Software Improvement Group

Software Improvement Group

AI – the Good, the Bad and preventing the Ugly

Luc Brandts

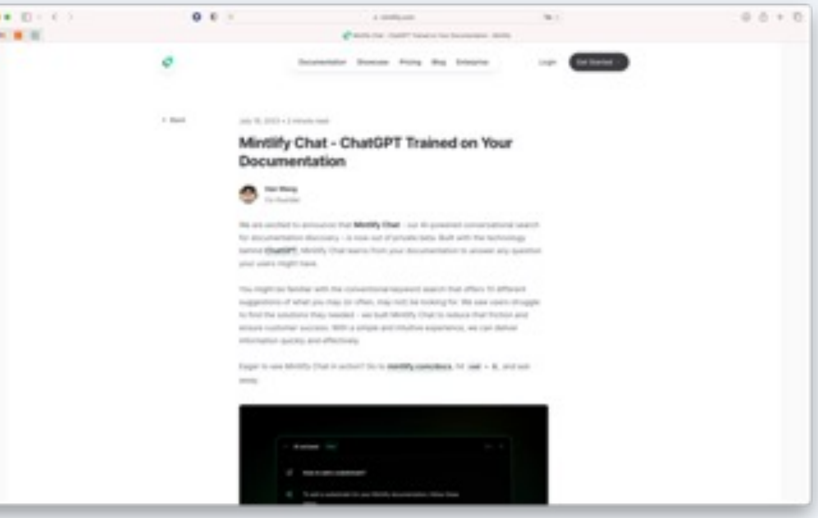
Contents

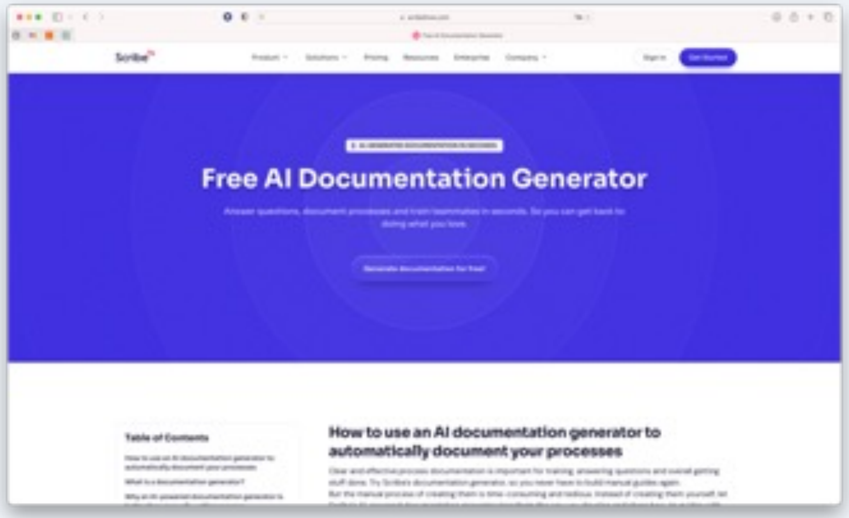
The impact of Generative AI on today's world

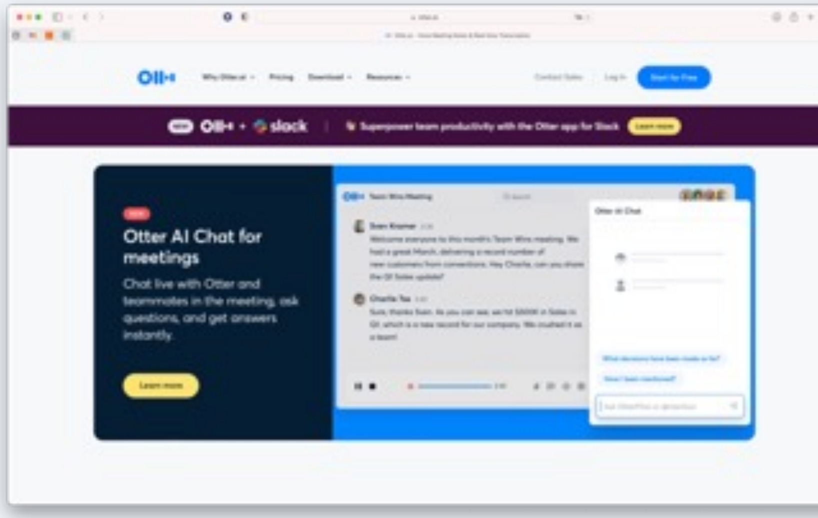
The impressive rise of ChatGPT and similar solutions,
the need to embrace GenAI

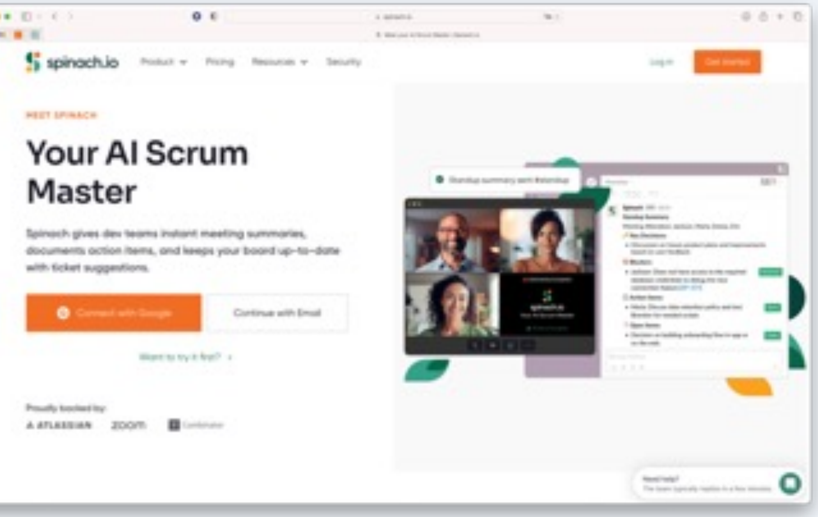
And, the need to manage the change.
Lessons from our benchmark, lessons from the past

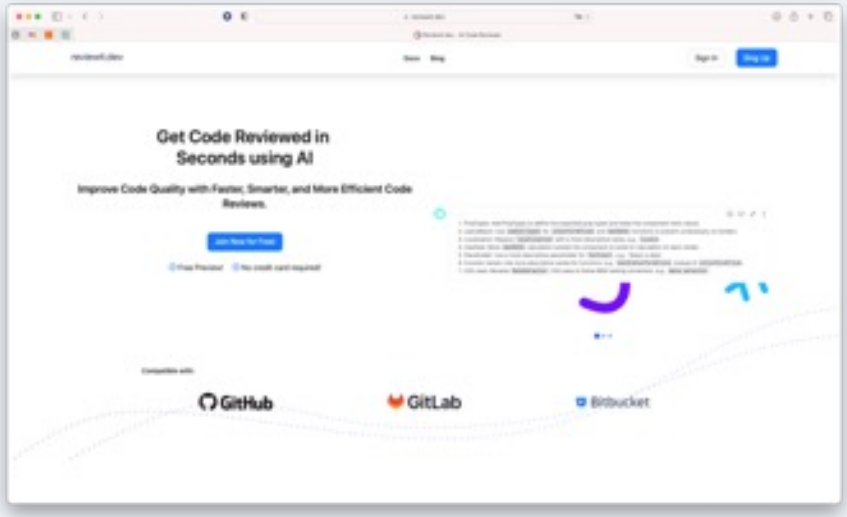
AI is increasingly making the life of software developers easy

