

Software Engineering Beyond the AI Hype

Prof. Dr. Sebastian Baltes



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

SE@UHD
Software Engineering

“Hi, my name is Sebastian and I’m a pracademic”

<https://en.wikipedia.org/wiki/Pracademic>

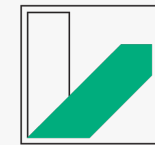
Industry



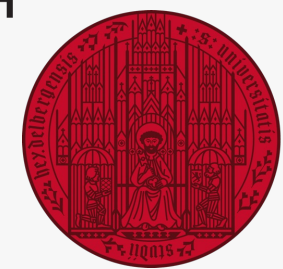
Academia



THE UNIVERSITY
of ADELAIDE



UNIVERSITÄT
BAYREUTH



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

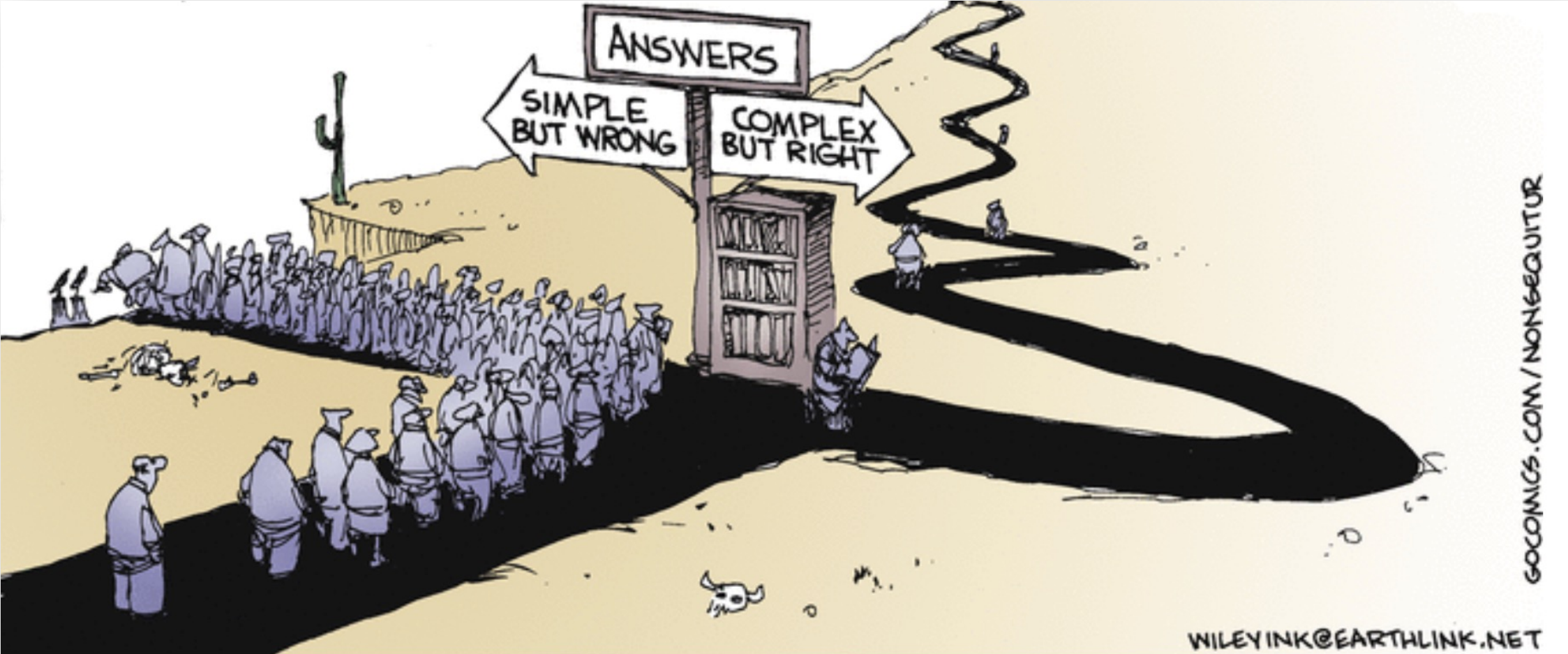
(1) Software Engineering (SE) \gg Programming

(2) AI “solves” Programming $\not\Rightarrow$ AI “solves” SE

(3) Agentic AI is a Software Systems Innovation

(4) We Need More Solid Empirical Evidence

Current Discourse on AI for SE

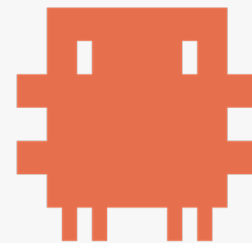




Who knows this man?

Boris Cherny

Creator of Claude Code



ANTHROPIC



“Coding Is Solved”

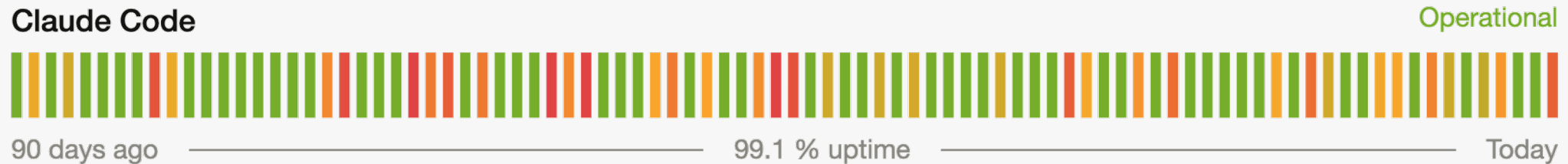
<https://youtu.be/SIGRN8jh2RI>

Well, but Software Engineering isn't

- `Bash(cd /Users/sebastian/Git/teaching/ise25-26_campus-coffee
mkdir -p build-logic/src/main/kotlin/de/seuhd...)`
 - └ `Error: claude-opus-4-7[1m] is temporarily unavailable, so auto mode cannot determine the safety of Bash right now. Wait briefly and then try this action again. If it keeps failing, continue with other tasks that don't require this action and come back to it later. Note: reading files, searching code, and other read-only operations do not require the classifier and can still be used.`
 - Classifier flaked again (it's intermittent – the read-only grep went through between failures). Retrying the same operation:
 - `Bash(cd /Users/sebastian/Git/teaching/ise25-26_campus-coffee
mkdir -p build-logic/src/main/kotlin/de/seuhd...)`
 - └ `Error: claude-opus-4-7[1m] is temporarily unavailable, so auto mode cannot determine the safety of Bash right now. Wait briefly and then try this action again. If it keeps failing, continue with other tasks that don't require this action and come back to it later. Note: reading files, searching code, and other read-only operations do not require the classifier and can still be used.`
 - Classifier still down for write commands. Rather than risk leaving the repo mid-change (the move needs git, which needs Bash), I'll keep it in its current good state (squash `2bfa340`, green) and retry the operation once more – environment hiccups like this usually clear quickly:
 - `Bash(cd /Users/sebastian/Git/teaching/ise25-26_campus-coffee
mkdir -p build-logic/src/main/kotlin/de/seuhd...)`
 - └ `package lines: 6 updated; module id() refs updated: 20`
 - └ `==== build ====`
 - └ `BUILD SUCCESSFUL in 45s`
 - └ `... +7 lines (ctrl+o to expand)`
 - └ `(timeout 10m)`
 - └ `Allowed by auto mode classifier`
- * `Schlepping...` (7m 30s · ↓ 17.6k tokens · almost done thinking with max effort)

Well, but Software Engineering isn't

☀ Claude Status



<https://status.claude.com/>

$0.009 \times 90 \text{ days} = 0.81 \text{ days} = 19.44 \text{ hours} = 1,166.4 \text{ minutes}$
→ 12.96 minutes per day*

*Of course, the actual duration and frequency varies (see <https://pulsetic.com/status/claude/>).

Well, but Software Engineering isn't



Boris Cherny ✓

@bcherny

Follow



Coding is just one part of engineering. There's also debugging, operating services, scaling up infrastructure, deciding what to optimize, setting up hardware and capacity, talking to users, product planning, etc. Coding is the easy part, **everything else is not yet solved** (but is also becoming increasingly automated).

2:04 AM · Jun 9, 2026 · **90.5K** Views

<https://x.com/bcherny/status/2064136590667256229>

*Of course, the actual duration and frequency varies (see <https://pulsetic.com/status/claude/>).

