seerene
serenity in software production

SEE. UNDERSTAND. IMPROVE.
Operational Excellence in Software Production

Leveraging the Power of AI-Based Code Production

Guiding Large-Scale Software Development Organizations Through the Risky Terrain of Generative AI

Dr. Johannes Bohnet
Founder & CEO of Seerene

May 15th, 2024, in Berlin

Executive Summit on the Evolution of Software Production

Software Production – Yesterday, Today, Tomorrow

…if software production would be like building a house…

Yesterday

- Manual process “stone by stone”
- Using small building blocks (reuse via methods/procedures)
- Programming languages like C/CPP

→Specialized knowledge and expertise about programming languages and about algorithms & data structures is required

Software Production – Yesterday, Today, Tomorrow

…if software production would be like building a house…

Today

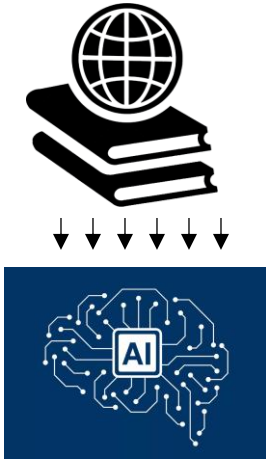
- Still manual process
- Using large building blocks
- Reuse from (open source) frameworks and libraries
- Rich fundus of out-of-the-box functionality

  Examples: TCP network connection, image processing algorithms, …

→ Specialized knowledge and expertise about the frameworks/libraries is required

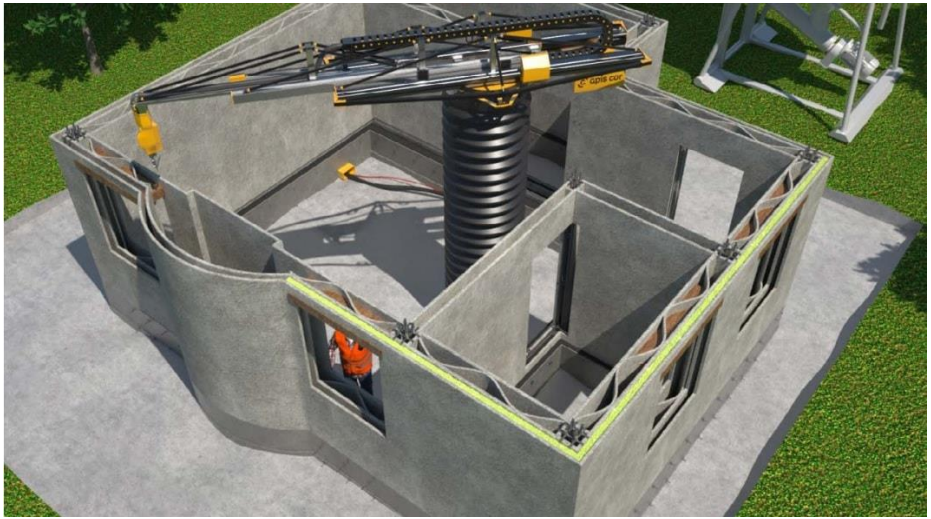
Software Production – Yesterday, Today, Tomorrow

…if software production would be like building a house…



Tomorrow

(has started already today)



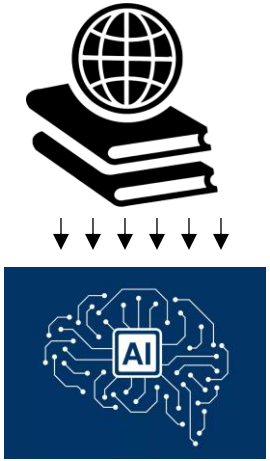
- **Specify the house**
- **Print the house**

- Reuse from building blueprints and best-practices that the AI (LLM) has learnt from world knowledge