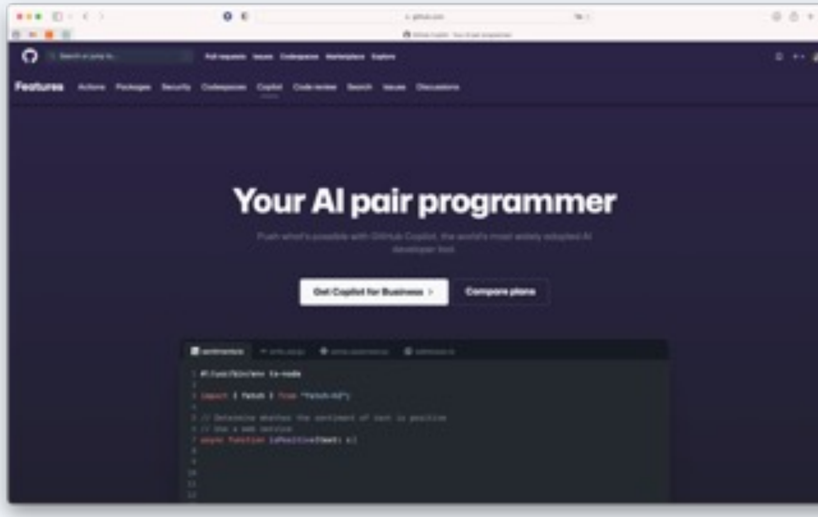



Scribe
Free AI Documentation Generator
How to use an AI documentation generator to automatically document your processes
Table of Contents


Otter
Otter + slack
Otter AI Chat for meetings
Chat live with Otter and teammates in the meeting, ask questions, and get answers instantly.
Learn more


spinach.io
MEET SPINACH
Your AI Scrum Master
Spinach gives dev teams instant meeting summaries, documents action items, and keeps your board up-to-date with ticket suggestions.
Continue with Google
Continue with Email
ATLASSIAN
zoom


Get Code Reviewed in Seconds using AI
Improve Code Quality with Faster, Smarter, and More Efficient Code Reviews.
GitHub
GitLab
Bitbucket



Warning

This presentation will **NOT** be about the super powers of generative AI

This is something we should all embrace

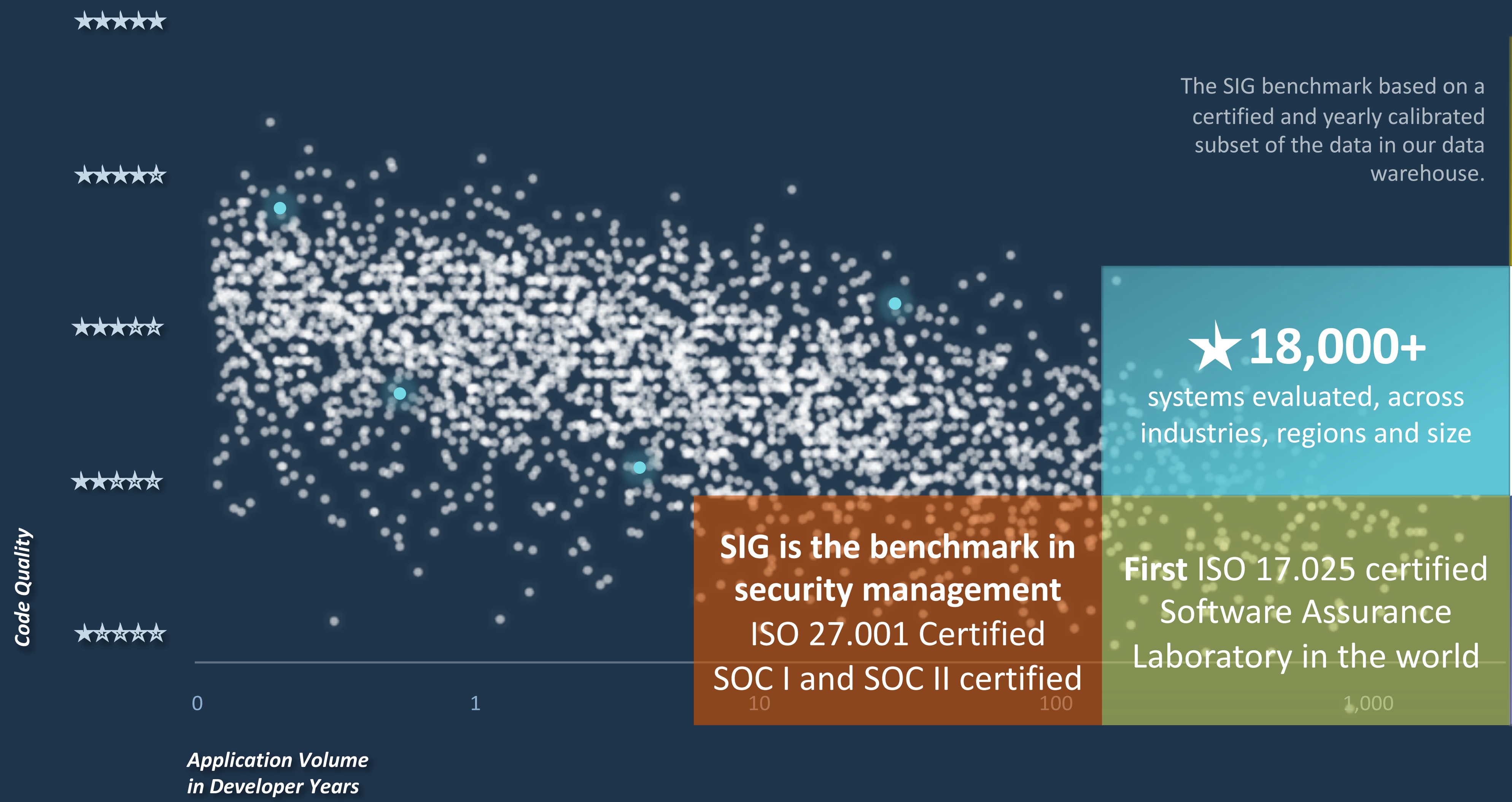
This presentation is about how we should manage the change

Software Improvement Group

SIG Background

Luc Brandts

Benchmarked approach to manage the change



The SIG benchmark based on a certified and yearly calibrated subset of the data in our data warehouse.

