News from San Diego + Intro to Multi-Agent Engineeering

15-113 – Monday 3/23

Quick poll on Database/SQLite HW6

- As we wait for class to start, please fill this out: https://bit.ly/3PpqugY

- Note that this is *not the most recent HW you did!* We'll discuss that exercise later, and we'll gather your HW7 thoughts on it on Wednesday.

News from San Diego + Intro to Multi-Agent Engineeering

15-113 – Monday 3/23

First off…

- Thanks so very much for your patience while I was away
- Extra thanks to the 15-113 TAs for steering the ship!
- My office hours are back at their classic times this week. As always, email me if you can’t make it to those and would like to set up a meeting.

So where was I?

- GenAI in CS Education consortium meeting on Practice and Research
  - Held at UC San Diego on the shores of La Jolla (no snow there)

- Colleges and universities and industries all came together to discuss these questions (paraphrased):
  - How does GenAI change how we code/create/explore today and in the future?
  - What are the key skills we need to develop?
  - How do we adapt our tools and curricula to empower you to learn those skills?

So where was I?

- GenAI in CS Education consortium meeting on Practice and Research
  - Held at UC San Diego on the shores of La Jolla (no snow there)

- Colleges and universities and industries all came together to discuss these questions (paraphrased):
  - How does GenAI change how we code/create/explore today and in the future?
  - What are the key skills we need to develop?
  - How do we adapt our tools and curricula to empower you to learn those skills?

So where was I?

- GenA
  Resea
  - Hel
- Colleg
  discus
  - Hov
    futu
  - Wh
  - Hov

the

those skills?

Sc

- Ge
  Re
  -

Some unsurprising confirmations

- GenAI is probably not going anywhere
- Github Copilot and others are using primarily AI-generated code to improve their GenAI tools, according to Github Copilot representative
- Industry is still *thrilled* to get more productivity out of you (but many of us worry about cognitive overload and burnout)
- Benario et al. 2025: We are approaching ubiquity
  - 90% of professional software engineers use GenAI for at least something
  - In the above study, 88.9% of students used GenAI, of which 90% learned entirely informally
  - Students use GenAI more productively for learning and creating after instruction
- Traditional human-studies research generally can't stay current (publication cycle is just too long)

On the education side

- Near-consensus that we *must* integrate and teach GenAI, and we have to figure it out fast.

- Likewise, we also need to understand and teach the costs of AI (monetary, environmental, psychological, and humanitarian)

- One immediate concern for students: Social isolation
  - "All roads lead to ChatGPT" (Hou et al. 2025)
    - Most questions are taken straight there
    - If you ask your friend, they'll tell you to "ask Chat"
    - …or, they'll ask Chat and send the results back to you
  - Selective disclosure of AI use is common
  - Lost opportunities for peer interaction and shared struggle
  - Magnified fears of rejection, "uselessness," or unreliability

Consensus: You are amazing

- I told people about you >>(o u o)<<
- Reps from other universities were genuinely amazed with your projects and the progress you've made in such a short time
- I genuinely believe you're leading the pack. Multiple universities are now looking at your success as they shape their future courses.
- Can't stop here though!

What do you need to know?

- It might depend on whether you want to be a Computer Scientist, a Software Engineer, or Something Else
- Consensus that you still need to know how to read and understand unfamiliar code. *Less* consensus on whether *everyone* needs write code.
- Career-critical skills for programmers:
  - Identifying and resolving unknowns/ambiguity
  - Creating detailed specifications
  - Problem decomposition (similar to Top Down Design)
  - Ability to read code and identify code quality
  - Debugging, Validation, and Verification
  - Deep critical thinking
- Skills that will open many more doors:
  - Navigating ethical considerations
  - Human-oriented design skills
  - Ability to write and modify code