→ Knowledge
<u>how to talk with the GenAI tool</u>
is required

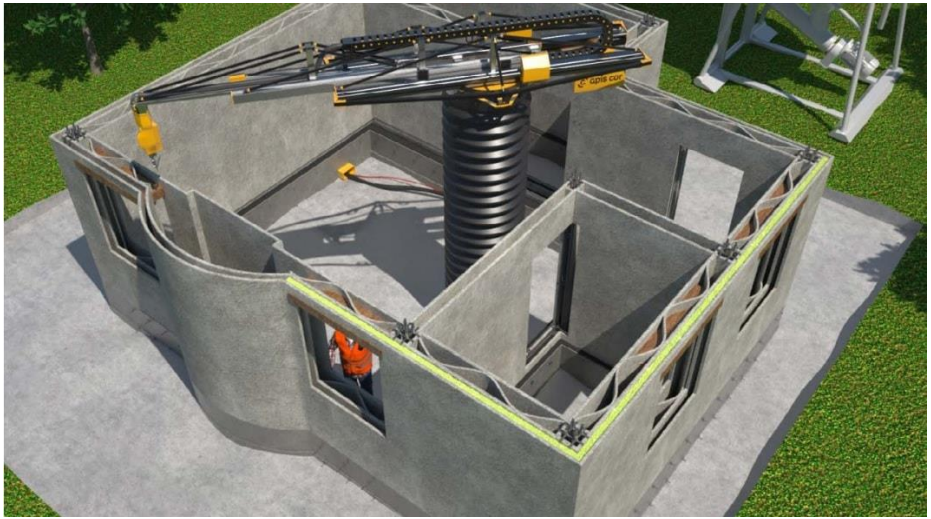
Software Production – Yesterday, Today, Tomorrow

…if software production would be like building a house…



Tomorrow

(has started already today)



- **Specify the house**
- **Print the house**

At first sight:
No need for humans as code producers anymore

Limitation of the Metaphor of “Building a House”

The metaphor holds only for software that addresses well-specifiable problems

Building a house addresses a commodity (mass) problem that has been solved very often in the past so that the user requirements are clear:

- Clear requirements for a garage
- Clear requirements for a family house
- Clear requirements for an office building
- …

Examples of software that addresses a commodity (mass) problem:

- Static website for a company
- Webshop for a small business
- HR software for a SME company
- …

AI will make the costly work of 3rd-party integrators & agencies obsolete

Most Software cannot be Specified Upfront

→ Nearly all software dev orgs worldwide & cross-industry went from waterfall to agile

Most software does not address a commodity (often-solved) problem. Instead:

Software addresses a very specific, unique business process of a company

Software is the innovative part of a physical product (car, IoT device, …)

Software is the innovative product itself

Producing such software means a journey with the users:

Finding out what the users want while producing the software.

Software is Never Finished

Inherent pressure to innovate software

World turns,
business processes &
opportunities change.

Users expect continuous innovation. Otherwise, they switch over to competitors.

Software must continuously be innovated, adapted, extended.

"House Building" Metaphor

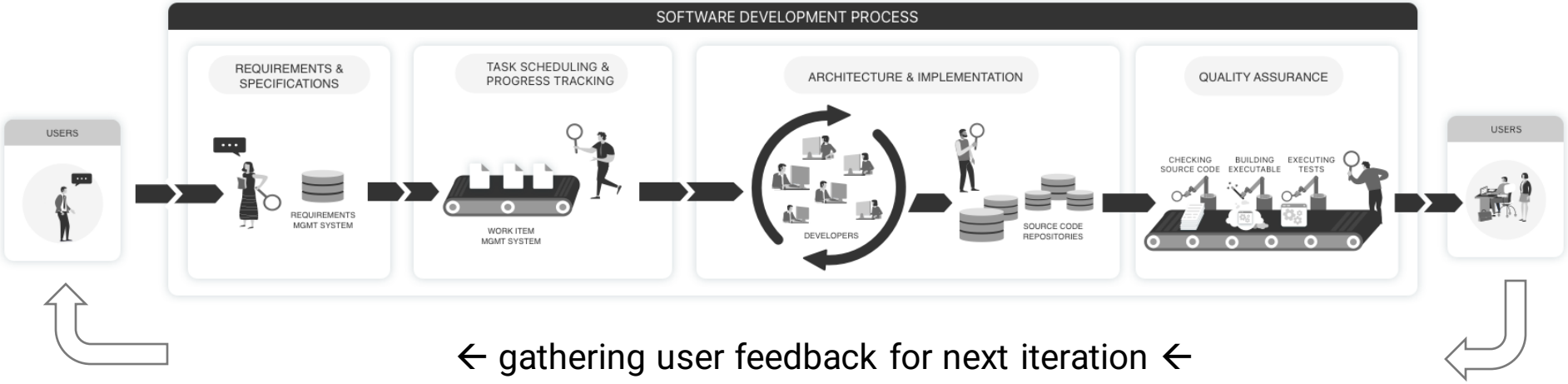
Never-ending series of small change requests such as:

- Family switch bedroom and bathroom.
  → Impact on water pipes and original architecture principles.

