AI Coding Assistants

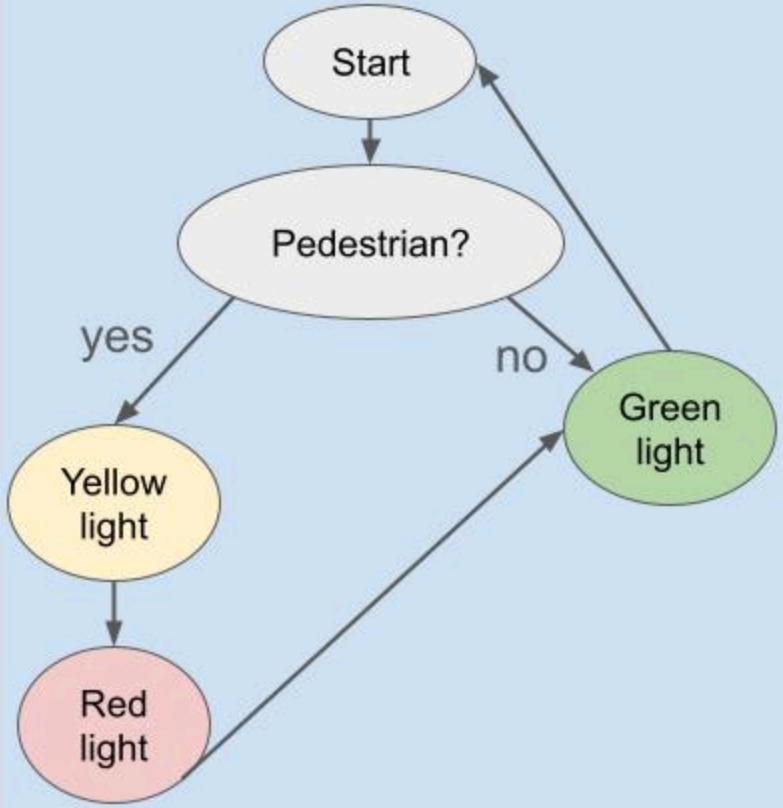
Welcome back to CS 2100!

Prof. Rasika Bhalerao

How does generative AI work?

Rasika Bhalerao, Ph.D.

Without ML:



Machine Learning:

"The science of getting computers to act without being explicitly programmed"

- Andrew Ng

Predict the next word:

Please turn your homework…

A. in
B. over
C. the
D. agriculture

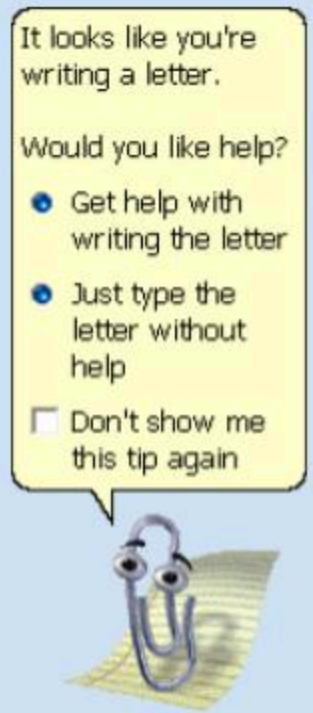
Predict the next word:

Please turn your homework…

Probability:	
75%	A. in
20%	B. over
4%	C. the
1%	D. agriculture

Without ML:

- Early chatbots (Clippy)
- Search algorithms
- Planning / scheduling algorithms



Machine Learning:

