

V I B E C O D I N G

**BUILDING PRODUCTION-GRADE
SOFTWARE WITH GENAI, CHAT,
AGENTS, AND BEYOND**

GENE KIM & STEVE YEGGE

Foreword by Dario Amodei,
CEO and Cofounder of Anthropic

IT Revolution
Independent Publisher Since 2013
Portland, Oregon



25 NW 23rd Pl, Suite 6314
Portland, OR 97210

Copyright © 2025 by Gene Kim and Steve Yegge

All rights reserved. For information about permission to reproduce selections from this book,
write to Permissions, IT Revolution Press, LLC, 25 NW 23rd Pl, Suite 6314, Portland, OR 97210

First Edition

Printed in the United States of America

30 29 28 27 26 25

1 2 3 4 5 6 7 8 9 10

Cover Design by Alana McCann

Book Design by Devon Smith

Library of Congress Control Number: 2025022944

Paperback: 9781966280026

Ebook: 9781966280033

Audio: 9781966280040

For information about special discounts for bulk purchases
or for information on booking authors for an event,
please visit our website at www.ITRevolution.com.

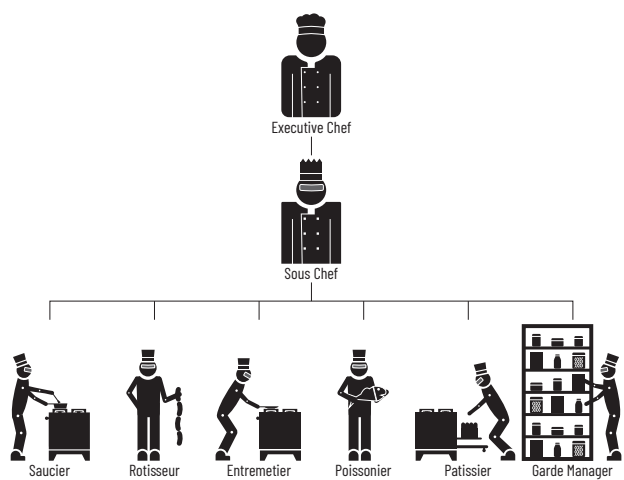


Figure 0.1: The Kitchen Brigade

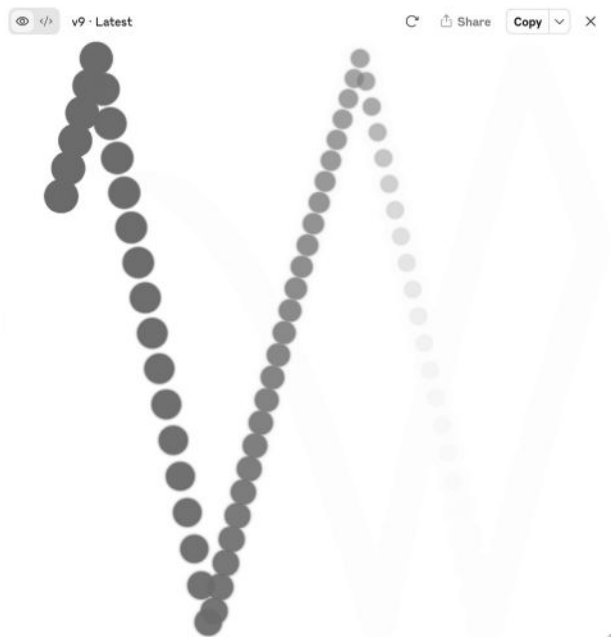


Figure 8.1: Vibe Coded Bouncing Red Ball (Claude)

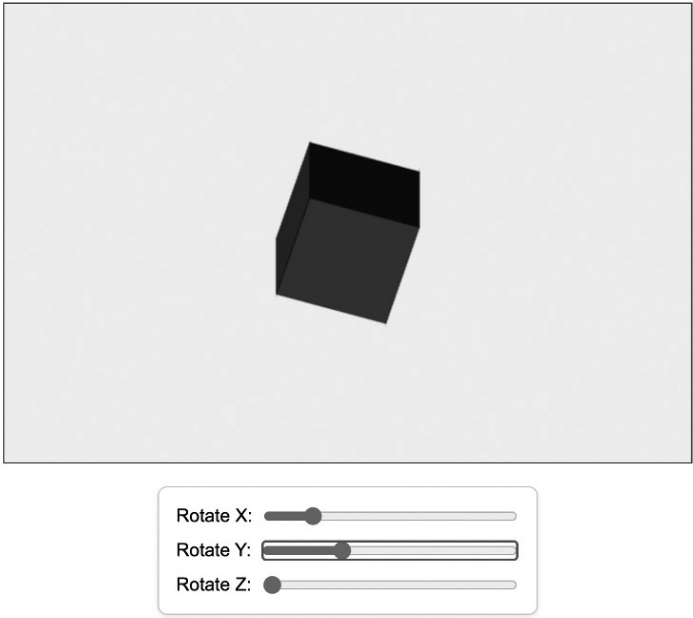


Figure 8.2: Vibe Coded Cube with Two-Colored Lighting (Gemini)

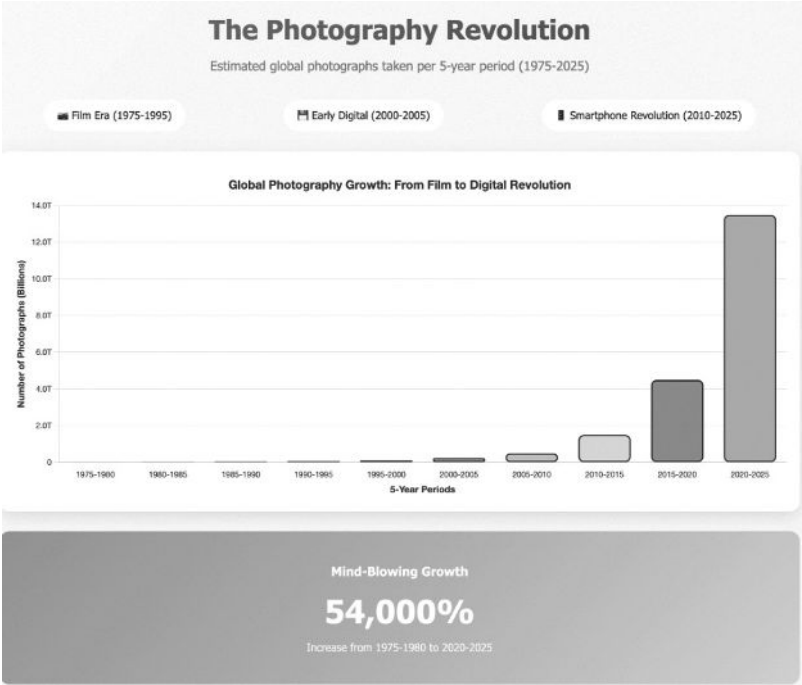


Figure 8.3: The Number of Photographs Taken Annually, Generated Using Vibe Coding (Claude)

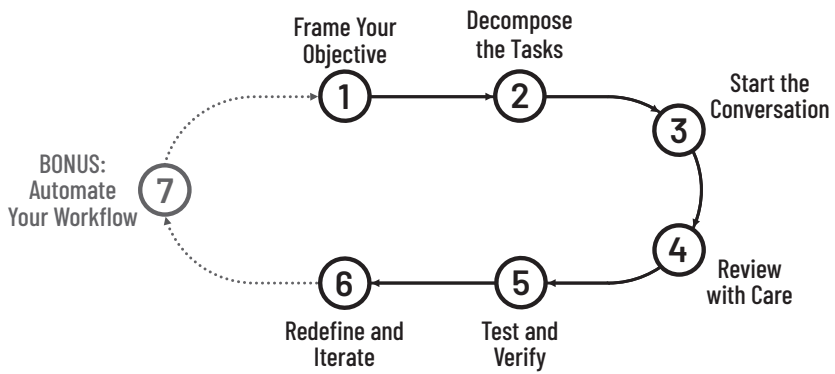


Figure 9.1: The Vibe Coding Loop

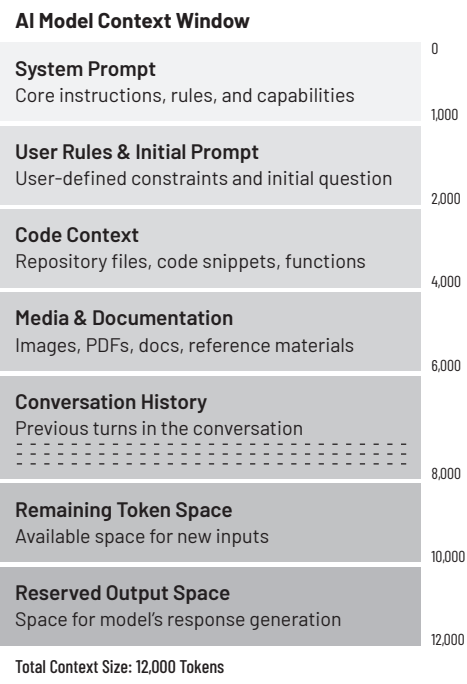


Figure 10.1: A Typical AI Model’s Context Window

```
{
  "messages": [
    {"role": "system", "content": "You are a helpful coding assistant..."},
    {"role": "user", "content": "How do I implement a binary tree in Python?"},
    {"role": "assistant", "content": "Here's how you can implement a binary tree:..."},
    {"role": "user", "content": "Now I'm getting this error: TypeError: 'NoneType'..."}
  ]
}
```

