychological safety	...is established contributing to higher motivation, less human mistakes and a sense of ownership.






DevOps Insights #2

Processes



Executable Digital Twins on the factory floor

5 highlights of what worked for us

	Automate	...repeating tasks as much as possible (meeting reports, pending work overview, problem reports, ticket clean-up ...).
	Review	...(part of) your processes regularly based on their effectiveness. Use trial and error to optimize.
	Communicate	...effectively, clearly and timely all the time (progress reports, software designs, problem reports, code reviews, requirements...)
	Clear & Lean	...processes are essential to allow everybody to follow them on auto-pilot.
	Assign	...each work item to one and only one person responsible to drive it to completion.

DevOps Insights #2






Processes

5 highlights

	Automate
	Review
	Communicate
	Clear & lean
	Assign



5 positive impacts we experienced

	Agility	...is maintained in the long run at team level and individual level. E.g. meeting content can be prepared during the meeting thus reducing preparation effort.
	Performance	...of the team is maximized by removing all unnecessary work and inaccurate task assignments.
	No stress	...during release or in urgent situations by following the processes. "Trust the system, follow the processes".
	Safety	...of the software creation is ensured for the entirety of the processes.
	Motivation	...of going to work knowing what needs to be done, why, by whom and by when.

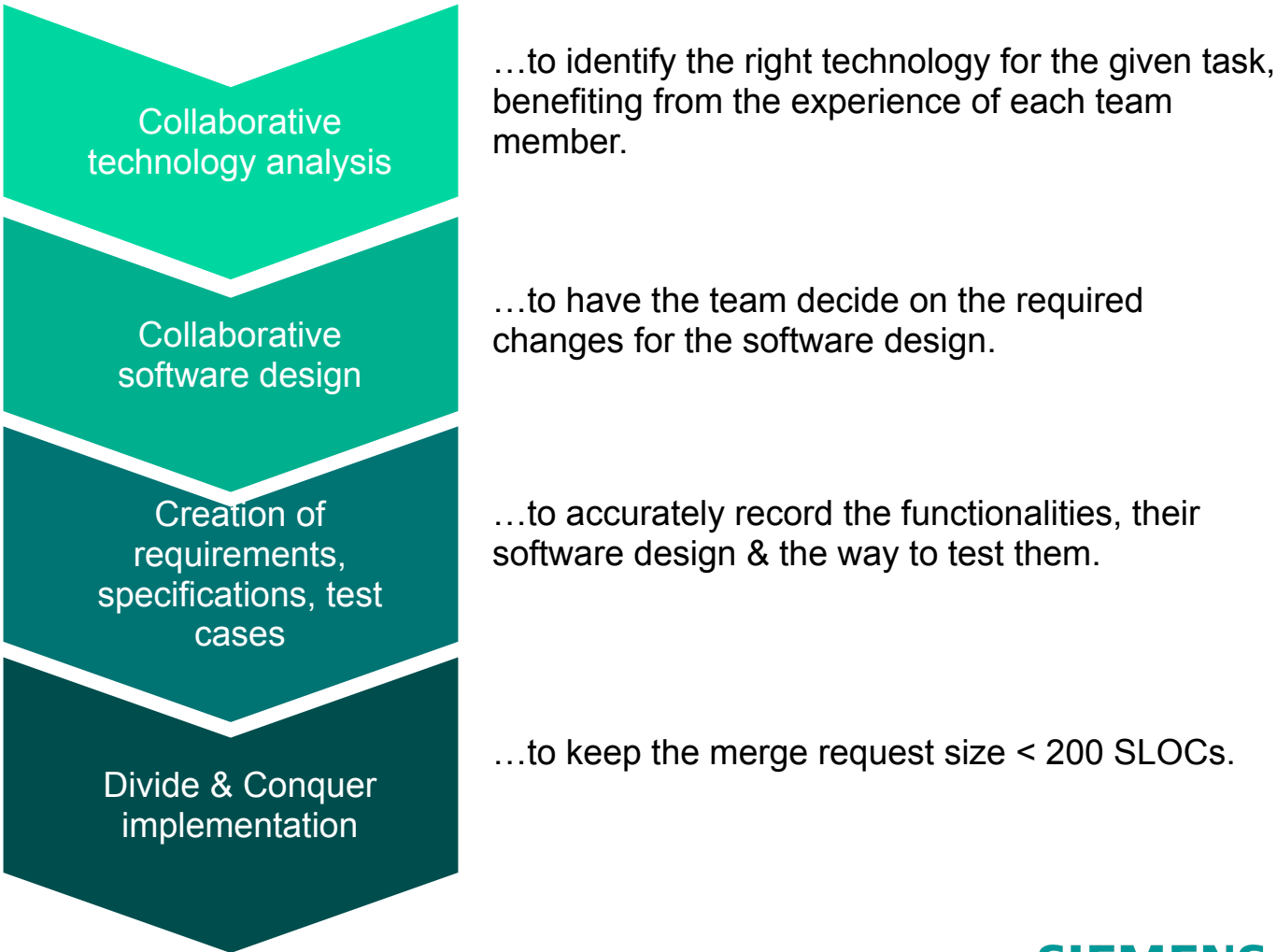
DevOps Insights #3

Software architecture for DevOps



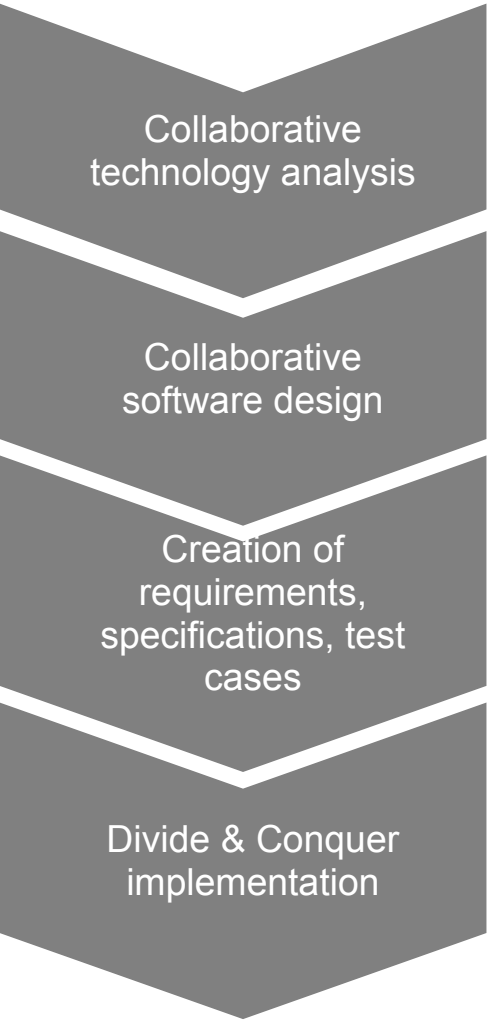
Accuracy validation of newly-created Executable Digital Twins

From ideation to creation. What worked for us



DevOps Insights #3

Software architecture for DevOps



4 positive impacts we experienced



Clarity

...of

- how the software architecture gets modified.
- how the software architecture is captured and can be consulted.



Team benefits

...by

- making the architecture a team effort.
- not having code ownership. Anyone can propose changes.
- enabling (almost) any developer to address any issue.



Micro-services

- 100+ orthogonal code repositories.
- ~5 MB code repository size (average).

