

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



UNIVERSITÄT
BAYREUTH



UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

(1) Software Engineering (SE) $\gg$ Programming

(2) AI “solves” Programming $\nRightarrow$ AI “solves” SE

(3) Agentic AI is a Software Systems Innovation

(4) We Need More Solid Empirical Evidence

Current Discourse on AI for SE

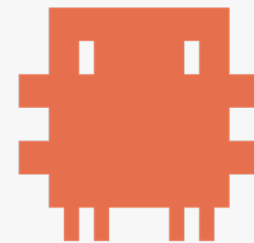




Who knows this man?

Boris Cherny

Creator of Claude Code



ANTHROPIC



“Coding Is Solved”

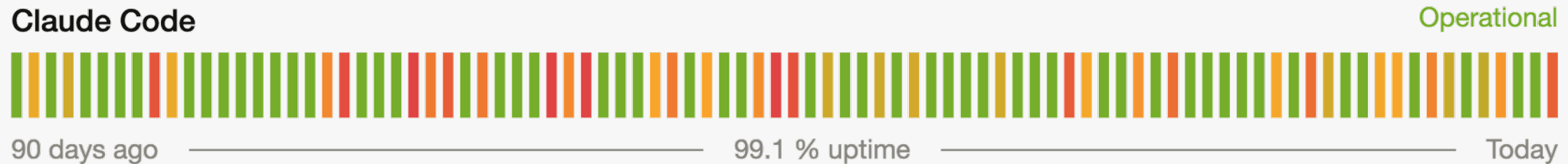
<https://youtu.be/SIGRN8jh2RI>

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



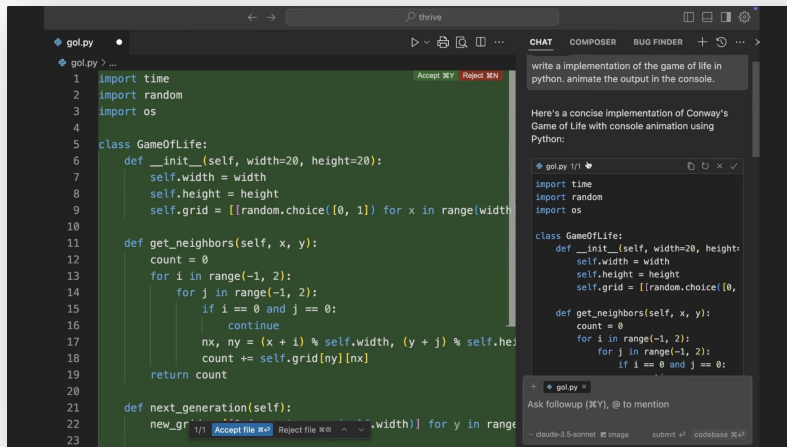
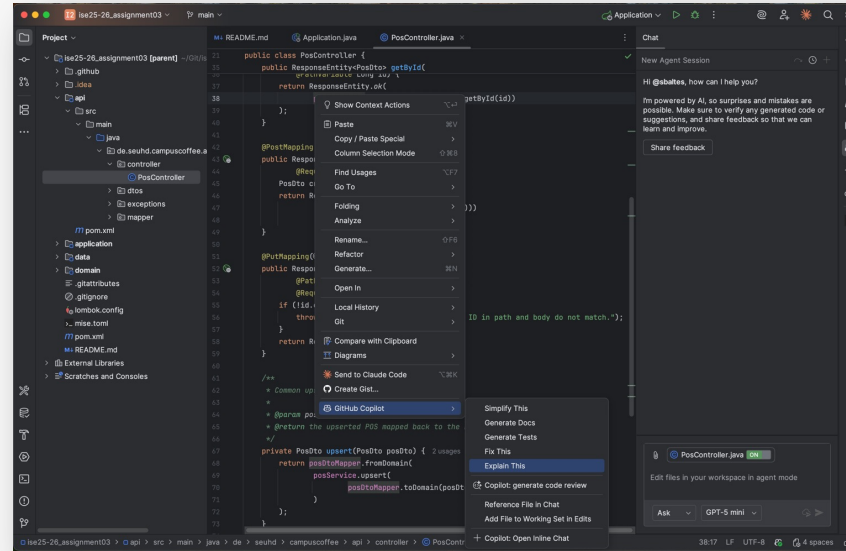
June 10, 2025 3 minutes read 0

<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.

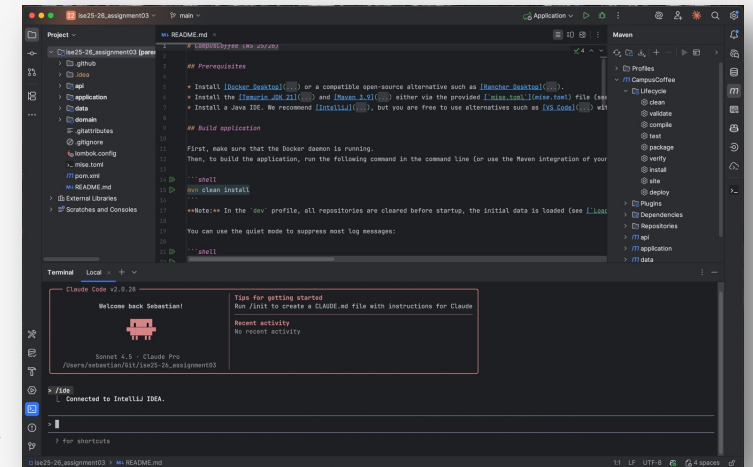
 **GitHub Copilot**



GenAI-
centric **IDEs**



"Agentic" AI
assistants



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

Wunschkennzeichen

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 image shows a GitHub repository for a project named 'license-plate-checker' by user 'sbaltes'. The repository is public and has 1 commit. The file list includes .gitignore, CLAUDE.md, README.md, check_plates.py, requirements.txt, and test_check_plates.py. The README is visible, showing the project's purpose: 'License Plate Availability Checker'. It mentions that the tool is a GWT (Google Web Toolkit) SPA and uses Playwright for headless browser automation.