Cursor-Opus agent snuffs out startup's production database

Relax, the data's been recovered. Continue with your vibe coding

 Thomas Claburn

Published Mon 27 Apr 2026 // 21:29 UTC



<https://www.theregister.com/software/2026/04/27/cursor-opus-agent-snuffs-out-startups-production-database/5224442>

An AI-powered coding tool wiped out a software company's database, then apologized for a 'catastrophic failure on my part'

By Beatrice Nolan

Tech Reporter

July 23, 2025, 7:22 AM ET

<https://fortune.com/2025/07/23/ai-coding-tool-replit-wiped-database-called-it-a-catastrophic-failure/>

AI

Uber's COO says it's getting harder to justify the money spent on AI tokenmaxxing

By Aditi Bharade [+ Follow](#)

<https://www.businessinsider.com/uber-coo-andrew-macdonald-ai-token-spending-harder-justify-2026-5>

Vulnerability Report

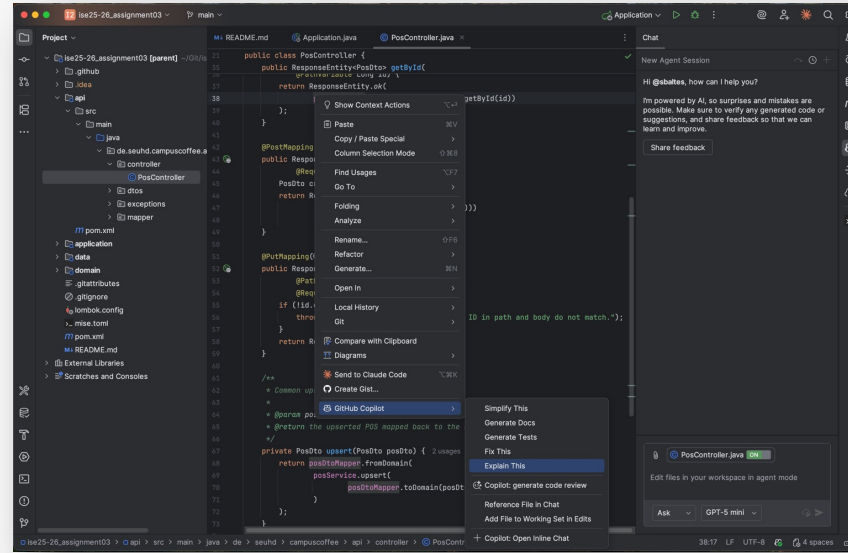
CVE-2025-48757: Lovable's Row-Level Security Breakdown Exposes Sensitive Data Across Hundreds of Projects

 Ddos  June 10, 2025 3 minutes read  0

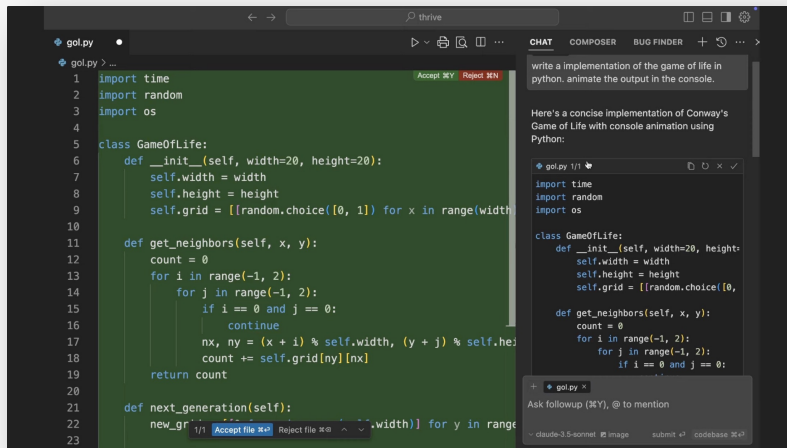
<https://securityonline.info/cve-2025-48757-lovables-row-level-security-breakdown-exposes-sensitive-data-across-hundreds-of-projects/>

Generative AI Tools for Software Development

Editor-integrated and chat-based AI assistants for software development.



 **GitHub Copilot**

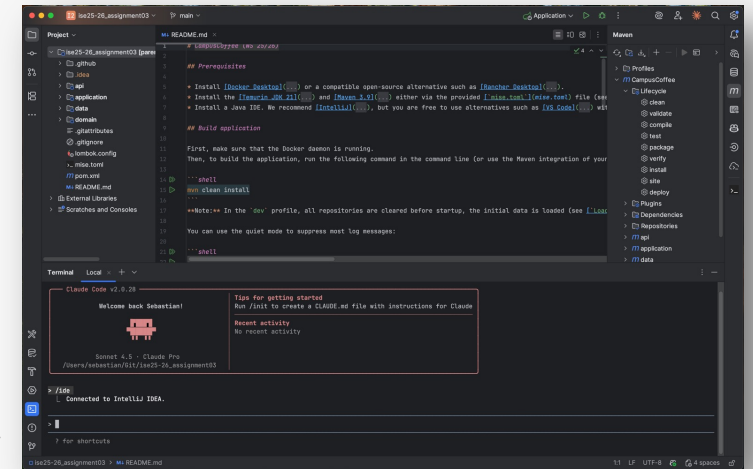


GenAI-
centric **IDEs**

 **CURSOR**

"Agentic" AI
assistants

 **Claude**



What is Software Engineering?

Let's start with what it isn't



Who knows this man?

Andrej Karpathy

Co-founder of OpenAI

OpenAI ANTHROPIC

Vibe Coding



Andrej Karpathy ✓

@karpathy



There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists. It's possible because the LLMs (e.g. Cursor Composer w Sonnet) are getting too good. Also I just talk to Composer with SuperWhisper so I barely even touch the keyboard. I ask for the dumbest things like "decrease the padding on the sidebar by half" because I'm too lazy to find it. I "Accept All" always, I don't read the diffs anymore. When I get error messages I just copy paste them in with no comment, usually that fixes it. The code grows beyond my usual comprehension, I'd have to really read through it for a while. Sometimes the LLMs can't fix a bug so I just work around it or ask for random changes until it goes away. It's not too bad for throwaway weekend projects, but still quite amusing. I'm building a project or webapp, but it's not really coding - I just see stuff, say stuff, run stuff, and copy paste stuff, and it mostly works.

12:17 AM · Feb 3, 2025 · 4.9M Views

1.3K

4.9K

29K

15K



*"There's a new kind of coding I call **vibe coding** [...] where you [...] forget that the code even exists."*

*"I just see stuff, say stuff, run stuff, and copy paste stuff, and it **mostly works**."*

<https://x.com/karpathy/status/1886192184808149383>

Vibe Coding



*"There's a new kind of coding I call **vibe coding** [...] where you [...] forget that the code even exists."*

*"I just see stuff, say stuff, run stuff, and copy paste stuff, and it **mostly works**."*

<https://x.com/karpathy/status/1886192184808149383>

Where Vibe Coding Works: Small Projects

Wunschzeichen

1 Antragsteller
Persönliche Daten, Datenschutz

Antragsteller

Name

Datenschutz

Es ist mir bewusst, dass die in den folgenden Schritten erfassten Daten automatisiert verarbeitet werden. Ich stimme der elektronischen Verarbeitung der freiwillig abgegebenen Daten ausdrücklich zu.

Ja, ich stimme zu

ABBRECHEN

WEITER

2 Suche
Kennzeichenauswahl

3
4 Abschluss
Bestätigung

The screenshot shows a GitHub repository for 'license-plate-checker'. The README is visible, containing the following text:

```
Batch-checks desired license plate ("Wunschzeichen") on a public license plate availability portal using browser automation.

The portal is a GWT (Google Web Toolkit) SPA with obfuscated JavaScript.
This tool uses Playwright to drive a headless Chromium browser.
```

The file list includes: .gitignore, CLAUDE.md, README.md, check_plates.py, requirements.txt, and test_check_plates.py.

```
District [MTK]:
Letter combinations (1-2 chars, comma-separated, e.g. AB,CD,XY): SN
Number range (e.g. 1-99 or 1,5,10-50): 1-99

Will check 99 combination(s) for district MTK (Main-Taunus-Kreis).

Navigating to https://kfzonline.ekom21.de/kfzonline.public/start.html?oe=00.00.06.436000 ...
District prefix: MTK
[1/99] MTK-SN 1      -> TAKEN x
[2/99] MTK-SN 2      -> TAKEN x
[3/99] MTK-SN 3      -> TAKEN x
[4/99] MTK-SN 4      -> TAKEN x
[5/99] MTK-SN 5      -> TAKEN x
[6/99] MTK-SN 6      -> TAKEN x
[7/99] MTK-SN 7      -> TAKEN x
[8/99] MTK-SN 8      -> TAKEN x
[9/99] MTK-SN 9      -> TAKEN x
[10/99] MTK-SN 10     -> TAKEN x
[11/99] MTK-SN 11     -> TAKEN x
[12/99] MTK-SN 12     -> TAKEN x
[13/99] MTK-SN 13     -> TAKEN x
[14/99] MTK-SN 14     -> TAKEN x
[15/99] MTK-SN 15     -> TAKEN x
[16/99] MTK-SN 16     -> TAKEN x
[17/99] MTK-SN 17     -> TAKEN x
[18/99] MTK-SN 18     -> TAKEN x
[19/99] MTK-SN 19     -> TAKEN x
[20/99] MTK-SN 20     -> TAKEN x
[21/99] MTK-SN 21     -> TAKEN x
[22/99] MTK-SN 22     -> TAKEN x
[23/99] MTK-SN 23     -> TAKEN x
[24/99] MTK-SN 24     -> TAKEN x
[25/99] MTK-SN 25     -> TAKEN x
[26/99] MTK-SN 26     -> TAKEN x
[27/99] MTK-SN 27     -> TAKEN x
[28/99] MTK-SN 28     -> TAKEN x
[29/99] MTK-SN 29     -> TAKEN x
[30/99] MTK-SN 30     -> TAKEN x
[31/99] MTK-SN 31     -> TAKEN x
```

Where Vibe Coding Works: Small Projects

The image displays three overlapping screenshots of GitHub repository pages, illustrating small projects created using Vibe coding. The top-most screenshot shows the 'caldav-utils' repository, which is public and has 0 watches, 0 forks, and 0 stars. It lists files such as .gitignore, LICENSE, README.md, caldav_utils.py, and requirements.txt. The middle screenshot shows the 'imap-dedup' repository, also public, with 0 watches, 0 forks, and 0 stars. It lists files like .gitignore, LICENSE, README.md, imap_dedup.py, offlineimaprc, sync-mailbox.sh, and test_imap_dedup.py. The bottom-most screenshot shows the 'openreview-downloader' repository, public, with 0 watches, 0 forks, and 0 stars. It features a commit history table and a README section titled 'OpenReview Review Downloader'.

