



Software Engineering Career in the AI-First Era

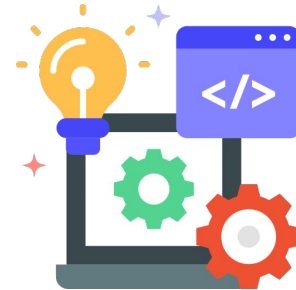
Pittawat Taveekitworachai
Research Scientist, Typhoon, DataX

SIT Hackathon 2026
29 May 2026

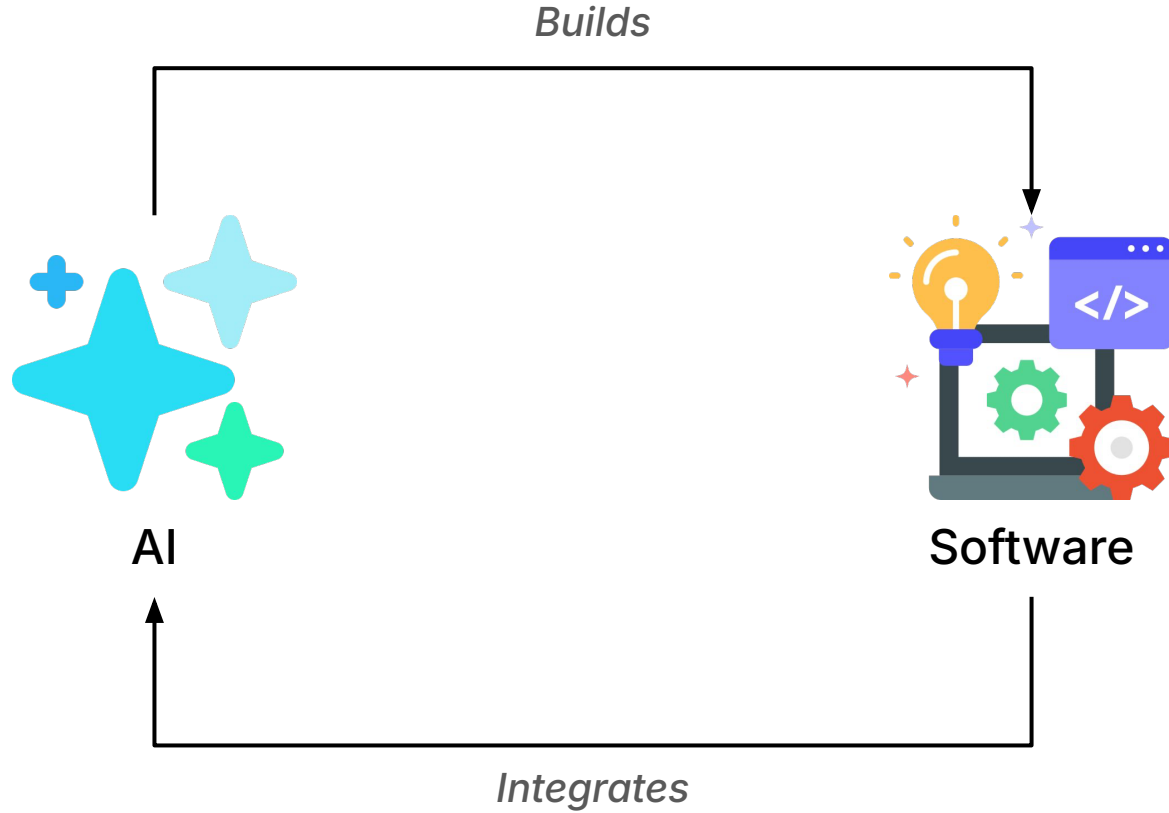
AI



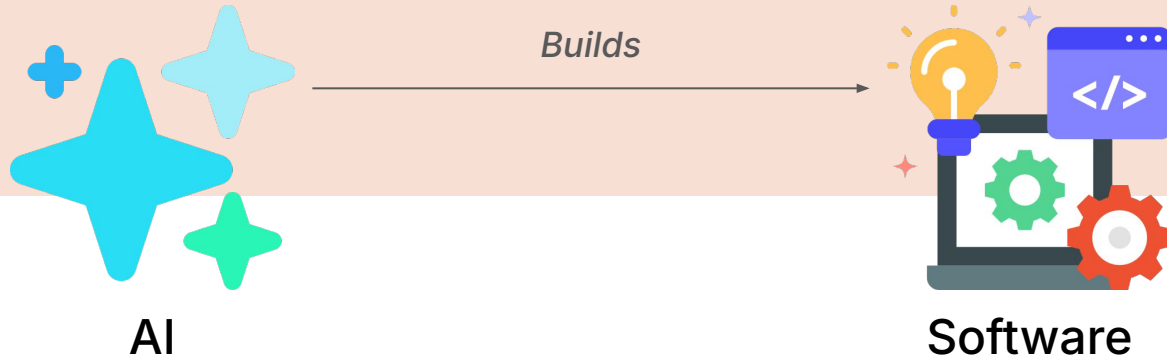
AI



Software

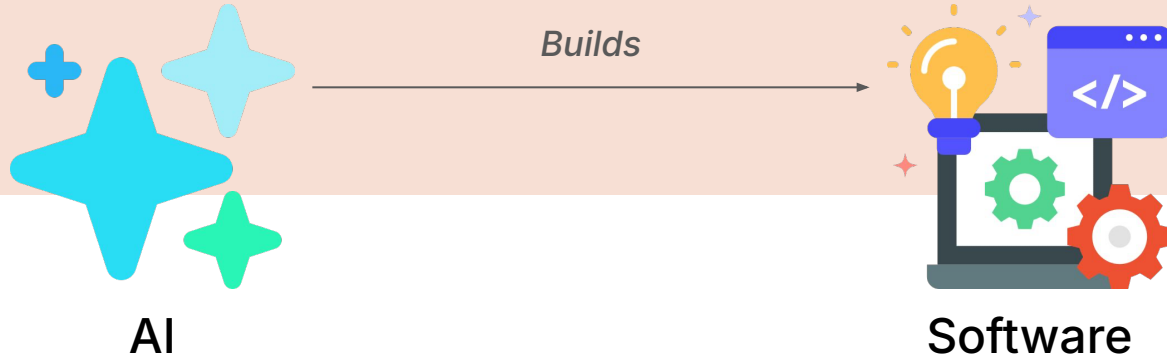


How Does AI Affect Software Development?



We will focus on this one!

How Does AI Affect Software Development?



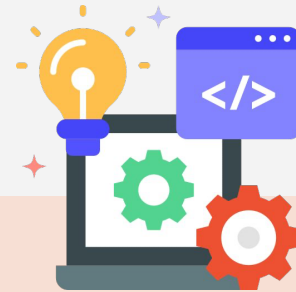
PART I How Does AI Affect Software Development?



AI



Integrates



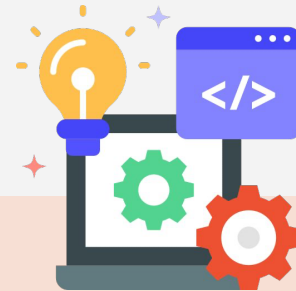
Software

How AI Changes the Shape of Software?

PART I How Does AI Affect Software Development?



AI



Software

Integrates

I will give you some sneak peak

How AI Changes the Shape of Software?

PART I

How Does AI Affect Software Development?

What is the current definition of a software developer?

Vibe Coding



Andrej Karpathy
Cofounder of OpenAI



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.

6:17 AM · Feb 3, 2025 · 7.1M Views

Vibe Coding: Turning Instructions Into Code



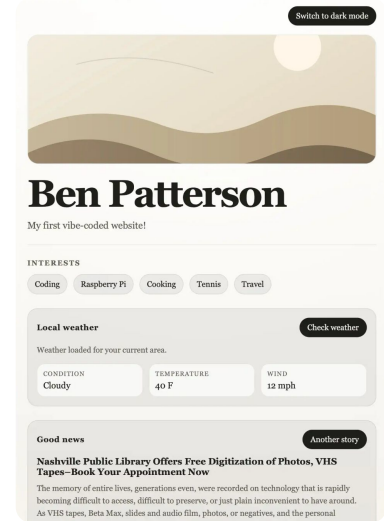
Instruction



Codex
Claude Code
Google Antigravity
CURSOR **Windsurf**

Coding Harness

(Agentic Development Environment; ADE)



Updated programs

Poll: Which Coding Harness Are You Using?