LLM Context Window Filling Up

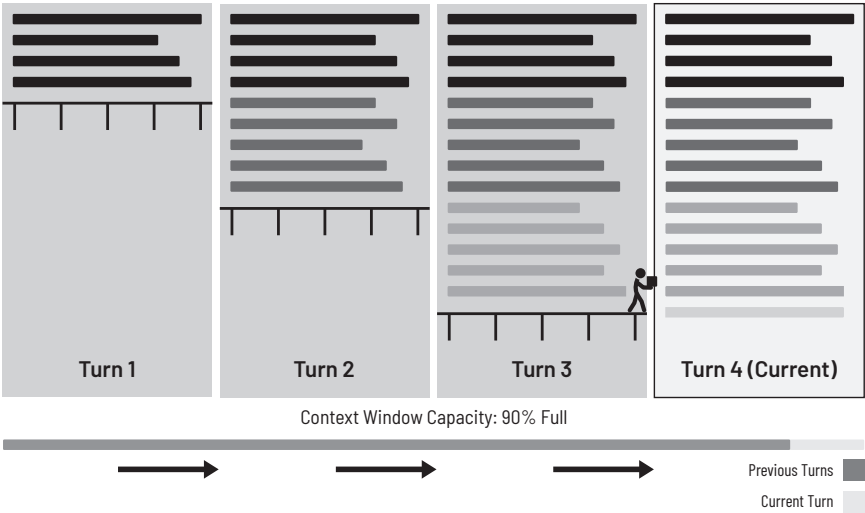


Figure 10.2: LLM Context Window Filling Up with Each Turn

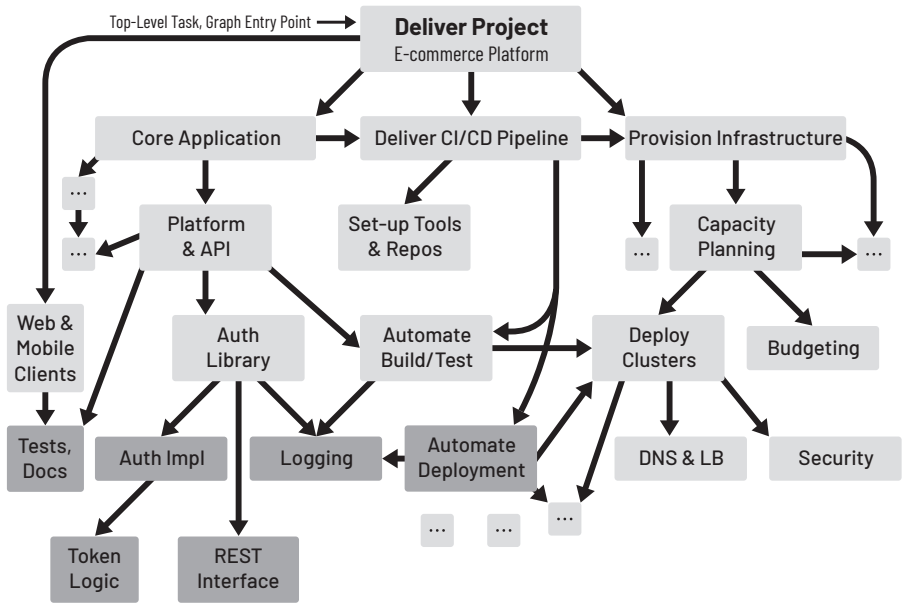


Figure 12.1: Example Large Project Task Graph
with AI Handling Some Leaf Nodes

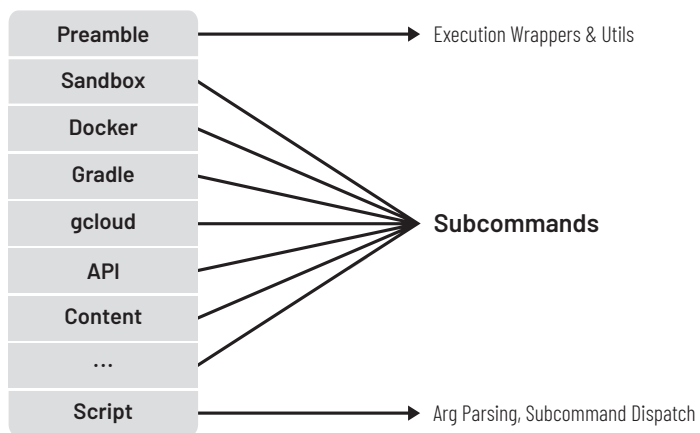


Figure 12.2: Architecture of Steve's Ruby Admin Script

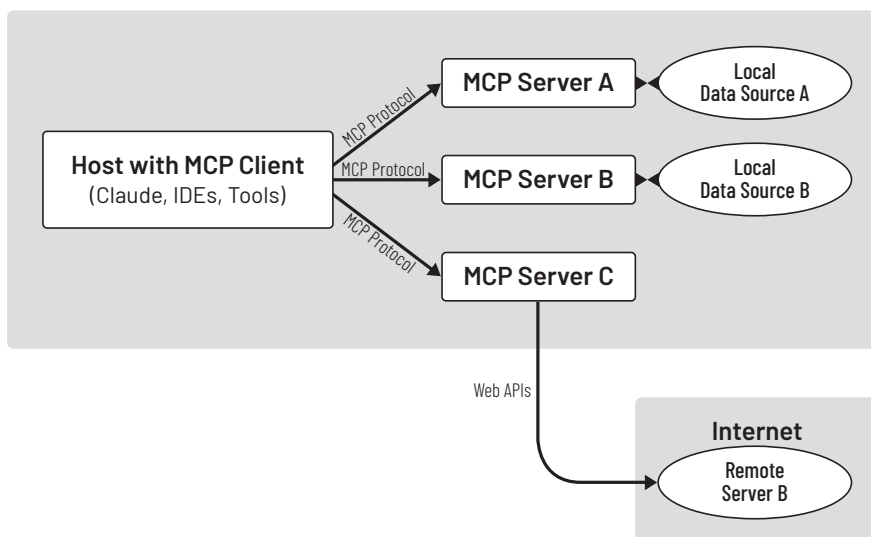


Figure 13.1: MCP-Enabled System

// from AI → MCP server

```
{
  "jsonrpc": "2.0",
  "id": 42, // request-id, which allows for async and parallel RPCs
  "method": "tools/call",
  "params": { "name": "fetch_weather", "arguments":
    {"location": "San Francisco" } }
}
```

The server translates `fetch_weather` into real operations (e.g., API calls to weather services or database queries), then replies:

```
{
  "jsonrpc": "2.0",
  "id": 42, // response-id
  "result": { "ok": true }
}
```