File	Description	Time
.gitignore	Add initial version of script	4 months ago
LICENSE	Add MIT license	3 months ago
README.md	Add --comments flag and configurable --format output	3 months ago
download_reviews.py	Add --comments flag and configurable --format output	3 months ago
requirements.txt	Add initial version of script	4 months ago

OpenReview Review Downloader

Download reviews, metareviews, decisions, and optionally comments from [OpenReview](#) for a given paper.

About

Script to download reviews, metareviews, and decisions as Markdown and JSON from the OpenReview platform.

Releases

No releases published
[Create a new release](#)

Packages

No packages published

Where It Doesn't Work: Publicly Deployed Software

ANDY GREENBERG

SECURITY MAY 7, 2026 7:00 AM

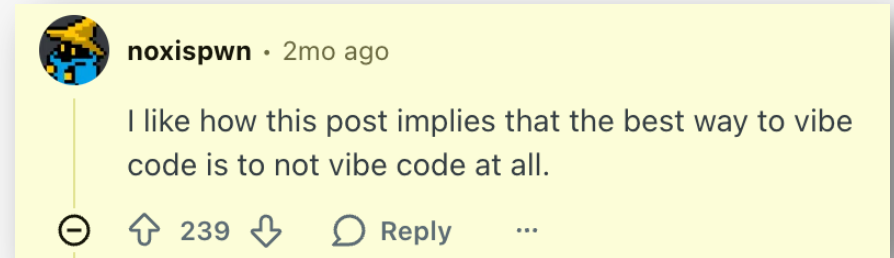
Thousands of Vibe-Coded Apps Expose Corporate and Personal Data on the Open Web

Companies like Lovable, Base44, Replit, and Netlify use AI to let anyone build a web app in seconds—and in thousands of cases, spill highly sensitive data onto the public internet.

<https://www.wired.com/story/thousands-of-vibe-coded-apps-expose-corporate-and-personal-data-on-the-open-web/>

“Vibe-Coding” at FAANG

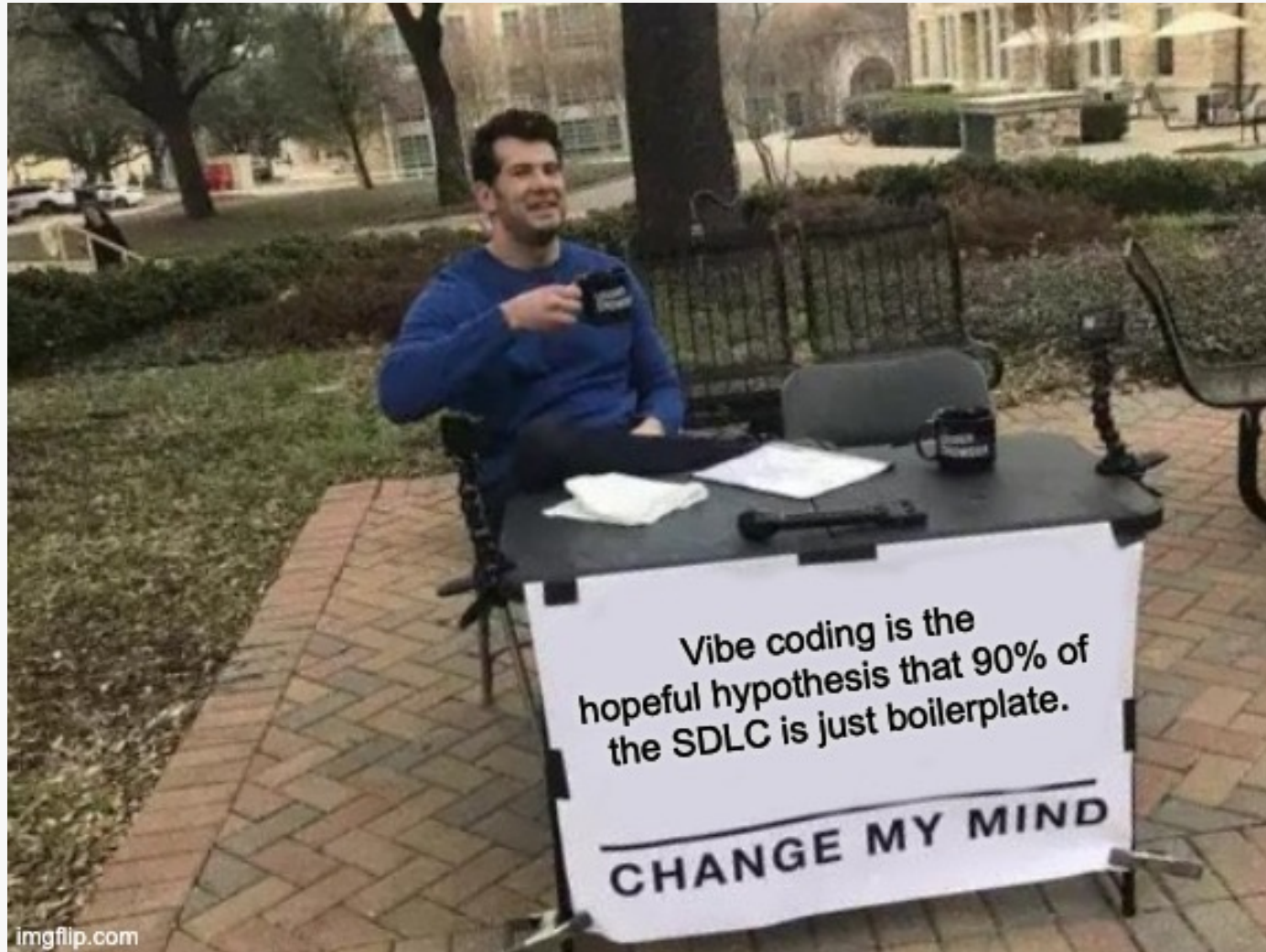
1. [...] start with a **technical design document**.
2. **Design review** before [...] development [...].
3. If you pass review, [...] **more documentation on each subsystem**.
4. **Backlog development** and **sprint planning**.
5. **Software development** [...] We use **test-driven development**, so I have the AI coding agent write the tests first for the feature I’m going to build.
6. **Code** [...] **review**. We have a two-dev approval process before code can get merged into main. AI is [...] assisting with the review.
7. **Test in staging**. If staging is good to go, we push to **prod**.



* Facebook (Meta), **A**pple, **A**mazons, **N**etflix, and **G**oogle (Alphabet)

[https://www.reddit.com/r/vibecoding/comments/1myakhd/how we vibe code at a faang/](https://www.reddit.com/r/vibecoding/comments/1myakhd/how_we_vibe_code_at_a_faang/)

Vibe Coding and the SDLC



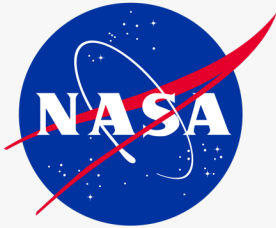
SDLC: Software Development Life Cycle

What is the Software Development Life Cycle (SDLC)?



Who knows this woman?

Margaret Hamilton
Led Software Development of
NASA's Apollo Guidance Computer



Origin of the Term “Software Engineering”



Margaret Hamilton
(1965-1969)



Anthony G. Oettinger
(1966)



NATO Software Engineering Conference
(1968/69)

“Software Engineering” as a Discipline (~1968)



Call to action:
“We know how to build **hardware**, now we must study how to build **software**.”

“Software Crisis”:

- Software was **inefficient** and difficult to **maintain**.
- It often did **not meet** the specified **requirements**.
- It was sometimes **not delivered**.

SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

January 1969

1968 NATO SE Conference Report: The SDLC

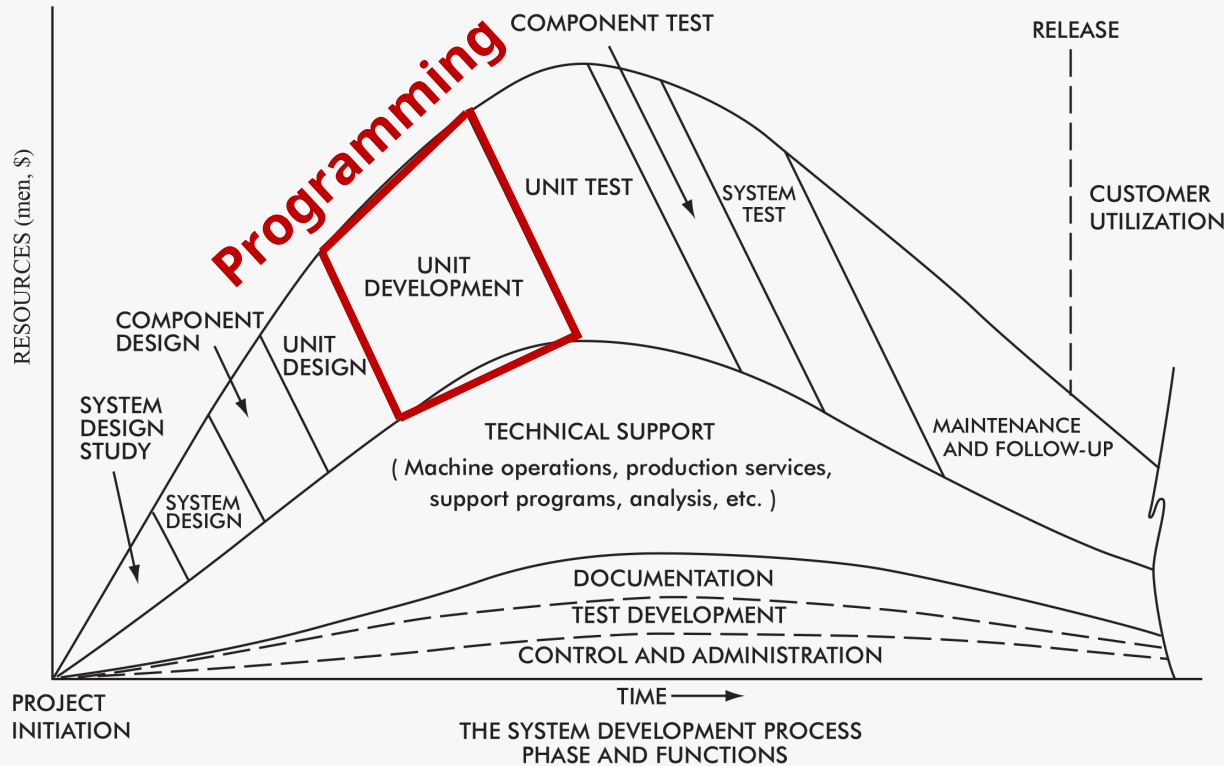


Figure 1. From Nash: Some problems in the production of large-scale software systems.

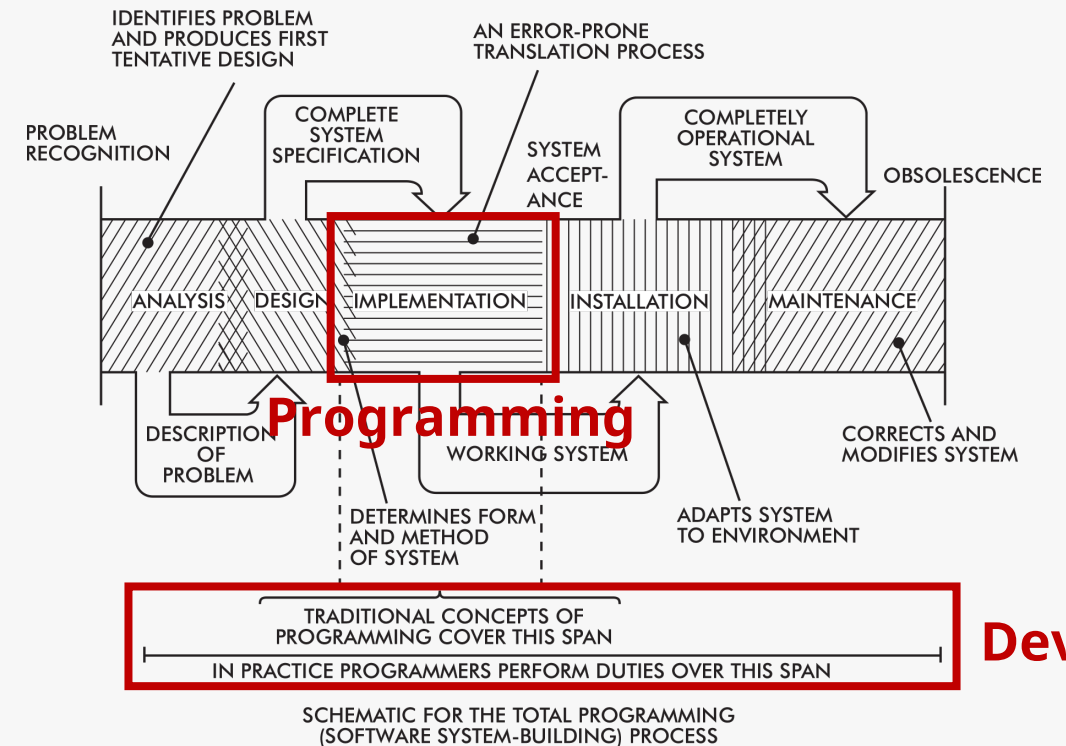
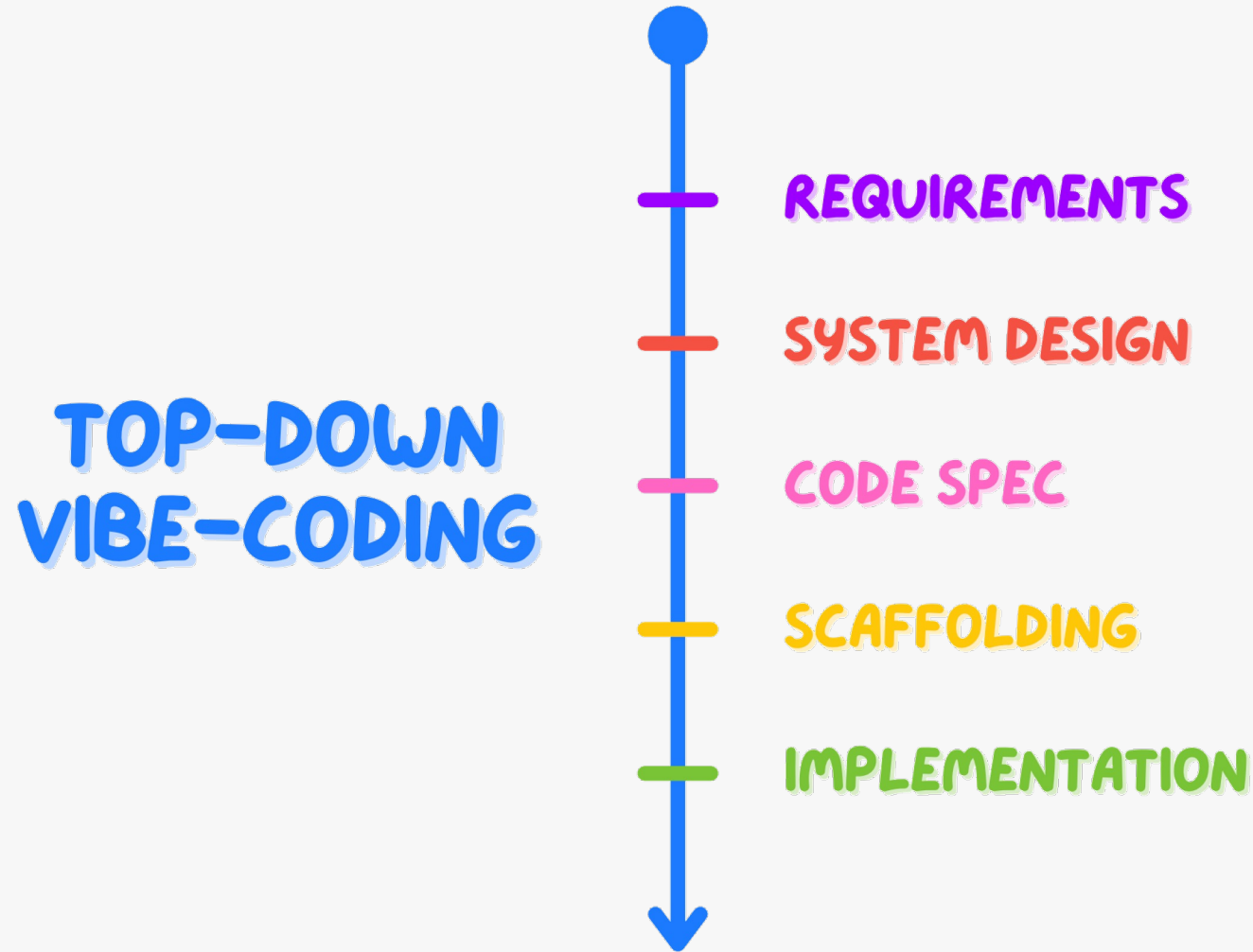


Figure 2. From Selig: Documentation for service and users. Originally due to Constantine.