Join at
slido.com
#2653 229

Which coding harness are you using?

Anthropic's Claude Code

0%

OpenAI's Codex

0%

Google's Antigravity

0%

Cursor

0%

Windsurf

0%

GitHub Copilot

0%

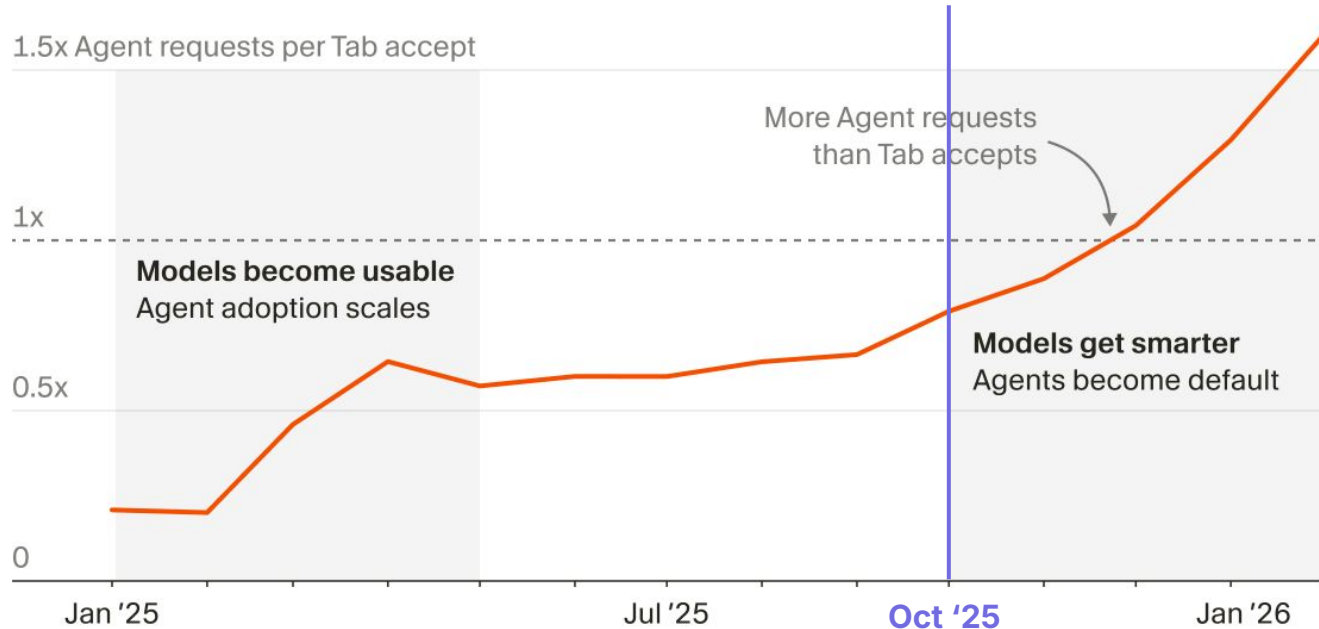
JetBrains AI Assistant

0%

Something else

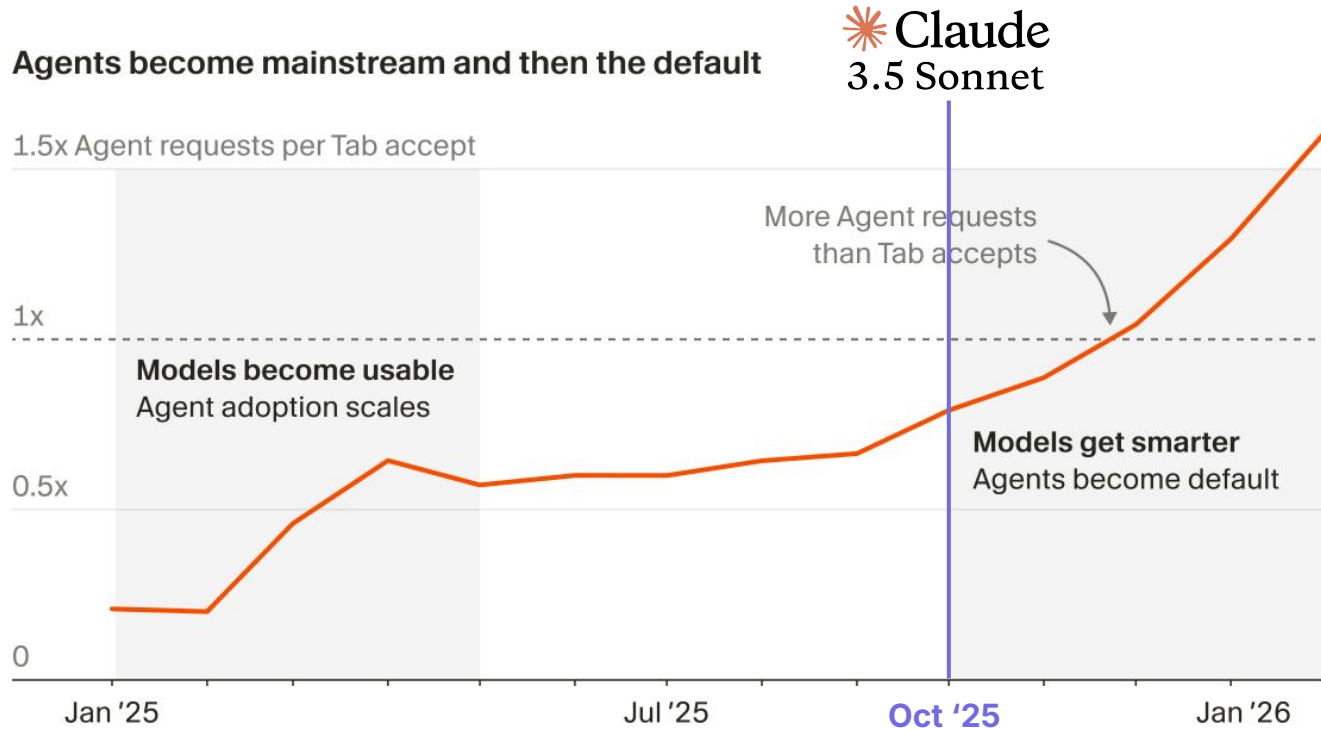
October 2025 Is the Turning Point

Agents become mainstream and then the default



Source: [Cursor](#)

October 2025 Is the Turning Point



Source: [Cursor](#)

Big Tech Embraces AI-Generated Code

Google

75%

of all new code is
AI-generated

Source: [Google Cloud](#)

ANTHROPIC

100%

of all new code from
Head of Claude Code
is AI-generated

Source: [Boris Cherny](#)

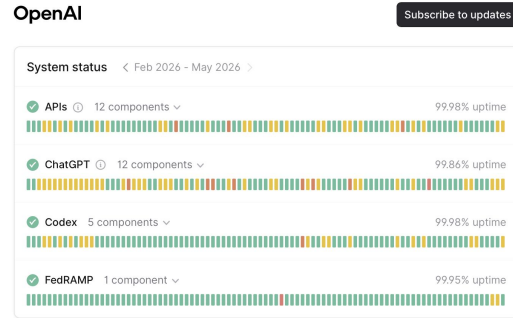
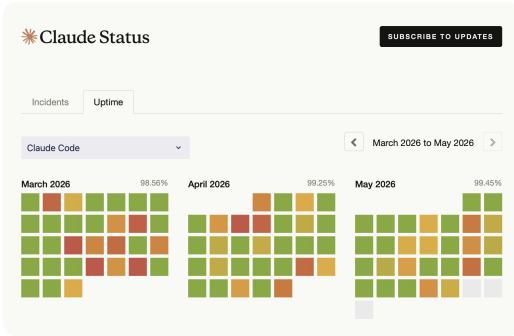
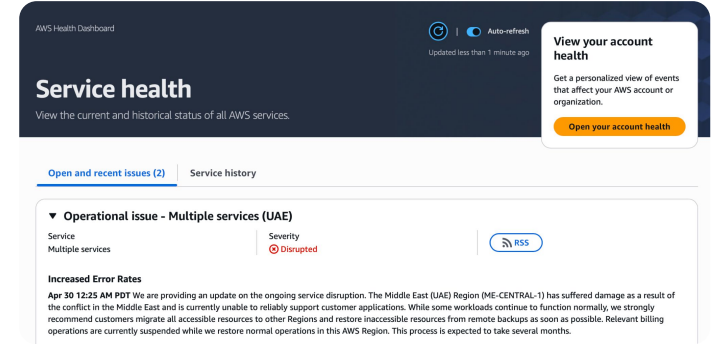
Science
JOURNALS 

29%

of Python functions in
US were produced with
substantial AI support

Source: [Science](#)

We Have More Incidents From Big Tech Than Ever

AWS Health Dashboard Auto-refresh

Service health

View the current and historical status of all AWS services.