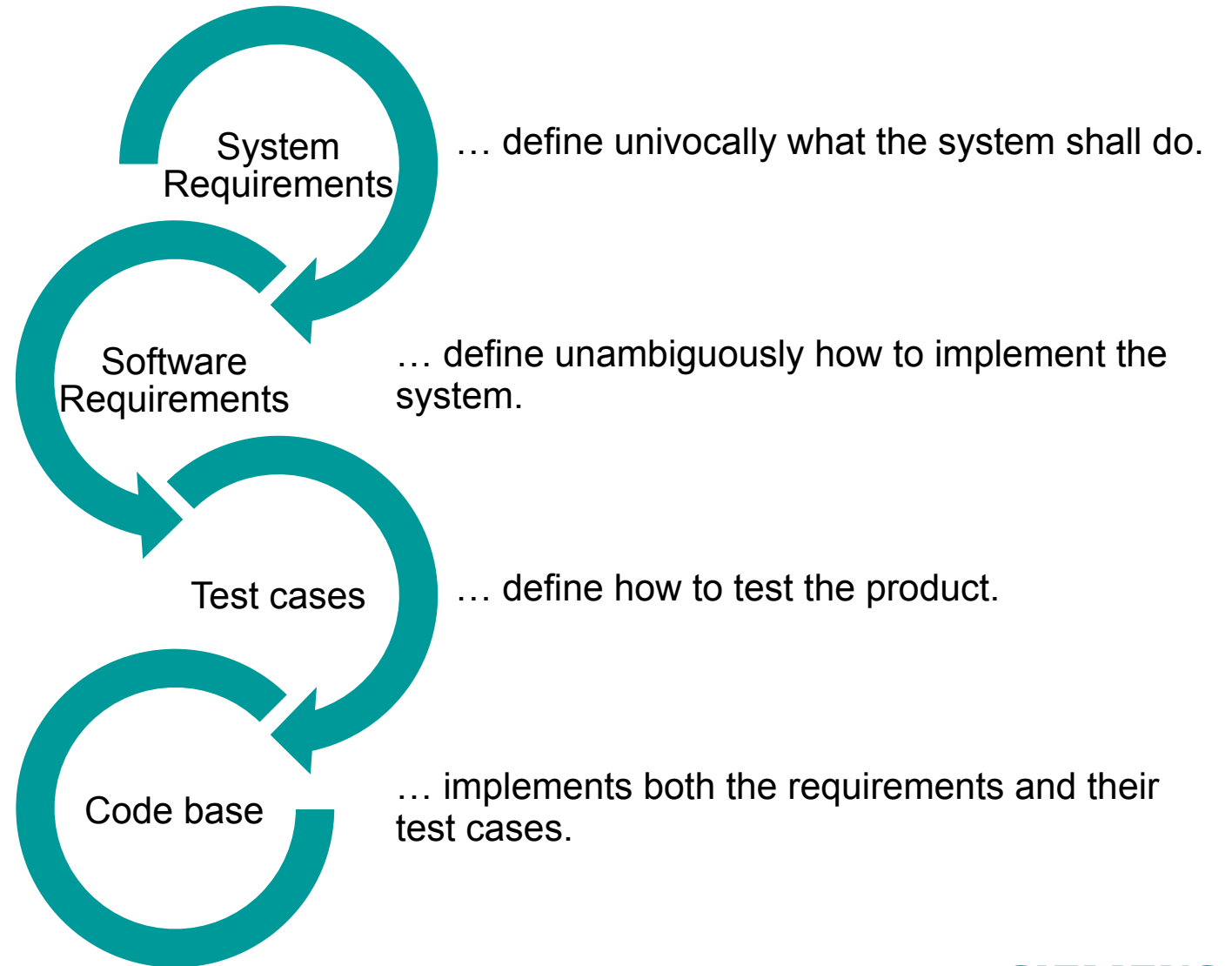


Creation flow

...of new features is maximized. Developers close ~10 issues per week. Often multiple developers can work in parallel on the same feature.