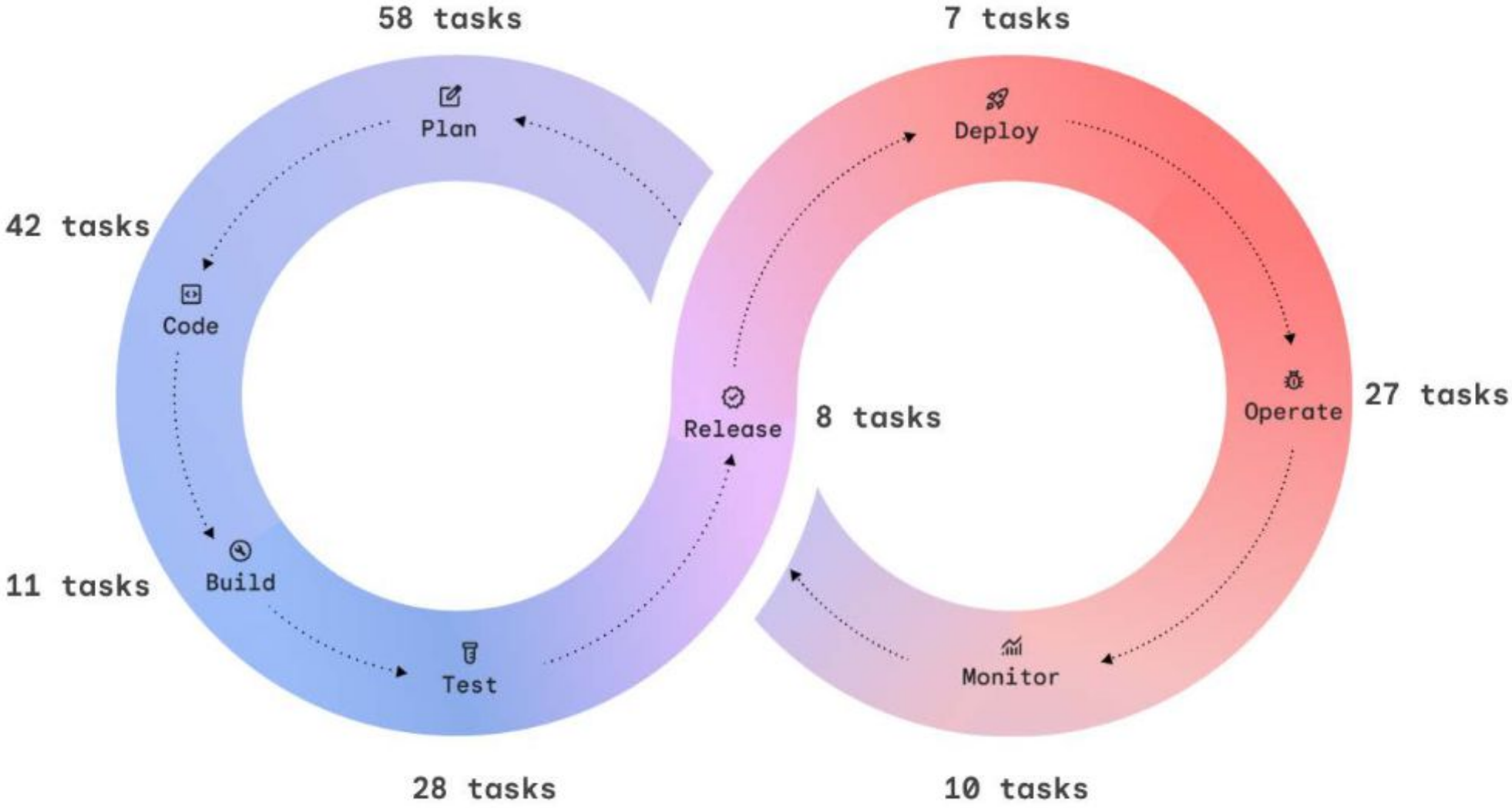
“What do professional software developers need to know to succeed in an age of Artificial Intelligence?”

- This is worth a read: (Kam et al. 2025)

- “We describe our research with 21 developers at the cutting edge of using AI, summarizing 12 of their work goals we uncovered, together with 75 associated tasks and the skills & knowledge for each, illustrating how developers use AI at work.”

- “In order to "future proof" developers ... on-the-job learning initiatives and computer science degree programs will need to target both "soft" skills and … technical skills & knowledge”