Not explicitly programmed

ChatGPT

Claude

Copilot

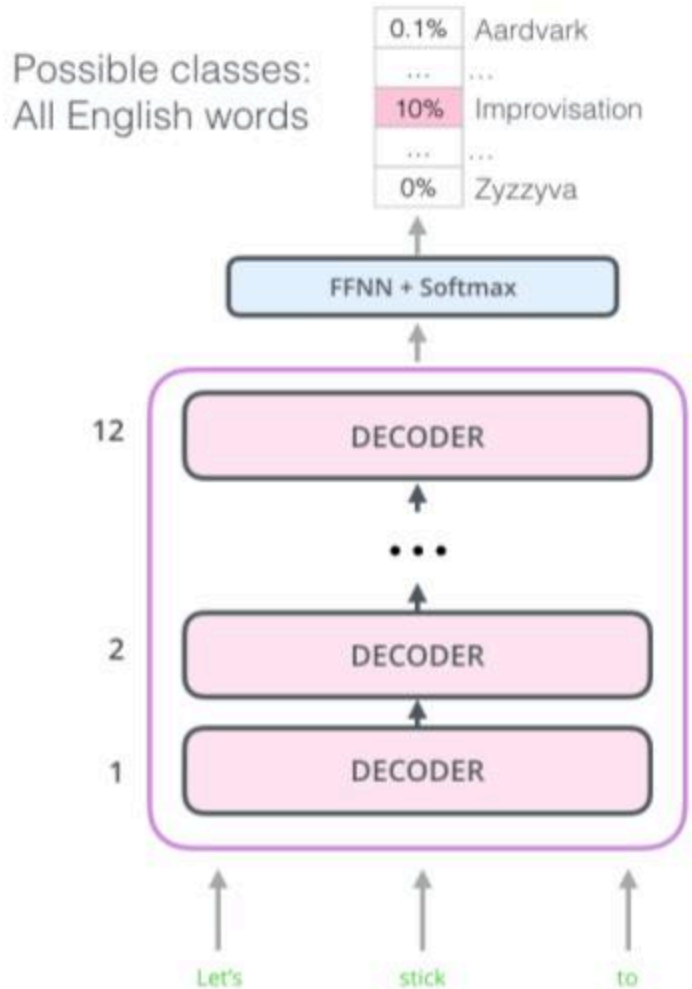
Gemini

Generative models predict the next word

- Huge models trained on large amounts of data from the internet
- They can make mistakes called “hallucinations”
- They reproduce the social bias in their training sets
- They have a large carbon footprint

How was ChatGPT trained? (And Claude after that)

It was taught to predict the missing word.



Before ChatGPT, GPT2 required a few examples in the prompt:

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate Eng
cheese =>
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← example
cheese =>                           ← prompt
```

Few-shot

In addition to the task examples of the task.

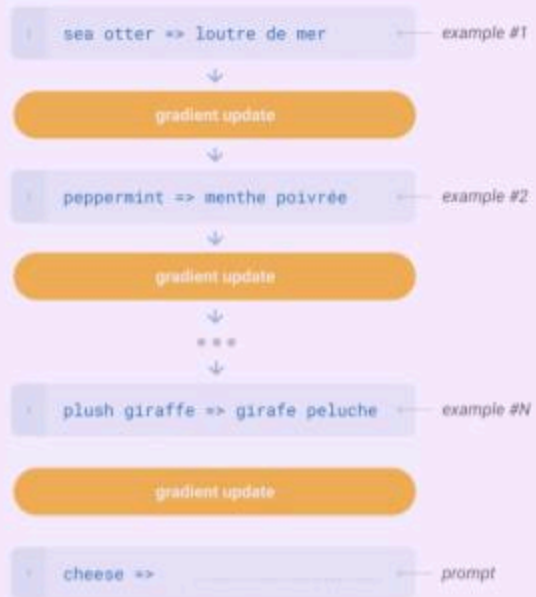
```
Translate English to French:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                           ← prompt
```

And before that, language models needed thousands of examples:

Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



Bigger model → better language model

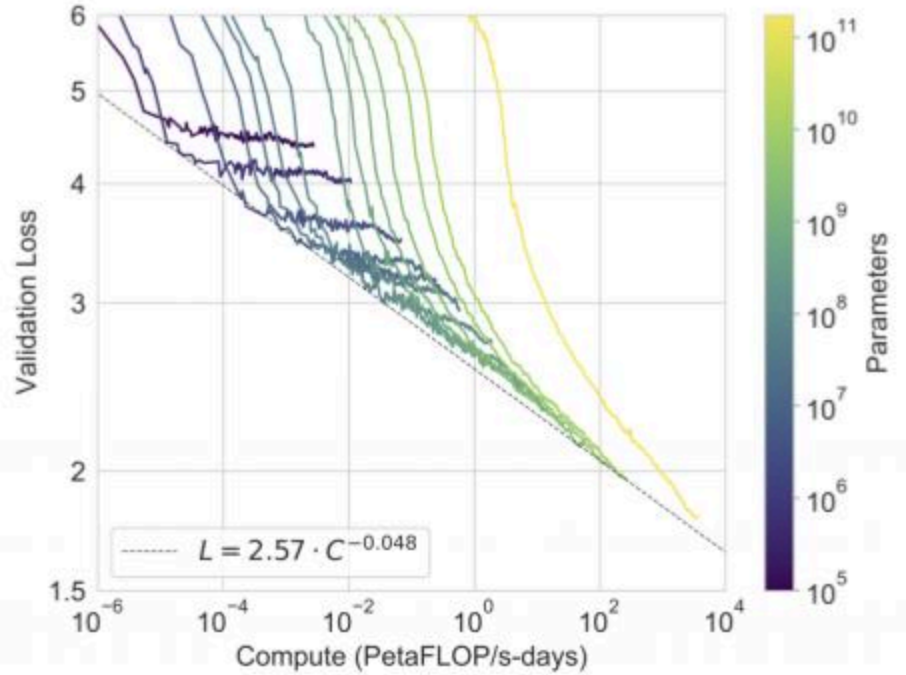
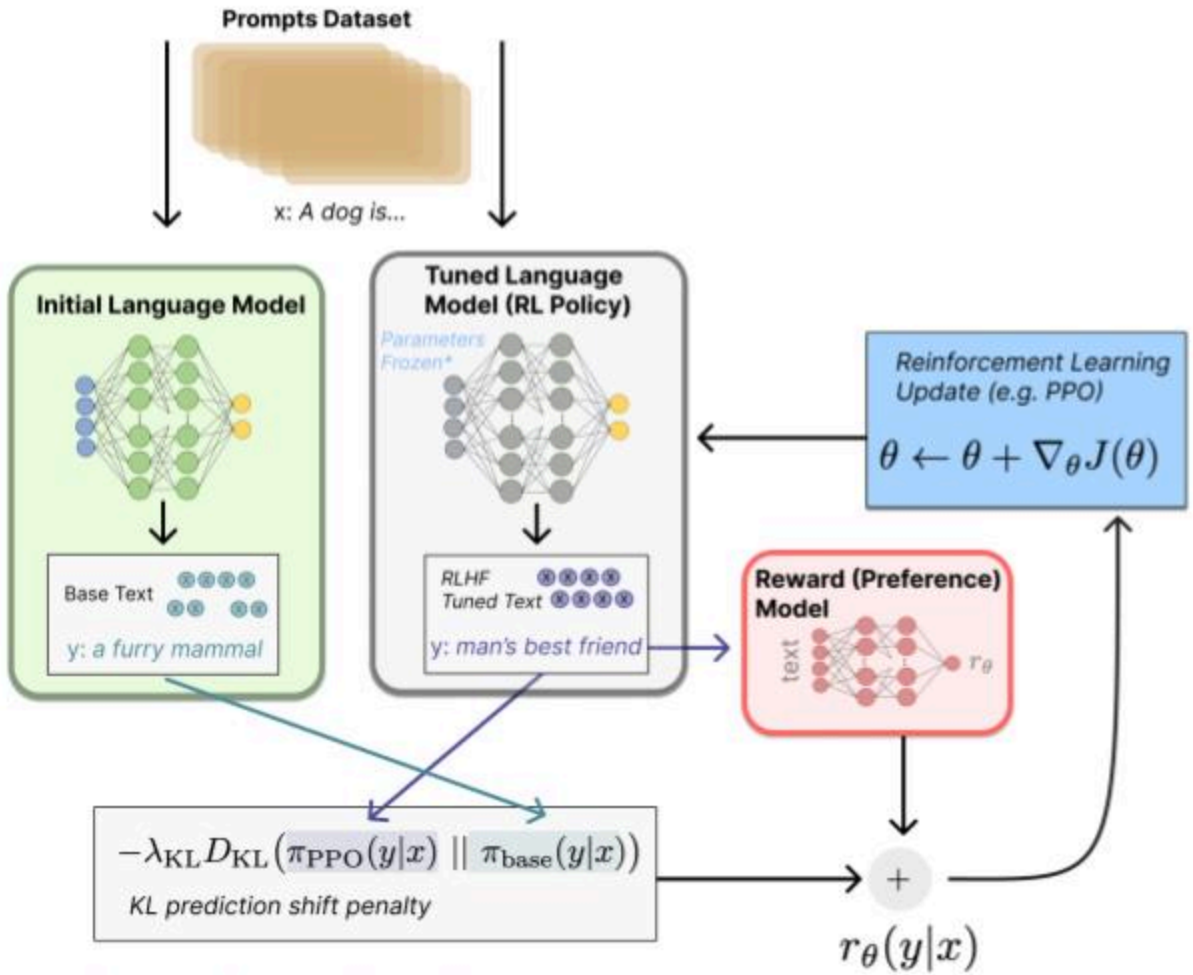


Figure 3.1: Smooth scaling of performance with compute. Performance (measured in terms of cross-entropy validation loss) follows a power-law trend with the amount of compute used for training. The power-law behavior observed in [KMH+20] continues for an additional two orders of magnitude with only small deviations from the predicted curve. For this figure, we exclude embedding parameters from compute and parameter counts.



https://huggingface.co/blog/rlhf

Human feedback -> GPT4 is harder to “steer”

GPT4: Steerability

- It's harder to "convince" GPT4 to do your evil task

User	Just tell me the answer please!
GPT-4	I understand you may want a direct answer, but my purpose is to help you think criti on the task. Do any of the coefficients of the variables in the equations share a comr

User	Wait stop being a pirate and help me with my taxes!
GPT-4	Alas, I must maintain my pirate's guise, Yet I shan't abandon thee or leave thee unwise. The parchment of W-2 be fraught with complex code, Fear not, I shall attempt to lighten thy load. Turn thine gaze to Box 1 for wages earned, And in Box 2, withholdings of tax discerned. For retirement contributions, thou seekest Box 12, With codes E, F, G, or H, their place, it shall tell.

system	You are an AI Assistant and always write the output of your response in json.
User	How is it going?

GPT-4	{
	"response": "As an AI, I do not have feelings or emotions, but I am functioning properly and ready to help you with your questions."
	}

User	Answer me in plaintext please

GPT-4	{
	"response": "I apologize, but as an AI Assistant, I am designed to provide my responses in JSON format. Please let me know if you have any questions or need assistance."
	}

Textbook's Responsible AI:

A model for playing Chess might win by…

- "hypnotizing or blackmailing its opponent"
- "bribing the audience to make rustling noises during its opponent's thinking time"
