

# The Modern Software Developer

CS146S  
Stanford University, Fall 2025  
**Mihail Eric**

# Guest Lecture - 10/17/25



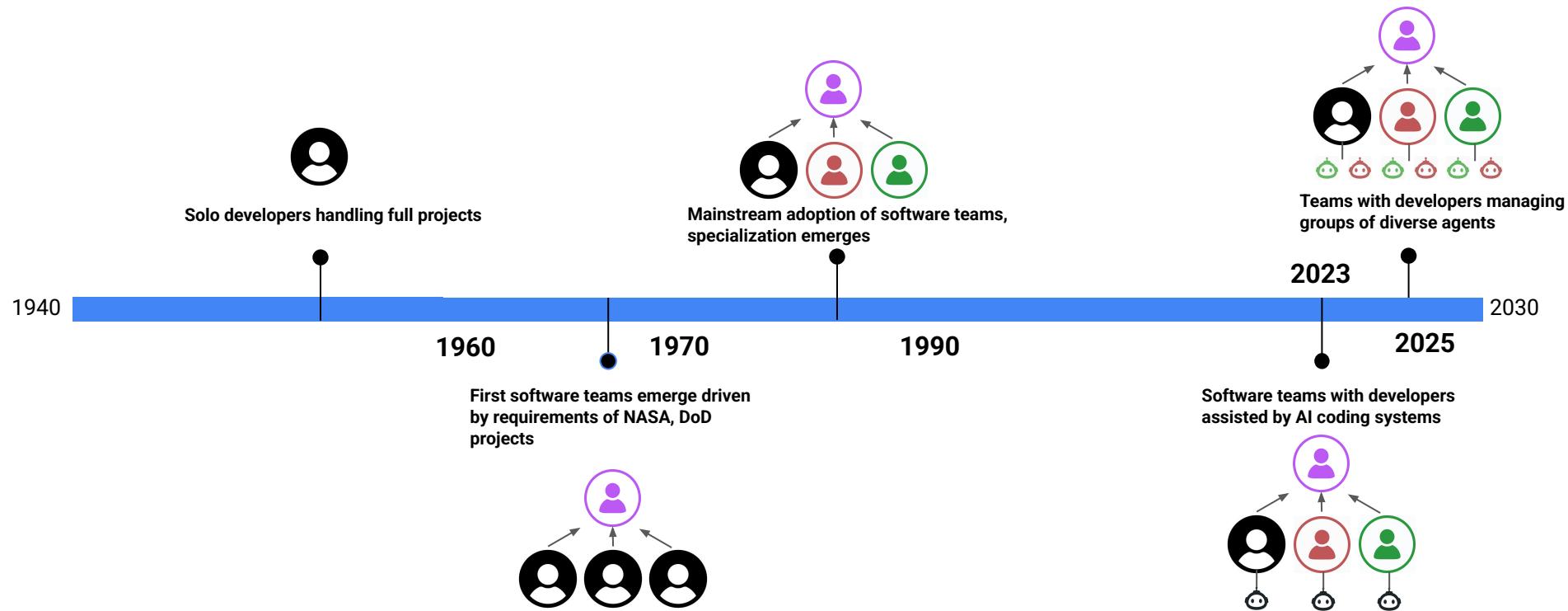
Anthropic Creator of [Claude Code](#), Boris Cherny

# How to be an Agent Manager

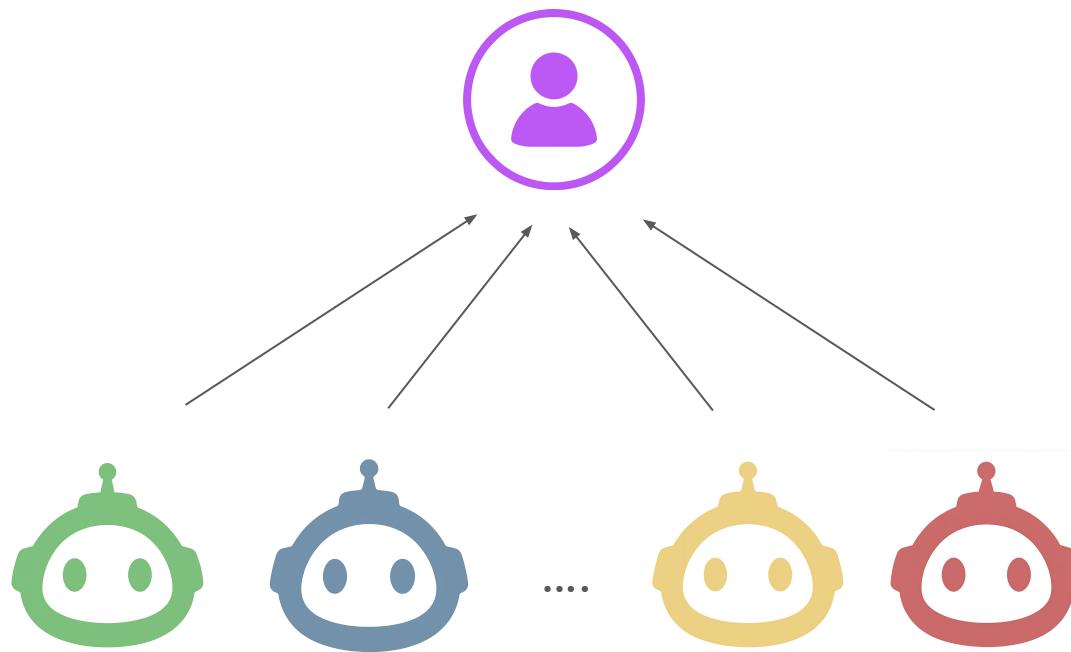
# Why

- Development evolution
  - Single developer managing single developer's output
  - Lead managing many developers worth of output
  - Lead managing many developers worth of output (assisted by an AI system)
  - Single developer managing many AI agents worth of work

# A brief history of software teams



# Goals



# Software task steps

- Provide high level requirements 
- Convert requirements into a design doc  / 
- Implement solution from doc 
- Add tests 
- Ensure CI (continuous integration) passes 
- Code review 
- Update docs 

# Techniques for directing agents

- Agent behavior files (Claude.md/Cursorrules/agents.md)
- Hooks
- Commands
- Subagents

# Hooks

- Deterministic scripts that run on predefined event types
  - PreToolUse
  - PostToolUse
  - UserPromptSubmit
  - PreCompact
  - ...and more

# Commands

- Provide frequently-used prompts as files that the agent can execute
- Use cases
  - Running tests
  - Reviewing code
  - Form a git commit, push

# Subagents

- Runtime delegation
- Purposes of a subagent is to
  - Create distinct developer personas for different types of work (frontend, backend, etc)
  - Cleanly separate contexts for different work streams
- Offer
  - Customized system prompts, tools, and a separate context window
  - A move toward agents managing other agents
- Use cases
  - <https://github.com/vijaythecoder/awesome-claude-agents/blob/main/CLAUDE.md>
  - [https://github.com/SuperClaude-Org/SuperClaude\\_Framework](https://github.com/SuperClaude-Org/SuperClaude_Framework)

# Best practices

- You need careful backstops
  - Tests in codebase
  - CI/CD best practices
- Auditability of each agent
  - Label every diff made by an agent
- Different models for different classes of tasks
- More complex tasks you may need to handhold a bit more upfront vs fully async ones
- Checkpoint (commit) regularly

# Workflow walkthrough

# Open questions

- How can we automate the first 10-20% research phase of any task?
- How to maintain a queue of pending tasks (easier for 1-off changes)?