Over time, the original purpose/architecture/function of the house evolves significantly.

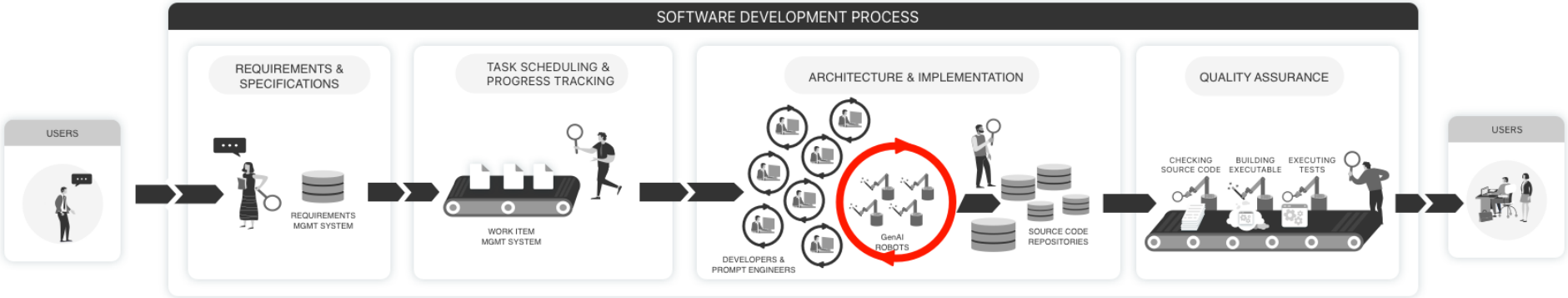
Software Development as Iterative Process

Development cycles bring innovation to the users incrementally



Software Development with GenAI-Assisted Code Production

Faster code production – an opportunity but also a risk



"House Building" Metaphor

Conventional Coding
→ Screwdriver

GenAI-assisted Coding
→ Screwrobot

Todays Studies about Productivity Boost with GenAI Code Production

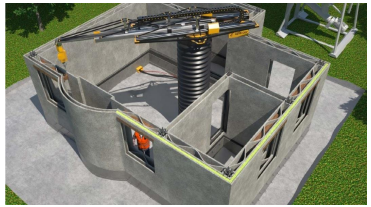
Attention – there is a bias

Typical study results:

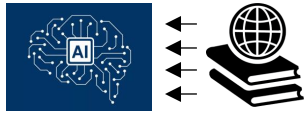
- Less expert knowledge needed to complete the task
- Faster task completion (up to 10x)

Bias:

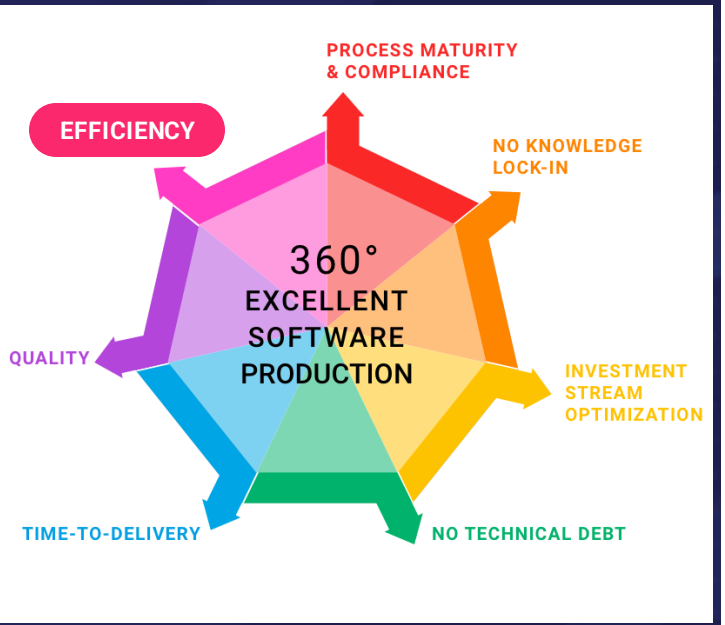
- Hackathon-like tasks:
- Building something from scratch
- Building a solution for a well-specified problem