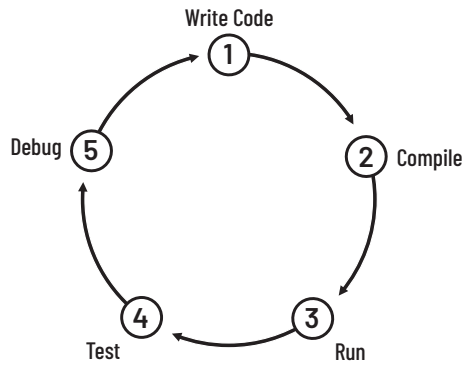


Figure 14.1: Traditional Developer Loop

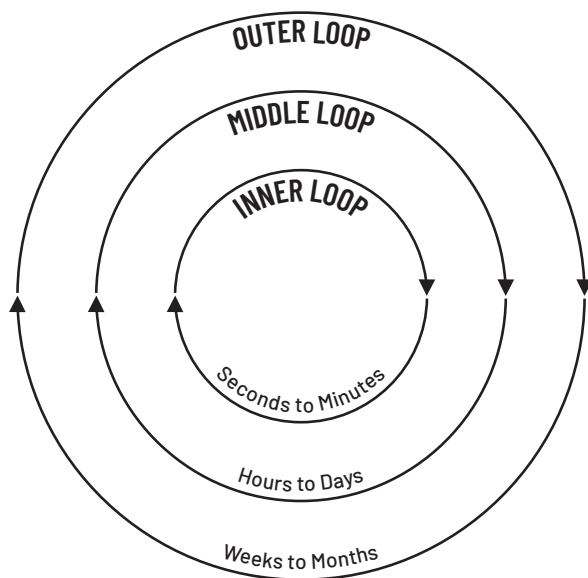


Figure 14.2: The Three Developer Loop Timescales

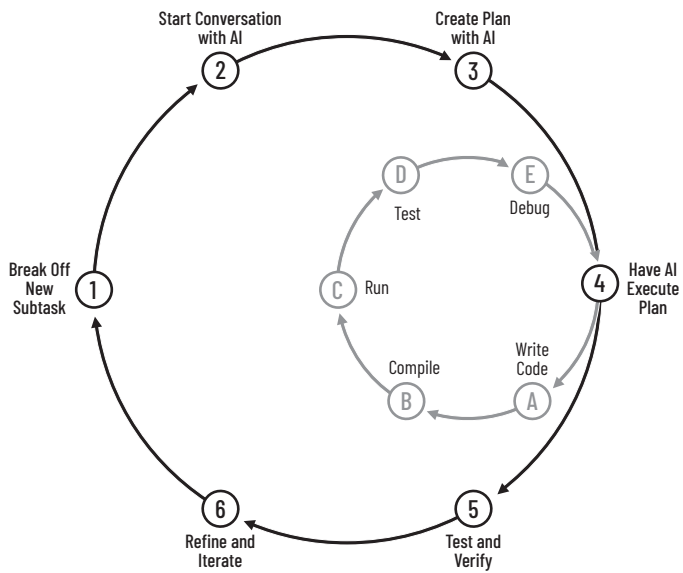


Figure 14.3: The Vibe Coding Developer Loop

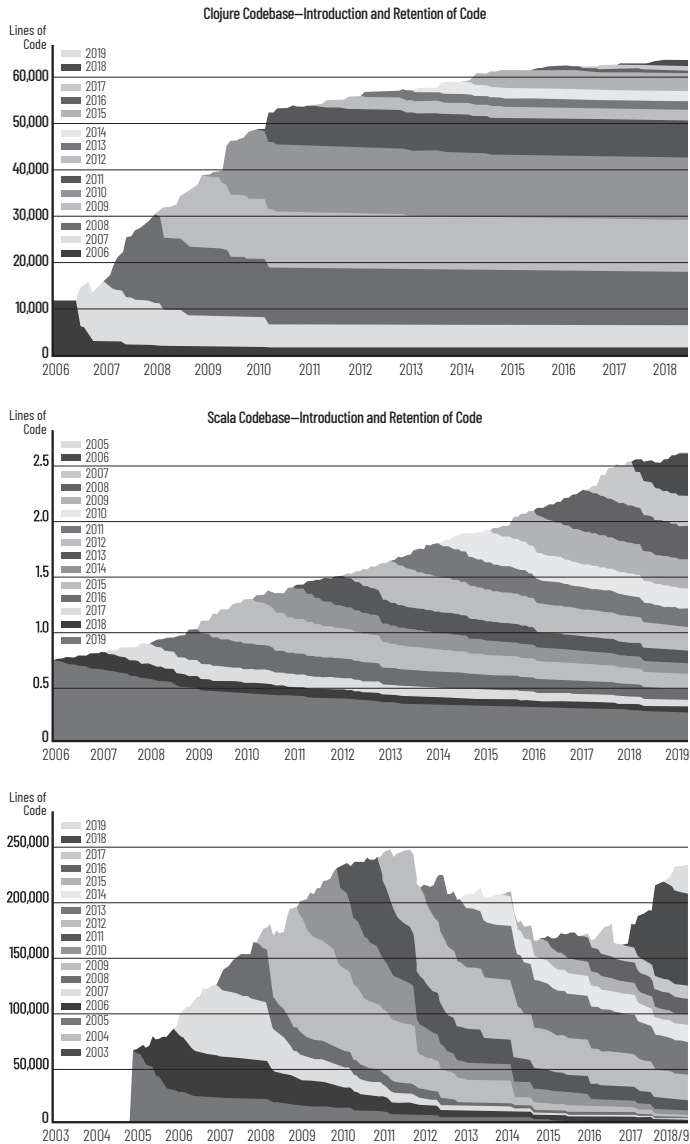


Figure 16.1: Code Survival Graphs for Clojure and Linux (High) and Scala (Low)

Source: Rich Hickey, “A History of Clojure.” Proceedings of the ACM on Programming Languages, 2020. <https://dl.acm.org/doi/pdf/10.1145/3386321>; SRC-d. “Hercules: Fast, Insightful and Highly Customizable Git History Analysis.” GitHub Repository, 2023. <https://github.com/src-d/hercules>.

Table 16.1: Vibe Coding Testing Strategies

		High Risk	Low Risk
Know Tech	Well	Deep white-box and black-box (exhaustive testing).	Light white-box, light black-box.
	Poorly	Deep black-box, light human white-box (code spot-checks are all you can do), heavy AI white-box.	Black box only (let it write some tests, then verify that the overall outputs “look right”).

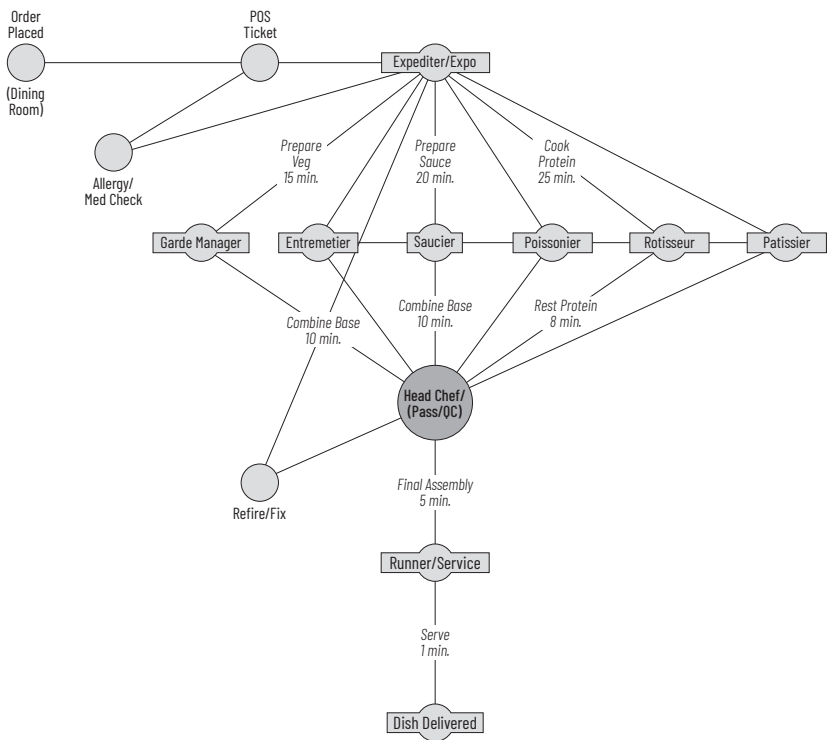


Figure 17.1: Parallelizing Kitchen Work with a Task Graph

GLOSSARY OF COMMON TERMS

Agent: An AI system designed to perform tasks autonomously with directed intent, often handling multiple subtasks and steps. Unlike LLMs, agents maintain state and can work independently toward specific goals.

API: Application programming interface.

API Key: This is your ticket for API access. It's a sequence of characters generated by the API provider and should be kept secret.

ChatGPT: A conversational AI model developed by OpenAI, based on the GPT (generative pre-trained transformer) family of models. Available through both web interface and API, it's widely used for code generation and explanation.

CHOP (Chat-Oriented Programming): A programming methodology where developers write code through natural language conversations with AI assistants, rather than writing code directly by hand.

Claude: An AI assistant developed by Anthropic, known for strong coding capabilities and detailed technical explanations. Available in several versions with varying capabilities and performance characteristics.

Code AI: An umbrella term encompassing all the ways people use Generative AI and LLMs to help their company's engineers, including Chat-Oriented Programming (CHOP), API-based automation, AI agents, assistants with agentic behavior, LLM-produced code indexes, and many other approaches that people are using to bring AI to software engineering.

Coding Assistant: A specialized AI tool designed to integrate directly into development environments (like VS Code or other IDEs), offering context-aware code suggestions, explanations, and modifications.

Context: In AI programming, the background information provided to AI about your code, requirements, and constraints. This includes code snippets, documentation, error messages, and previous conversation history that helps AI understand the current task.

Context Window: The amount of text an AI model can consider at once

when generating responses, typically measured in tokens. This includes both the conversation history and any provided code or documentation.

Dynamic Context: Temporary, task-specific information that changes frequently during development, such as current problem descriptions, intermediate code versions, and debugging information.

Foundation Model: A large AI model trained on vast amounts of data that serves as the base for various AI applications. Examples include GPT-4, Claude, and Llama.