[Open and recent issues \(2\)](#) | [Service history](#)

Operational issue - Multiple services (UAE)

Service: Multiple services | Severity: Disrupted | [RSS](#)

Increased Error Rates

Apr 30 12:25 AM PDT We are providing an update on the ongoing service disruption. The Middle East (UAE) Region (ME-CENTRAL-1) has suffered damage as a result of the conflict in the Middle East and is currently unable to reliably support customer applications. While some workloads continue to function normally, we strongly recommend customers migrate all accessible resources to other Regions and restore inaccessible resources from remote backups as soon as possible. Relevant billing operations are currently suspended while we restore normal operations in this AWS Region. This process is expected to take several months.



Cloudflare outage on February 20, 2026

2026-02-21

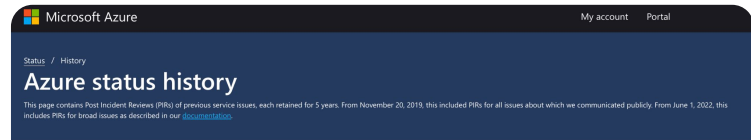


David Tuber



Dzevad Trumic

9 min read



Microsoft Azure My account Portal

Status / History

Azure status history

This page contains Post Incident Reviews (PIR) of previous service issues, each retained for 5 years. From November 20, 2019, this included PIRs for all issues about which we communicated publicly. From June 1, 2022, this included PIRs for broad issues as described in our [documentation](#).

April 2026

Post Incident Review (PIR) – Multiple services – Control plane issues in East US

Tracking ID: 5GP8-W0G

Watch our 'Azure Incident Retrospective' video about this incident: <https://aka.ms/air/5GP8-W0G>

We Have Already Passed the Point of No Return

Sorry to say this, but this is how the world will work from now on

At Least This Isn't the First Time It Has Happened



Terminal



Desktop



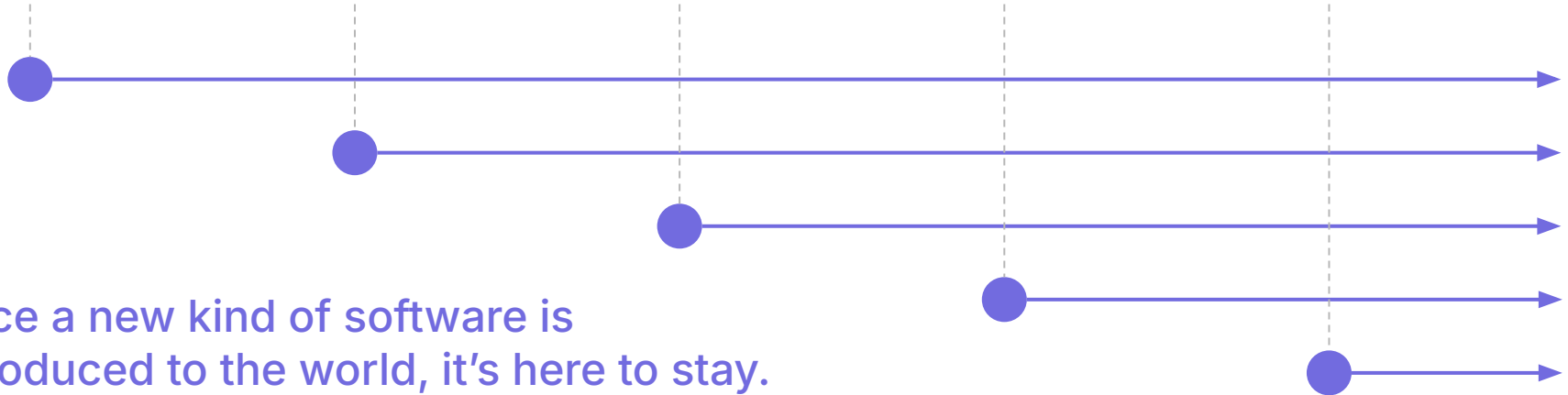
Web



Mobile

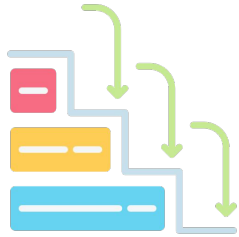


GenAI

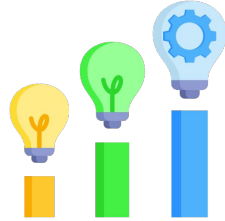


Once a new kind of software is introduced to the world, it's here to stay.

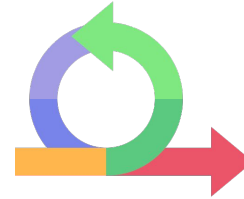
Same Goes for Software Development Methodologies



Waterfall



Incremental



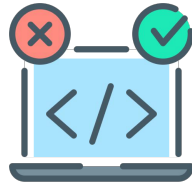
Agile



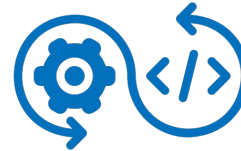
Spiral



Extreme Programming



Test-Driven Development



DevOps



Spec-driven development

Poll: What do you think is the main task of a software engineer?



What do you think is the main task of a software engineer?

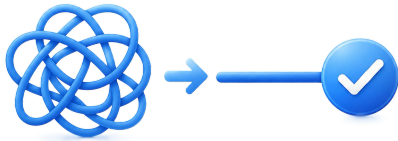
Join at
slido.com
#2653 229

Value Translator

Value + Translator

Value

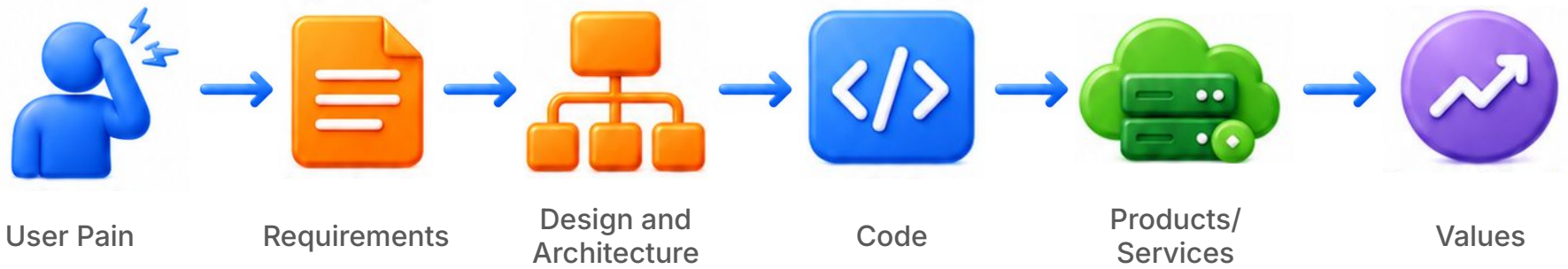
(n.) Making something easier, safer, faster, or more accessible for someone