350M+
Lines of Code per week analyzed and added to the benchmark database

★18,000+
systems evaluated, across industries, regions and size

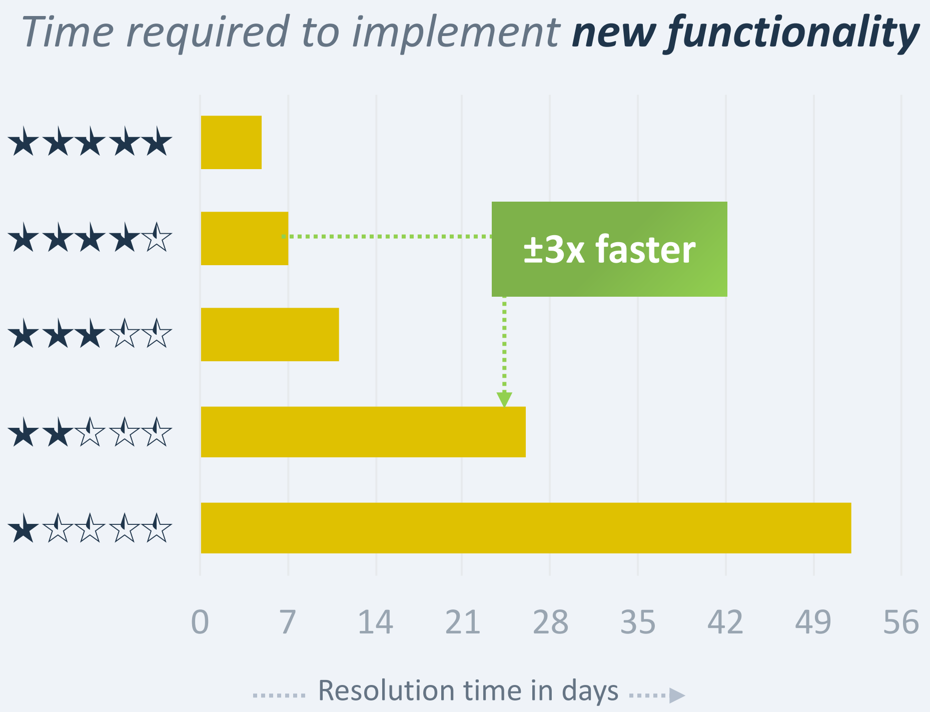
200+ billion
Lines of code in data warehouse, representing the largest benchmark of its kind

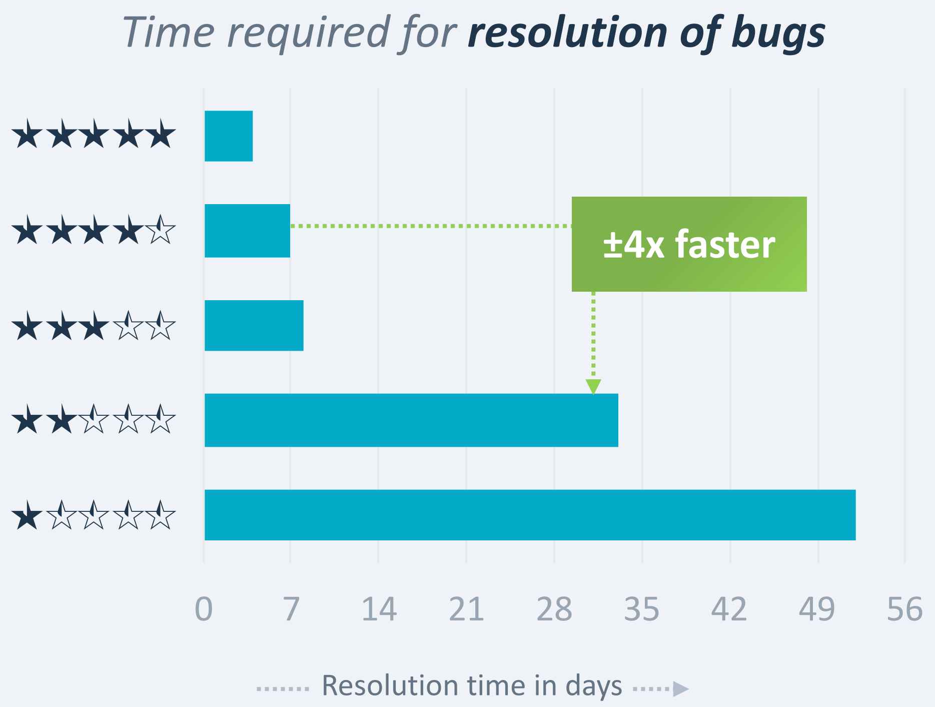
SIG is the benchmark in security management
ISO 27.001 Certified
SOC I and SOC II certified

First ISO 17.025 certified Software Assurance Laboratory in the world

300+
Technologies supported, leading the market by a (very) large margin

Build quality is a strong predictor for costs and risks





A ★★★★☆ system requires **3.5 times less effort*** to maintain than a ★★☆☆☆ star system

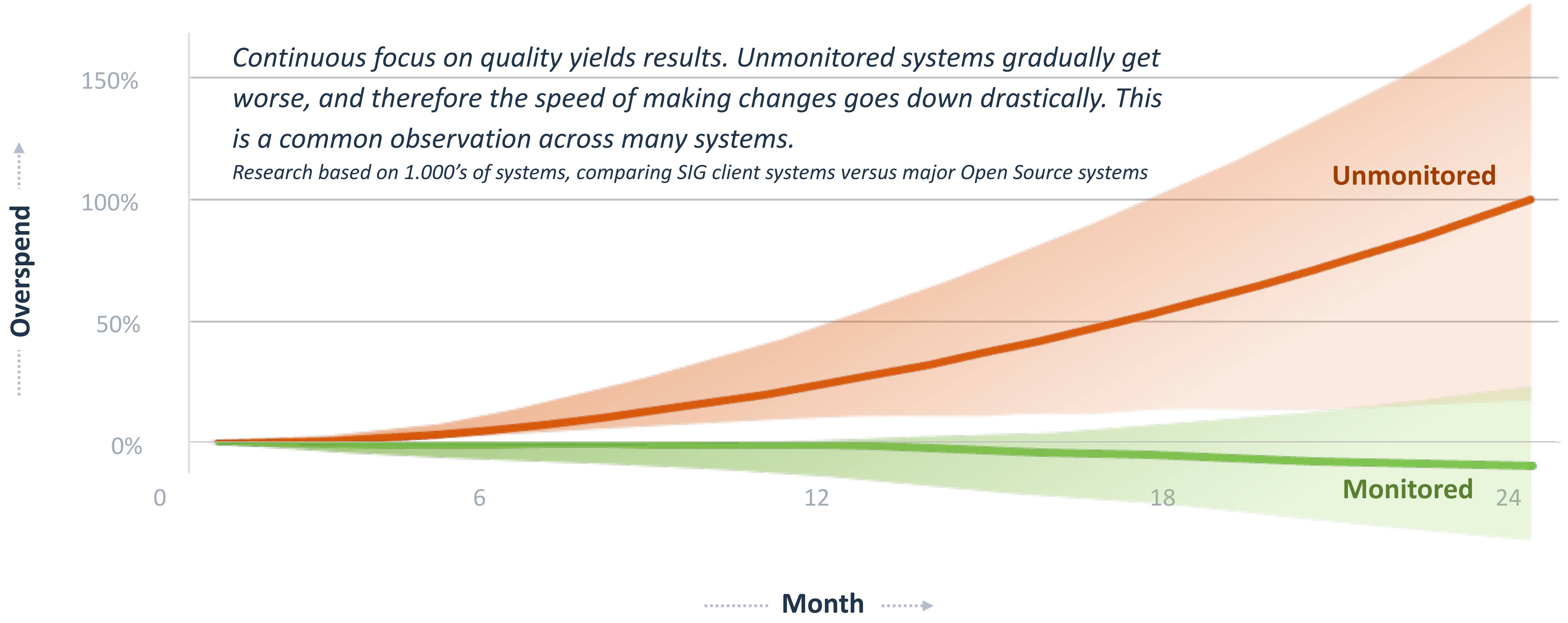
* **Source:** "Faster issue resolution with higher technical quality of software", Software Quality Journal, 2011