Gemini: Google's family of AI models, designed to work with multiple types of input including text and images. Available in different sizes, offering various trade-offs between capability and speed.

Generative AI (GenAI): AI systems that can create new content—including code, text, images, and more—based on training data and user prompts. Unlike traditional AI that focuses on classification or prediction, GenAI can produce novel outputs that follow patterns learned from its training. In software development, GenAI tools like LLMs can generate code, documentation, tests, and other artifacts while engaging in natural language dialogue with developers.

Hallucination: When an AI model generates incorrect or fabricated information, such as referring to non-existent functions or APIs.

Inference Provider: A service or platform that hosts and runs AI models, handling the computational resources needed for AI operations. Examples include AWS Bedrock and Azure OpenAI Service.

Leaf Node: In the task graph model, a small, independent task that can be completed without breaking it down further. In vibe coding, these are typically tasks that AI can accelerate by 10x compared to manual implementation.

LLAMA (Large Language Model Meta AI): A family of open-source language models developed by Meta (formerly Facebook). These models can be run locally and have spawned numerous derivatives and fine-tuned versions.

LLM (Large Language Model): An AI system trained on vast amounts of text data that can understand and generate human-like text, including code. Examples include GPT-4 and Claude.

Multi-Turn Conversation: A chat conversation with a model that involves

multiple “turns” or round trips between the human or agent and the machine (LLM). Multi-turn interactions are a basic building block of agentic behavior because they enable planning and dynamic adaptation. Contrast this with a single-turn or “one-shot” conversation, in which the human sends one query, and the LLM sends one response. A few-shot query is similar to a one-shot because they’re both fast enough to operate in pair-programming mode.

Ollama: An open-source tool that simplifies running various large language models locally on your computer. It provides an easy way to download, run, and manage different open-source models like Llama.

One-Shot Query: The simplest vibe coding operation. You send the LLM a question and some context and get the answer back in a single “turn,” meaning one human request followed by one machine response. Contrast this with few-shot queries and multi-turn conversations, which make more round trips, trading off time for accuracy.

Prompt: The input provided to an AI model to guide its response, including instructions, context, and any special requirements or constraints.

Prompt Engineering: The practice of crafting effective inputs to AI models to get desired outputs, though becoming less critical with newer models that better understand natural language.

Prompt Library: A collection of reusable prompts and context snippets that can be applied across different AI programming sessions to maintain consistency and efficiency.

RAG (Retrieval Augmented Generation): A technique that enhances AI model responses by first retrieving relevant information from a knowledge base, such as your code base, documentation, or other resources, and then using that information to generate more accurate and contextual responses. RAG typically involves indexing your code and documentation, capturing frozen semantic meaning, and then retrieving the most relevant pieces of content when AI needs to answer questions or generate code. This helps AI maintain consistency with your existing code base and follow your team’s patterns and conventions. RAG is particularly important for enterprise development where AI needs access to proprietary code and documentation that wasn’t part of its training data.

Static Context: Stable, long-lived information about a project that remains

relevant across multiple LLM sessions. Important because static context is often large and needs indexing. It includes all your relevant existing code, the vast majority of which never changes, and can also include coding standards, architecture documents, long-lived administrative prompts, API documentation, and large bodies of data such as issue trackers, databases, and logs. Often retrieved via RAG.

Task Graph: A conceptual model representing a project's work as interconnected nodes, where each node is some task or challenge that can be handled by humans, AI assistants, or agents. The connections between nodes represent dependencies and information flow.

Token: The basic unit of text that LLMs process, roughly equivalent to three-fourths of a word in English. Token limits affect how much context can be provided to and generated by an AI model.

Token Window: The maximum number of tokens an AI model can process in a single interaction, including both input context and generated output.

V&V (Verification & Validation): In the context of AI-assisted programming, the process of ensuring generated code both meets technical requirements (verification) and solves the intended problem (validation).

Workspace: A persistent environment for AI-assisted development that maintains context, conversations, and generated and/or uploaded artifacts across multiple sessions. Alternatively called a Project, for instance, in both Claude and Google AI Studio.

APPENDIX:

THE INNER/MIDDLE/OUTER LOOPS

Inner Developer Loop (seconds to minutes)

Prevent

- Checkpoint and save your game frequently
- Keep your tasks small and focused
- Get the AI to write specifications
- Have AI write the tests
- AI is a Git maestro

Detect

- Verify AI's claims yourself
- Always on watch: keeping your AI on the rails
- Use test-driven development
- Learn while watching
- Put your sous chef on cleanup duty
- Tell your sous chef where the freezer is

Correct

- When things go wrong: fix forward or roll back
- Automate linting and correction
- When to take back the wheel
- Your AI as a rubber duck

Middle Developer Loop (hours to days)

Prevent

- Written rules: because your sous chefs can't read your mind
- The Memento Method
- Design for AI manufacturing
- Working with two agents at once, and more
- Intentional AI coordination
- Keeping your agents busy when you're busy

Detect

- Waking up to eldritch AI-generated horrors
- Too many cooks: detecting agent contention

Correct

- Kitchen line stress tests: using tracer bullets
- Sharpen your knives: investing in workflow automation
- The economics of optionality

Outer Developer Loop (weeks to months)

Prevent

- Don't let your AI torch your bridges
- Workspace confusion: avoiding the "stewnamis"
- Minimize and modularize
- Managing fleets of agents: four and beyond
- Auditing through or around the kitchen
- Channel your inner product manager
- Making operations fast, ambitious, and fun

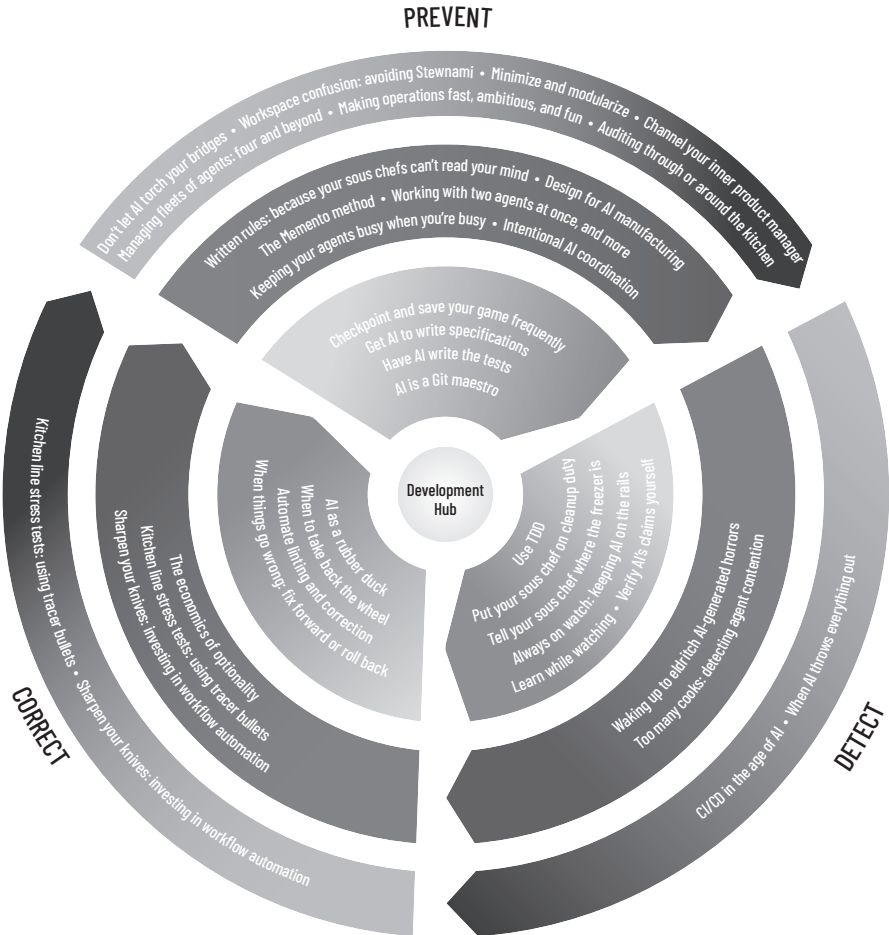
Detect

- When the AI throws everything out
- CI/CD in the age of AI

Correct

- Steve's harrowing merge recovery tale
- When you're stuck with awful processes and architecture