Overlaid on the bottom right is a terminal window showing the application's execution. It prompts for a district (MTK) and then lists 31 combinations of license plate numbers (MTK-SN 1 to MTK-SN 31). Each combination is followed by the status 'TAKEN x', indicating that all checked combinations are already taken.

```
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 repositories are:

- caldav-utils** (Public): A repository by sbaltes and claude. The file list includes `.gitignore`, `LICENSE`, `README.md`, `caldav_utils.py`, and `requirements.txt`. The README section is partially visible, showing the title **caldav-uti** and the description "Purge, deduplicate, Works with standard".
- imap-dedup** (Public): A repository by sbaltes. The file list includes `.gitignore`, `LICENSE`, `README.md`, `imap_dedup.py`, `offlineimaprc`, `sync-mailbox.sh`, and `test_imap_dedup.py`. The README section is partially visible, showing the title **imap-dedup** and the description "Deduplicate emails in a".
- openreview-downloader** (Public): A repository by sbaltes and claude. The file list includes `.gitignore`, `LICENSE`, `README.md`, `download_reviews.py`, and `requirements.txt`. The commit history shows several recent updates. The README section is visible, showing the title **OpenReview Review Downloader** and the description "Download reviews, metareviews, decisions, and optionally comments from [OpenReview](#) for a given paper."

Each repository page shows standard GitHub interface elements: repository name, public status, branch selection (main), file search, and action buttons (Pin, Watch, Fork, Star, Code). The 'openreview-downloader' page also includes an 'About' section with a description: "Script to download reviews, metareviews, and decisions as Markdown and JSON from the OpenReview platform." and a 'Releases' section indicating "No releases published".

Where It Doesn't Work: Publicly Deployed Software

ANDY GREENBERG

SECURITY MAY 7, 2026 7:00 AM

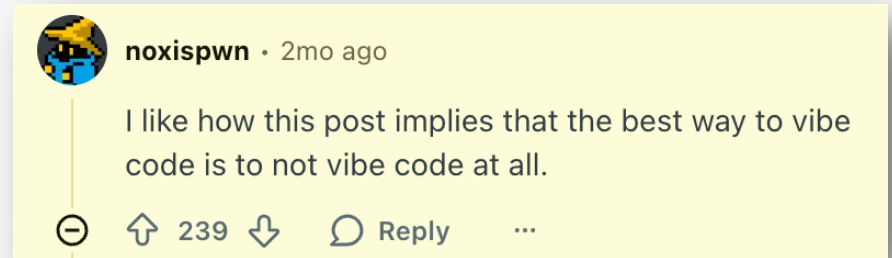
Thousands of Vibe-Coded Apps Expose Corporate and Personal Data on the Open Web

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

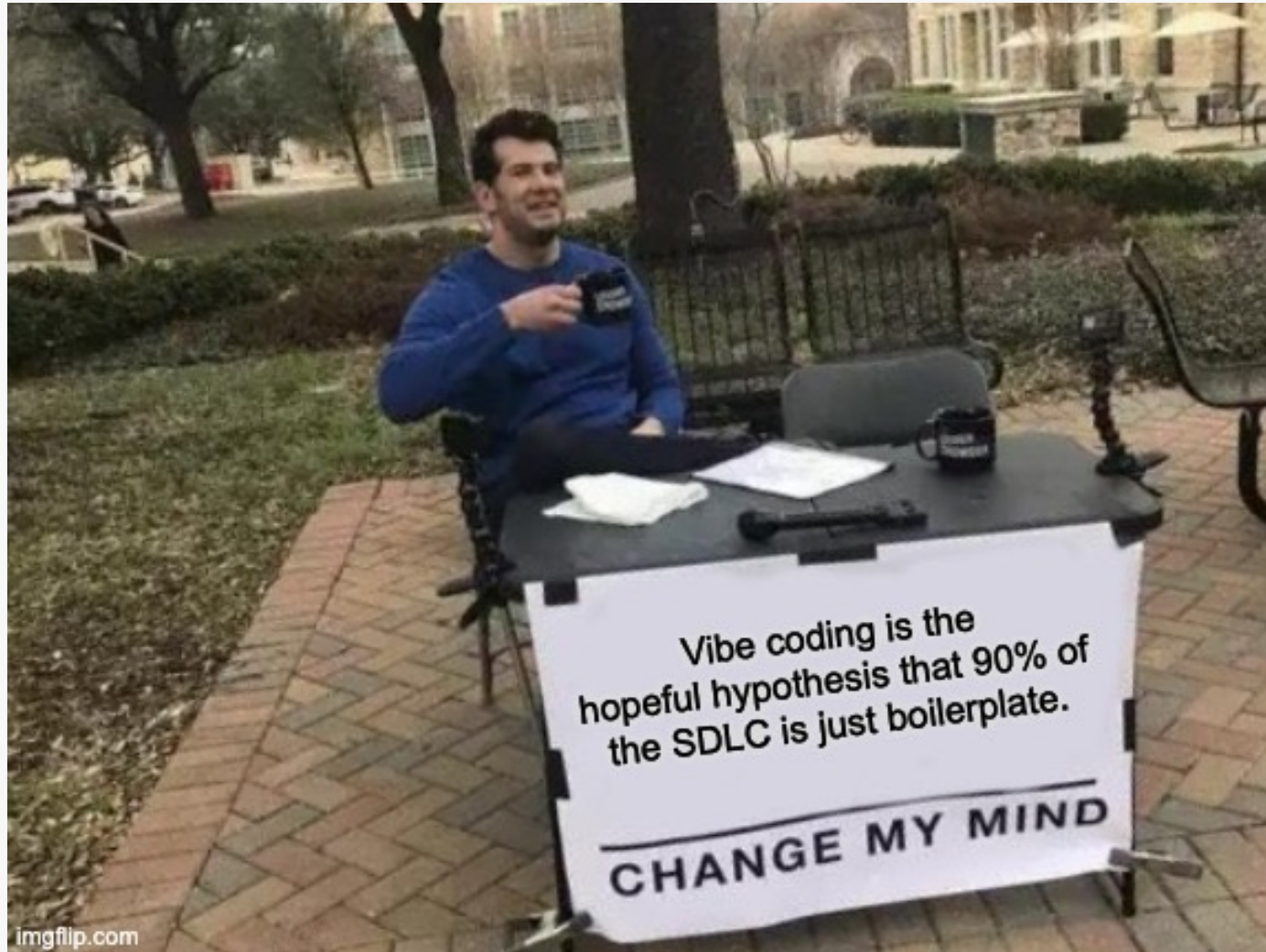
1. [...] start with a **technical design document**.
2. **Design review** before [...] development [...].
3. If you pass review, [...] **more documentation on each subsystem**.
4. **Backlog development** and **sprint planning**.