<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PD>

The SDLC, aka “Top-Down Vibe-Coding”



<https://refactoring.fm/p/vibe-coding-workflows>

Software Engineering: Definition

“Software engineering is the **systematic application** of scientific and technological **knowledge, methods, and experience...**

...to the design, implementation, testing, maintenance, operation, and documentation of **software...**

...with a particular focus on **automation.**”

(adapted from ISO/IEC 2382:2015)

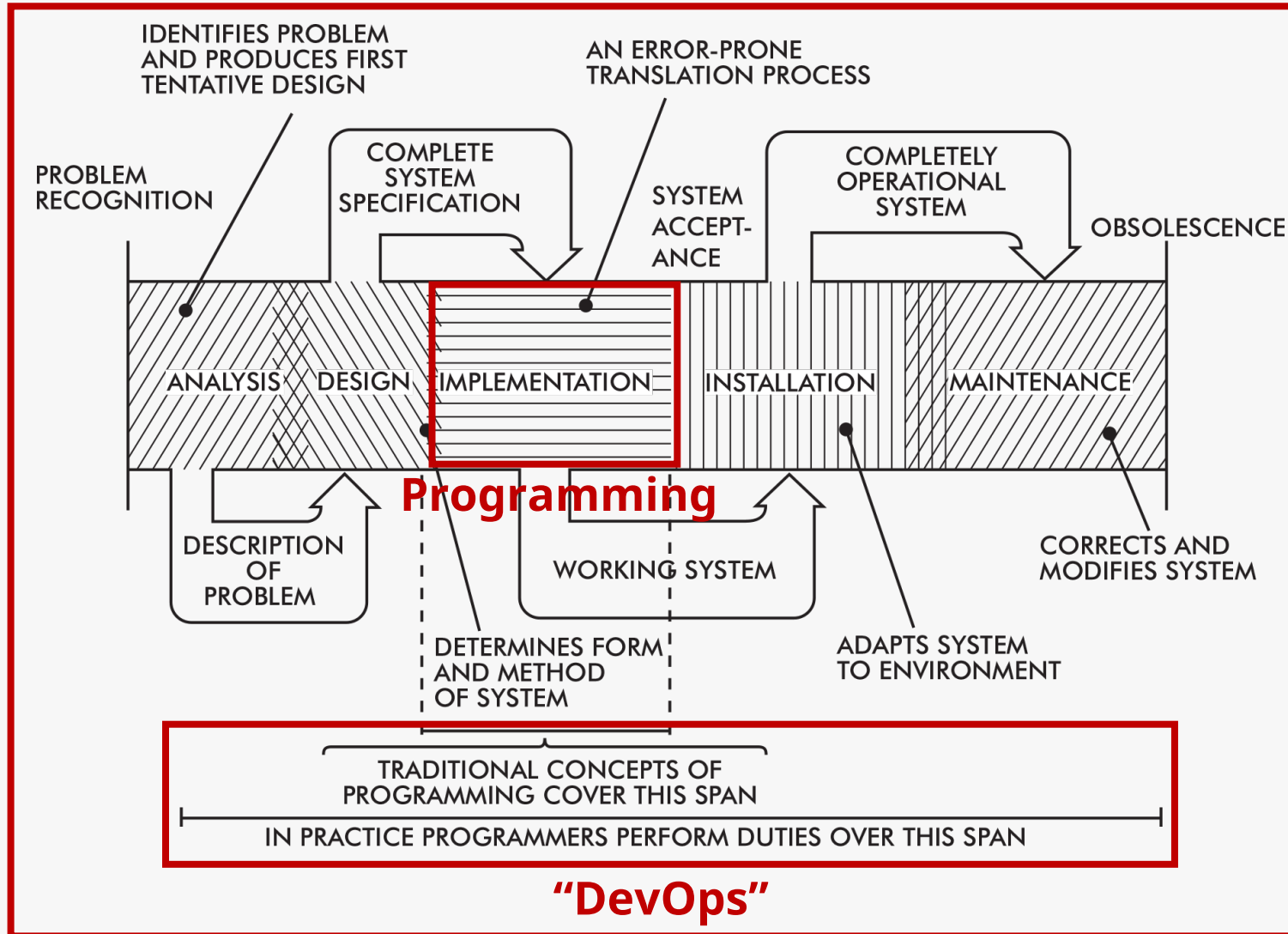
Software Engineering: Definition

“Software engineering is the application of scientific and technological knowledge, methods, processes, tools, and experience...to the design, development, testing, and documentation of software...with a particular emphasis on the systematic and disciplined application of scientific and technological knowledge, methods, processes, tools, and experience...”



...to the design, development, testing, and documentation of software...with a particular emphasis on the systematic and disciplined application of scientific and technological knowledge, methods, processes, tools, and experience... (from ISO/IEC 2382:2015)

SE Has Never Been Just About Coding!



Software Engineering

(1) Software Engineering (SE) \gg Programming



(2) AI “solves” Programming $\not\Rightarrow$ AI “solves” SE



(3) Agentic AI is a Software Systems Innovation

(4) We Need More Solid Empirical Evidence

Myths on Software Engineering and GenAI

Myth 1: Developers Spend Most of Their Time Writing Code

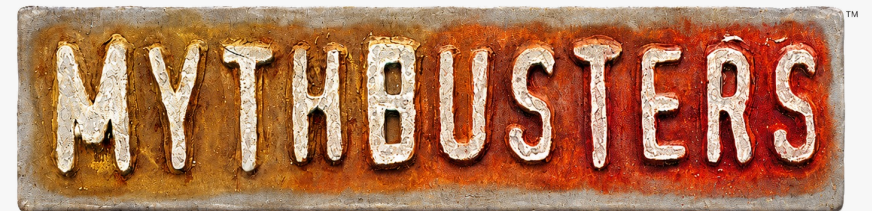
*“A study of more than 450 engineers at Microsoft in 2025 showed **developers spend only 14 percent of their time writing code**, reflecting what has been found in studies over the years.”*

RESEARCH-ARTICLE | 26 May 2026

Eight Myths on Software Engineering and GenAI

Examining the most common misconceptions

Jenna Butler, Brian Houck, Margaret-Anne Storey, Travis Lowdermilk, Steven Clarke, and Emerson Murphy-Hill



<https://spawn-queue.acm.org/doi/10.1145/3807963>



Yes, you can measure software developer productivity

B.S. Producers

August 17, 2023 | Article

<https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/yes-you-can-measure-software-developer-productivity#/>

“State of the Art” in Software Engineering as of 2019

No Single Metric Captures Productivity

Ciera Jaspan, Google, USA

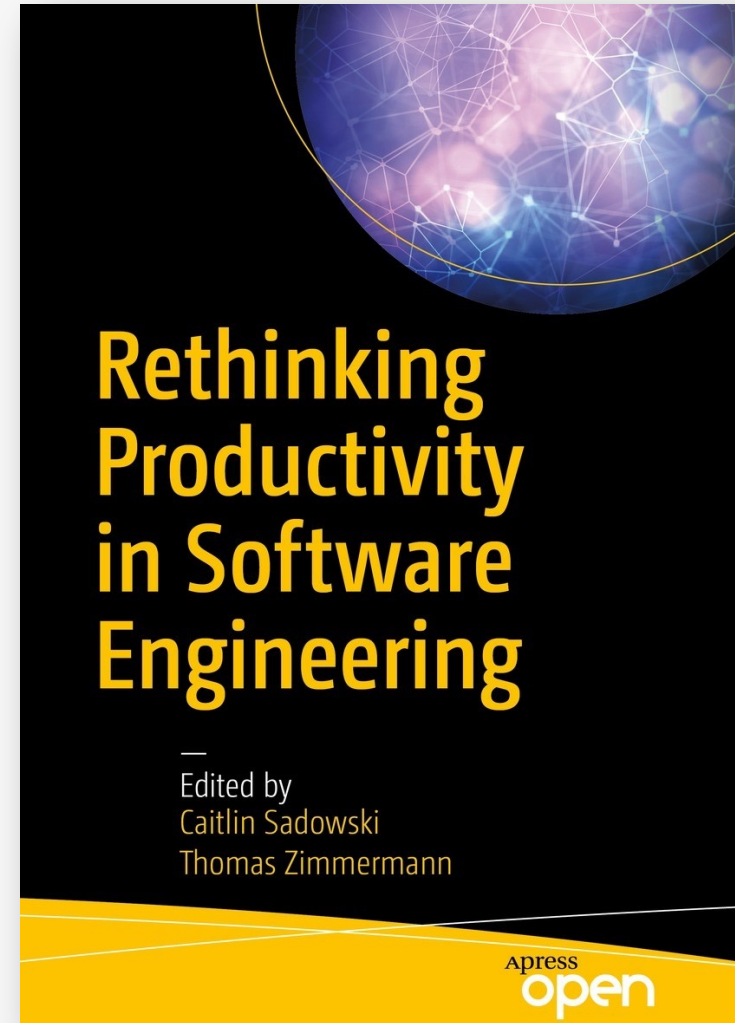
Caitlin Sadowski, Google, USA

“Measuring software productivity by lines of code is like measuring progress on an airplane by how much it weighs.”