Consistent monitoring of each system leads to lower cost of ownership

Overspend on cost of ownership on Maintenance

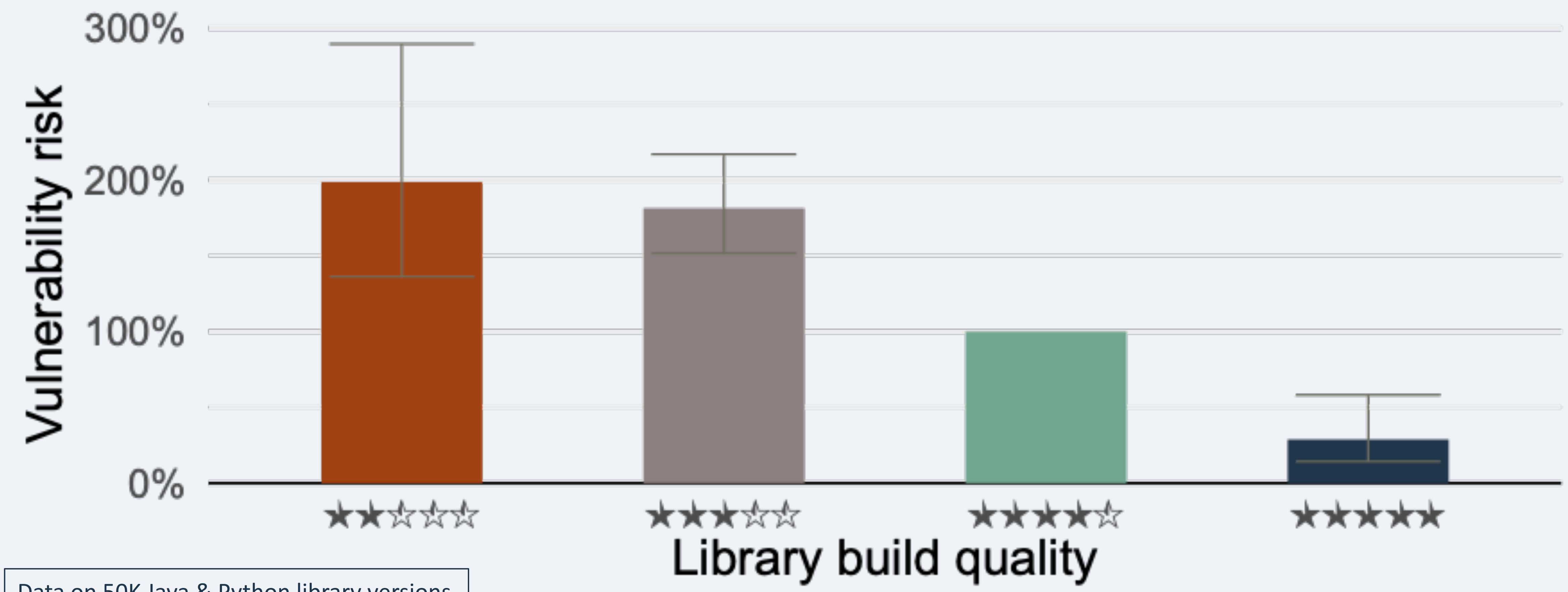
Continuous focus on quality yields results. Unmonitored systems gradually get worse, and therefore the speed of making changes goes down drastically. This is a common observation across many systems.

Research based on 1.000's of systems, comparing SIG client systems versus major Open Source systems



//

Security risk in bad quality software up to 10 times higher



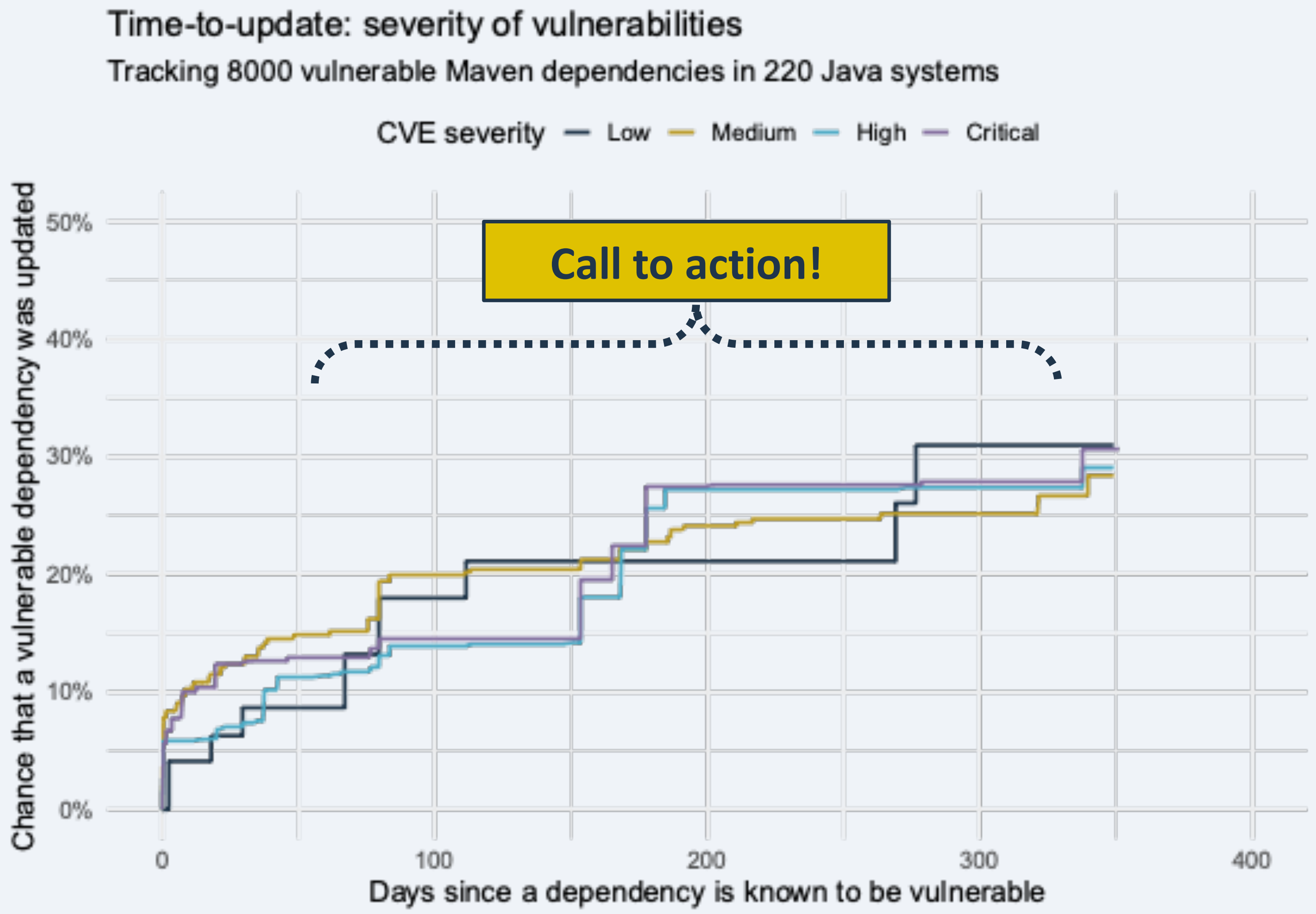
Data on 50K Java & Python library versions