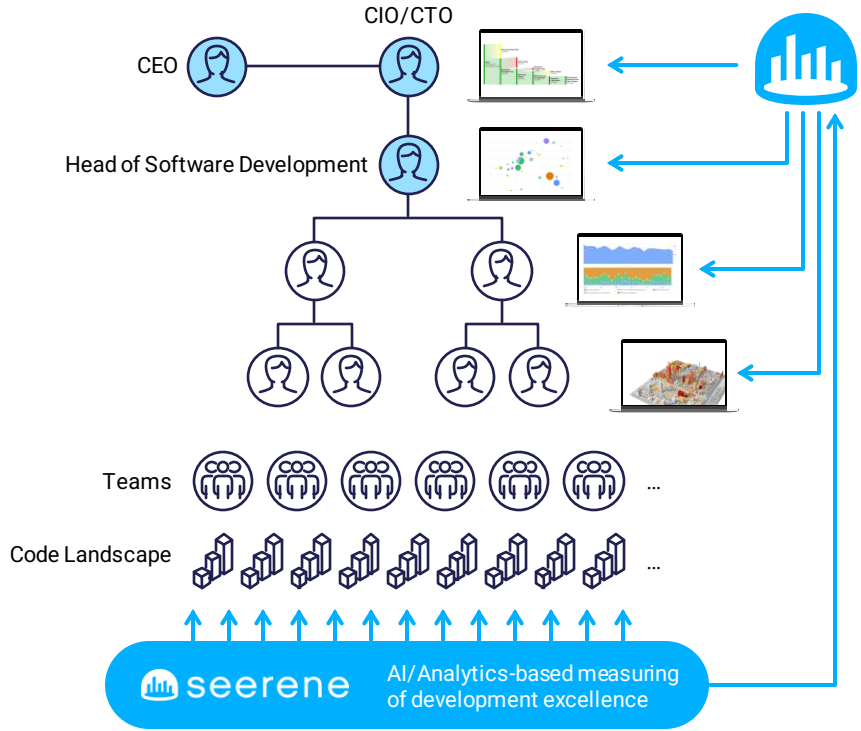
- Specify the house
- Print the house



See.
Understand.
Improve.