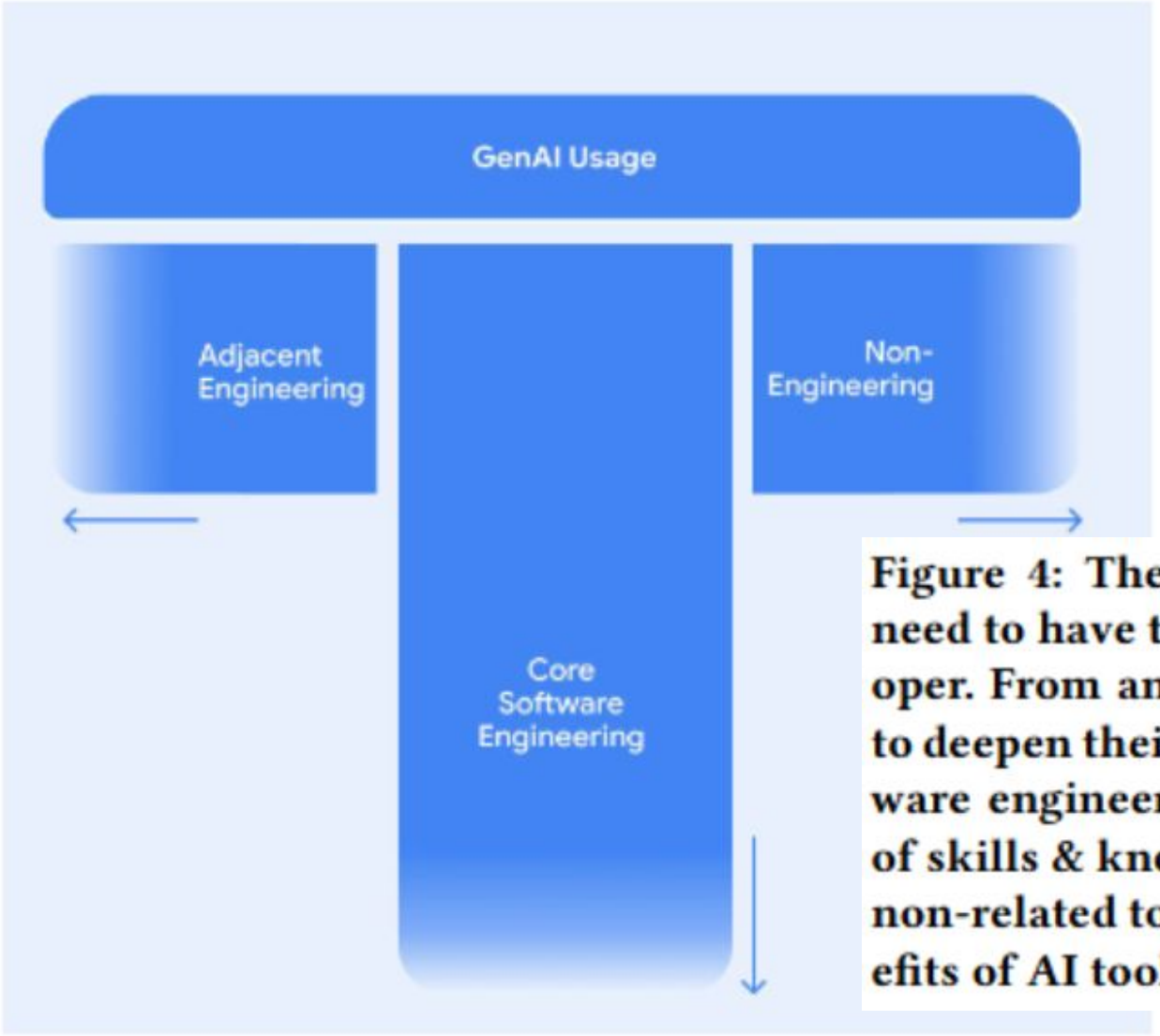
“What do professional software developers need to know (continued)”



“What do professional software developers need to know (continued)”

- **Plan** (58 tasks): project/technical-level planning and investigation prior to detailed implementation.
- **Code** (42): generating code, developing tests, and configuring the necessary system setup.
- **Build** (11): transforming source code into a deployable artifact.
- **Test** (28): evaluating the system at various levels and using multiple types of testing to ensure quality.
- **Release** (8): making a version of the system available.
- **Deploy** (7): making a system operational in a specific environment.
- **Operate** (27): ongoing process of keeping a system running smoothly. It overlaps with other stages when tasks involve coordination and task management work.

“What do professional software developers need to know (continued)”