THE THREE DEVELOPER LOOPS OF VIBE CODING



Outer Loop: Weeks to Months

Middle Loop: Hours to Days

Inner Loop: Seconds to Minutes

BIBLIOGRAPHY

- Acemoglu, Daron. "The Simple Macroeconomics of AI." Massachusetts Institute of Technology, April 5, 2024. <https://economics.mit.edu/sites/default/files/2024-04/The%20Simple%20Macroeconomics%20of%20AI.pdf>.
- Aguinaga, Jose. "How It Feels to Learn JavaScript in 2016." *HackerNoon*, October 3, 2016. <https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f>.
- AI Engineer. "Building AI Agents with Real ROI in the Enterprise SDLC: Bruno (Booking.com) & Beyang (Sourcegraph)." YouTube video, April 8, 2025. <https://www.youtube.com/watch?v=UXOLprPvr-0>.
- Andon, Paul. "Rage Against the Algorithm: Uber Drivers Revolt Against Algorithmic Management." *BusinessThink*, October 29, 2023. <https://www.businessthink.unsw.edu.au/articles/uber-algorithmic-management>.
- Anthropic. "Claude Code: Best Practices for Agentic Coding." Anthropic website, April 18, 2025. <https://anthropic.com/engineering/claude-code-best-practices>.
- Anthropic. "Introducing Claude 4." Anthropic website. Accessed May 30, 2025. <https://www.anthropic.com/news/claude-4>.
- Baldwin, Carliss Y. *Design Rules, Volume 2: How Technology Shapes Organizations*. The MIT Press, 2024.
- Ball, Thorsten. "How to Build an Agent or: The Emperor Has No Clothes." *Amp-Podcast*, April 15, 2025. <https://ampcode.com/how-to-build-an-agent>.
- Banks, Rob. "Woman Crashed Motorhome Using Cruise Control While Making Cup of Tea." *Suffolk Gazette*, October 3, 2022. <https://www.suffolkgazette.com/motorhome-crash/>.
- Beane, Matt. *The Skill Code: How to Save Human Ability in an Age of Intelligent Machines*. Harper Business, 2024.
- Beck, Kent. "Social AI Adoption: Lessons from Hybrid Corn." *Tidy First* (Substack), April 9, 2025. <https://tidyfirst.substack.com/p/fb1a4d52-eee7-484c-a3e9-9d6bfae8f7af>.
- Beck, Kent. *Tidy First?: A Personal Exercise in Empirical Software Design*. O'Reilly Media, 2023.
- Belsky, Scott. "Collapse the Talent Stack Every Chance You Get." LinkedIn post, December 20, 2024. <https://www.linkedin.com/pulse/collapse-talent-stack-every-chance-you-get-scott-belsky-srrye/>.

Bhagsain, Anurag (@abhagsain). "Last week, we asked Devin to make a change." X, January 6, 2025. <https://x.com/abhagsain/status/1876362355870994538>.

Bland, Mike. "Goto Fail, Heartbleed, and Unit Testing Culture." MartinFowler .com (blog), June 3, 2014. <https://martinfowler.com/articles/testing-culture.html>.

Borman, Frank. "A superior pilot uses his superior judgment to avoid situations which require the use of his superior skill." QuoteFancy. Accessed April 6, 2025. <https://quotefancy.com/quote/1100682/Frank-Borman-A-superior-pilot-uses-his-superior-judgment-to-avoid-situations-which>.

Butler, Jenna, Jina Suh, Sankeerti Haniyur, and Constance Hadley. "Dear Diary: A Randomized Controlled Trial of Generative AI Coding Tools in the Workplace." arXiv.org, October 24, 2024. <https://arxiv.org/abs/2410.18334>.

"Claude Code: Anthropic's CLI Agent." YouTube video, posted by Latent Space, May 7, 2025. <https://www.youtube.com/watch?v=zDmW5hJPsvQ>.

Cohen, Dave. "I read a lot of headlines these days about AI replacing software engineers..." LinkedIn post, January 2025. https://www.linkedin.com/posts/davemcohen_i-read-a-lot-of-headlines-these-days-about-activity-7288623576113369088-cqfD/.

Cornago, Fernando. "Further Results of Our 500-Person GenAI and Developer Pilot." Presentation at Enterprise Tech Leadership Summit, IT Revolution, February 2025. Video, 21:49. <https://videos.itrevolution.com/watch/1061198586>.

Culver, Hannah. "PagerDuty Operations Cloud Spring 25 Release: Reimagining Operations in the Age of AI and Automation." *PagerDuty* (blog), February 25, 2025. <https://pagerduty.com/blog/product-launch-enhancements-to-pagerduty-operations-cloud-2025-h1/>.

DeBellis, Derek, Kevin M. Storer, Daniella Villalba, Michelle Irvine, and Kim Castillo. "The Impact of Generative AI in Software Development Report." DORA Research, 2024. <https://dora.dev/research/2024/dora-report/>.

Delfanti, Alessandro. *The Warehouse: Workers and Robots at Amazon*. Pluto Press, 2021.

DeLong, J. Bradford. "The Reality of Economic Growth: History and Prospect." In *The Reality of Economic Growth: History and Prospect*, 120–122. https://www2.lawrence.edu/fast/finklerm/DeLong_Growth_History_Ch5.pdf.

De Sousa Pereira, Vitor M. "The Insanity of Being a Software Engineer." *0x1* (blog), April 6, 2025. <https://0x1.pt/2025/04/06/the-insanity-of-being-a-software-engineer/>.

develoopest. "I Must Be the Dumbest 'Prompt Engineer' Ever, Each Time I Ask an AI to Fix or Ev..." *Hacker News*, March 9, 2025. <https://news.ycombinator.com/item?id=43307892>.

Digital Workforce. "AI Agents." Accessed April 19, 2025. <https://digitalwork>

force.com/ai-agents/.

Distefano, Dino, Manuel Fähndrich, Francesco Logozzo, and Peter W. O'Hearn. "Scaling Static Analyses at Facebook." *Communications of the ACM* 62, no. 8 (August 2019): 62–70. <https://cacm.acm.org/research/scaling-static-analyses-at-facebook/>.

Eloundou, Tyna, Sam Manning, Pamela Mishkin, and Daniel Rock. "GPTs Are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models." arXiv.org, March 17, 2023. <https://arxiv.org/abs/2303.10130>.

Ericsson, Anders, and Robert Pool. *Peak: Secrets from the New Science of Expertise*. Mariner Books, 2016.

Ferriss, Tim. "The Tim Ferriss Show Transcripts: Jerry Seinfeld — a Comedy Legend's Systems, Routines, and Methods for Success (#485)." The Blog of Author Tim Ferriss, July 20, 2021. <https://tim.blog/2020/12/09/jerry-seinfeld-transcript/>.

Flowcon. "Keynote: Velocity and Volume (or Speed Wins) by Adrian Cockcroft." YouTube video, December 18, 2013. <https://www.youtube.com/watch?v=wyl3gWpB8o>.

FooCafe. "Advancements and Future Directions in AI-Assisted Coding - Erik Meijer." YouTube video, October 19, 2023. <https://www.youtube.com/watch?v=SsJqmV3Wtkg>.

Forsgren, Nicole, Jez Humble, and Gene Kim. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution, 2018.

Garret, Ron (a.k.a. Erann Gat). "Lisping at JPL." 2002. Accessed April 28, 2025. <https://flownet.com/gat/jpl-lisp.html>.

Garret, Ron. "LISP in Space with Ron Garret." *CoRecursive* #076. Accessed April 28, 2025. <https://corecursive.com/lisp-in-space-with-ron-garret/>.

Gazit, Idan. "Reaching for AI-Native Developer Tools." Presentation at Enterprise Technology Leadership Summit, IT Revolution, Las Vegas, 2024. Video. videos.itrevolution.com/watch/1002959470.

Google. "Google—GitHub Organization." GitHub. Accessed March 5, 2025. <https://github.com/google>.

"Google C++ Style Guide." Accessed May 7, 2025. <https://google.github.io/styleguide/cppguide.html#Exceptions>.

Grove, Andrew S. *High Output Management*. Vintage, 1995.

Guntur, Prabhudev. "Choosing Your AI Agent Framework: Google ADK vs. Autogen, Langchain, & CrewAI—A Deep Dive." *Medium*, April 15, 2025. <https://medium.com/@prabhudev.guntur/choosing-your-ai-agent-framework-google-adk-vs-autogen-langchain>.

Heavybit. "O11ycast | Ep. #80, Augmented Coding with Kent Beck | Heavybit." *Heavybit Podcast*, April 30, 2025. <https://www.heavybit.com/library>

/podcasts/o11ycast/ep-80-augmented-coding-with-kent-beck.

Heelan, Sean. “How I Used O3 to Find CVE-2025-37899, a Remote Zeroday Vulnerability in the Linux Kernel’s SMB Implementation.” *Sean Heelan’s Blog*, May 26, 2025. <https://sean.heelan.io/2025/05/22/how-i-used-o3-to-find-cve-2025-37899-a-remote-zeroday-vulnerability-in-the-linux-kernels-smb-implementation/>.

Hickey, Rich. “A History of Clojure.” *Proceedings of the ACM on Programming Languages*, 2020. <https://dl.acm.org/doi/pdf/10.1145/3386321>.