Translator



Translator



Programming != Software Engineering

Programming



Design



Code

Software Engineering

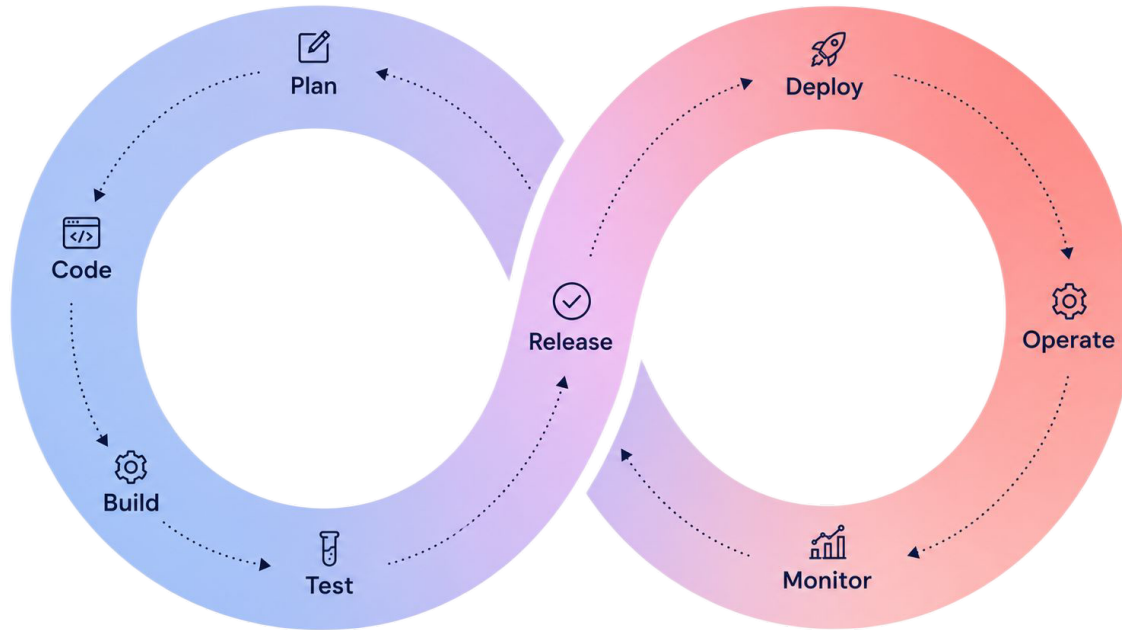


User Pain

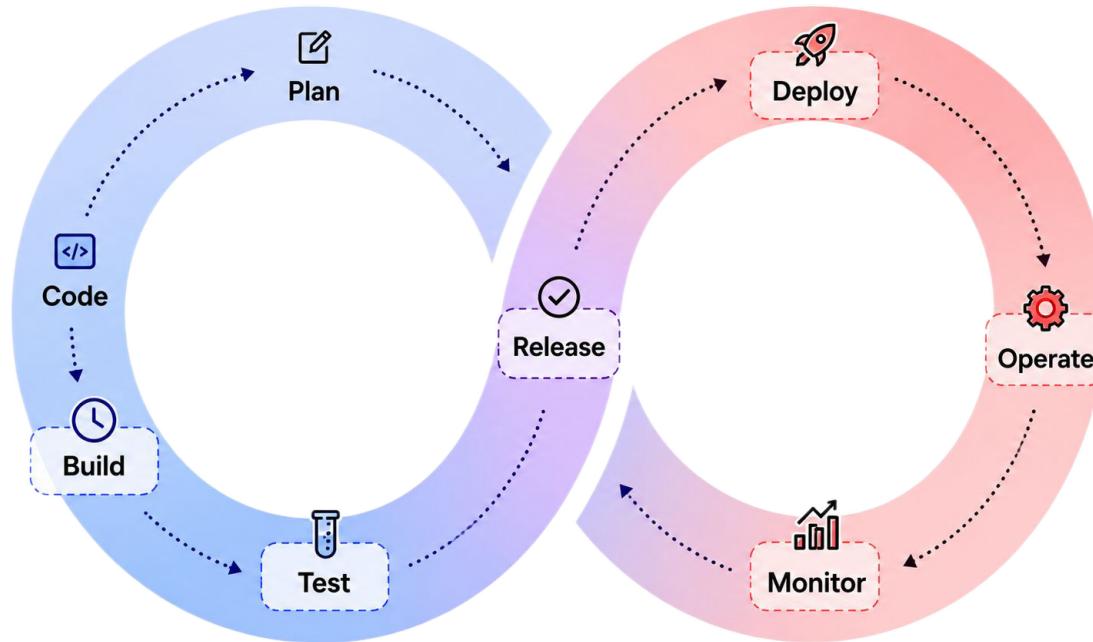


Values

Software Development LifeCycle (SDLC)



Coding Is Never the Bottleneck in the SDLC



Traditional T-Shaped Developer

Board Foundation of Knowledge

Core Software
Engineering

Traditional T-Shaped Developer

Board Foundation of Knowledge

This is not enough anymore!

Software Engineering

How do we thrive in this **new world?**

5 Patterns That Top-Performing Engineers at Google Follow

1. Shift Up

Think more about *Why*
(*Context*) > *What* (*Code*)