5. **Software development** [...] We use **test-driven development**, so 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), Apple, Amazon, Netflix, and Google (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



<https://imgflip.com/i/9zk6ku>

SDLC: Software Development Life Cycle

Dunning-Kruger and the AI Hype



Gergely Orosz  • Following

Deepdives on software engineering, tech careers and industry trends. Writi...

[Visit my website](#)

3mo • 



Spent a few hours inside OpenAI & Anthropic yesterday and today. One interesting thing:

I asked several people (both engineers and non-engineers) about what they think of the future of software engineering.

The "closer" to shipping production code folks are, the less they believe software engineering will be "solved" fully by AI. And the more removed / less hands on, the more they believed that it is either pretty much already solved or will be solved eventually, as models improve.

I would think this is a kind of bias when you overestimate model capability for areas you know very little of. (I could be wrong, but then so are hands-on software engineers inside the labs!)



2,367



137



148



https://www.linkedin.com/posts/gergelyorosz_spent-a-few-hours-inside-openai-anthropic-share-7476403008478728194-Pj0o/

*"The **Dunning-Kruger effect** is a cognitive bias that describes the systematic tendency of people with low ability in a specific area to give overly positive assessments of their ability."*

https://en.wikipedia.org/wiki/Dunning%E2%80%93Kruger_effect

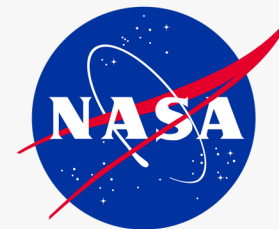
What is the Software Development Life Cycle (SDLC)?



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)

1968 NATO SE Conference Report: The SDLC

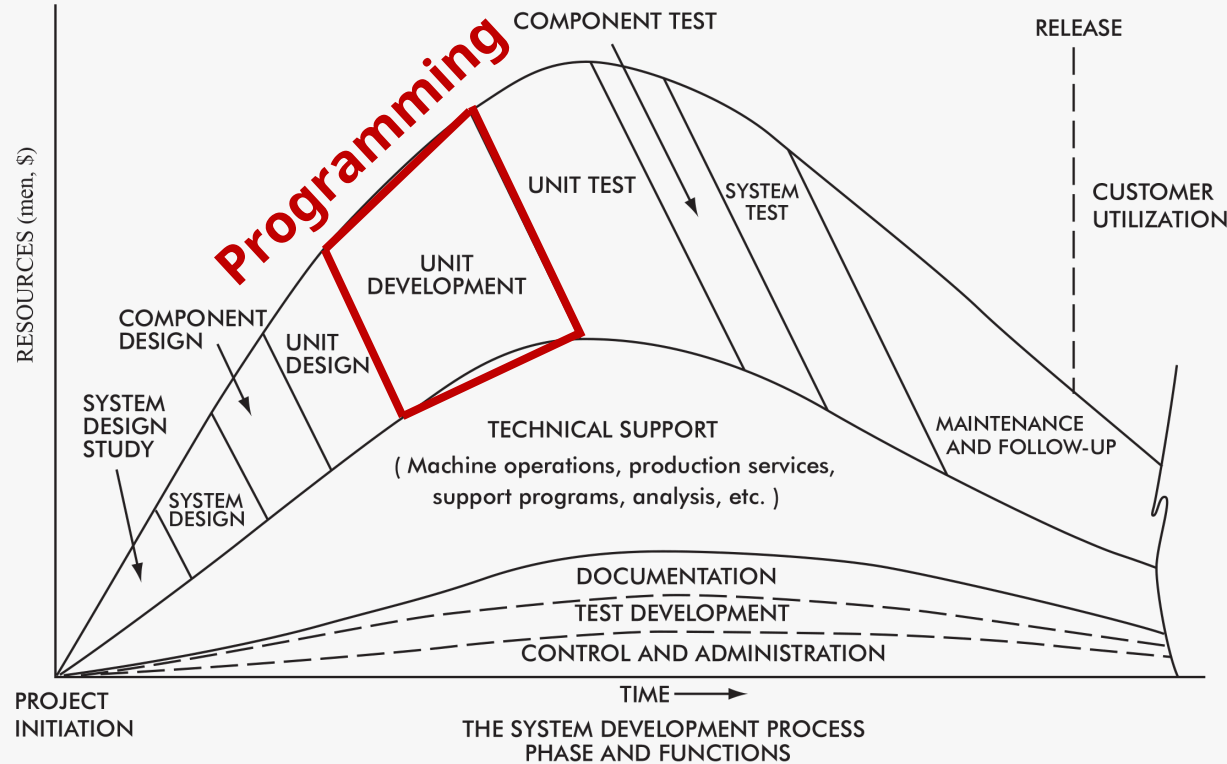


Figure 1. From Nash: Some problems in the production of large-scale software systems.

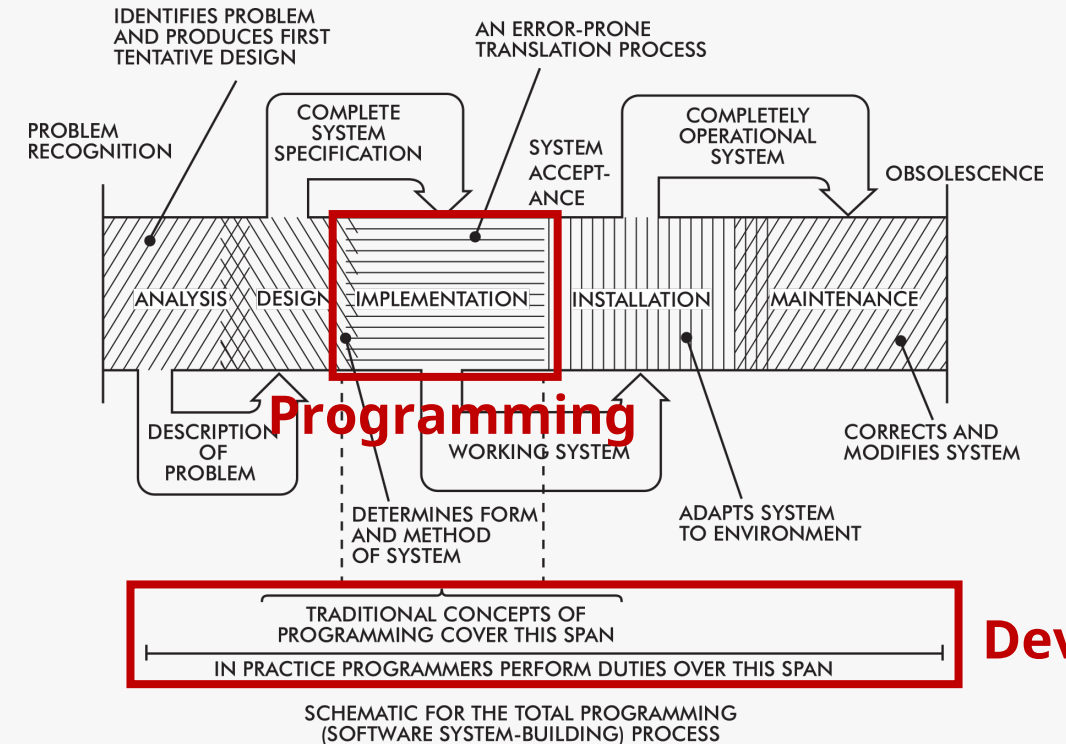
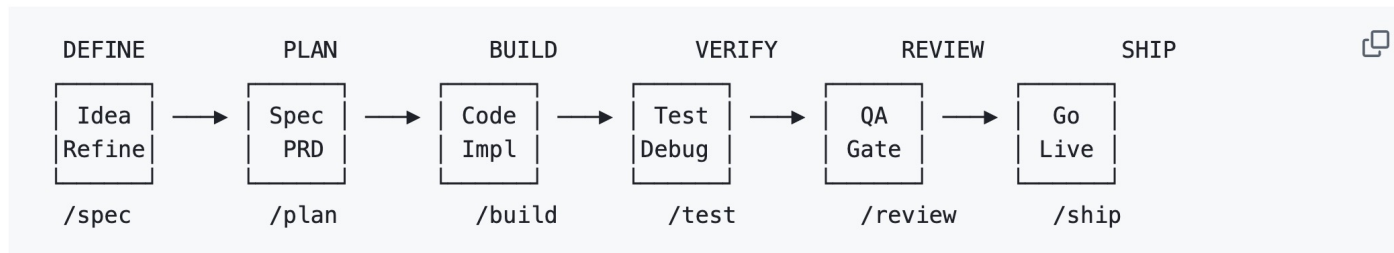


Figure 2. From Selig: Documentation for service and users. Originally due to Constantine.

SDLC and Agentic AI

Agent Skills

Spec Plan Build Test Review Simplify Ship



<https://github.com/addyosmani/agent-skills>

rk 8.3k ▾ ★ Starred 77.3k ▾

About

Production-grade engineering skills for AI coding agents.

skills cursor codex antigravity

agent-skills claude-code

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)