Software Improvement Group

Open Source

Luc Brandts

The severity of vulnerabilities is not a major factor whether updates are done

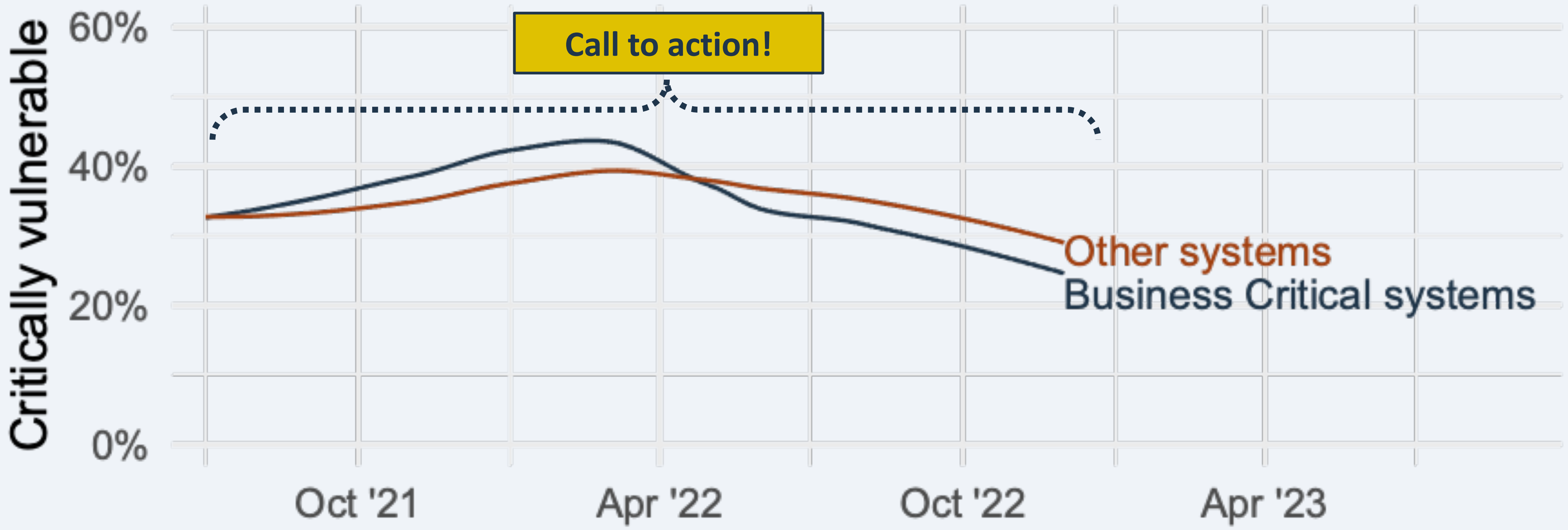


Key findings:

- Users of known vulnerable open-source libraries are not updating quickly, even if vulnerabilties are critical.
- **70% are still using known vulnerable Java libraries after a year has passed.**
- In many cases, security updates are available that can be implemented by development teams.

//

Go to zero vulnerabilities, business critical systems first



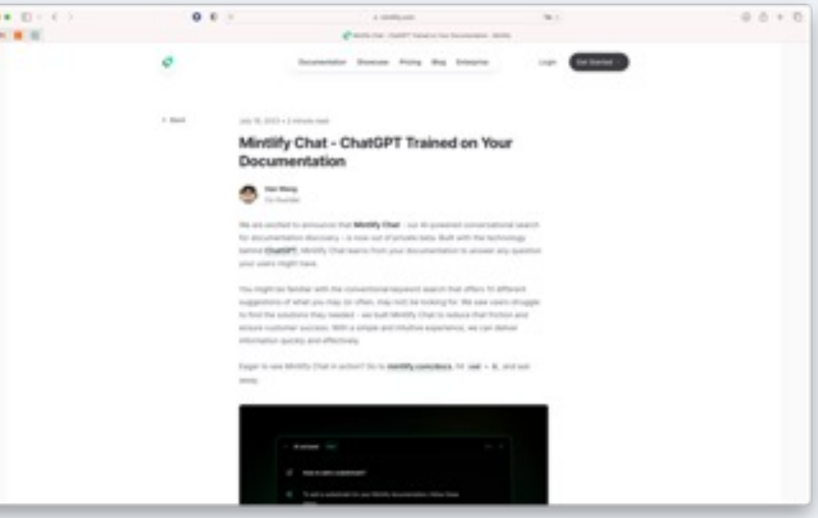
Software Improvement Group

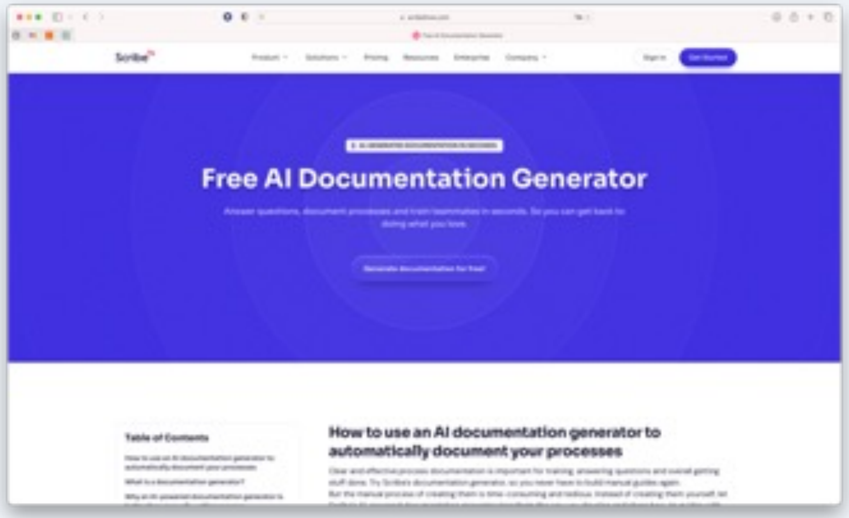
AI and data-analysis systems
So, how about AI?