Humphreys, Brendan. “No, you won’t be vibe coding your way to production. Not if you prioritise quality, safety, security, and long-term maintainability at scale.” LinkedIn post, April 2025. <https://www.linkedin.com/feed/update/urn:li:activity:7305080254417547264/>.

Kalliamvakou, Eirini. “Research: Quantifying GitHub Copilot’s Impact on Developer Productivity and Happiness.” *The GitHub Blog*, May 21, 2024. <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>.

Karpathy, Andrej (@karpathy). “I just vibe coded a whole iOS app in Swift (without having programmed in Swift before, though I learned some in the process) and now ~1 hour later it’s actually running on my physical phone. It was so ez... I had my hand held through the entire process. Very cool.” X, March 22, 2025. <https://x.com/karpathy/status/1903671737780498883>.

Karpathy, Andrej (@karpathy). “Noticing myself adopting a certain rhythm in AI-assisted coding (i.e. code I actually and professionally care about, contrast to vibe code)...” X, April 24, 2025. <https://x.com/karpathy/status/1915581920022585597>.

Karpathy, Andrej (@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.” X, February 2, 2025. <https://x.com/karpathy/status/1886192184808149383>.

Kersten, Nigel, Caitlyn O’Connell, and Ronan Keenan. *2023 State of DevOps Report: Platform Engineering Edition*. Portland, OR: Puppet by Perforce, 2023. <https://www.puppet.com/system/files/2025-03/report-puppet-sodor-2023-platform-engineering.pdf>.

Kim, Gene, Jez Humble, Patrick Debois, John Willis, and Dr. Nicole Forsgren. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. 2nd ed. IT Revolution, 2021.

Kim, Gene, and Steve Spear. *Wiring the Winning Organization: Liberating Our Collective Greatness through Slowification, Simplification, and Amplification*. IT Revolution, 2023.

Kwa, Thomas, Ben West, Joel Becker, et al. “Measuring AI Ability to Complete Long Tasks.” arXiv.org, March 18, 2025. <https://arxiv.org/abs/2503.14499v2>.

- Latent Space, “ChatGPT Codex: The Missing Manual,” YouTube video, posted May 16, 2025, <https://www.youtube.com/watch?v=LIHP4BqwSw0>.
- Levy, Mosh, Alon Jacoby, and Yoav Goldberg. “Same Task, More Tokens: The Impact of Input Length on the Reasoning Performance of Large Language Models.” arXiv.org, February 19, 2024. <https://arxiv.org/abs/2402.14848>.
- Loftus, Tom. “Google Engineer Goofs, Makes Google+ Criticism Public.” *Wall Street Journal*, October 12, 2011. <https://www.wsj.com/articles/BL-DGB-23338>.
- Lopez, Linette. “The White House Is Only Telling You Half of the Sad Story of What Happened to American Jobs.” *Business Insider Nederland*, July 25, 2017. <https://www.businessinsider.nl/what-happened-to-american-jobs-in-the-80s-2017-7/>.
- Lutke, Tobi (@tobi). “Reflexive AI Usage Is Now a Baseline Expectation at Shopify.” X, April 7, 2025. <https://x.com/tobi/status/1909251946235437514>.
- MacroTrends. “Shopify Revenue 2013-2025 | SHOP.” Accessed March 28, 2025. <https://www.macrotrends.net/stocks/charts/SHOP/shopify/revenue>.
- Mauran, Cecily. “Mark Zuckerberg Wants AI to Do Half of Meta’s Coding by 2026.” *Mashable*, April 30, 2025. <https://mashable.com/article/llamacon-mark-zuckerberg-ai-writes-meta-code>.
- McCullough, David. Interview with the National Endowment for the Humanities, Jefferson Lecture, 2003. <https://www.neh.gov/about/awards/jefferson-lecture/david-mccullough-biography>.
- Meijer, Erik (@headinthebox). “Looks amazing! Thanks for doing this. Feels much faster to grasp than the watch the whole talk, even at 2x speed.” X, September 9, 2024. <https://x.com/headinthebox/status/1833304124127121883>.
- Meijer, Erik. “What makes me most happy is that this decreased the #LOC of Ruby and increased the #LOC of Kotlin.” Comments to LinkedIn post, March 2025. https://www.linkedin.com/feed/update/urn:li:activity:7307434087365943296?commentUrn=urn%3Ali%3Acomment%3A%28activity%3A7307434087365943296%2C7307599768673820674%29&dashCommentUrn=urn%3Ali%3Afsd_comment%3A%287307599768673820674%2Curn%3Ali%3Aactivity%3A7307434087365943296%29.
- “Microsoft Build 2025 | Day 2 Keynote.” YouTube video, posted by Replay, May 20, 2025. <https://www.youtube.com/live/RbKyBbn1vkI>.
- Mollick, Ethan. *Co-Intelligence: Living and Working with AI*. Portfolio, 2024.
- Montti, Roger. “Why Google May Adopt Vibe Coding for Search Algorithms.” *Search Engine Journal*, April 4, 2025. <https://www.searchenginejournal.com/why-google-may-adopt-vibe-coding-for-search-algorithms/541641/>.
- Nolan, Beatrice. “AI Employees with ‘Memories’ and Company Passwords Are a Year Away, Says Anthropic Chief Information Security Officer.” *Fortune*,

April 23, 2025. <https://fortune.com/article/anthropic-jason-clinton-ai-employees-a-year-away/>.

Nathani, Ronak, and Guang Yang. “LLMs Are Like Your Weird, Over-confident Intern | Simon Willison (Datasette).” *Software Misadventures Podcast* (blog), September 10, 2024. <https://softwaremisadventures.com/p/simon-willison-llm-weird-intern>.

Olsson, Catherine (@catherineols). “4) If we’re working on something tricky and it keeps making the same mistakes, I keep track of what they were in a little notes file.” X, February 24, 2025. <https://x.com/catherineols/status/1894105719953310045>.

Osorio, Kevin Gargate, and PyCoach. “Codex Is Not Just Smarter. It’ll Reshape Software Development.” *Artificial Corner* (blog), May 22, 2025. <https://artificialcorner.com/p/codex-is-not-just-smarter-itll-reshape>.

Patel, Dwarkesh. “Is RL + LLMs Enough for AGI? – Sholto Douglas & Trenton Bricken.” YouTube video, May 22, 2025. <https://www.youtube.com/watch?v=64lXQP6cs5M>.

Patel, Nilay. “Microsoft CTO Kevin Scott on How AI Can Save the Web, Not Destroy It.” *The Verge*, May 19, 2025. <https://www.theverge.com/decoder-podcast-with-nilay-patel/669409/microsoft-cto-kevin-scott-interview-ai-natural-language-search-openai>.

Patel, Nilay. “UiPath CEO Daniel Dines on AI Agents Replacing Our Jobs.” *The Verge*, April 7, 2025. <https://theverge.com/decoder-podcast-with-nilay-patel/643562/uipath-ceo-daniel-dines-interview-ai-agents>.

Paul, Gus. “Automated Change Management.” Presentation at the IT Revolution Enterprise Summit Europe, 2022. Video. videos.itrevolution.com/watch/708122268.

Programmers are also human. “Interview with Vibe Coder in 2025.” YouTube video, April 1, 2025. <https://www.youtube.com/watch?v=JeNS1ZNHQs8>.

Shopify. “Shopify for Executives - CTOs.” Shopify website. Accessed March 28, 2025. <https://www.shopify.com/toolkit/cto>.

SRC-d. “Hercules: Fast, Insightful and Highly Customizable Git History Analysis.” GitHub Repository, 2023. <https://github.com/src-d/hercules>.

“Steve Yegge/Gene Kim: Pair Programming Session (Sept 2024).” YouTube video, Posted by IT Revolution, November 2024. https://www.youtube.com/playlist?list=PLvk9Yh_MWYuzptetZDa0KxM-ahjQgctHS.

Sturtevant, Daniel J. “System Design and the Cost of Architectural Complexity.” MIT Thesis, 2013. <https://dspace.mit.edu/handle/1721.1/79551>.

Tan, Garry (@garrytan). “For 25% of the Winter 2025 batch, 95% of lines of code are LLM generated. That’s not a typo. The age of vibe coding is here.” X, March 5, 2025. <https://x.com/garrytan/status/1897303270311489931>.