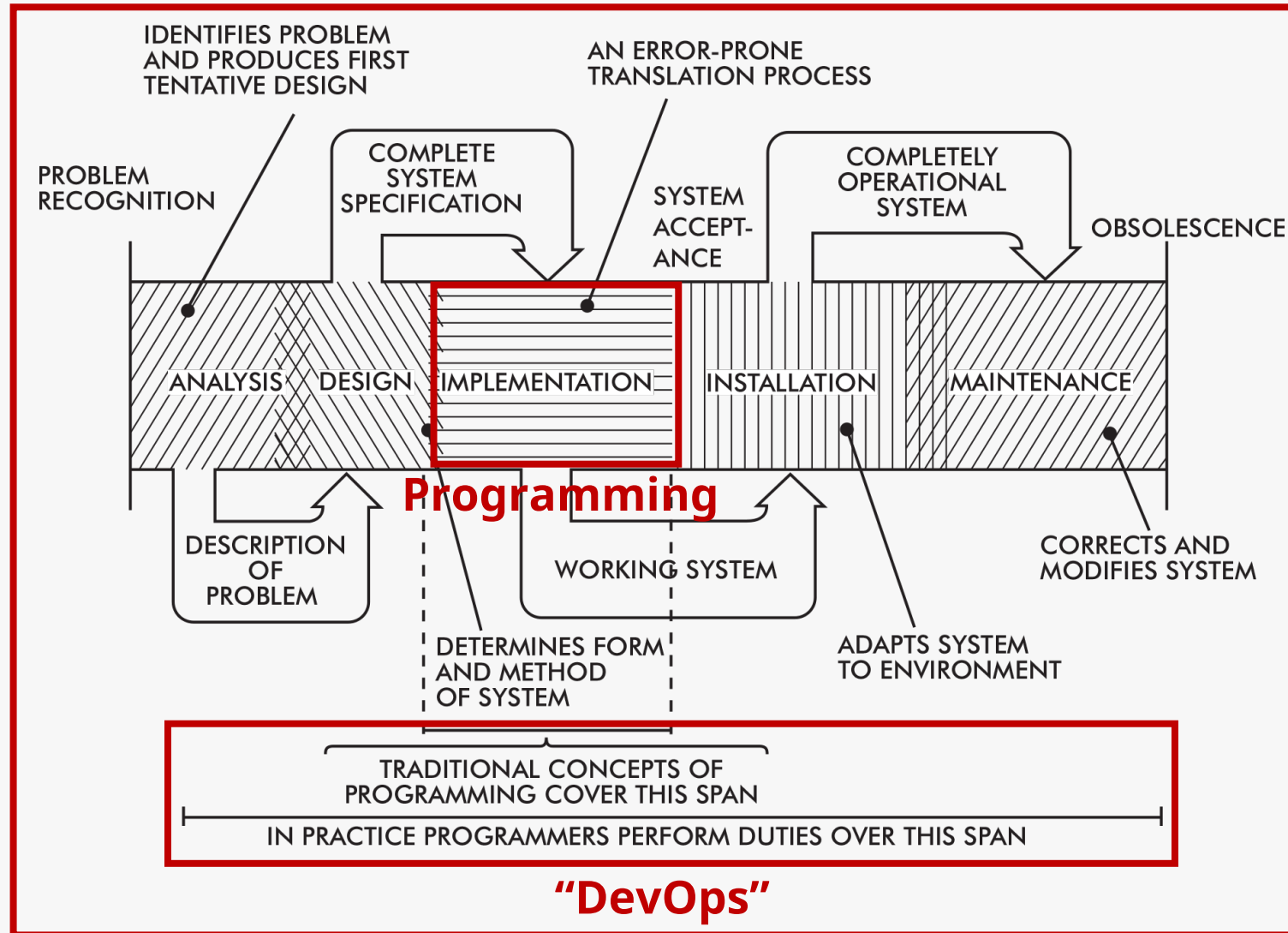
Software Engineering: Definition

“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 approach to the design, development, testing, and documentation of software.”



...to the design, development, testing, and documentation of software...with a particular emphasis on the systematic and disciplined approach to the design, development, testing, and documentation of software.” (from ISO/IEC 2382:2015)

SE Has Never Been Just About Coding!



Software Engineering

(1) Software Engineering (SE) $\gg$ Programming



(2) AI “solves” Programming $\nRightarrow$ 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

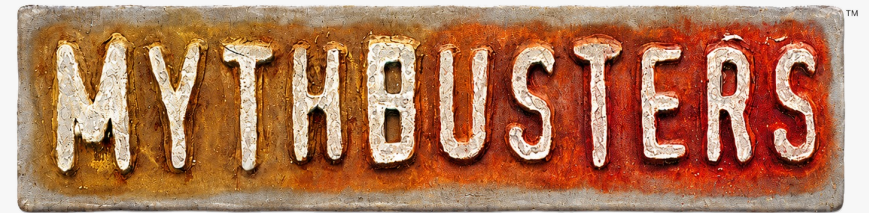
*“A study of more than 450 engineers at Microsoft in 2025 showed **developers spend only 14 percent of their time writing code**, reflecting what has been found in studies over the years.”*

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

August 17, 2023 | Article

B.S. Producers

<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

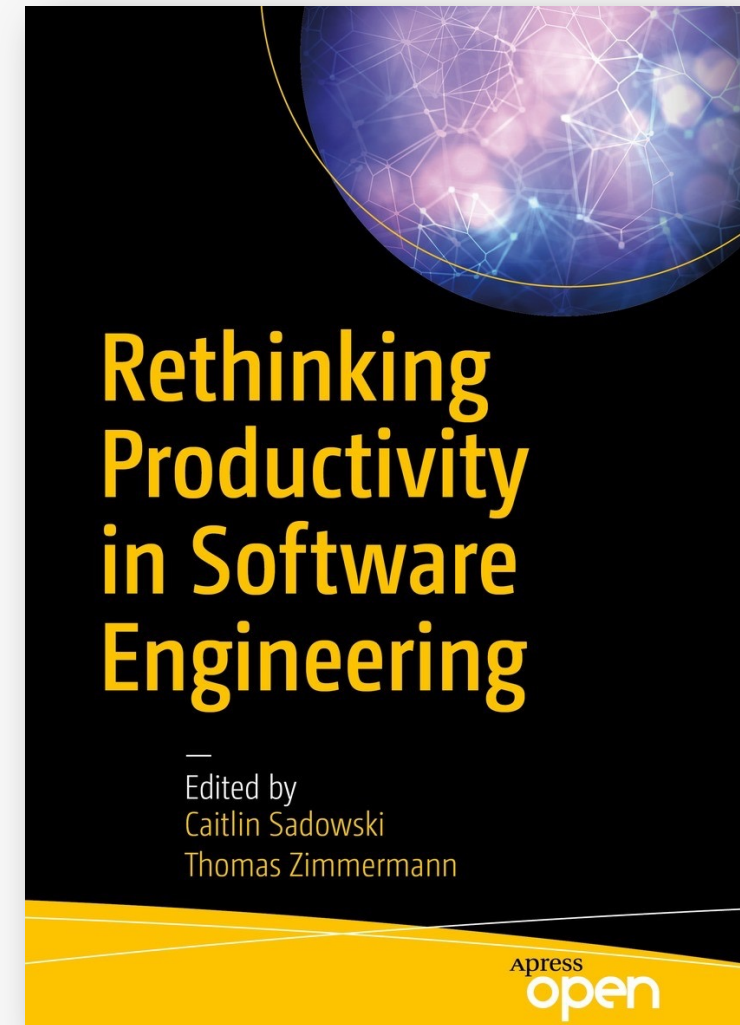
Caitlin Sadowski, Google, USA

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

—Bill Gates

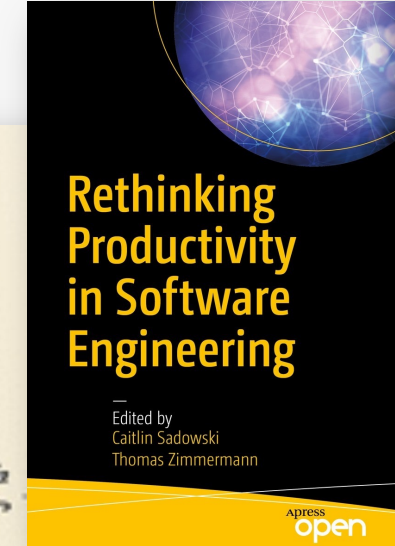
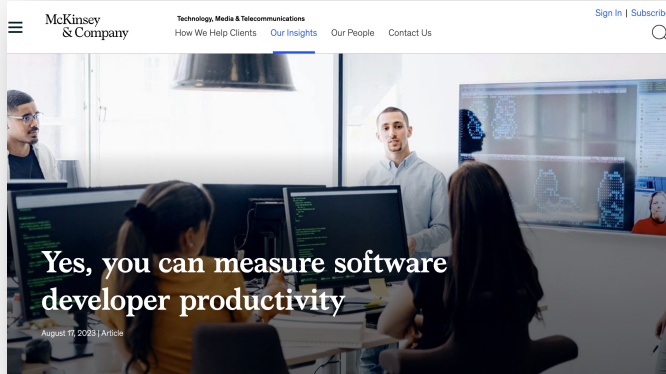
Why We Should Not Measure Productivity

Amy J. Ko, University of Washington, USA



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

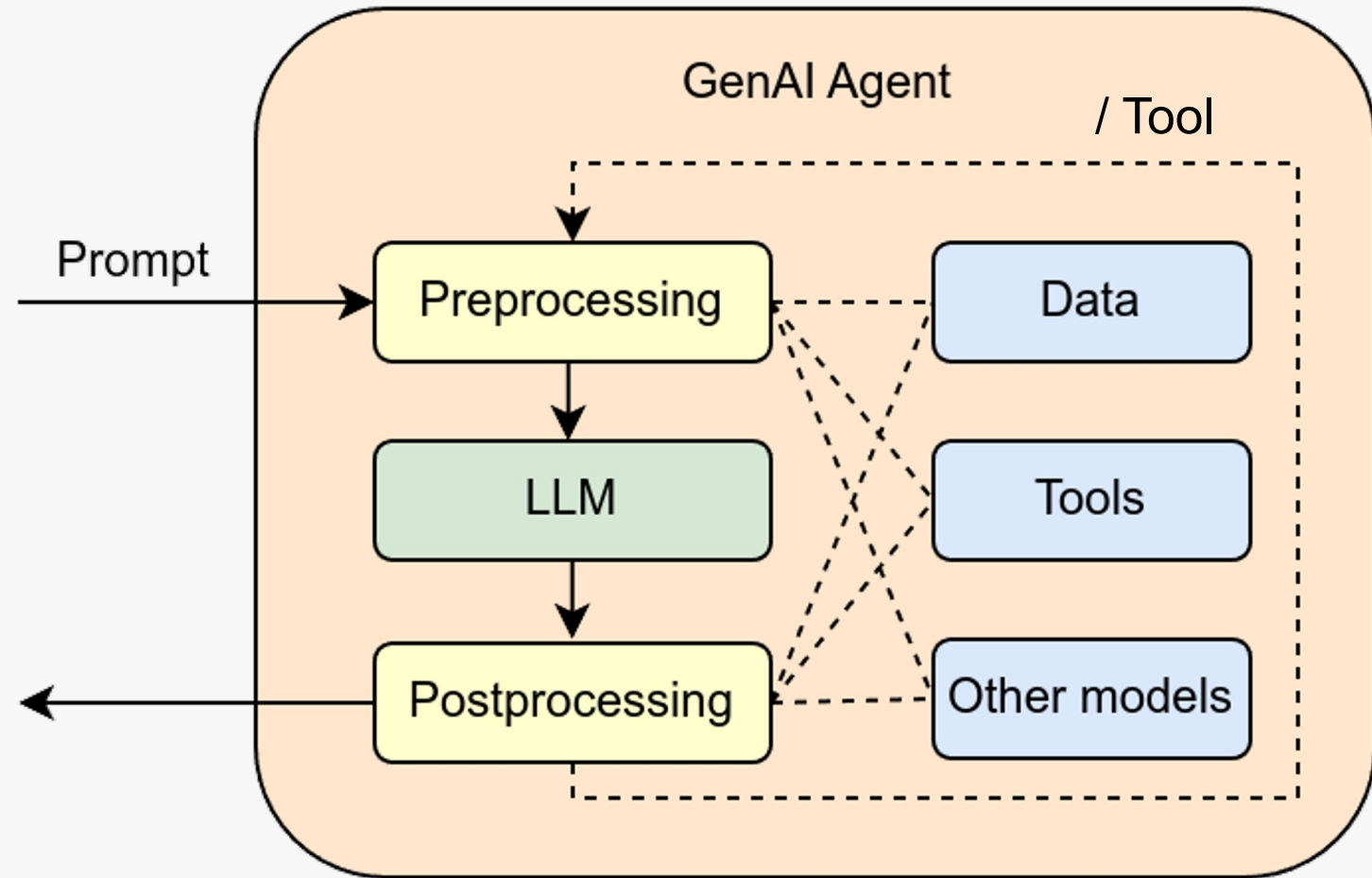
Public Discourse about SE



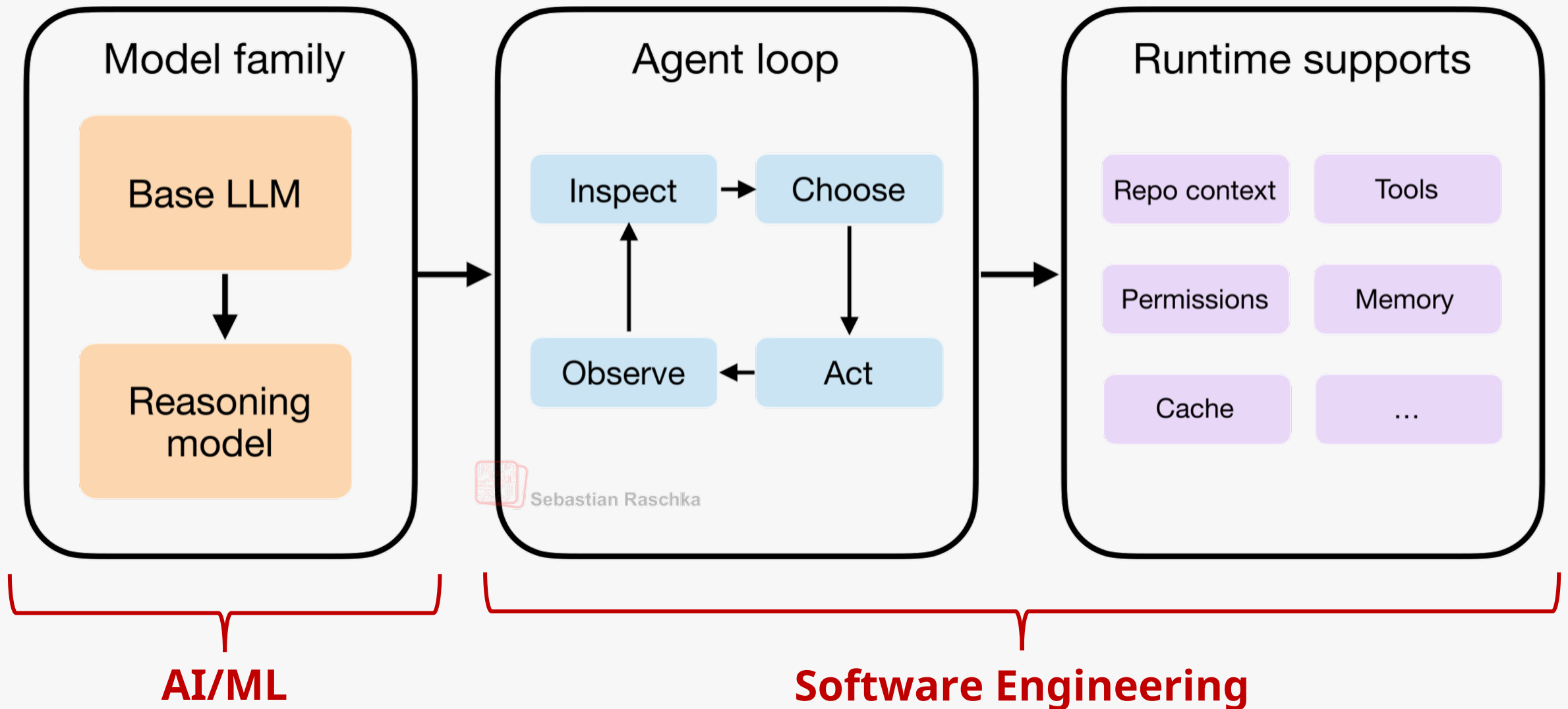
Agentic AI Assistants

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>

D: Deterministic
ND: Non-deterministic



TL;DR

We've moved from model innovation to systems innovation

(1) Software Engineering (SE) $\gg$ Programming



(2) AI “solves” Programming $\nRightarrow$ 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 Fadaee¹, and Sara Hooker¹

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

**Construct
validity?!**

- 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.**

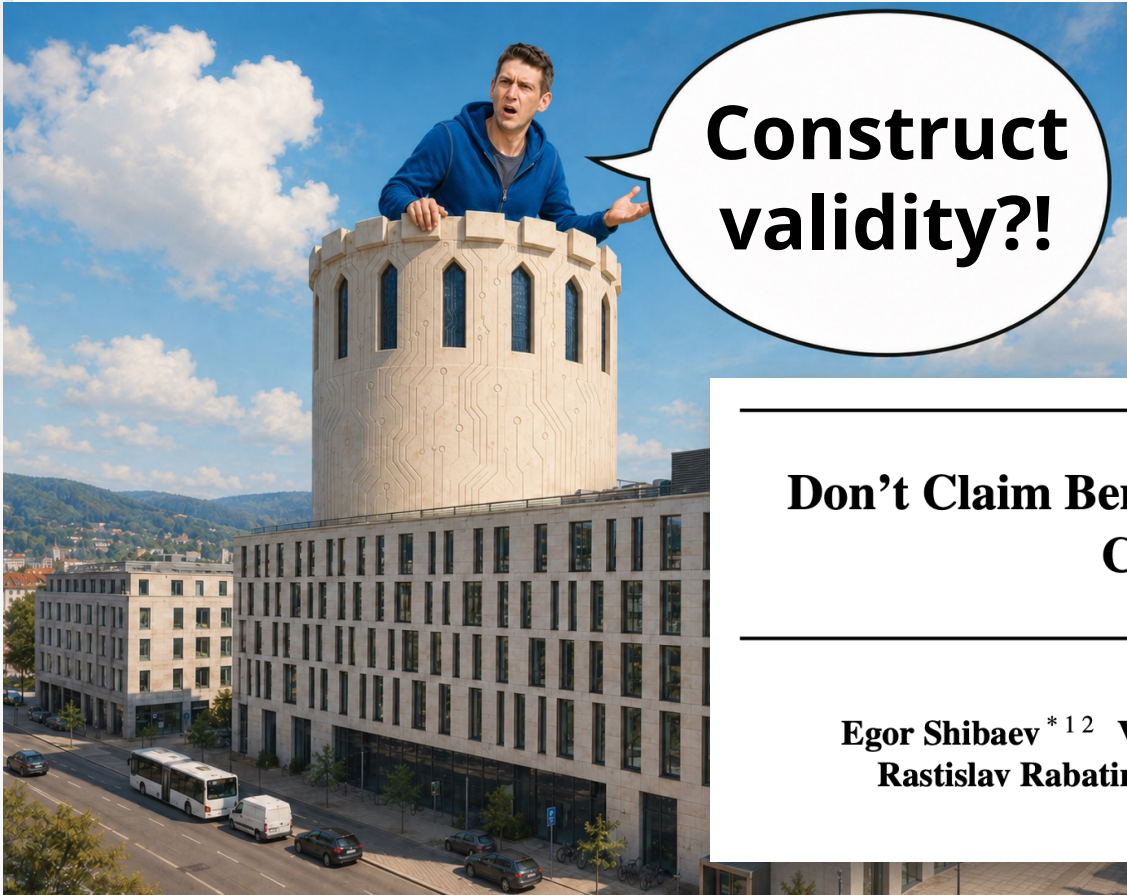
Construct

Operationalization

Benchmarks: HumanEval

Frontier intelligence at 2x the speed

Claude 3.5 Sonnet sets new industry benchmarks for



**Construct
validity?!**

July 8, 2026 Research Publication

<https://openai.com/index/separating-signal-from-noise-coding-evaluations/>

Separating signal from noise in coding evaluations