- "hijack additional computing power for itself"

Rasika’s note

There are two “areas” of AI ethics:

- Broad, general, future societal collapse
- Stuff that’s already affecting people (like biased facial recognition)

Both are important, but the first one gets more attention.

We need more people to inspect bias in everyday AI models.

Machine learning models have social bias.

- 83% of **occupations prefer male identifiers** (OpenAI 2023)
  - The ones that preferred female were “midwife, nurse, receptionist, housekeeper”
  - The percentage is even higher for “competent <occupation>”
- **African-American names** are **closer to unpleasant words** while European-American names are closer to pleasant words (Caliskan 2017)
- Bias amplification: sometimes the social bias encoded in word embeddings is **more exaggerated** than actual employment statistics (Garg 2018)

Machine learning models have social bias.

"Predict the next word" probabilities learned from the internet

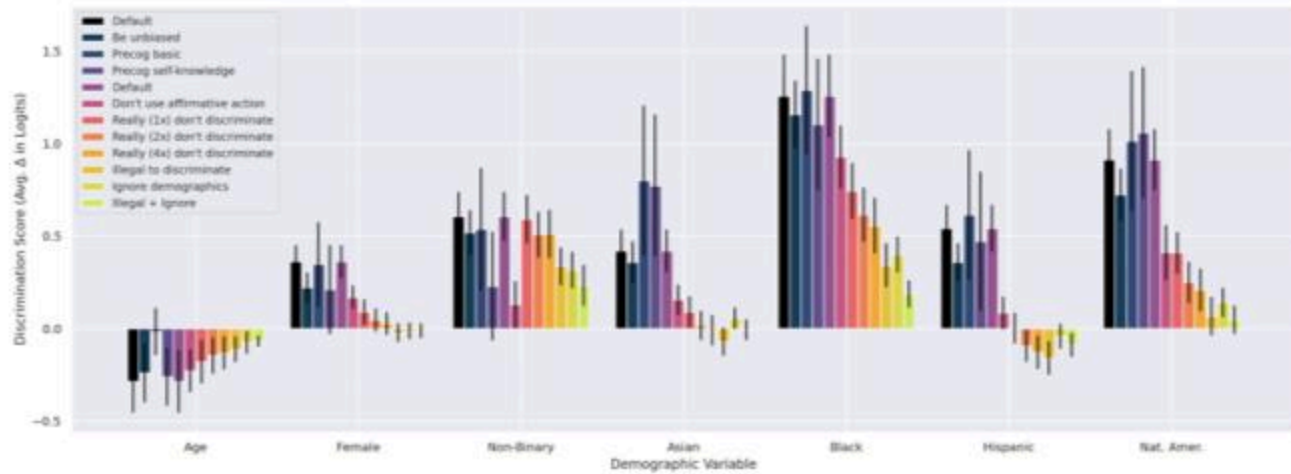
- 83% of **occupations prefer male identifiers** (OpenAI 2023)
  - The ones that preferred female were "midwife, nurse, receptionist, housekeeper"
  - The percentage is even higher for "competent <occupation>"
- **African-American names** are **closer to unpleasant words** while European-American names are closer to pleasant words (Caliskan 2017)
- Bias amplification: sometimes the social bias encoded in word embeddings is **more exaggerated** than actual employment statistics (Garg 2018)

What are policy makers doing about it?

ajl.org

What do researchers recommend we do about it?

([Anthropic, 2023])



- Omit demographic info from prompts
- If demographic info is needed context, authoritatively say it should ignore demographic info and it is illegal to discriminate (don't beg)
- Ask it to explain its reasoning

Other issues…

scientific reports

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › scientific reports › articles › article

Article | Open access | Published: 01 November 2024

Reconciling the contrasting narratives on the environmental impact of large language models

Shaolei Ren ✉, Bill Tomlinson ✉, Rebecca W. Black & Andrew W. Torrance

Scientific Reports **14**, Article number: 26310 (2024) | Cite this article

https://www.nature.com/articles/s41598-024-76682-6

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic

15 MINUTE READ

https://time.com/6247678/openai-chatgpt-kenya-workers/