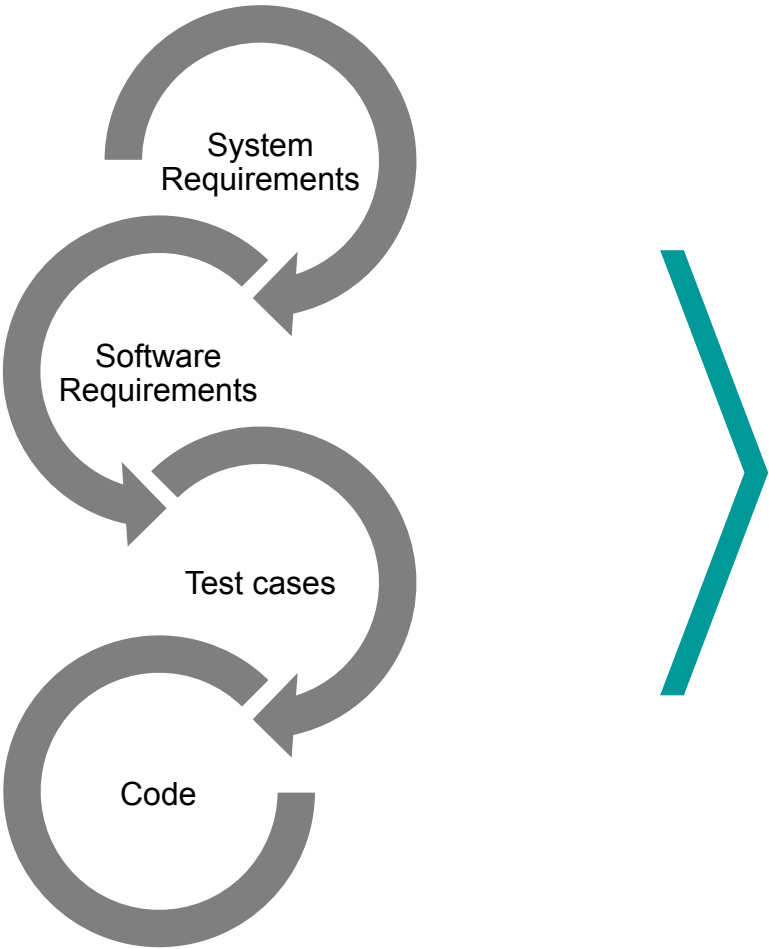
DevOps Insights #4

Requirement, specification and test case management

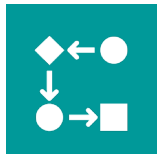


DevOps Insights #4

Requirement, specification and test case management



4 positive impacts we experienced



Traceability

... from requirement to code and vice versa. Vital when developing safety critical systems.



Ground truth

... of what shall be in the product, how to implement it, and how to test it. Requirements, code & tests evolve together.



Second brain

... that accumulates the product and team knowledge over years, next to the decisions taken and by whom.