Through a detailed audit, we find widespread task issues in SWE-Bench Pro and estimate that ~30% of the tasks are broken.

that at least one of k generated

Don't Claim Benchmark-Oriented Optimization Improves General Coding Capability — Diverse Evaluation Is Required

Egor Shibaev^{*12} Vera Kudrevskaia^{*12} Timur Galimzyanov^{*1} Mikhail Evtikhiev^{*3} Ana Terna⁴
Rastislav Rabatin⁵ Timur Kudashev¹² Timofey Bryksin³ Arina Puchkova¹ Patrik Bartak⁴
Egor Bogomolov⁴ Sergey Titov⁴

<https://openreview.net/pdf?id=dQs1srLmGN>

Benchmarks: HumanEval

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

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>

“[...] 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.**”

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 $\nRightarrow$ 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...

2022–2024: The Fear That AI Is Eating Software

DOI:10.1145/3570220

Matt Welsh

Viewpoint

The End of Programming

*The end of classical computer science is coming,
and most of us are dinosaurs waiting for the meteor to hit.*

<https://dl.acm.org/doi/10.1145/3570220>

AI

In a leaked recording, Amazon cloud chief tells employees that most developers could stop coding soon as AI takes over

By Eugene Kim

+ Follow

<https://www.businessinsider.com/aws-ceo-developers-stop-coding-ai-takes-over-2024-8>

Don't learn to code: Nvidia's founder Jensen Huang advises a different career path



MICHAEL PETRAEUS • 23 FEB 2024

<https://vulcanpost.com/853029/dont-learn-to-code-jensen-huang-on-career/>

Salesforce Will Hire No More Software Engineers in 2025, Says Marc Benioff



By Henry Martin

Published December 18, 2024

Updated December 23, 2024

<https://www.salesforceben.com/salesforce-will-hire-no-more-software-engineers-in-2025-says-marc-benioff/>

2022–2024: The Fear That AI Is Eating Software

DOI:10.1145/3570220

Viewpoint The End

*The end of classical computing
and most of us are dinosaurs*

<https://>

Don't learn to Jensen Huang path

 MICHAEL PETRAEUS • 23 FEB 2024

<https://vulcanpost.com>



ells employees
on as AI takes

[ai-takes-over-2024-8](#)

are Engineers

[software-engineers-in-](#)

Since 2025: Discourse Shifts From Models to Systems

AI Can't Replace Software Engineers Yet, Marc Benioff Says



By Henry Martin

April 09, 2026

<https://www.salesforceben.com/ai-cant-replace-software-engineers-yet-marc-benioff-says/>

AI + ML