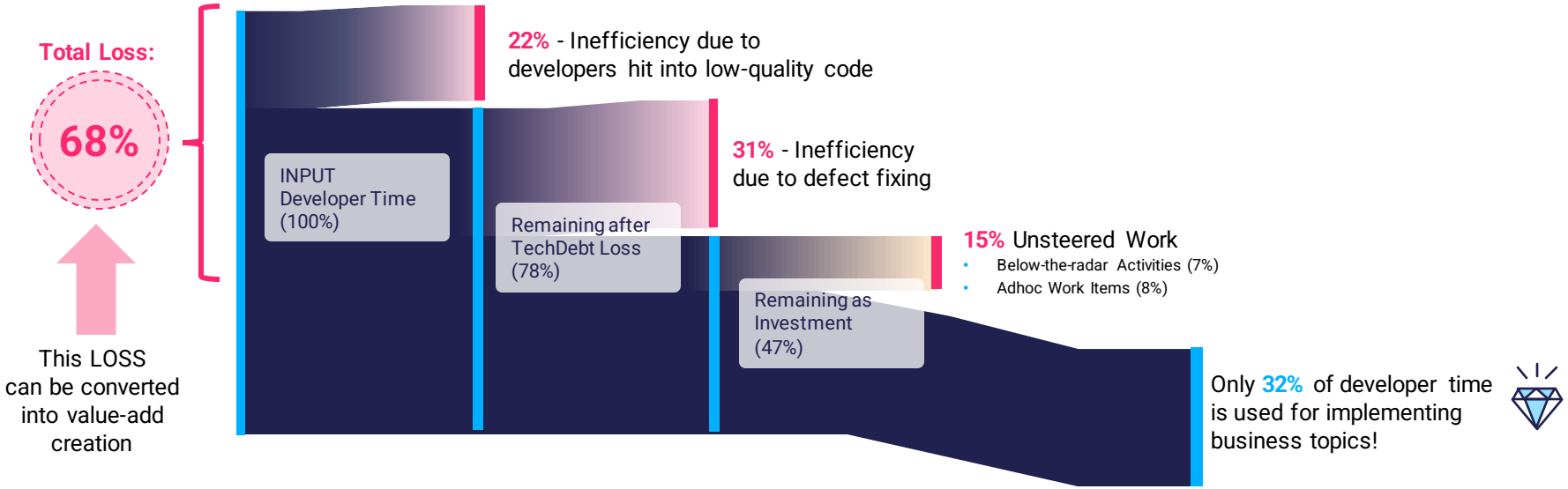


Towards Operational Excellence in Software Production



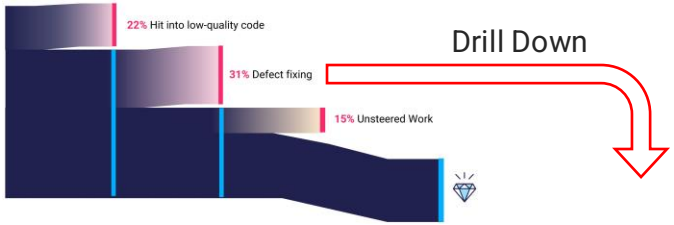
Efficiency in Software Production

How much developer time is left for creating value-add?