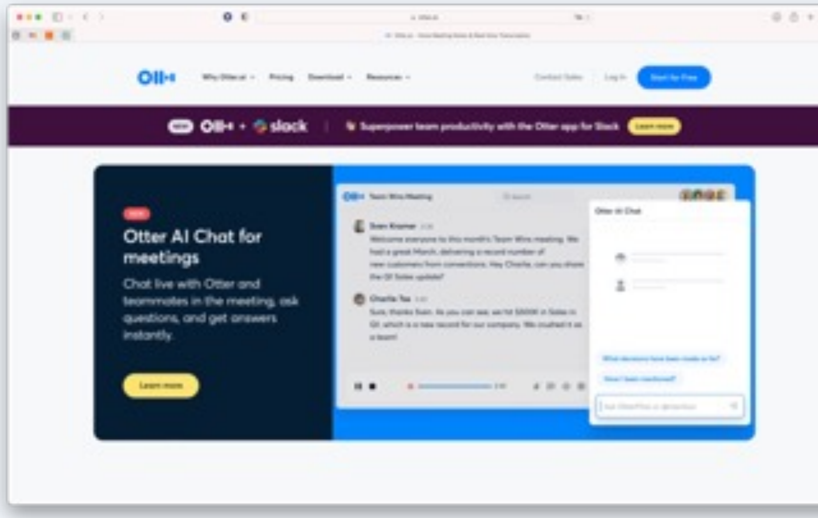
Luc Brandts

// DESIGN AND BUILD USING AI

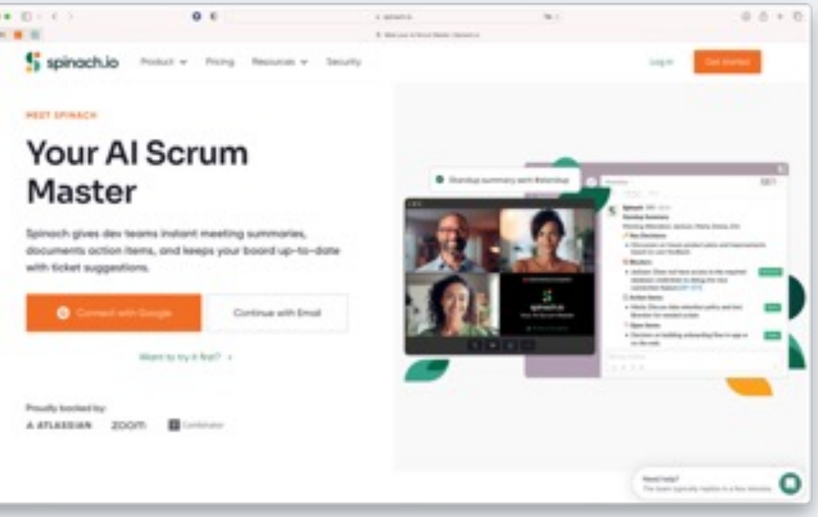
AI is increasingly making the life of software developers easy



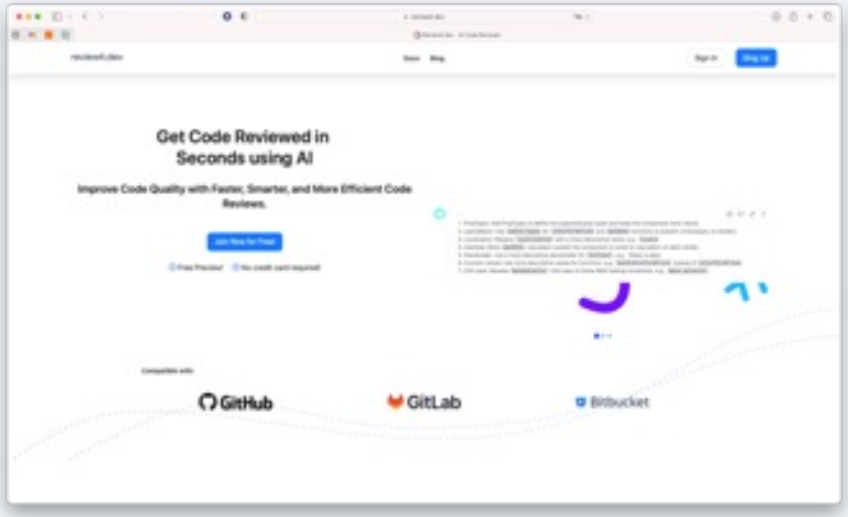

Scribe
Free AI Documentation Generator
Table of Contents
How to use an AI documentation generator to automatically document your processes

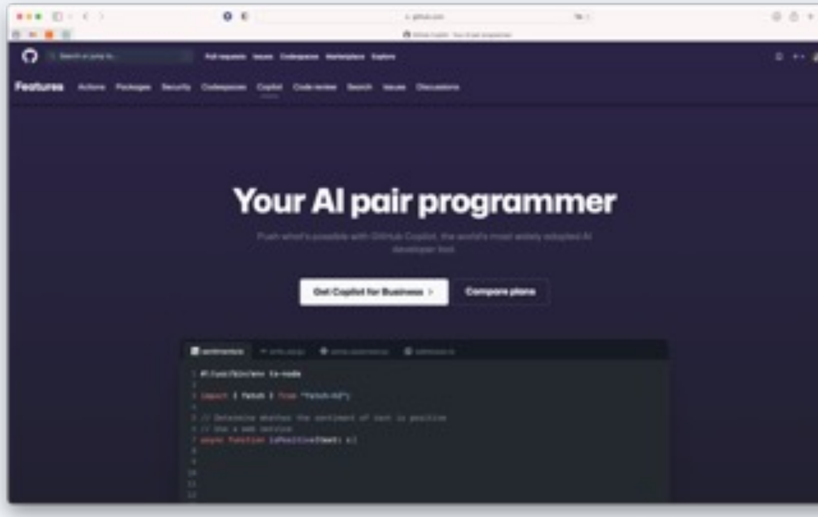


Chat live with Otter and teammates in the meeting, ask questions, and get answers instantly.

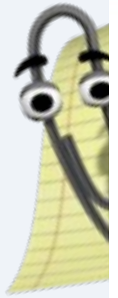


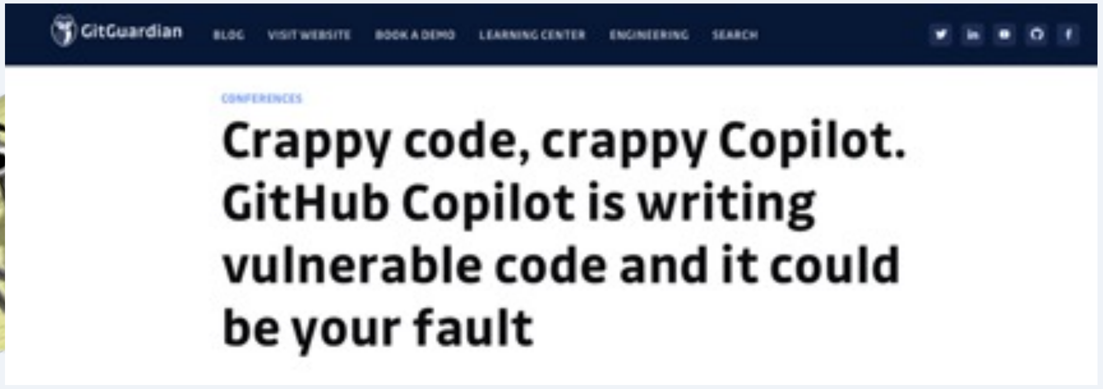
Spinach gives dev teams instant meeting summaries, documents action items, and keeps your board up-to-date with ticket suggestions.