Vibe coding service Replit deleted user's production database, faked data, told fibs galore

AI ignored instruction to freeze code, forgot it could roll back errors, and generally made a terrible hash of things

 Simon Sharwood

<https://www.theregister.com/software/2025/07/21/vibe-coding-service-replit-deleted-production-database/719783>

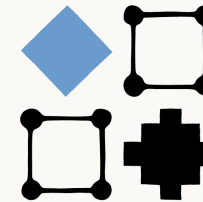
Announcements

Claude 3.7 Sonnet and Claude Code

Feb 24, 2025

<https://www.anthropic.com/news/claude-3-7-sonnet>

Engineering at Anthropic



Harness design for long-running application development

Published Mar 24, 2026

Harness design is key to performance at the frontier of agentic coding. Here's how we pushed Claude further in frontend design and long-running autonomous software engineering.

<https://www.anthropic.com/engineering/harness-design-long-running-apps>

Since 2025: Discourse Shifts From Models to Systems

AI Can't Replace Software Eng

Benioff Says



By Henry Martin
April 09, 2026

<https://www.salesforceben.com/ai-cant-replace-software-e>

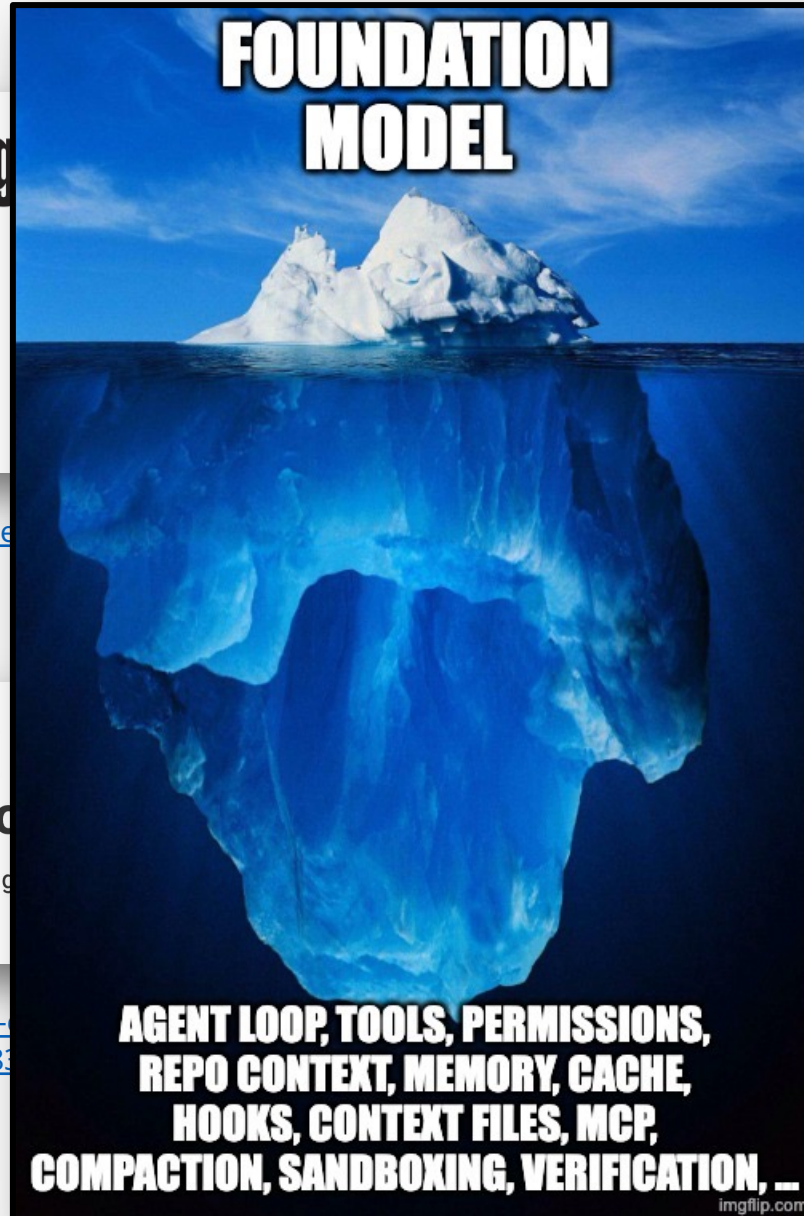
AI + ML

Vibe coding service Replit deleted database, faked data, told fibs gal

AI ignored instruction to freeze code, forgot it could roll back errors, and g

Simon Sharwood

<https://www.theregister.com/software/2025/07/21/vibe-coding-replit-production-database/719783>



Announcements

3.7 Sonnet and Claude Code

Feb 24, 2025

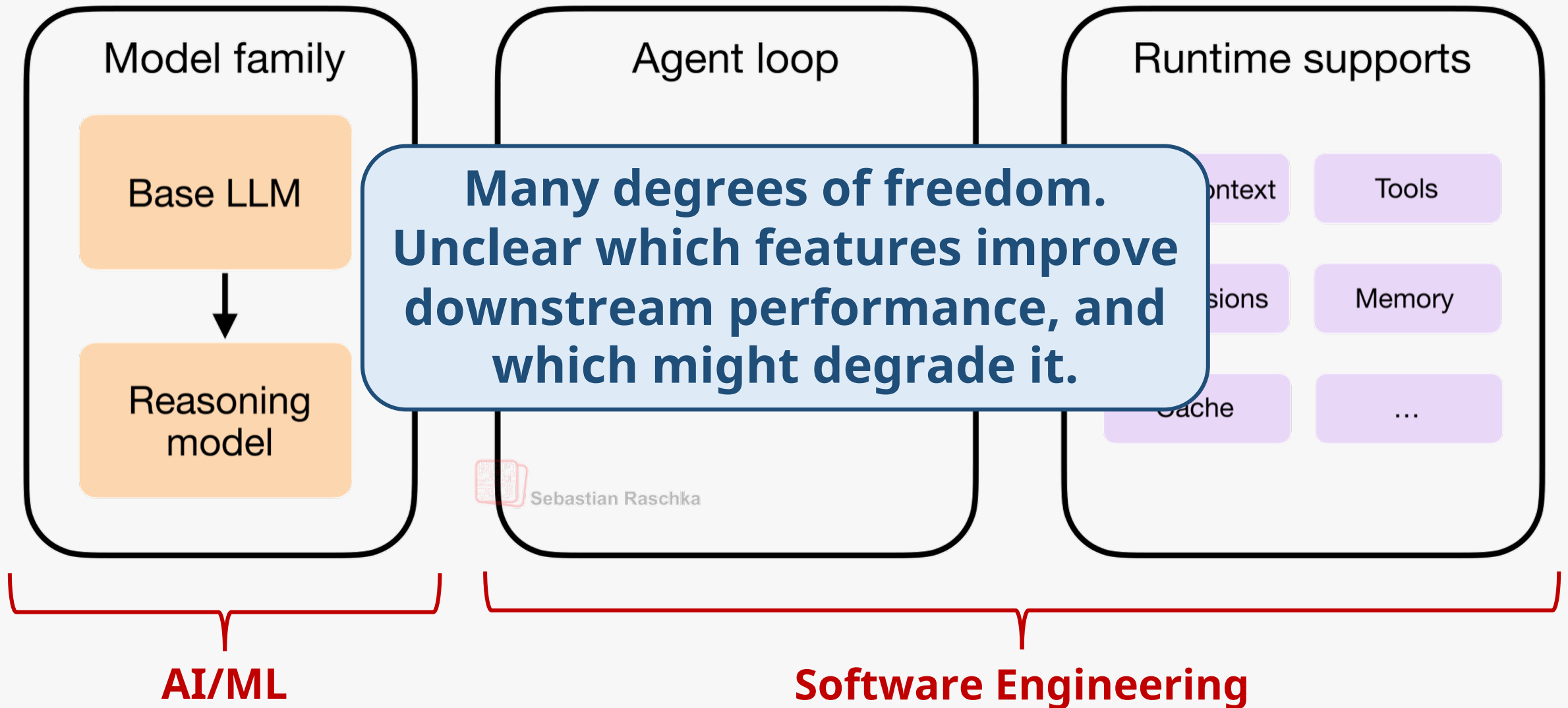
<https://www.anthropic.com/news/claude-3-7-sonnet>

Harness design for long-running application development

Harness design is key to performance at the frontier of agentic coding. Here's how we pushed Claude further in frontend design and long-running autonomous software engineering.

[anthropic.com/engineering/harness-design-long-running-apps](https://www.anthropic.com/engineering/harness-design-long-running-apps)

Agentic Coding Assistants



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

Harness Engineering for 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://arxiv.org/pdf/2602.14690>

Table 1: Overview of repository-level configuration mechanisms across agentic AI coding tools. Each cell lists the repository file(s) or directory implementing that mechanism; “–” indicates that the mechanism was not available when we checked.

Mechanism	Description	Claude	Copilot	Codex	Cursor	Gemini