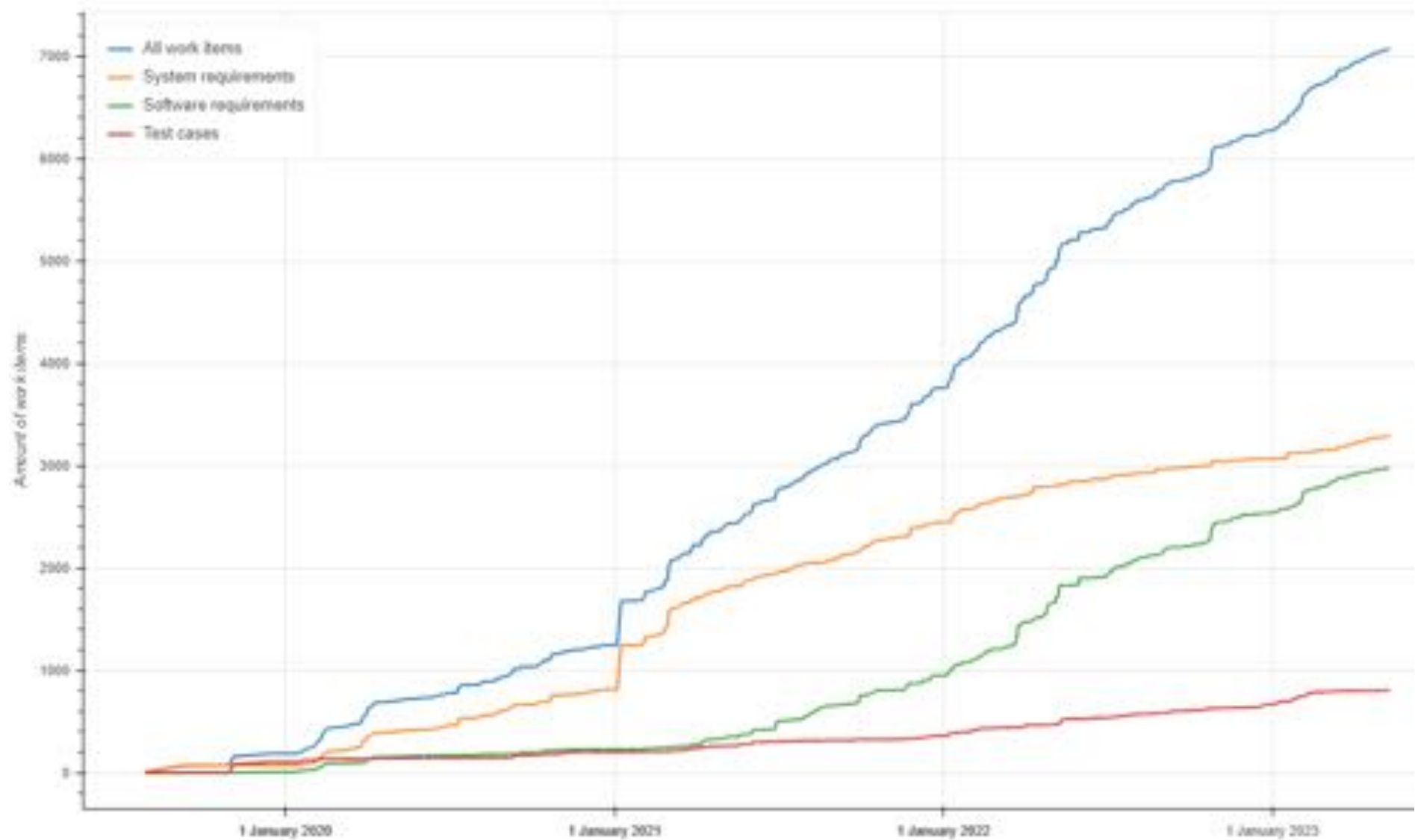
Clarity

... for all stakeholders of

- what the product needs to do
- how it is implemented
- how it is tested

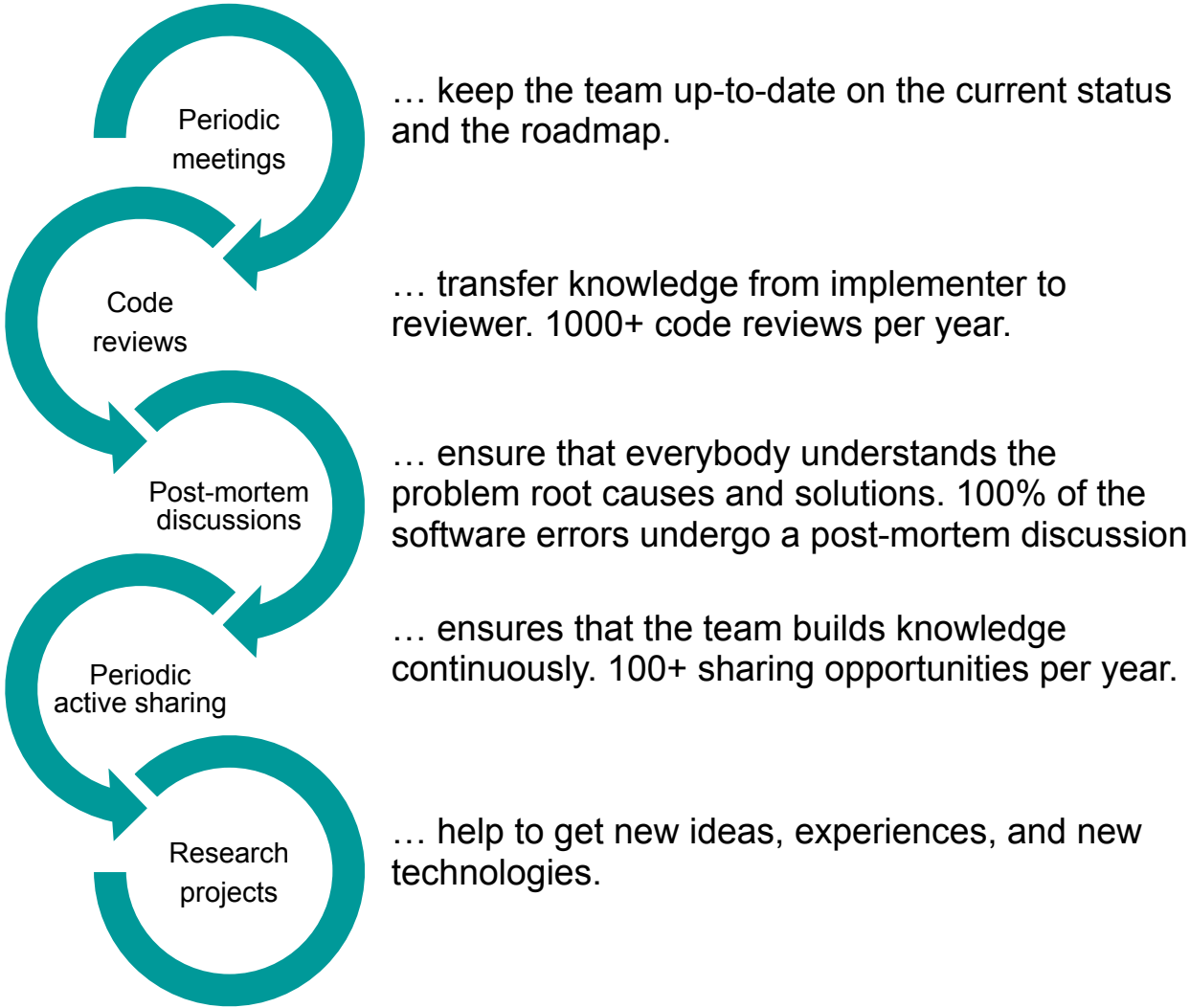
DevOps Insights #4

Requirement, specification and test case management



DevOps Insights #5

Continuous learning



DevOps Insights #6

Automatic exploratory testing



Executable Digital Twin down to the production line



- Tests only find what they are designed to find.
- The imagination of testers has (human) limits.
- We want to detect unknown software errors in the products.

Automatically explore the test space using:

- Fuzzing
 - Mutation testing
 - Design Space Exploration heuristics
-
- Software errors are found before revealing themselves in production.
 - Continuous test suite improvements.