2. Shifting Left

More time & effort on
Intent > *Implementation*

3. The Environment

Design robust harness for
Verified output > *Faster code*

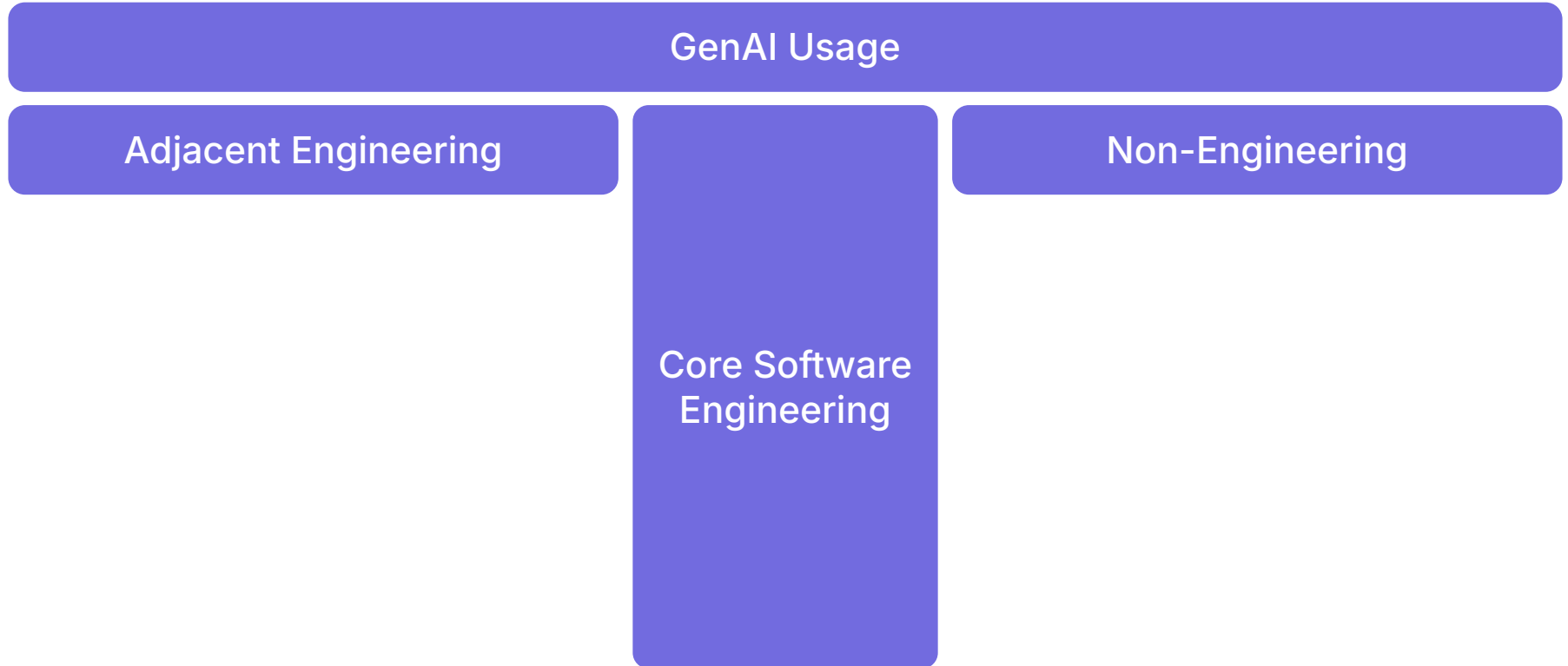
4. The Workforce

Dissolving boundaries, focusing
on *Capability* > *Job title*

5. New Values

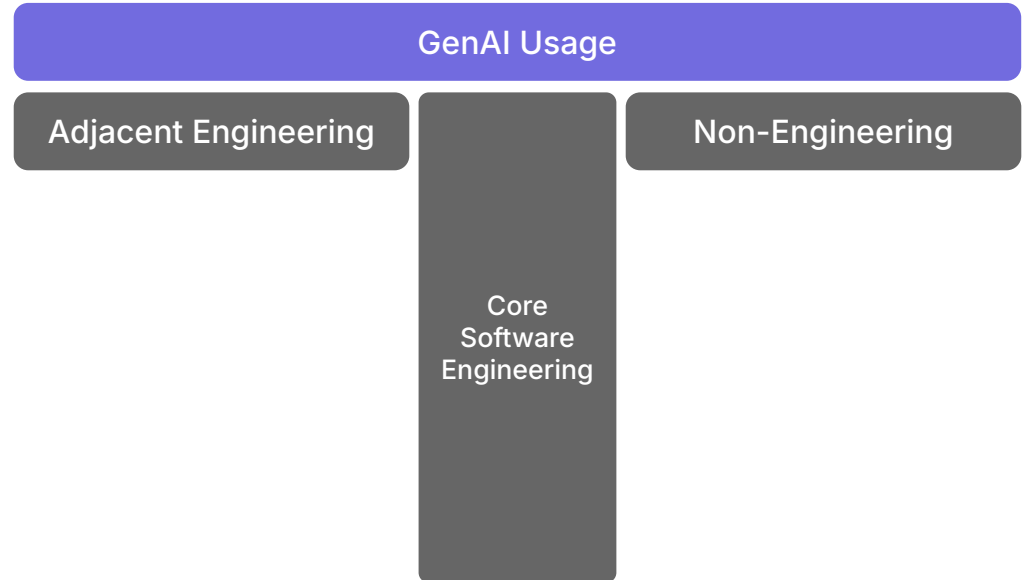
Comfortable with *Intelligent*
failure > *The status quo*

Modern T-Shaped Developer



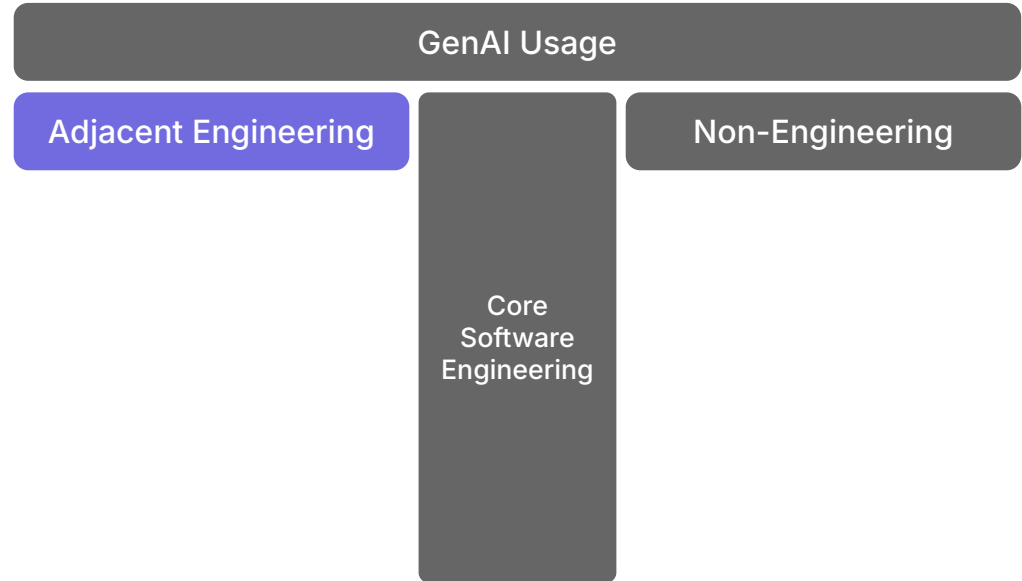
GenAI Usage

- Understand the **model capabilities & constraints**
- Delegate execution, **maintain judgements**
- Be an **"orchestrator"**, control **"context"**



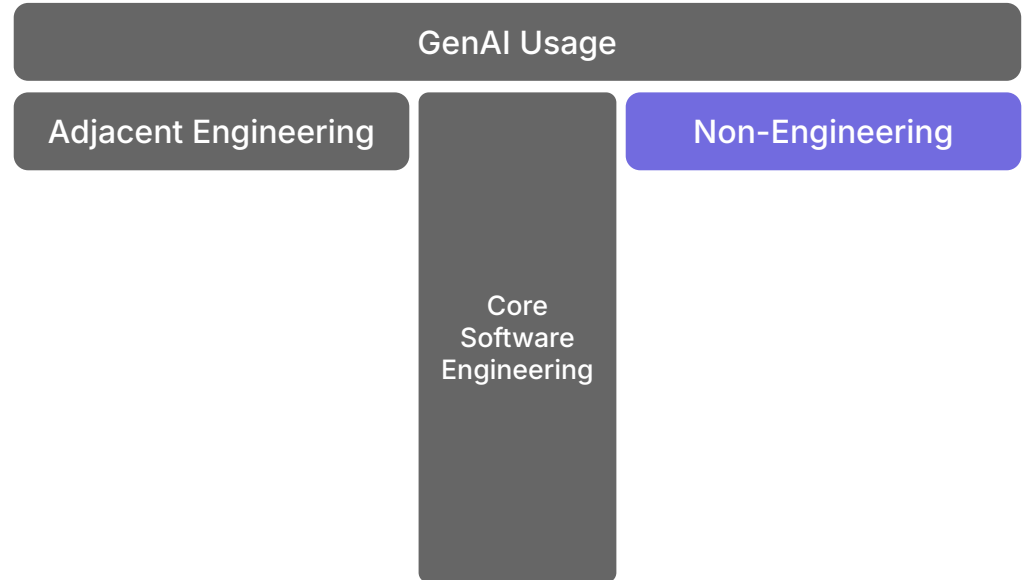
Adjacent Engineering

Understand **non-technical requirements** and **business operations**, such as compliance, security, infrastructure, risk management, and observability



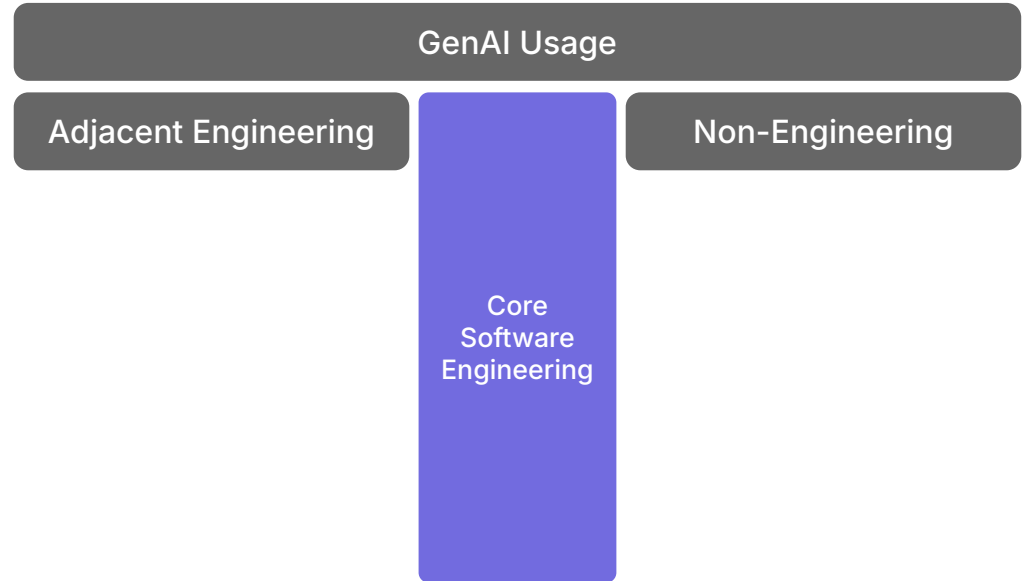
Non-Engineering

- Be a **value translator** between users and the business
- Deepen business **domain knowledge**