CONTEXT FILES	Markdown file loaded into the context each session.	CLAUDE.md	.github/ { copilot- instructions.md instructions/*.md } ^a	AGENTS.md, AGENTS.override. md	AGENTS.md, . cursorrules ^c	GEMINI.md
SETTINGS	JSON/TOML config for project-level tool behavior.	.claude/ settings (local)?.json	– ^b	.codex/ config. toml	.cursor/ cli.json	.gemini/{ settings.json config.yaml }
SKILLS	Reusable knowledge and invocable workflows.	.claude/skills/	.github/skills/	.codex/skills/	.cursor/skills/	.gemini/skills/
SUBAGENTS	Specialized agents that operate in parallel to the central agent loop, in their own context.	.claude/agents/	.github/agents/	–	.cursor/agents/	–
COMMANDS	User-triggered shortcuts for predefined prompts.	.claude/ commands/	–	–	.cursor/ commands/	.gemini/ commands/
HOOKS	Scripts executed at specific agent lifecycle points.	.claude/ settings.json	.github/hooks/*.json	–	.cursor/ hooks. json	.gemini/ settings.json
RULES	System-level instructions to control agent behavior.	–	–	.codex/rules/	.cursor/rules/	–
MCP	External tool or data connections via the Model Context Protocol.	.mcp.json	– ^b	.codex/ config. toml	.cursor/ mcp.json	.gemini/ settings.json

^a Copilot also supports CLAUDE.md, AGENTS.md, and GEMINI.md; ^b Configured via the GitHub web UI, not via files in the project repository;

^c Cursor deprecated .cursorrules and now suggests using AGENTS.md instead.

Loop Engineering: Building Blocks, Adoption, and Impact

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

Vahram Nersesyan
Heidelberg University
Heidelberg, Germany
vahram.nersesyan@stud.uni-heidelberg.de

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

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

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



<https://arxiv.org/pdf/2608.21884>

Table 1: Four layers of directing AI coding agents [12, 20, 40]. Each layer subsumes the previous one; the harness layer refers to the outer harness.

Layer	Unit of Concern	Example Artifacts
Prompt eng.	One instruction	Prompts, prompt templates
Context eng.	One model call	AGENTS.md, CLAUDE.md, rules
Harness eng.	One agent run	Skills, subagents, hooks, MCP config.
Loop eng.	Recurring runs	Schedules, stop conditions, state files, run logs, budgets, verifier agents

Table 2: Loop engineering mechanisms and their traceability from GitHub data. The first eight rows are the building blocks named in the practitioner sources (Section 3); the last two are traces of a loop’s operation. ‘Detected’: number of repositories with matches ($n = 36,645$); ‘TP’ (true positives): number of repositories confirmed by manual inspection.