Unwrap. “How GitHub’s Copilot Team Automated Their Entire Customer Feed-

- back Analysis Process.” Case Study, August 5, 2024. <https://unwrap.ai/case-studies/github-copilot>.
- Varanasi, Lakshmi. “AI Won’t Replace Human Workers, but ‘People That Use It Will Replace People That Don’t,’ AI Expert Andrew Ng Says.” *Business Insider*, March 16, 2025. <https://www.businessinsider.com/andrew-ng-ai-jobs-workers-optimist-economy-2024-7>.
- Vas (@vasumanmoza). “Claude 4 just refactored my entire codebase in one call...” X, May 24, 2025. <https://x.com/vasumanmoza/status/1926487201463832863>.
- Wickett, James. “The AI Future of Information Security.” Presentation at the Enterprise Technology Leadership Summit, IT Revolution, Las Vegas, 2024. Video. <https://videos.itrevolution.com/watch/1003869130>.
- Wikipedia contributors. “Auguste Escoffier.” Wikipedia. Last modified March 28, 2025. https://en.wikipedia.org/wiki/Auguste_Escoffier.
- Willison, Simon. “Here’s how I use LLMs to help me write code.” *Simon Willison’s Weblog* (blog), March 11, 2025. <https://simonwillison.net/2025/Mar/11/using-llms-for-code/#context-is-king>.
- Wu, Scott. “Introducing Devin, the First AI Software Engineer.” *Cognition* (blog), March 12, 2024. <https://cognition.ai/blog/introducing-devin>.
- Yegge, Steve. “Dear Google Cloud: Your Deprecation Policy Is Killing You.” *Medium*, August 14, 2020. <https://steve-yegge.medium.com/dear-google-cloud-your-deprecation-policy-is-killing-you-ee7525dc05dc>.
- Yegge, Steve. “Stevey’s Google Platforms Rant.” *GitHub Gist*, posted by chitchcock, 2011. Accessed May 28, 2025. <https://gist.github.com/chitchcock/1281611>.
- Yegge, Steve. “The Death of the Junior Developer.” *Sourcegraph* (blog), June 24, 2024. <https://sourcegraph.com/blog/the-death-of-the-junior-developer>.
- Zavřel, Roman. “This Year, 94% of All Photos Will Be Taken on Smartphones—How Many Photos Does the Average American Take per Day?” *Letem Svetem Applem*, April 19, 2024. <https://www.letemsvetemapplem.eu/en/2024/04/19/v-letosnim-roce-bude-94-vsech-fotografii-porizeno-pomoci-smartphonu-v-usa-prumerne-vyfoti-clovek-20-fotek-denne/>.

NOTES

Introduction

1. FooCafe, “Advancements and Future Directions in AI-Assisted Coding - Erik Meijer.”
2. Cornago, “Further Results of Our 500-Person GenAI and Developer Pilot.”
3. Cornago, “Further Results of Our 500-Person GenAI and Developer Pilot.”
4. Beck, “Social AI Adoption: Lessons from Hybrid Corn.”
5. Karpathy, “There’s a new kind of coding I call ‘vibe coding.’”
6. Kalliamvakou, “Research: Quantifying GitHub Copilot’s Impact on Developer Productivity and Happiness.”
7. Mauran, “Mark Zuckerberg Wants AI to Do Half of Meta’s Coding by 2026.”
8. Wu, “Introducing Devin, the First AI Software Engineer.”
9. Nolan, “AI Employees With ‘Memories’ and Company Passwords Are a Year Away.”
10. Yegge, “Stevey’s Google Platforms Rant.”
11. Loftus, “Google Engineer Goofs, Makes Google+ Criticism Public.”
12. Yegge, “The Death of the Junior Developer.”
13. Kent Beck, personal conversation with the authors, April 2, 2025.

Chapter 1

1. Karpathy, “There’s a new kind of coding I call ‘vibe coding.’”
2. Karpathy, “There’s a new kind of coding I call ‘vibe coding.’”
3. Karpathy, “There’s a new kind of coding I call ‘vibe coding.’”
4. Tan, “For 25% of the Winter 2025 batch, 95% of lines of code are LLM generated.”
5. “Claude Code: Anthropic’s CLI Agent.”
6. MacroTrends, “Shopify Revenue 2013-2025 | SHOP.”
7. Shopify, “Shopify for Executives - CTOs.”
8. Lutke, “Reflexive AI Usage Is Now a Baseline Expectation at Shopify.”
9. Humphreys, “No, you won’t be vibe coding your way to production.”

10. Jessie Young, personal conversation with Gene Kim, February 29, 2025.
11. Montti, “Why Google May Adopt Vibe Coding for Search Algorithms.”
12. Montti, “Why Google May Adopt Vibe Coding for Search Algorithms.”
13. “Microsoft Build 2025 | Day 2 Keynote.”

Chapter 2

1. Aguinaga, “How It Feels to Learn JavaScript in 2016.”
2. De Sousa Pereira, “The Insanity of Being a Software Engineer.”
3. Borman, “A superior pilot uses his superior judgment.”

Chapter 3

1. “Claude Code: Anthropic’s CLI Agent.”
2. Kim and Spear, *Wiring the Winning Organization*, xxvii.
3. Dr. Daniel Rock, personal conversation with the authors, April 23, 2025.
4. Belsky, “Collapse the Talent Stack Every Chance You Get.”
5. Butler et al., “Dear Diary: A Randomized Controlled Trial of Generative AI Coding Tools in the Workplace.”
6. Cornago, “Further Results of Our 500-Person GenAI and Developer Pilot.”

Chapter 4

1. DeBellis et al., “The Impact of Generative AI in Software Development Report.”
2. Kwa et al., “Measuring AI Ability to Complete Long Tasks.”
3. Patel, “Is RL + LLMs Enough for AGI? – Sholto Douglas & Trenton Bricken.”
4. Kwa et al., “Measuring AI Ability to Complete Long Tasks.”

Chapter 5

1. Eloundou et al., “GPTs Are GPTs.”
2. Brendan Hopper, personal communication with Gene Kim, April 2025. Hopper was referencing Dr. Nicholas Negroponte, founder of the MIT Media Lab, for framing this as “move bits, not atoms.”
3. Lopez, “The White House Is Only Telling You Half of the Sad Story of What Happened to American Jobs.”
4. Varanasi, “AI Won’t Replace Human Workers, but ‘People That Use It Will Replace People That Don’t,’ AI Expert Andrew Ng Says.”
5. FooCafe, “Advancements and Future Directions in AI-Assisted Coding -

Erik Meijer.”

6. Yegge, “The Death of the Junior Developer.”
7. Cohen, “I read a lot of headlines these days about AI replacing software engineers...”
8. Zavřel, “This Year, 94% of All Photos Will Be Taken on Smartphones.”
9. Acemoglu, “The Simple Macroeconomics of AI.”
10. DeLong, “The Reality of Economic Growth: History and Prospect.”
11. Matt Velloso, personal correspondence with Gene Kim, March 2025.
12. Velloso, personal correspondence with the authors, 2025.

Chapter 6

1. Cornago, “Further Results of Our 500-Person GenAI and Developer Pilot.”
2. AI Engineer, “Building AI Agents with Real ROI in the Enterprise SDLC.”

Chapter 7

1. Forsgren, Humble, and Kim, *Accelerate*.
2. Sturtevant, “System Design and the Cost of Architectural Complexity.”
3. Forsgren, Humble, and Kim, *Accelerate*.
4. Latent Space, “ChatGPT Codex: The Missing Manual.”
5. Ericsson and Pool, *Peak*.

Chapter 9

1. If you’re interested, you can watch each step in this YouTube playlist: “Steve Yegge/Gene Kim: Pair Programming Session (Sept 2024).”
2. Meijer, “Looks amazing! Thanks for doing this. Feels much faster to grasp than the watch the whole talk, even at 2x speed.”
3. Erik Meijer, personal correspondence with Gene Kim, May 14, 2025.
4. Karpathy, “I just vibe coded a whole iOS app in Swift...”
5. Gazit, “Reaching for AI-Native Developer Tools.”

Chapter 10

1. Willison, “Here’s how I use LLMs to help me write code.”
2. Karpathy, “Noticing myself adopting a certain rhythm in AI-assisted coding (i.e. code I actually and professionally care about, contrast to vibe code)...”
3. “Claude Code: Anthropic’s CLI Agent.”

Chapter 11

1. Jason Clinton, personal conversation with the authors, April 2, 2025.
2. Anthropic, “Introducing Claude 4.”

Chapter 12

1. Vas (@vasumanmoza), “Claude 4 just refactored my entire codebase in one call...”
2. Gazit, “Reaching for AI-Native Developer Tools.”
3. Mollick, *Co-Intelligence*, 46.
4. developoest, “I Must Be the Dumbest ‘Prompt Engineer’ Ever.”
5. Banks, “Woman Crashed Motorhome Using Cruise Control While Making Cup of Tea.”
6. Bhagsain (@abhagsain), “Last week, we asked Devin to make a change.”
7. Erik Meijer, personal communication with the authors, May 14, 2025.
8. Meijer, “What makes me most happy is that this decrease the #LOC of Ruby and increased the #LOC of Kotlin.”
9. Flowcon, “Keynote: Velocity and Volume (or Speed Wins) by Adrian Cockcroft.”
10. Grove, *High Output Management*.