—Bill Gates

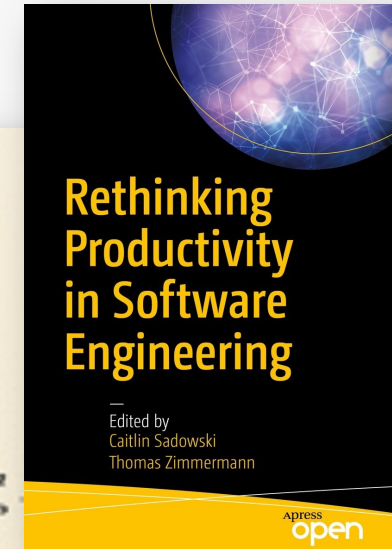
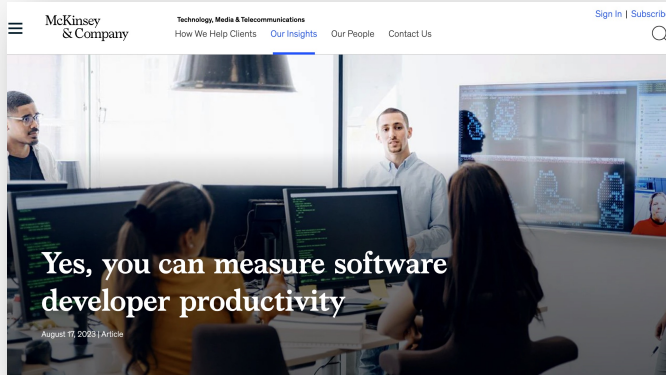
Why We Should Not Measure Productivity

Amy J. Ko, University of Washington, USA



<https://link.springer.com/book/10.1007/978-1-4842-4221-6>

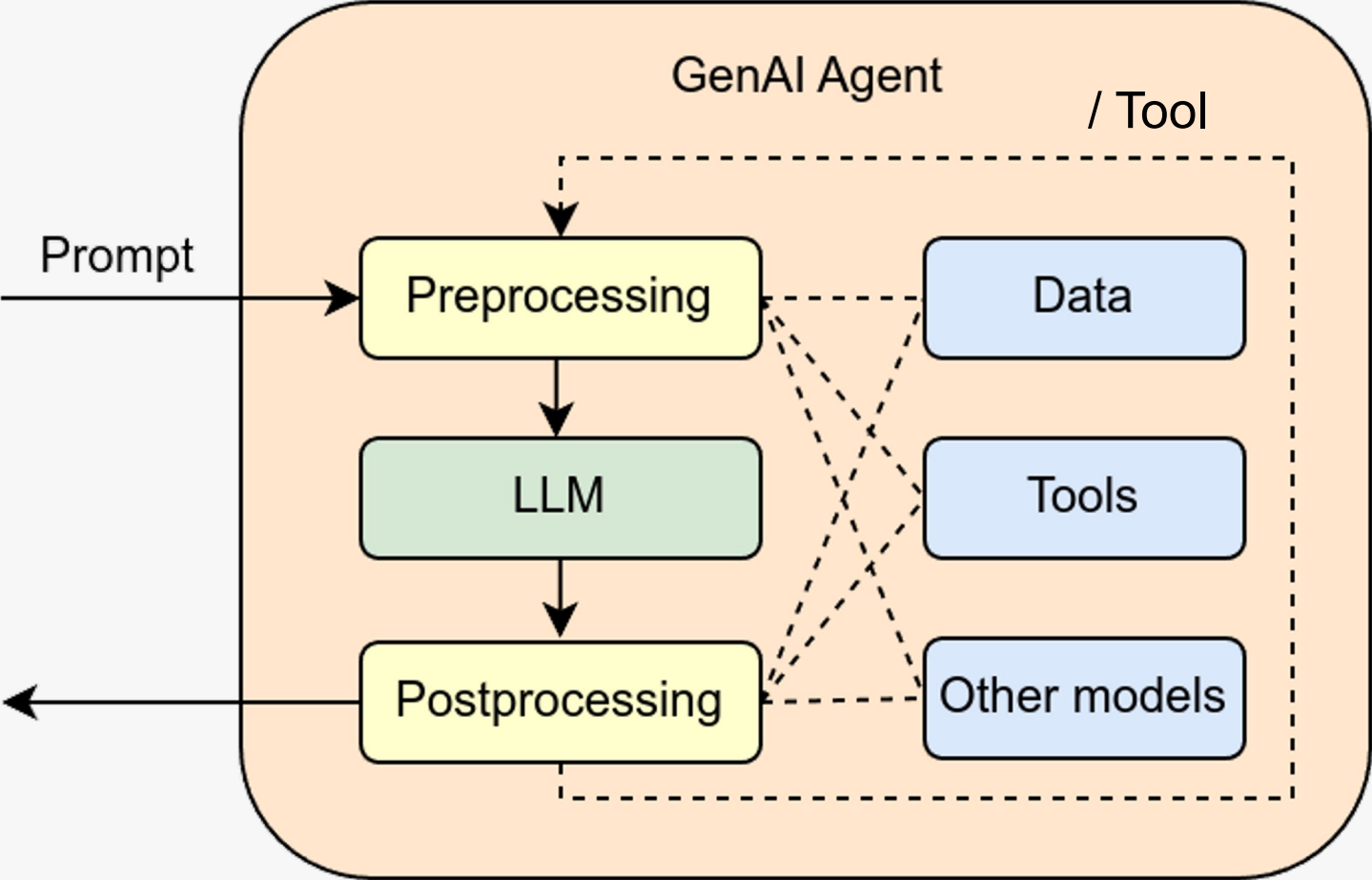
Public Discourse about SE



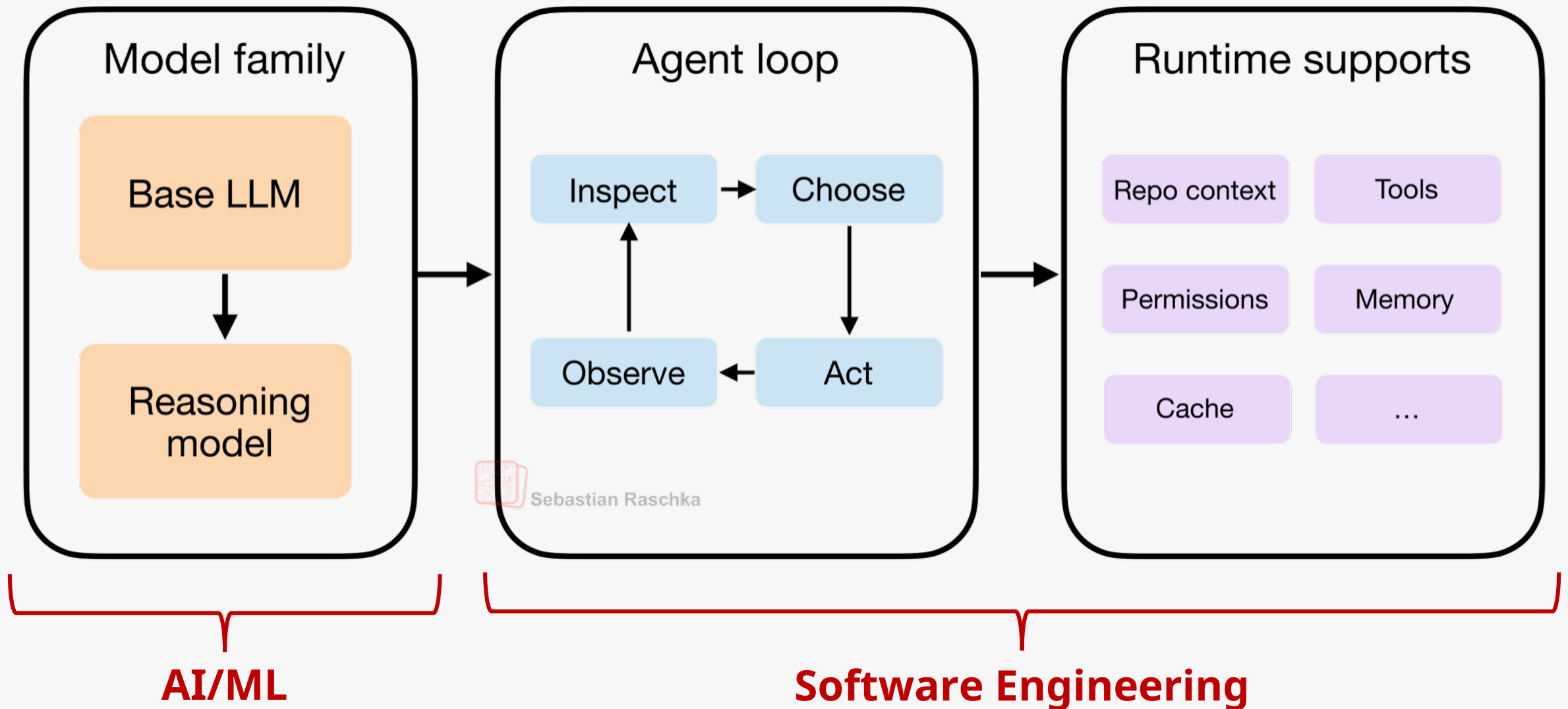
Agentic AI Assistants

Generative Artificial Intelligence (GenAI) Tools

The quality of the **context** provided as part of the prompts is crucial.