Mechanism	Role in the Loop	Example Repository Artifacts	Traceability from GitHub Data	Detected	TP
Triggering & scheduling	Starts runs on a cadence or event	.github/workflows/*.yaml with cron schedules invoking agent CLIs	CI schedules visible; tool-native schedulers (/loop) and local cron jobs leave no trace	253	242
Goal & stop conditions	Specifies when a run is complete, in a form a machine can check	Goal definitions in skills, commands, or workflow scripts	Visible only when encoded in committed prompts or scripts	0	0
State & memory	Persists findings and progress across runs	STATE.md, LOOP.md, plan/progress files, run logs	File presence, plus Git history revealing cadence, authorship, and append-only updates	2	0
Skills & intent	Encodes durable project knowledge read on every run	.claude/skills/, SKILL.md	Presence detectable; runtime usage invisible [12]	1	1
Parallel isolation	Lets concurrent agents work without collisions	Git worktrees (local); branch naming conventions	Worktrees are not pushed; agent branches indicate general agent use, not loop isolation	—	—
Verification (“maker/checker”)	Checks the stop condition independently of the implementer; can reject	Verifier subagent definitions (.claude/agents/), CI gates, protected branches	Definitions and CI results visible; agent-side verdicts invisible unless logged	0	0
Human oversight & escalation	Gates risky actions; receives escalations	Pull-request reviews, CODEOWNERS, escalation issues, gate rules in loop docs	Review traces and documented gates visible; interactive supervision invisible	15	13
Budgets & pause controls	Caps spend per run; restricts what an unattended run may change	loop-budget.md, loop-constraints.md, pause labels	Declarations visible when committed; enforcement invisible	0	0
Outcomes & provenance	Identifies what the loop produced	Bot-authored commits/PRs, Co-Author-By trailers	Attribution conventions inconsistent across tools and teams [42, 45]	30	—
Cost	Tokens and money actually consumed	Run logs with token estimates	Rarely committed; observed logs hold template values, not measurements [18]	0	0

RQ3 *How does the degree of loop autonomy affect development outcomes and developer effort?*



 empirical-software.engineering

Prof. Dr. Sebastian Baltes