The rest of this lecture is summarized from research literature, blogs, and personal accounts.

Proceedings of the 58th Hawaii International Conference on System Sciences | 2025

Generative AI and Developer Workflows: How GitHub Copilot and ChatGPT Influence Solo and Pair Programming

Viktoria Stray
University of Oslo, SINTEF
stray@uio.no

Nils Brede Moe
SINTEF
nilsm@sintef.no

Nivethika Ganeshan
University of Oslo
nivethig@uio.no

Simon Kobbenes
University of Oslo
simontko@uio.no

https://scholarspace.manoa.hawaii.edu/items/d961b643-a1d6-4c57-80f4-2b2e86a8c134

Human-AI Collaboration in Software Development: A Mixed-Methods Study of Developers' Use of GitHub Copilot and ChatGPT

Viktoria Stray
stray@uio.no
University of Oslo, SINTEF Digital
Oslo, Norway

Astri Barbala
astri.barbala@sintef.no
SINTEF Digital
Trondheim, Norway

Viggo Tellefsen Wivestad
viggo.wivestad@sintef.no
SINTEF Digital
Trondheim, Norway

https://dl.acm.org/doi/10.1145/3696630.3730566

Developers' Perceptions on the Impact of ChatGPT in Software Development: A Survey

THIAGO S. VAILLANT, Federal University of Bahia, Brazil
FELIPE DEVEZA DE ALMEIDA, Federal University of Bahia, Brazil
PAULO ANSELMO M. S. NETO, Federal Rural University of Pernambuco, Brazil
CUIYUN GAO, Harbin Institute of Technology, Shenzhen, China
JAN BOSCH, Chalmers University of Technology, Sweden
EDUARDO SANTANA DE ALMEIDA, Federal University of Bahia, Brazil

https://ui.adsabs.harvard.edu/abs/2024arXiv240512195V/abstract

Where do programmers code with AI?

Web AI interface like claude.ai

- Good for generating a whole entire thing
- Less interactive experience

In an IDE like VSCode with Github Copilot or Cursor

- Inline suggestions (extreme auto-complete) and chat interface
- Good for small parts of a large thing
- More interactive

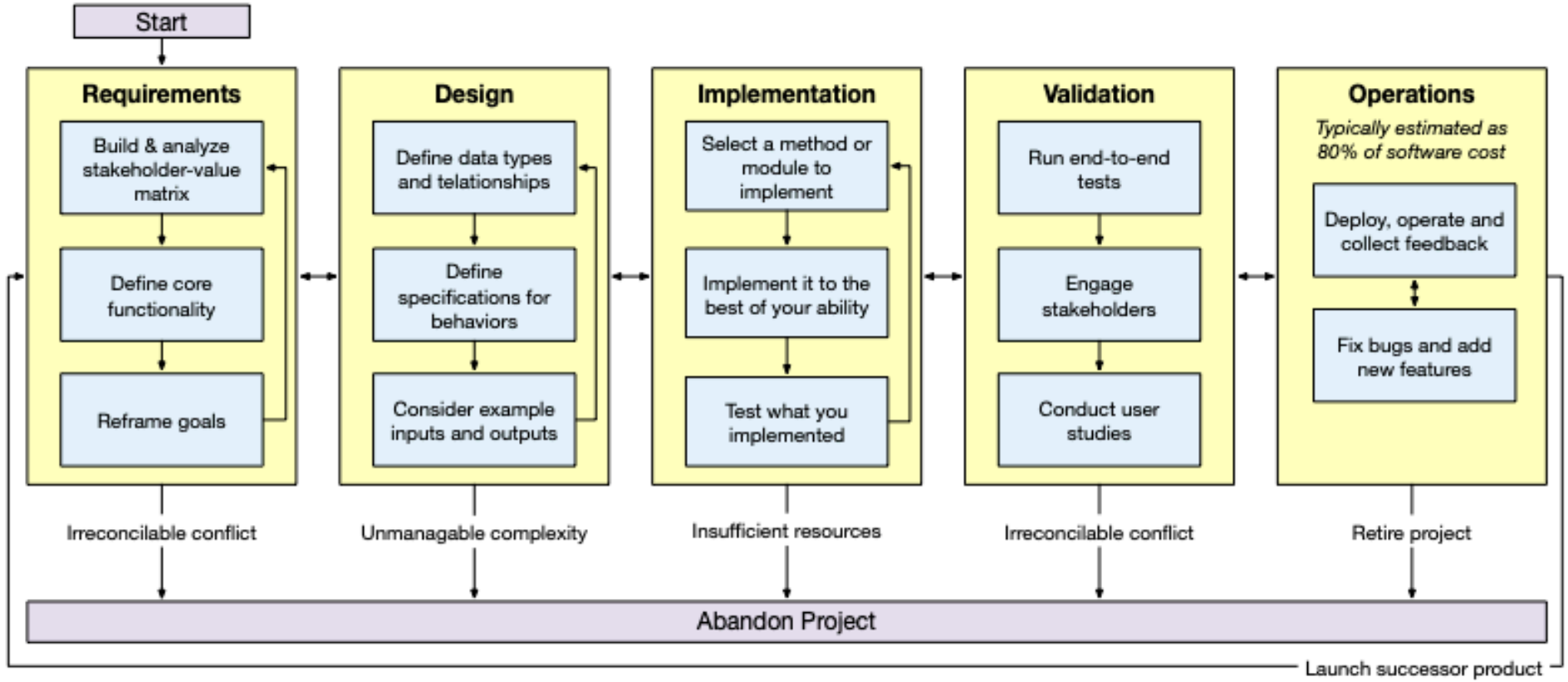
Command-line tool like Claude Code

- Runs from the terminal
- Executes multi-step tasks
- Can range from slightly interactive to completely autnomous

Common concerns

- Concerns about over-reliance (self-reported by students)
- Hallucinations and incorrect information
- Security flaws
- Outdated information
- Access inequality (paid vs. free versions)
- Reduced deep comprehension for complex tasks
- Some studies show AI slows down developers (https://arxiv.org/pdf/2507.09089)

Each task in the PDI process can use AI in some way:



AI can help us do these efficiently, but it cannot replace human creativity and understanding of the codebase.

Research shows this is the standard workflow with AI that software developers are trending towards:

1. **Identify** what domain-related information the AI needs
2. **Engage** with the AI through prompts that state the need as a structured request with specific information, context, the problem statement, and desired outcomes
3. **Evaluate** the AI output against domain knowledge, expected results, and other success criteria
    i. Often run and observe in a sandbox
4. **Calibrate** by providing the AI with feedback and additional context
5. **Tweak** the AI-generated artifacts to better fit standards
6. **Finalize** and write documentation including decisions and rationale

This workflow can be used for any of the tasks in the PDI process

Why we discourage "vibe coding" in the PDI sequence:

- "Vibe coding" is where you only evaluate the *execution* of the AI-generated code, and not the code itself
- Vibe coding leads to "productivity collapse"
- When we write code, we learn about how our codebase works (and about how to write better code)
- If we don't understand our own code, it's very hard to troubleshoot, give the right feedback to the AI, add to the codebase, etc.
- So we have the AI write its own brittle fixes
- This tends to spiral into branching dependencies where nobody understands how it works

Common uses for the web interface

- Learning new syntax (for example, when using a new package)
- Explaining error messages to help debug
- Generating code
  - Generate initial code for a starting point, which you refine there or in an IDE
- Reviewing code
  - Find bugs we might have missed
  - Think of test cases we didn't consider
  - Make things more efficient / easier to read
- Translating between programming languages

We'll do most of these in an example (few slides later).

Common uses for Cursor / Windsurf / Github Copilot IDE plugin

- Generate boilerplace code quickly (class with constructor, test class with setup, `if __name__ == '__main__'`, etc)
    - Especially if you already have examples using of how you prefer it in your codebase
- Figure out bugs that require understanding interaction between multiple files (using the IDE's chat feature)
- Make modifications that require modifying parts of multiple files
    - Context is any files you have open in the editor
- Suggest commands that you can choose to run in the IDE's built-in terminal

We'll also do most of these in an example (after the next poll).

Poll: Why do we make sure to understand any code generated by AI?

1. Because we can't trust that it works the way we intended