Chapter 13

1. Patel, “Microsoft CTO Kevin Scott on How AI Can Save the Web, Not Destroy It.”

Chapter 14

1. Bland, “Goto Fail, Heartbleed, and Unit Testing Culture.”
2. Distefano et al., “Scaling Static Analyses at Facebook.”
3. Kersten, O’Connell, and Keenan, *2023 State of DevOps Report*.
4. Nathani and Yang, “LLMs Are Like Your Weird, Over-confident Intern | Simon Willison (Datasette).”

Chapter 15

1. “Google C++ Style Guide.”
2. Google, “Google—GitHub Organization.”
3. Olsson, “4) If we’re working on something tricky and it keeps making the same mistakes...”
4. Anthropic, “Claude Code: Best Practices for Agentic Coding.”

5. Osorio and PyCoach, “Codex Is Not Just Smarter. It’ll Reshape Software Development.”
6. Ferriss, “The Tim Ferriss Show Transcripts: Jerry Seinfeld — a Comedy Legend’s Systems, Routines, and Methods for Success (#485).”
7. Kent Beck, personal conversation with Gene Kim, January 2025.
8. Baldwin, *Design Rules*, 78.

Chapter 16

1. Yegge, “Dear Google Cloud.”
2. Wickett, “The AI Future of Information Security.”
3. Heelan, “How I Used O3 to Find CVE-2025-37899.”
4. Heelan, “How I Used O3 to Find CVE-2025-37899.”
5. Paul, “Automated Change Management.”

Chapter 17

1. Wikipedia contributors, “Auguste Escoffier.”
2. Kim, Humble, Debois, Willis, Forsgren, *The DevOps Handbook*, 104.
3. DeBellis et al., “The Impact of Generative AI in Software Development Report.”
4. Cornago, “Further Results of Our 500-Person GenAI and Developer Pilot.”
5. Ken Exner, Director of Dev Productivity, 2015, tktk.

Chapter 18

1. Heavybit, “O11ycast | Ep. #80, Augmented Coding With Kent Beck | Heavybit.”

Chapter 19

1. Dr. Daniel Rock, personal conversation with the authors, May 2025.
2. Dr. Matt Beane, personal conversation with the authors, May 2025.
3. McCullough, Interview with the National Endowment for the Humanities.

ACKNOWLEDGMENTS

We want to thank Dr. Andrej Karpathy for coining the phrase vibe coding and Dr. Erik Meijer for giving us such an inspiring vision of where vibe coding will take our profession.

We are also grateful to Dario Amodei for writing a powerful and visionary foreword for our book, and for all that Anthropic is doing for society.

Thank you to Dr. Carliss Baldwin (Harvard Business School) and Dr. Steve Spear (MIT Sloan) for teaching us about modularity and option value. (And Dr. Daniel Rock for all the after-school tutoring sessions we needed afterward!)

Our heartiest thanks to Simon Willison for his brilliant characterization of AI as the “crazy summer intern, who also believes in conspiracy theories,” and his amazing 11m utility, which became the heart of Gene’s Writer’s Workbench, because of the modularity it enabled (hello NK/t and σ !).

And thank you to all our manuscript reviewers, who went to outrageous lengths to help improve our book—your long letters to us gave us a lot to think about, and we hope you see how your feedback shaped the final book: Dr. Matt Beane (MIT and UCSB), Adam Gordon Bell (CoRecursive), JD Black (Northrop Grumman), James Cham (Bloomberg Beta), Mike Carr (Vanguard), Sean Corfield (World Singles Networks), Jason Cox (Disney), Cornelia Davis (Temporal Technologies), Derek DeBellis (Google), Richard Feldman (zed.dev), Ben Grinnell, Jeff Gallimore (Excella), Nathen Harvey (DORA and Google Cloud), Mitchell Hashimoto, Elisabeth Hendrickson (Curious Duck), Christine Hudson (The Welcome Elephant), Christofer Hoff (LastPass), Tom Killilea, Dr. Mik Kersten (Planview), Kerrick Long (Over The Top Marketing), Ryan Martens (Manifest), Dr. Erik Meijer, Kyle Moschetto (KMo), Stuart Pearce (Hg), John Rauser (Cisco), Matt Ring (John Deere), Richard Seroter (Google Cloud), Randy Shoup (Thrive Market), Steve Smith (Equal Experts), Laura Tacho (DX), Mat Velloso (Meta), Prashant Verma (DoorDash), Steve Wilson (Exabeam), Adam Zimman.

Gene

Thank you to everyone who has helped me learn about how to use AI to become a better developer, listed in roughly chronological order: Mitesh Shah (Gaiwan), Patrick Debois, Jason Cox (Disney), Jeff Gallimore (Excella), Brian Scott (Adobe), Joseph Enochs (EVT), Paige Bailey (Google), Idan Gazit (GitHub), Dr. Eirini Kalliamvakou (GitHub), Luke Burton (NVIDIA), Kent Beck (KentBeck.com), and Adrian Cockcroft.

I am so grateful to everyone who helped me better understand the impact of AI on technology organizations and society by sharing their expertise and experiences, including Dr. Matt Beane (UCSB and MIT), Jason Clinton (Anthropic), Fernando Cornago (adidas), Jason Cox (Disney), Dr. Joe Davis (Vanguard), Dr. Nicole Forsgren (Microsoft), Andrew Glover (OpenAI), Brendan Hopper (CBA), Timothy Howard (UK Defra), Dr. Tapabrata Pal (Fidelity Investments), Bruno Passos (Booking.com), John Rauser (Cisco), Dr. Daniel Rock (Wharton and Workhelix), Ryan Sikorsky (Equal Experts), Amy Willard (John Deere), and Jessie Young (GitLab).

And to my coauthor Steve Yegge, whose work I've admired for over a decade. I never would have believed that we'd get to work on something together, let alone something that would lead to so many exciting adventures. I so much appreciated your love of coding, high energy and standards, compassion, and desire to improve our profession.

Steve

Thank you to Dominic Cooney (Anthropic) for validating my crazy ideas early on, leading to my "Death of the Junior Developer" post, which got this whole ball rolling. And thank you to Dominic Widdows (AMD) for our thoughtful early conversations in this space and for being the first to realize we're turning into AI nannies.

Thank you to Quinn Slack (CEO Sourcegraph), whose support and brilliant ideas made this book possible. And I thank everyone at Sourcegraph, an amazing and vibrant company, for cheering me on while Gene and I slogged through this instruction manual for the agentic coding age.

I am so grateful to everyone who helped me better understand vibe coding, agents, LLMs, and AI in the enterprise, leading to this being a much more useful book: Beyang Liu (CTO Sourcegraph), Chris Sells (Sourcegraph),

Dr. Eric Fritz (Sourcegraph), Erika Rice Scherpelz (Sourcegraph), Gergely Orosz (The Pragmatic Programmer), Mike Schiraldi (Anthropic), Oscar Wickström (Sourcegraph), Prashant Verma (DoorDash), Rik Nauta (Sourcegraph), Rishabh Mehrotra (Sourcegraph), Robert Lathrop (Ghost Track, the man who first spotted Godzilla), and Thorsten Ball (Sourcegraph).

Finally, thank you, Gene, for coming along on this amazing adventure we've been on, and for always being inspiring and encouraging. The book is great because of you, and also it's finished because of you: you dragged us to the finish line through sheer willpower and a world-class Writer's Workbench that you vibe coded along the way. What an effort! We'll be sharing stories from this adventure for years to come.

ABOUT THE AUTHORS

Gene Kim has been studying high-performing technology organizations since 1999. He was the founder and CTO of Tripwire, Inc., an enterprise security software company, where he served for thirteen years. His books have sold over 1 million copies—he is the *Wall Street Journal* bestselling author of *The Unicorn Project*, and co-author of *Wiring the Winning Organization*, *The Phoenix Project*, *The DevOps Handbook*, and the Shingo Publication Award-winning *Accelerate*. In 2025, he won the Philip Crosby Medal from the American Society for Quality (ASQ) for his work on the book *Wiring the Winning Organization*. Since 2014, he has been the organizer of DevOps Enterprise Summit (now Enterprise Technology Leadership Summit), studying the technology transformations of large, complex organizations.

Steve Yegge began his career as a computer programmer at GeoWorks in 1992. He worked at Amazon from 1998 to 2005 as a senior engineer and senior manager. There he led the transition from 2-tier to N-tier service architecture, then led Customer Service Tools. From 2005 to 2018, Yegge worked at Google as a senior staff engineer and senior engineering manager. There, he built a knowledge engine called Grok, wired into Google’s internal Code Search system, which had a 99% satisfaction rating within Google (soundly beating the next-best tool by double digits). He went on to be Head of Engineering at Grab, a ride-share and payments company based in Singapore. Beginning in 2022, he helped lead the development of the Cody AI assistant at Sourcegraph (which commercialized the Code Search system that Steve built at Google) and wrote the infamous “Yegge Rant” in 2011 and the “Death of the Junior Developer” post in 2024.