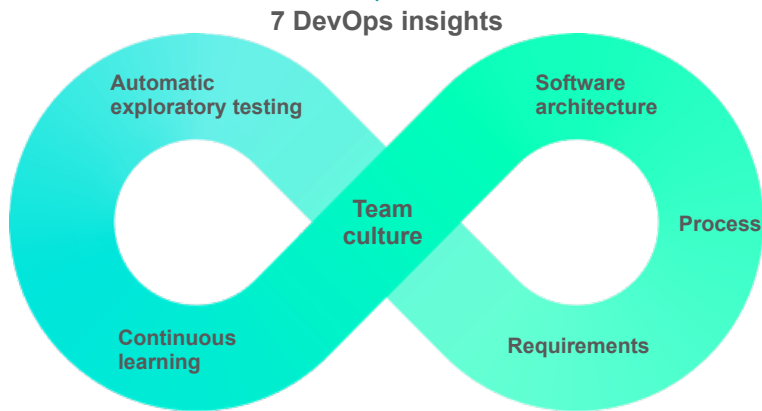
DevOps Insights for the Executable Digital Twin

Conclusion



Satisfactory DevOps outcome and outlook

- The products are starting to scale together with the variety in Digital Twins, in their connection to the physical world and in application use-cases.
- The product development (software architecture, processes...) is ready to scale for the coming years.



DevOps Insights for the Executable Digital Twin

Open points

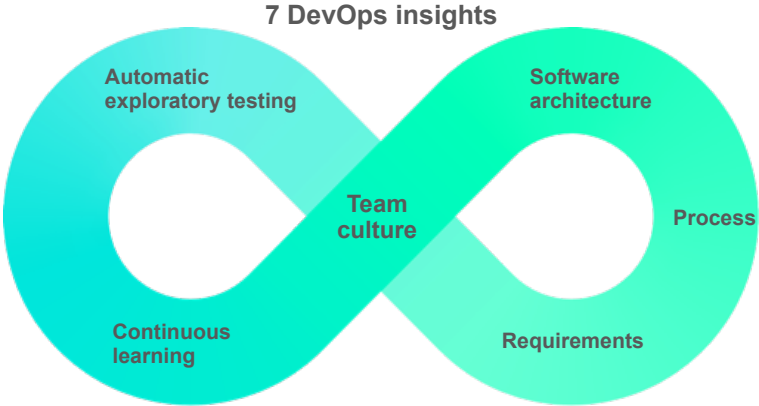
The biggest and more sustainable improvements in our DevOps efficiency are believed to have been achieved via DevOps **Insights #1 (Team culture)** and **Insights #2 (Processes)**.

Both insights relate to organizational dynamics and individual interactions:
“Yet we don’t seem to be able to find a good KPI to measure them.”

How to express and measure the value of

- a value-driven mindset?
- clearly-defined tasks?
- a responsible behavior?

Are these even measurable?



DevOps Insights #1: Team culture

	Knowledge
	Team goals
	Value mindset
	Expertise
	Responsibility
	Openness

DevOps Insights #2: Processes

	Automate
	Review
	Communicate
	Clear & lean
	Assign

DevOps Insights for the Executable Digital Twin

Roland Pastorino

Product Manager Model-based System Testing

Bernhard Spath

Sr Software Engineer Model-based System Testing

Part of this work is conducted within the EFFECTS project (HBC.2021.0010) funded by VLAIO (Flanders Innovation & Entrepreneurship) and supported by Flanders Make, the strategic research center for the manufacturing industry..