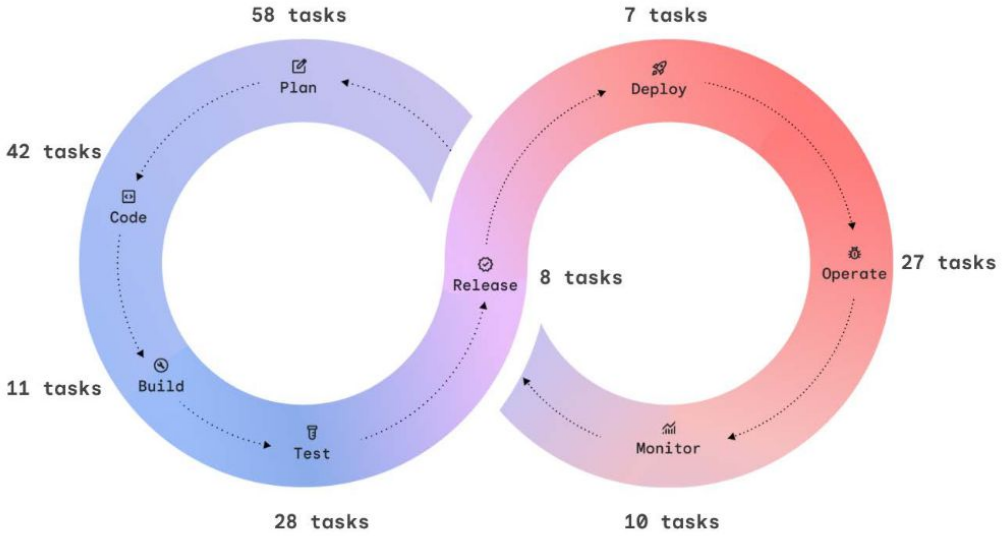


Figure 4: The AI-enhanced developer (of tomorrow) will need to have the skills & knowledge of today’s senior developer. From an upskilling perspective, developers will need to deepen their depth of skills & knowledge in the "core software engineering" domain, as well as widen their breadth of skills & knowledge in adjacent domains both related and non-related to engineering, in order to fully realize the benefits of AI tools while managing the risks.

What AI tools and strategies are being used in industry today?

- Less browser-based chat-style tools
- More Copilot, Codex, Claude Code
  - (just as many of you have been doing)
- More “hybrid” software that *integrates* calls to AI to perform tasks
- Multi-agent software engineering
  - Using multiple agents with well-scoped contexts to handle separate tasks, like code creation, documentation, verification, parts of validation
  - (This is our next topic)

If time, let's discuss HW7 with each other

- We tasked you with improving and ideally finishing someone else's AI-generated code. How'd that go?
- Did you find anything more difficult than you imagined?
- Regardless, was there anything your contributor did that made your task easier?
- Was there anything you wish they had done in retrospect, or anything you wish you had done before handing off your code?

How Prof. Kosbie and I write exams in 15-112

Big problem: It's hard to review something that you've written yourself, and we *really* need exams to have the right difficulty and zero mistakes /oversights. So, as an example…

- **David** writes the first draft in a temporary format
- **Mike** receives the first draft, makes high level adjustments to difficulty and content, and formats it
- **Our review lead** sends the draft to multiple TA reviewers
- **The TA reviewers** (in parallel) scour the exam for problems
- **The review lead** collects and consolidates the reviews
- **David** makes final adjustments based on the collected review

Imagine a similar arrangement of agents:

- **The human engineer** writes a detailed specification for an app
- **Agent 1** creates the initial code
- **Agent 2** dispatches agents **3a, 3b, and 3c**
  - **Agent 3a** checks that the specification is met and all test cases pass
  - **Agent 3b** checks that the code does not pose security risks
  - **Agent 3c** checks for code quality and style
- **Agent 2** consolidates, and prioritizes results from **3a, 3b, and 3c**
- **The human engineer** decides which of the issues to address and then addresses them (likely with the help of AI)
- **Rinse and repeat as necessary**

Quick discussion:
Why might this be a good strategy?

“9 Parallel AI Agents That Review My Code”

- Hamilton Greene, Software Engineer
- This article goes into more depth about one person’s workflow
- It’s much more complicated than we want to dive into right now, but it’s a good read.

Let’s look at this week’s assignment

We’ll post this ASAP after class

Let’s get agentic

15-113 – Wednesday 3/25

Quick poll on Code Hand-off HW7

- Please fill this out:
  https://bit.ly/4uQnhqP

- This is the most recent homework, where you passed your code off to someone else, and tried to fix another person's code.

Pushed from Monday, let's discuss HW7

- We tasked you with improving and ideally finishing someone else's AI-generated code. How'd that go?
- Did you find anything more difficult than you imagined?
- Regardless, was there anything your contributor did that made your task easier?
- Was there anything you wish they had done in retrospect, or anything you wish you had done before handing off your code?

Remember the agentic conversation

- **The human engineer** writes a detailed specification for an app
- **Agent 1** creates the initial code
- **Agent 2** dispatches agents **3a, 3b, and 3c**
  - **Agent 3a** checks that the specification is met and all test cases pass
  - **Agent 3b** checks that the code does not pose security risks
  - **Agent 3c** checks for code quality and style
- **Agent 2** consolidates, and prioritizes results from **3a, 3b, and 3c**
- **The human engineer** decides which of the issues to address and then addresses them (likely with the help of AI)
- **Rinse and repeat as necessary**

Today we’re just going to try to do some cool stuff with this. First we need something to build. Any ideas?