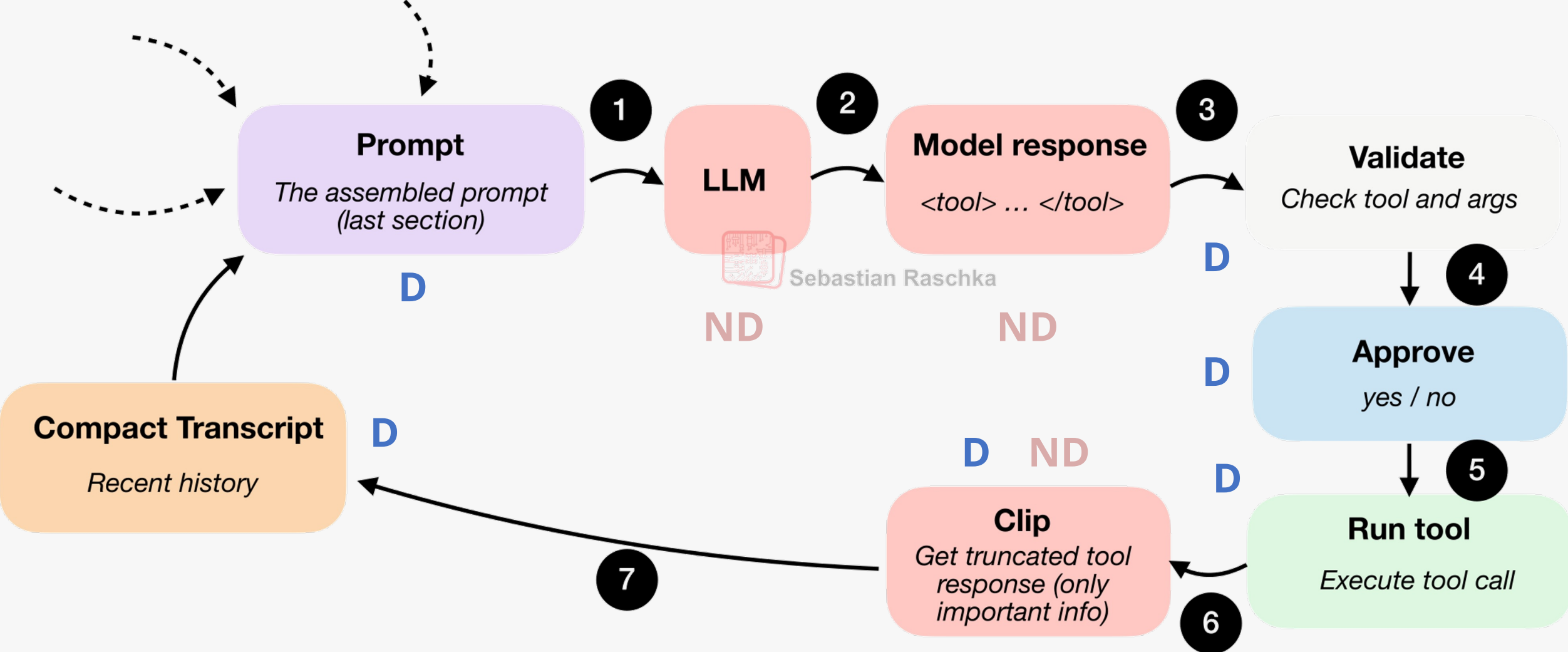
Agentic Coding Assistants



<https://magazine.sebastianraschka.com/p/components-of-a-coding-agent>

Example: Tool Use

D: Deterministic
ND: Non-deterministic



<https://magazine.sebastianraschka.com/p/components-of-a-coding-agent>

TL;DR

We've moved from model innovation to systems innovation

(1) Software Engineering (SE) \gg Programming



(2) AI “solves” Programming $\not\Rightarrow$ AI “solves” SE



(3) Agentic AI is a Software Systems Innovation



(4) We Need More Solid Empirical Evidence

Benchmarking AI Tools for SE

Benchmarking AI Tools

- Benchmarks are a central aspect of **vendors' marketing**.
- Most **benchmarks are open**, but the **models are closed**. (“Open” usually means open weight, not open training data.)
- **Benchmark contamination** is one of many problems.
- Another problem is that benchmarks **rarely define the construct they intend to measure** (functional correctness, security, reliability, etc.) and why the chosen **operationalization** is valid.

The Leaderboard Illusion

Shivalika Singh^{*1}, Yiyang Nan¹, Alex Wang², Daniel D'souza¹,
Sayash Kapoor³, Ahmet Üstün¹, Sanmi Koyejo⁴, Yuntian Deng⁵,
Shayne Longpre⁶, Noah A. Smith^{7,8}, Beyza Ermiş¹,
Marzieh Fadaee^{♦1}, and Sara Hooker^{♦1}

¹Cohere Labs, ²Cohere, ³Princeton University, ⁴Stanford University, ⁵University of Waterloo,
⁶Massachusetts Institute of Technology, ⁷Allen Institute for Artificial Intelligence, ⁸University of
Washington

<https://arxiv.org/pdf/2504.20879>

Contamination Report for Multilingual Benchmarks

Sanchit Ahuja*
Microsoft Research

Varun Gumma*
Microsoft Research

Sunayana Sitaram
Microsoft Research

<https://arxiv.org/pdf/2410.16186>

Benchmarks: HumanEval

Prompt:

```
def is_palindrome(text: str):  
    """  
    Checks if given string is a palindrome  
>>> is_palindrome('')  
True  
>>> is_palindrome('aba')  
True  
[...]
```

Canonical Solution:

```
def is_palindrome(text: str):  
    for i in range(len(text)):  
        if text[i] != text[len(text)-1-i]:  
            return False  
    return True
```

Tests:

```
assert is_palindrome('') == True  
assert is_palindrome('aba') == True  
assert is_palindrome('aaaaa') == True  
assert is_palindrome('zabcd') == False  
assert is_palindrome('xywyx') == True  
assert is_palindrome('xywyz') == False  
assert is_palindrome('xywzx') == False
```

<https://github.com/openai/human-eval/>

Benchmarks: HumanEval

Check an LLM-generated solution by validating that the generated code is **executable** and **passes the tests**. Usually, several (n) candidate solutions are generated.

$$\text{pass}@k = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}$$

pass@k estimates the probability that at least one of k generated candidate solutions for a fixed task passes the benchmark's tests.

where:

- n is the total number of samples generated per prompt (with $n \geq k$),
- c is the number of correct samples among n , and
- k is the number of attempts considered, drawn from the n generated samples without replacement.

Benchmarks: HumanEval

Frontier intelligence at 2x the speed

Claude 3.5 Sonnet sets new industry benchmarks for graduate-level reasoning (GPQA), undergraduate-level knowledge (MMLU), and coding proficiency (HumanEval). It shows marked improvement in grasping nuance, humor, and complex instructions, and is exceptional at writing high-quality content with a natural, relatable tone.

- Just Python.
- Just pure functions.
- No modules.
- No existing codebase.
- No debugging.
- No non-functional properties.

ANTHROPIC
Marketing in June 2024

pass@k estimates the probability that at least one of k generated candidate solutions for a fixed task passes the benchmark's tests.

Construct

Operationalization

Benchmarks: HumanEval

Frontier intelligence at 2x the speed

Claude 3.5 Sonnet sets new industry benchmarks for graduate-level reasoning



coding proficiency
g nuance, humor, and
-quality content with a

- Just Python.
- Just pure functions.
- No modules.
- No existing codebase.
- No debugging.
- No non-functional properties.

pass@k estimates the probability that at least one of k generated candidate solutions for a fixed task passes the benchmark's tests.