Efficiency in Software Production

Actionable Insights: Seerene identifies the hotspots in the code architecture



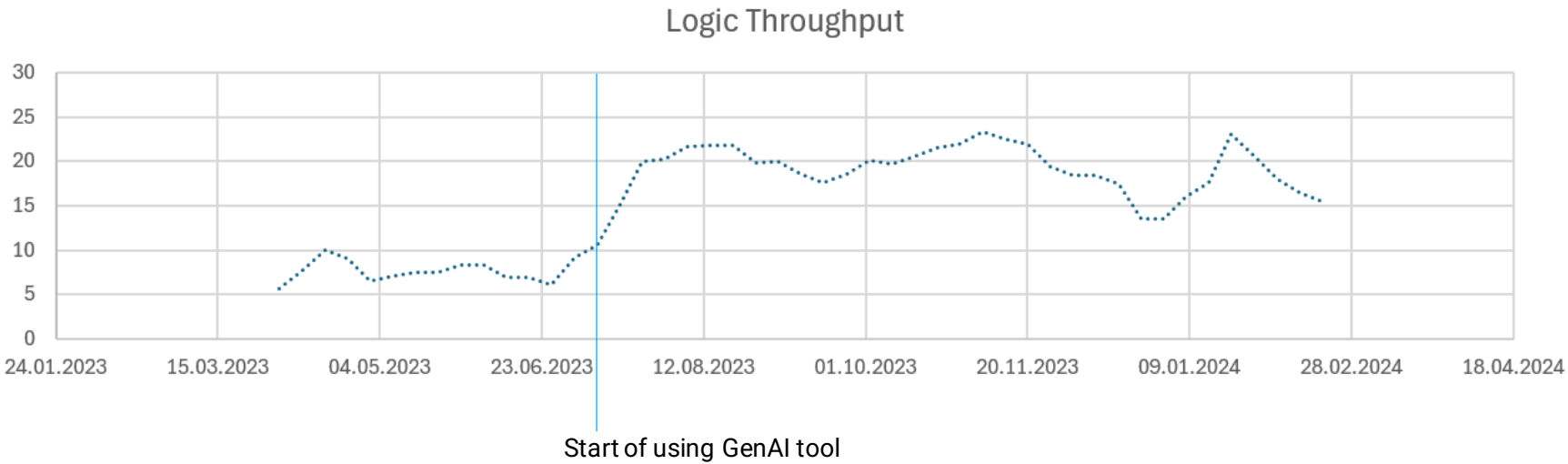
Drill Down

Code unit that consumes high amount of developer time for defect fixing

Source code of the software visualized as city

We are Building up a Benchmark for Software Production Excellence

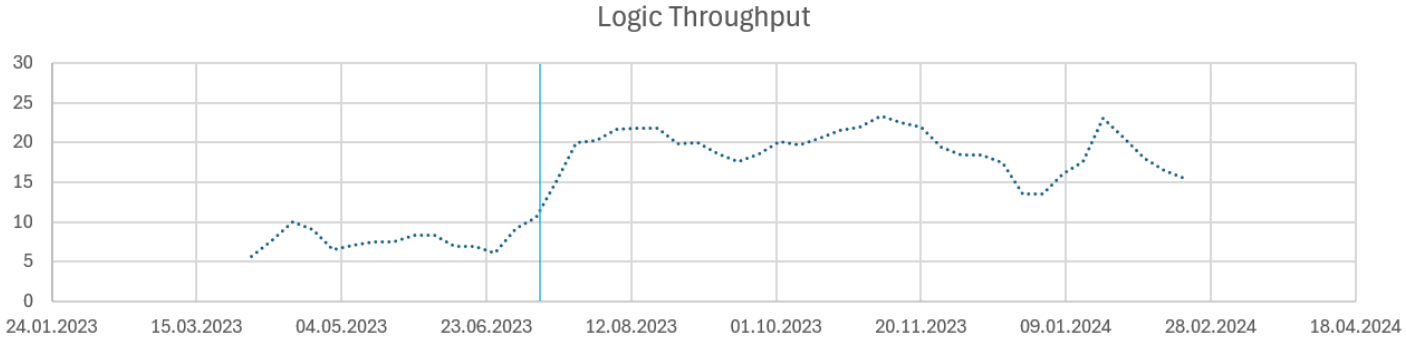
Preliminary observations of impact of GenAI-assisted code production

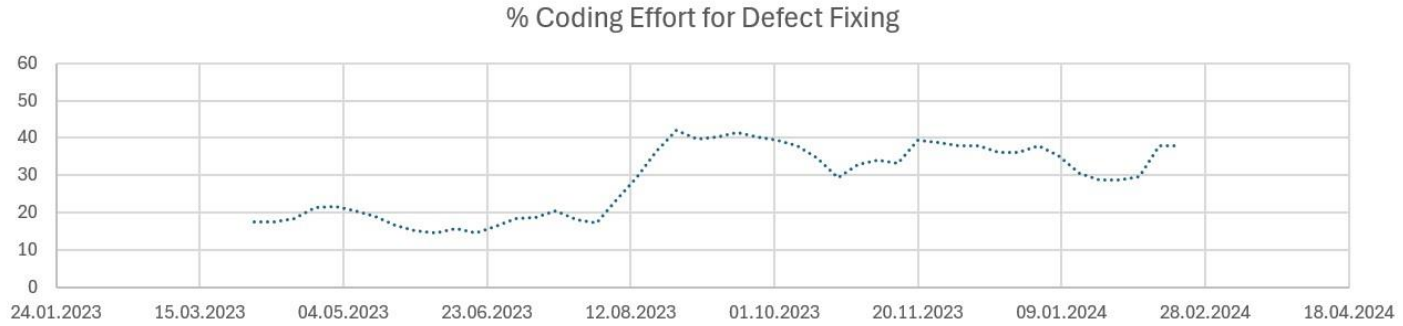


Preliminary observations:
→ approx. 2-3 times more code (logic statements) produced per developer per week

Faster Code Production Requires Better Test Automation

Quality assurance must be prepared and must keep up with the faster production speed