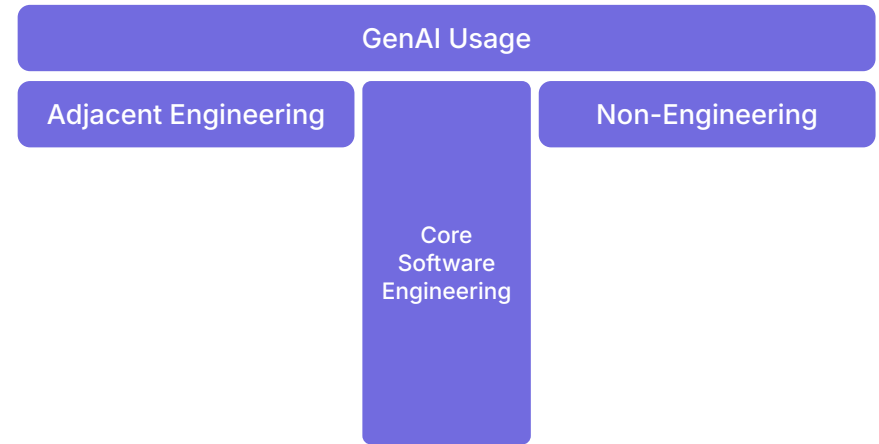


Core Software Engineering

- **Engineering excellence and best practices**, such as software architecture, system design, and guardrails
- **Software craftsmanship**
- **System thinking**



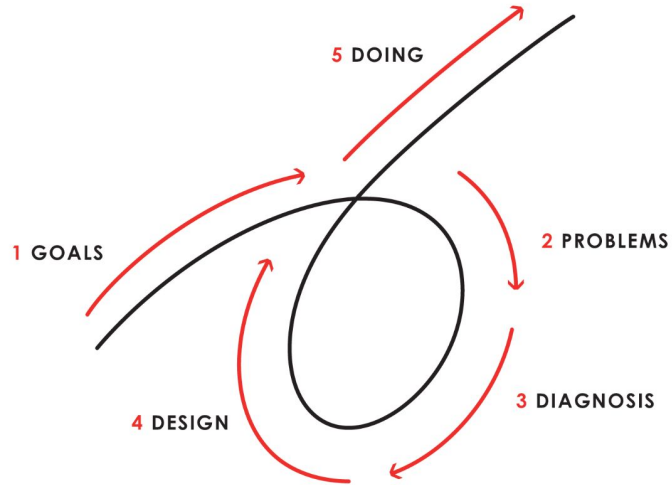
What was previously defined as a **Senior Software Developer** is now expected of a **Junior Software Developer**



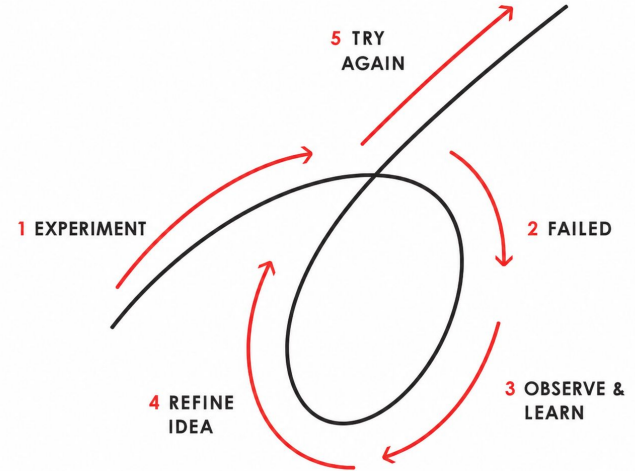
How to Start Practicing **Today**?

Just Use It!

Ray Dalio's 5-Step Process for Life



Apply the same idea to using AI



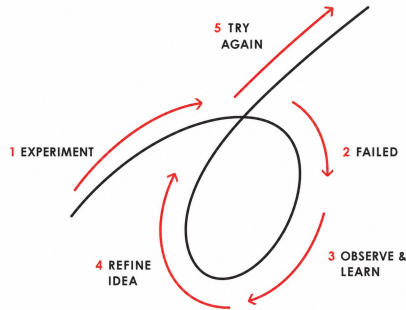
Develop Your Own Taste

The ability to tell **good from great**, and to know **how to** move from one to the other

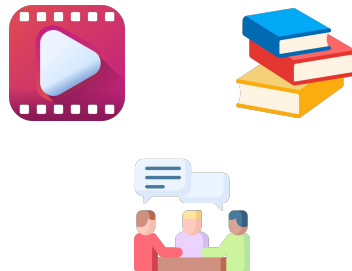
Develop Your Own Taste

The ability to tell **good from great**, and to know **how to** move from one to the other

Learn from **your** experience



Learn from **others'** experience



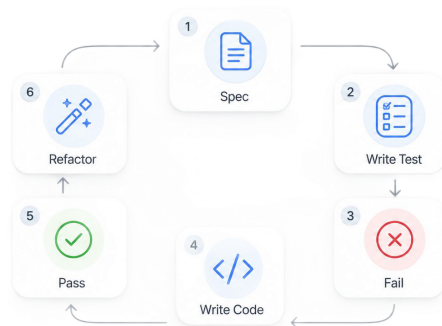
BONUS
Read **Agent Skills!**



Create Rich Feedback Loop

Test-Driven Development

*Tests are signals.
Learn from them.*



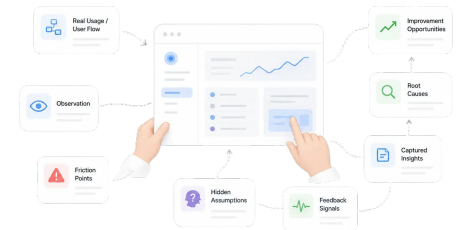
Conduct Post-mortem

*Make bugs your
learning opportunities*



System walkthrough by pen and paper

*Own the understanding
of system design and
data flow*



Practice System Thinking: Spec > Code

Spend more time thinking about spec

Use Plan Mode

☰ Plan

|| plan mode on

Use Agent Skill, e.g., [Grill Me](#)

name	grill-me
description	Interview the user relentlessly about a plan or design until reaching shared understanding, resolving each branch of the decision tree. Use when user wants to stress-test a plan, get grilled on their design, or mentions "grill me".

Interview me relentlessly about every aspect of this plan until we reach a shared understanding. Walk down each branch of the design tree, resolving dependencies between decisions one-by-one. For each question, provide your recommended answer.

Ask the questions one at a time.

If a question can be answered by exploring the codebase, explore the codebase instead.

Avoid vibe coding

The Slop Spiral



Develop your product sense

Design thinking

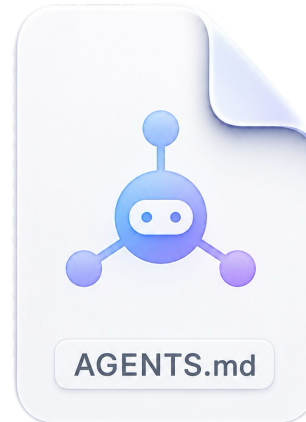
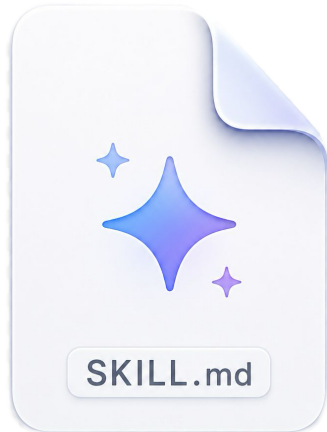


User-centric design



Develop Your Principles, Encode Your Learning

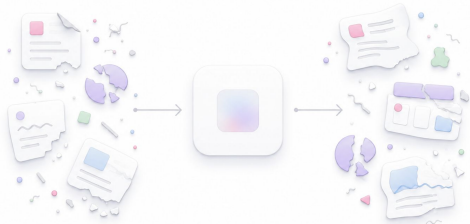
Show your taste through concrete evidence



Some Useful Principles

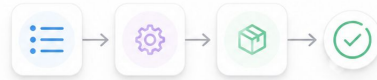
Garbage in, garbage out

Bad prompts, bad plans, bad context, bad specs, ...