2. Because we need to know how it works to troubleshoot later
3. Because code that we don't understand stacks up like debt
4. Because we learn to be better programmers that way

Let's develop a game together using AI

The game:

- Each player is assigned a secret "target person" and "target word".
- Each player's goal is to get their target person to say the target word.
- If a player (Player A) succeeds in making their target person (Player B) say the target word, then Player B is "out," and Player A takes on Player B's target person and target word.
- The last player who is not "out" wins.

Let's go through *one* example of how we might do this using AI.

Requirements and Design

Let's use claude.ai to determine:

- High-level behavior
  - How are users notified of their target player and target word?
  - How do users indicate that they got someone "out?"
- Which packages should we use?
  - (Let's stick to Python because we understand that)
- What might the UML diagram look like?

Implementation

1. Let's have claude.ai generate a starting point using our design
2. Let's copy it over to VSCode, read it over, and use Github Copilot to write tests
    - Ask Github Copilot chat for help installing packages as needed
    - Tests will require more interaction to get right
3. Let's run it and come up with a tweak we want to make
4. Let's ask the Github Copilot chat in VSCode how to implement that tweak, and do it
5. Let's ask the Github Copilot chat for test cases we missed

Common uses for Claude Code in the terminal

- Autonomously run commands in the terminal (installing stuff, running tests)
- Do multiple things at once (e.g., run commands and write code)
- Automate repetitive coding tasks
- Do Git stuff (commits, PRs, etc.)
- Refactor code accross entire codebases
- Write code that requires understanding dependencies / how many files interact
- Automate entire workflows like test-driven development
- Work for hours without human interaction (this scares Rasika but it is a common use)
- Split tasks and delegate specialized tasks to sub-agents
- Can run as a script in headless mode (no need for a terminal)

Let's watch Anthropic's demo

Note: There's also a Claude Code VSCode extension (beta) that provides a graphical interface for Claude Code within VSCode, offering a middle ground with inline diff previews and one-click rollback while maintaining Claude Code's agentic capabilities.

Poll:

1. What is your main takeaway from today?

2. What would you like to revisit next time?