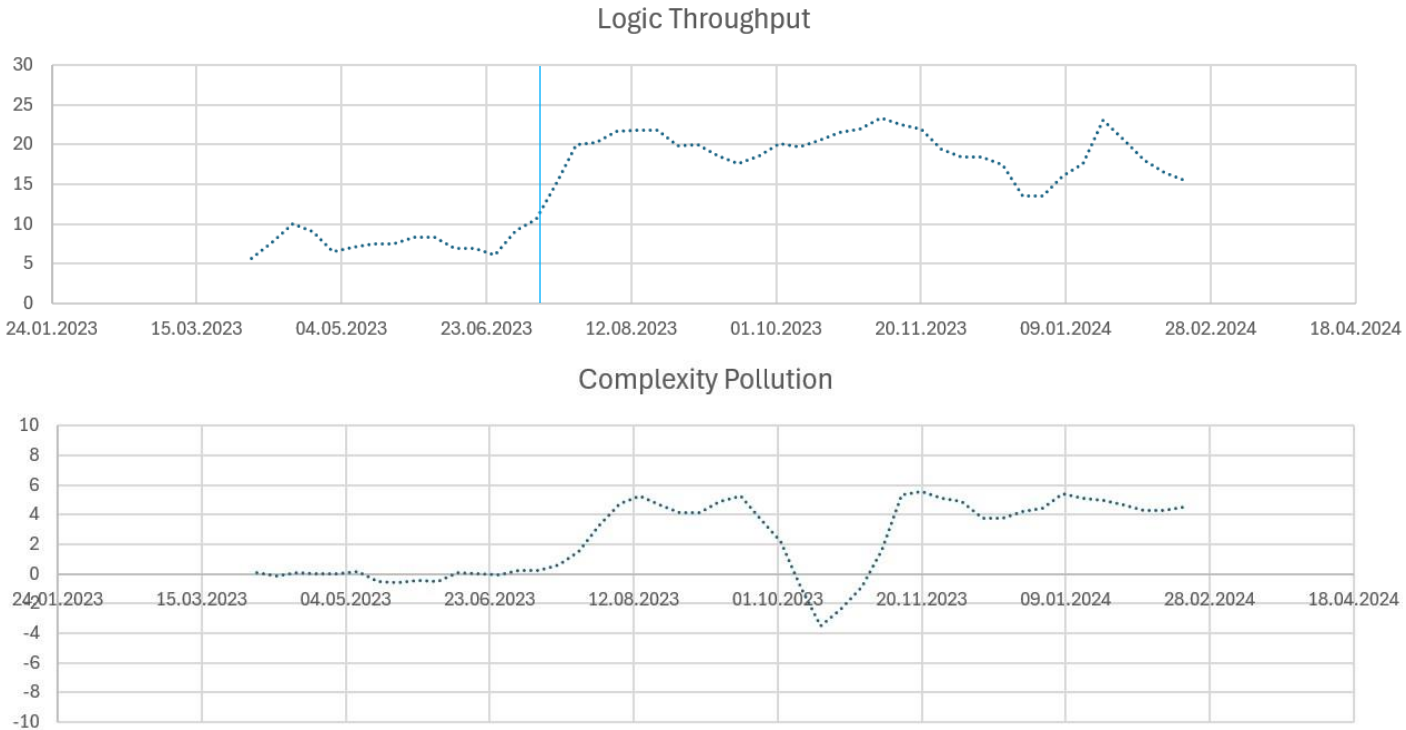




Preliminary observations:
→ Defect fixing efforts go up by approx. factor 1.5

Maintainability Risk: GenAI-Tools Tend to Create Nested Code

Nested code (if, if, if, else, if, if, …) is difficult to understand and also to secure by tests



Preliminary observations:

→ Drastic increase of “code lines in deep nesting level” per developer per week

Summary

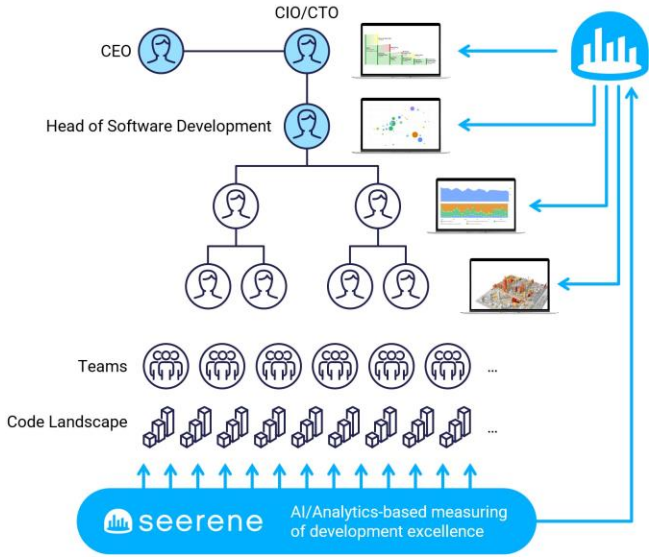
GenAI-based Code Production: Opportunity for boosting productivity but it also bears risk

- GenAI code production tools can drastically boost the speed of software development
- However, faster code production will backfire
  - …if quality assurance (test automation, code review processes, …) is not equally scaled up
  - …if developers don't have the necessarily high skillset to use these tools
    They must be able to assess the effects of the GenAI-based code proposals on code quality and architecture.

Recommendation

- Providing your developers with GenAI tools
- BUT also introduce measurement and governance across your organization
  - Measure development excellence per business unit, per department, per team
  - Proactively identify org areas that need upskilling

And: Contact me if you want to participate in our benchmark initiative



Thank You!

Dr. Johannes Bohnet

johannes.bohnet@seerene.com

www.seerene.com