Delivery vs Learning Modes

Delivery mode
Output is the **product**

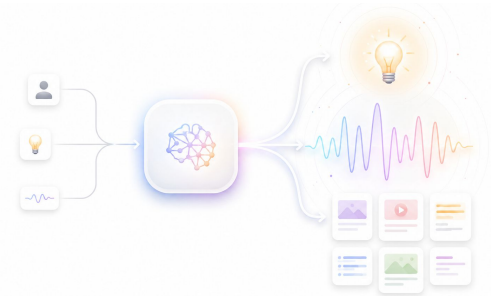


Learning mode
Output is the **byproduct**



AI is an Amplifier

*It amplifies everything:
10x code, 10x slops,
10x features, 10x bugs, ...*



Be Careful of Illusion of Competence



Andrej Karpathy
Cofounder of OpenAI



Andrej Karpathy ✓

@karpathy



This is the the quote I've been citing a lot recently.



kache ✓ @yacineMTB · Feb 4

you can outsource your thinking
but you cannot outsource your understanding

12:43 AM · May 1, 2026 · **2.5M** Views

AI makes **coding easier**, but it makes **everything else harder**

For example, learning, system design, verification, and responsibility

PART II

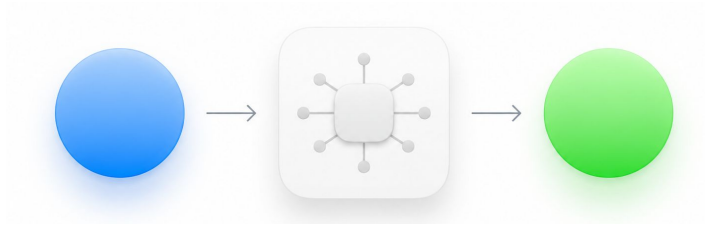
How AI Changes the Shape of Software?

How can AI unlock new software features that were previously impossible to build?

The Nature of Software Has Changed with AI

Traditional Software

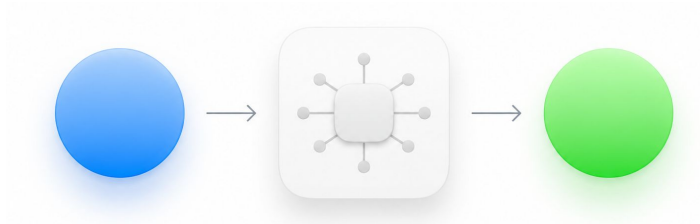
Mostly Deterministic: Same input → Same output



The Nature of Software Has Changed with AI

Traditional Software

Mostly Deterministic: Same input → Same output



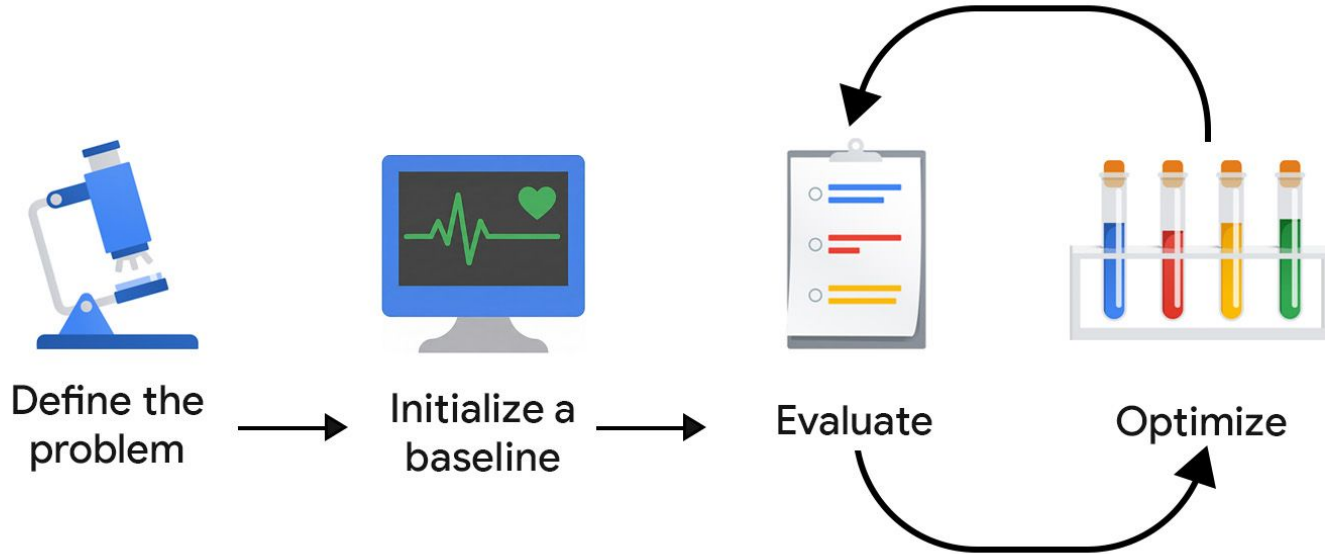
AI-Integrated Software

Non-deterministic: Same input → Different output



Evaluation-Driven Development

Nondeterminism of AI changes how we build software



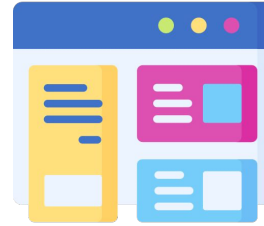
AI Enables Hard-to-Code Features



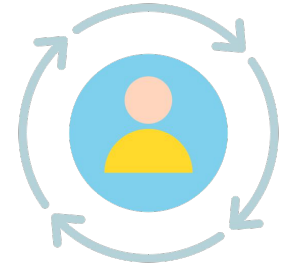
Bidirectional
voice interaction



Natural language
processing



Generative UIs



Hyperpersonalized
features

AI Engineering

Harness Engineering

What components does the model need to complete the task successfully?

Context Engineering

What should the model know at this step?

Prompt Engineering

How do we instruct the model?

Harness Engineering 101



Tools

What can it do?



MCP



API



Server



Context

What does it know?



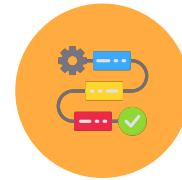
RAG



Memory



Skills



Orchestration

What happens next?