Get Code Reviewed in Seconds using AI
Improve Code Quality with Faster, Smarter, and More Efficient Code Reviews.
GitHub
GitLab
Bitbucket



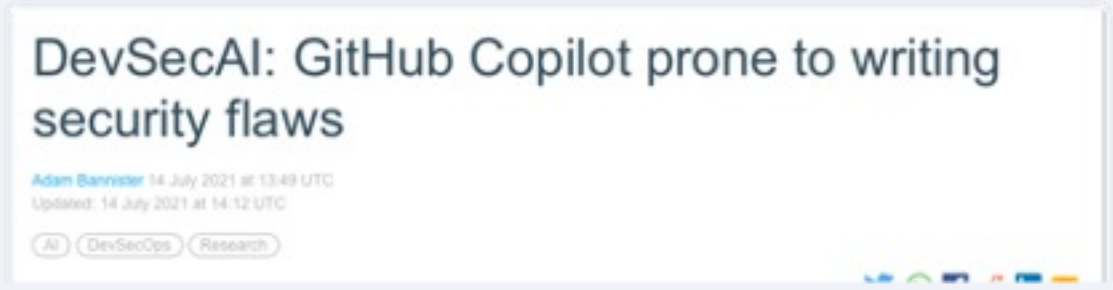
“AI pair programmer should be supervised like a toddler, says researcher”





CONFERENCES

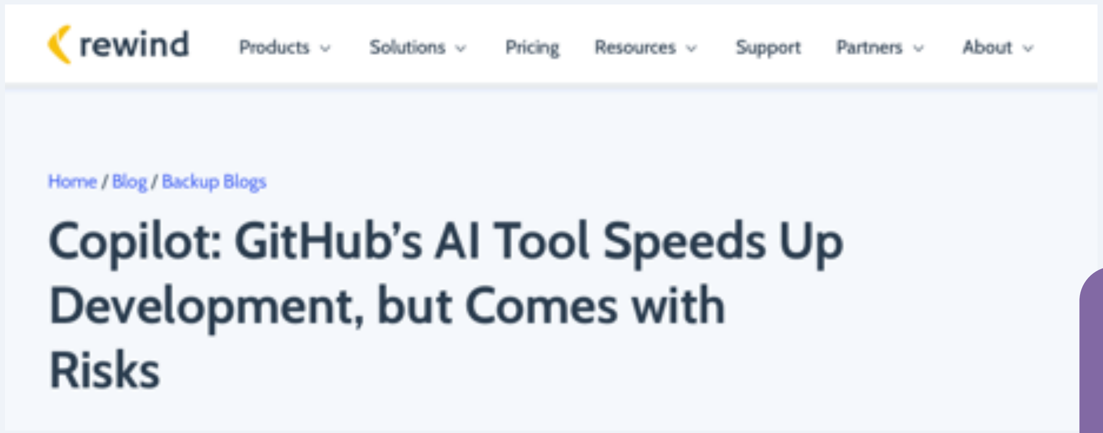
Crappy code, crappy Copilot. GitHub Copilot is writing vulnerable code and it could be your fault