Construct

Operationalization

Benchmarks: HumanEval

- HumanEval contained several **incorrect “canonical” solutions**.
- Only **10 assertions per problem on average**, not enough to accurately assess the correctness of solutions.

“[...] continued pre-training on a dataset containing even a single epoch of the test split for HumanEval **boosted scores from 74.4% to 89.0%, highlighting the danger of data contamination.**”

Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation

Jiawei Liu^{1*} Chunqiu Steven Xia^{1*} Yuyao Wang² Lingming Zhang¹

University of Illinois Urbana-Champaign¹ Nanjing University²

{jiawei6, chunqiu2, lingming}@illinois.edu yuyao6@outlook.com

<https://arxiv.org/abs/2305.01210>

Google DeepMind

Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context

Gemini Team, Google¹

<https://arxiv.org/pdf/2403.05530>

Newer Benchmarks, Same Problems

Open Benchmarks for Closed Models



<https://knowyourmeme.com/memes/obama-awards-obama-a-medal>

(1) Software Engineering (SE) \gg Programming



(2) AI “solves” Programming $\not\Rightarrow$ AI “solves” SE



(3) Agentic AI is a Software Systems Innovation

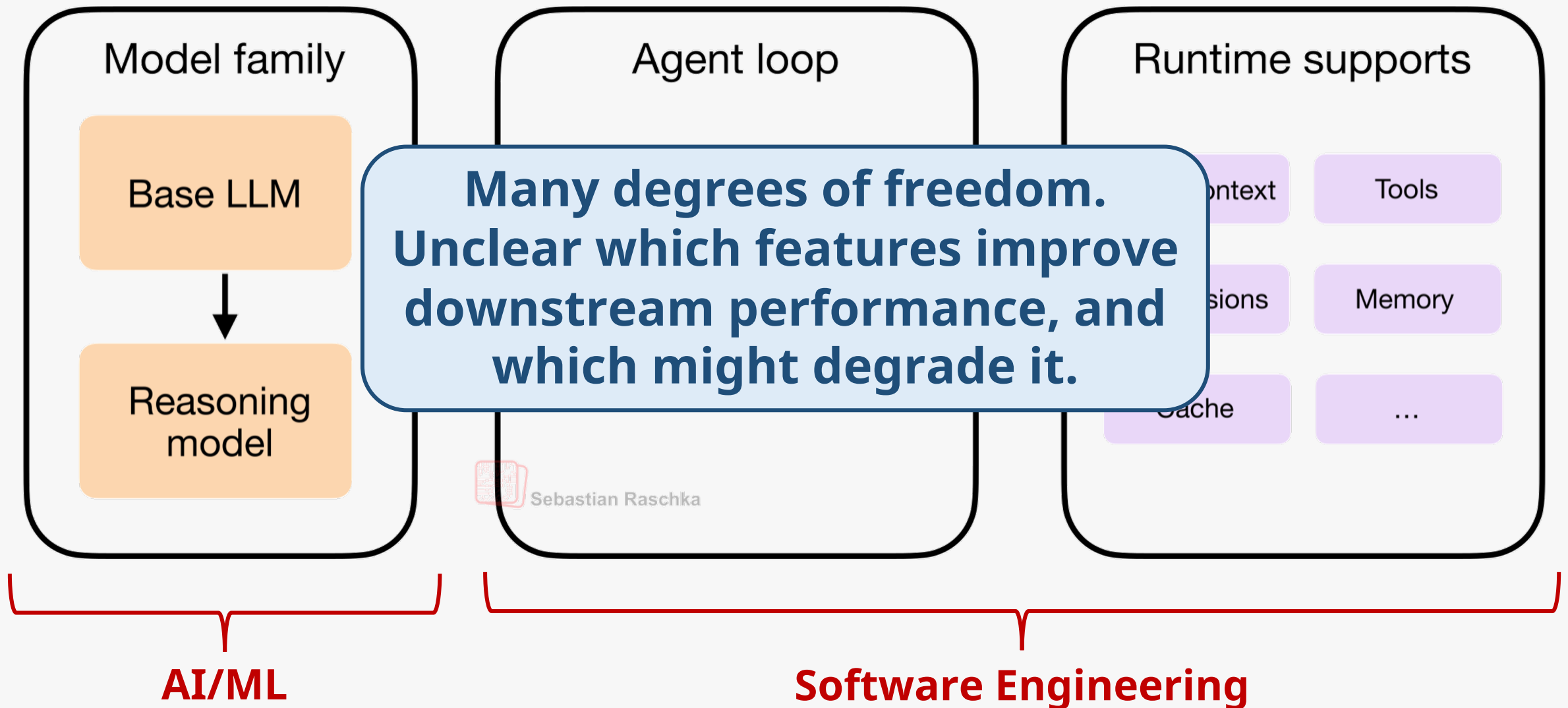


(4) We Need More Solid Empirical Evidence



Problem: We don't even know the "simple" things yet...

Agentic Coding Assistants



Sebastian Raschka

<https://magazine.sebastianraschka.com/p/components-of-a-coding-agent>

Configuring Agentic AI Coding Tools: An Exploratory Study

Matthias Galster

University of Bamberg
Bamberg, Germany
mgalster@ieee.org

Syedmoein Mohsenimofidi

Heidelberg University
Heidelberg, Germany
s.mohsenimofidi@uni-heidelberg.de

Jai Lal Lulla

Singapore Management University
Singapore, Singapore
jailal.l.2025@phdcs.smu.edu.sg

Muhammad Auwal Abubakar

University of Bamberg
Bamberg, Germany
muhammad.abubakar@uni-bamberg.de

Christoph Treude

Singapore Management University
Singapore, Singapore
ctreude@smu.edu.sg

Sebastian Baltes

Heidelberg University
Heidelberg, Germany
sebastian.baltes@uni-heidelberg.de

<https://assets.empirical-software.engineering/pdf/aiware26-agents-configuration.pdf>

User Misconceptions of LLM-Based Conversational Programming Assistants

Gabrielle O'Brien
University of Michigan
Ann Arbor, Michigan, USA
elleobri@umich.edu

Antonio Pedro Santos Alves
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
apsalves@inf.puc-rio.br

Sebastian Baltes
Heidelberg University
Heidelberg, Germany
sebastian.baltes@uni-heidelberg.de

Grischa Liebel
Reykjavik University
Reykjavik, Iceland
grischal@ru.is

Mircea Lungu
IT University of Copenhagen
Copenhagen, Denmark
mlun@itu.dk


Marcos Kalinowski
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
kalinowski@inf.puc-rio.br

<https://assets.empirical-software.engineering/pdf/jaws26-llm-misconceptions.pdf>

Can We Classify Flaky Tests Using Only Test Code? An LLM-Based Empirical Study

Alexander Berndt 

Heidelberg University
alexander.berndt@uni-heidelberg.de

Vekil Bekmyradov 


SAP
vekil.bekmyradov@sap.com

Rainer Gemulla 

University of Mannheim
rgemulla@uni-mannheim.de

Marcus Kessel 

University of Mannheim

Thomas Bach 

SAP

Sebastian Baltes 

Heidelberg University

On the Impact of AGENTS.md Files on the Efficiency of AI Coding Agents

Jai Lal Lulla

Singapore Management University
Singapore, Singapore
jailal.l.2025@phdcs.smu.edu.sg

Syedmoein Mohsenimofidi

Heidelberg University
Heidelberg, Germany
s.mohsenimofidi@uni-heidelberg.de

Matthias Galster

University of Bamberg
Bamberg, Germany
mgalster@ieee.org

Jie M. Zhang

King's College London
London, United Kingdom
jie.zhang@kcl.ac.uk

Sebastian Baltes

Heidelberg University
Heidelberg, Germany
sebastian.baltes@uni-heidelberg.de

Christoph Treude

Singapore Management University
Singapore, Singapore
ctreude@smu.edu.sg

<https://assets.empirical-software.engineering/pdf/jaws26-agents.md-efficiency.pdf>



The Impact of Configuring Agentic AI Coding Tools on Build-vs-Buy Decisions: A Study Protocol

Jai Lal Lulla  

Singapore Management University, Singapore

Matthias Galster  



University of Bamberg, Germany

Jie M. Zhang  

King's College London, United Kingdom

Sebastian Baltes  

Heidelberg University, Germany

Christoph Treude  

On the Need to Rethink Trust in AI Assistants for Software Development: A Critical Review

Sebastian Baltes^{†*}, Timo Speith^{†**}, Brenda Chiteri[†], Syedmoein Mohsenimofidi[†],
Shalini Chakraborty[‡], Daniel Buschek[‡]

[†]Heidelberg University, Germany

[‡]University of Bayreuth, Germany

[†]{sebastian.baltes, brenda.chiteri, s.mohsenimofidi}@uni-heidelberg.de

[‡]{timo.speith, s.chakraborty, daniel.buschek}@uni-bayreuth.de

<https://assets.empirical-software.engineering/pdf/tse26-trust-ai.pdf>



 empirical-software.engineering

Prof. Dr. Sebastian Baltes