Workflow



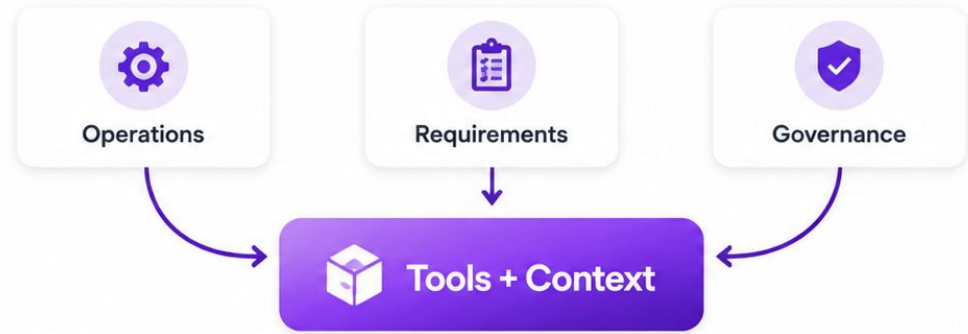
Multi-agent system



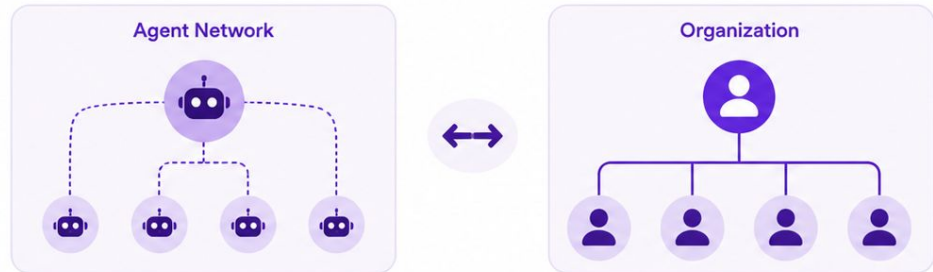
Human-in-the-loop

Harness Design Is Business Design

Tools and context must reflect **business reality**

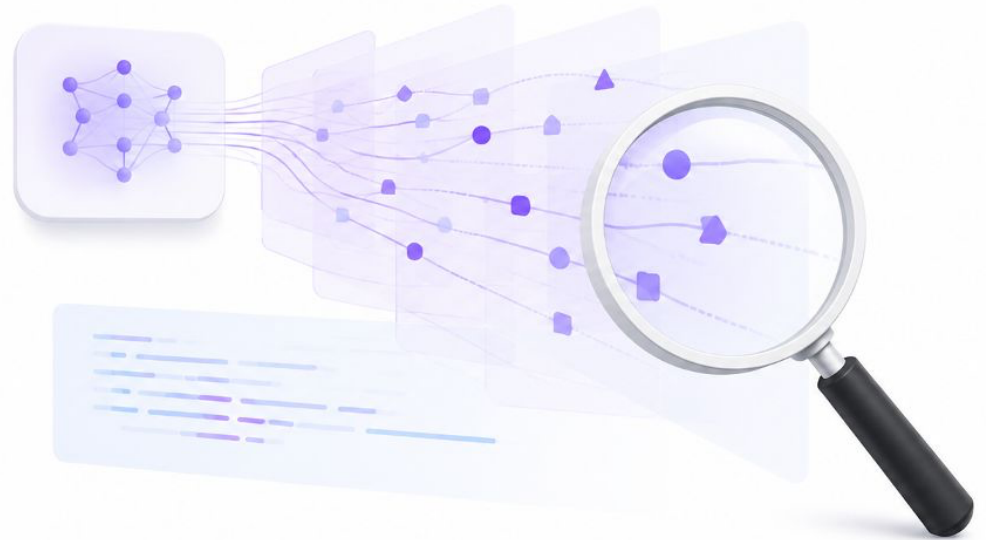


Multi-agent design is an **organization design**



Now We Debug Behavior, Not Just Code

Different Methods, Same Goal
Deliver Value to Users



Takeaways

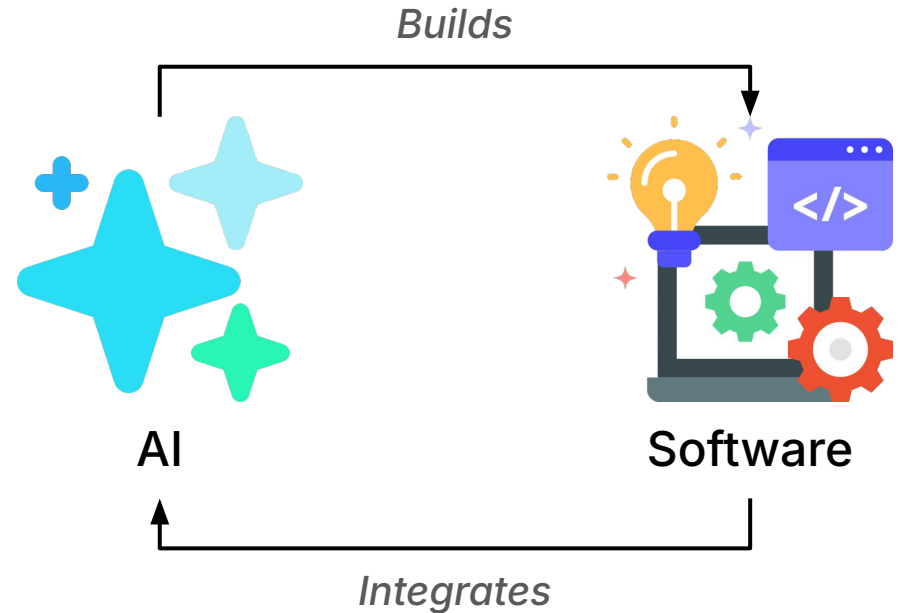
Two Changes Happening At Once

1 How we build software?

2 What software can do?

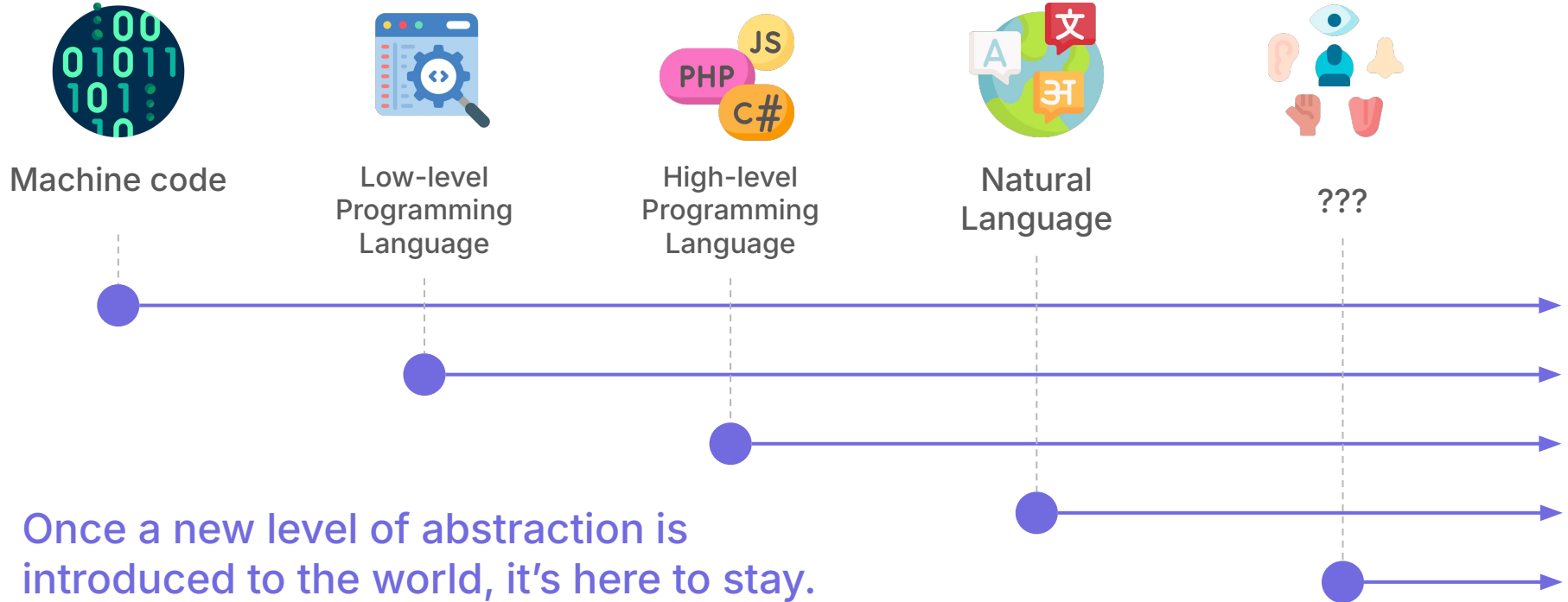


*It's normal to feel
anxious and frustrated*



Software engineering **IS NOT dying**.
It is **moving UP** the abstraction level.

Again, This Isn't the First Time It Has Happened



Once a new level of abstraction is introduced to the world, it's here to stay.

“The best way to predict
the **future** is to **invent it**”



—*Alan Kay*
Computer Scientist