DevSecAI: GitHub Copilot prone to writing security flaws

Adam Bannister 14 July 2021 at 13:49 UTC
Updated: 14 July 2021 at 14:12 UTC

AI DevSecOps Research

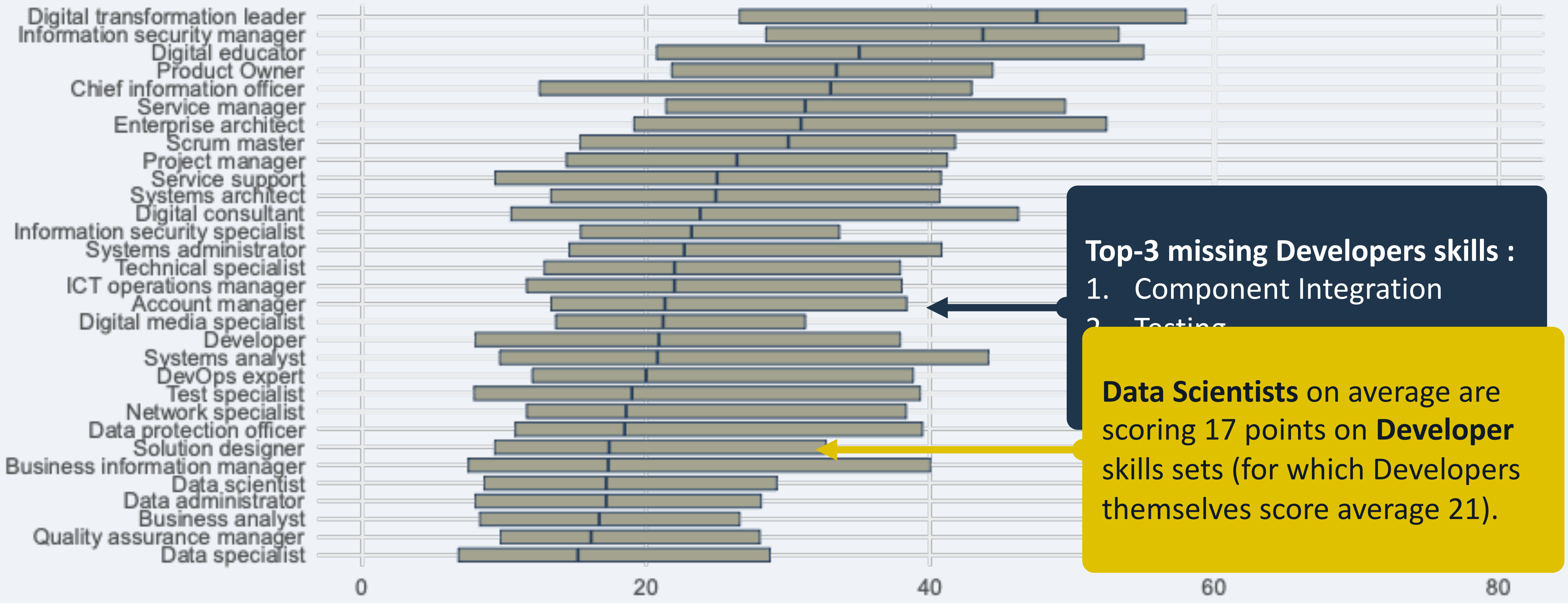


Home / Blog / Backup Blogs

Copilot: GitHub’s AI Tool Speeds Up Development, but Comes with Risks

Oh, and yes… these images were generated by AI

Analyse the skill gaps of all job roles



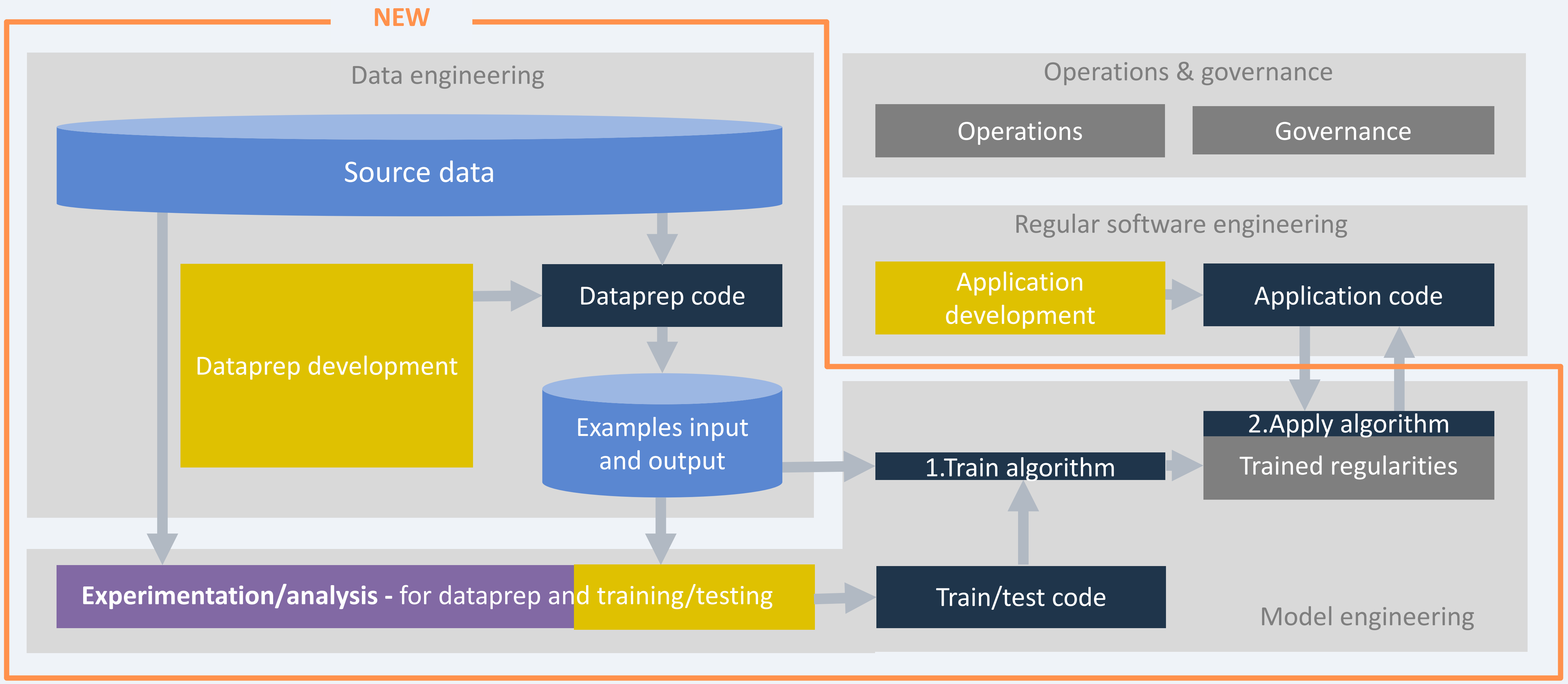
Top-3 missing Developers skills :

1. Component Integration
2. Testing

Data Scientists on average are scoring 17 points on **Developer** skills sets (for which Developers themselves score average 21).

EXIN Astride results for 5,500 participants.

How do we engineer machine learning?



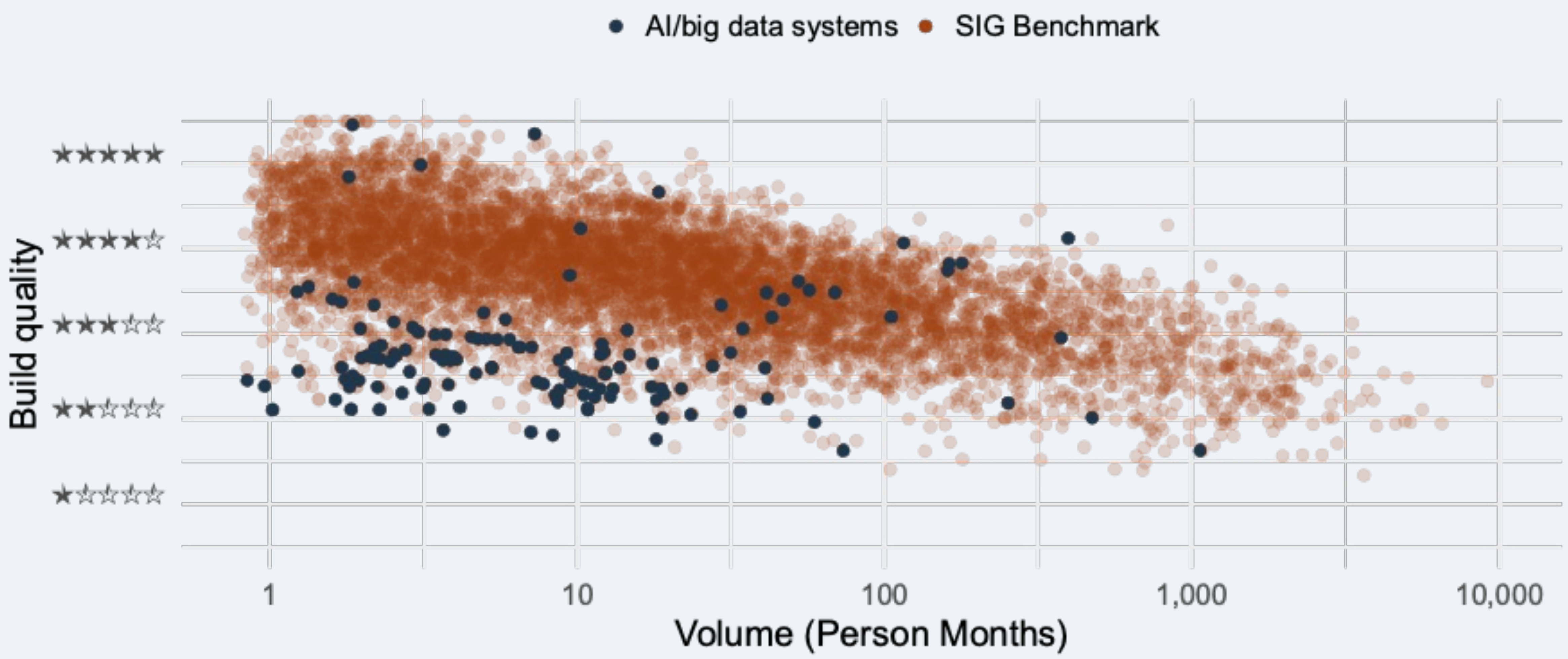
🟦 =Data
⬛ =Code
🟨 =Writing code
🟪 =Exploring & experimenting

Educate and coach data scientists in software engineering

```
GREATEST(IIF(ISNULL(i_RS_VLD_FM_DT),TO_DATE(v_
LOGC_RSVD_VAL_UNKNOWN, 'YYYY-MM-DD HH24:MI:'),i_
RS_VLD_FM_DT),IIF(ISNULL(i_RS_VLD_FM_DT_fauit),
TO_DATE(v_LOGC_RSVD_VAL_UNKNOWN, 'YYYY-MM-DD
HH24:MI:SS'), i_RS_VLD_FM_DT_fauit),IIF(ISNULL(i_
RS_VLD_FM_DT_xref_sol),TO_DATE(v_LOGC_RSVD_VAL_
UNKNOWN,'YYYY-MM-DD HH24:MI:SS'),i_RS_VLD_FM_DT_
xref_sol))
```

```
Greatest      ( MakeValidDate(i_RS_VLD_FM_DT),
                MakeValidDate(i_RS_VLD_FM_DT_fauit),
                MakeValidDate(i_RS_VLD_FM_DT_xref_sol))
```

Currently build AI/big data systems are the future legacy



Just an example of what goes wrong

Test-code Ratio

AI/big data	Median	Recommended
1.5%	43%	80%

Software
Improvement
Group

How to manage the change

Luc Brandts

//
Key take-aways

- Embrace new technology, embrace AI
- Understand AI is like any new technology: it has strong points, but it is not without its mistakes
- Understand that roles change from creation to review
- Make sure you understand it before you use it (at least know the pitfalls)
- Apply rigorous methods to ensure it works
  - Don’t destroy the innovation with red tape
  - But also don’t destroy your future with stuff that doesn’t scale or last