

Today's plan

1. What is Software Engineering?
2. How to Succeed in CPSC 310
3. Project Design
4. Important Notes

what is
see?

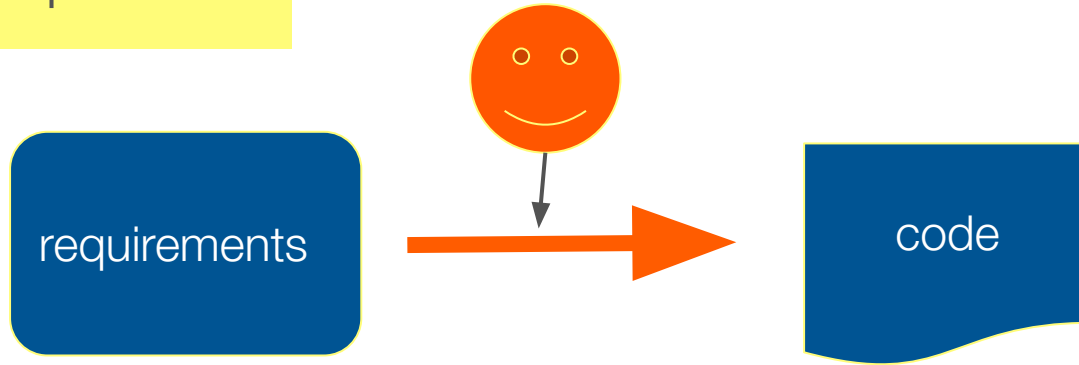
210-Centric view

The process of transforming a mental plan of desired actions for a computer into a representation that can be understood by the computer.

— Jean-Michel Hoc and Anh Nguyen-Xuan

CPSC 210

- Small Code
- Single-version
- Solo
- One prompt



310-Centric view

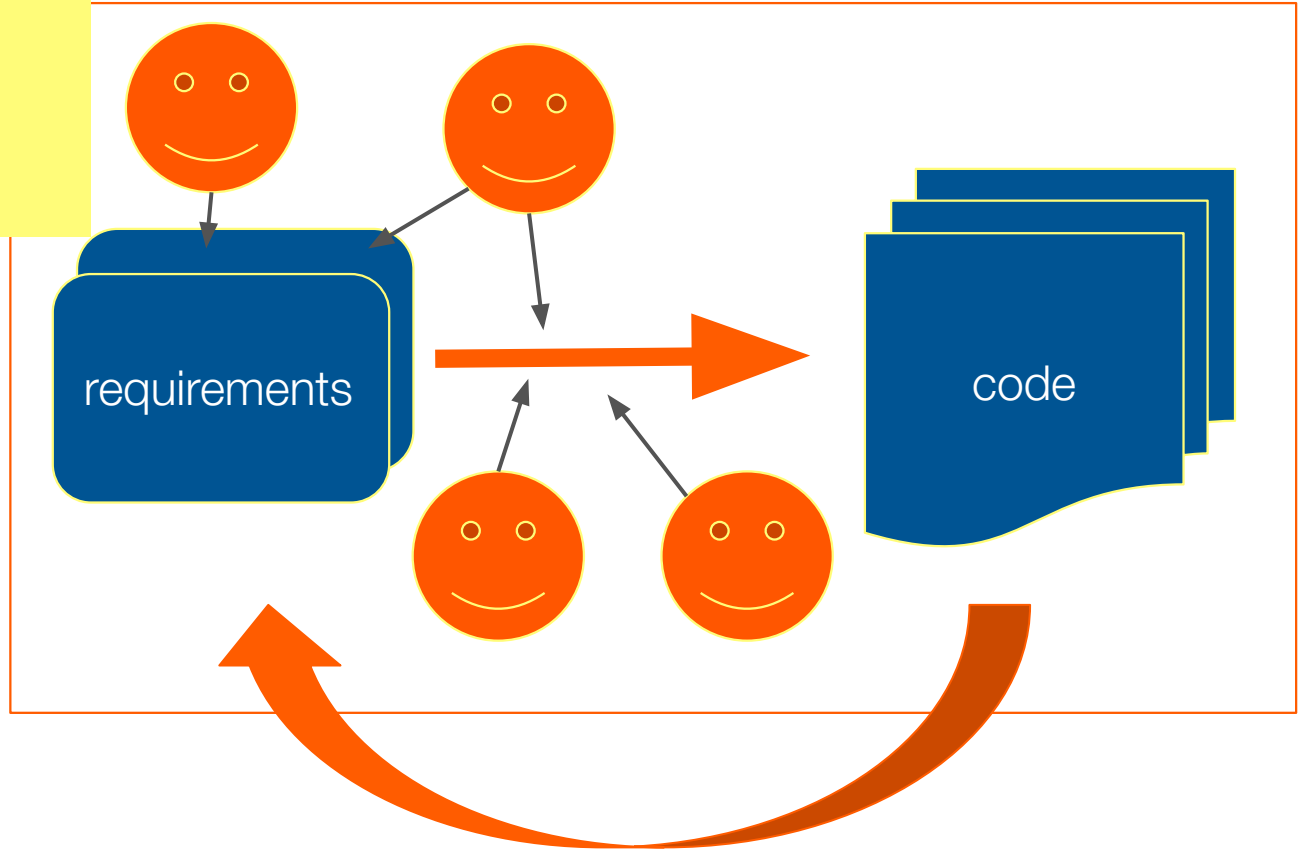
Socio-technical

The process of transforming a mental plan of desired actions for a computer into a representation that can be understood by the computer.

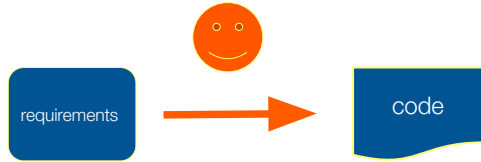
— Jean-Michel Hoc and Anh Nguyen-Xuan

- Large codebase
- Multi-version
- Collaborative
- Changing Requirements

CPSC 310

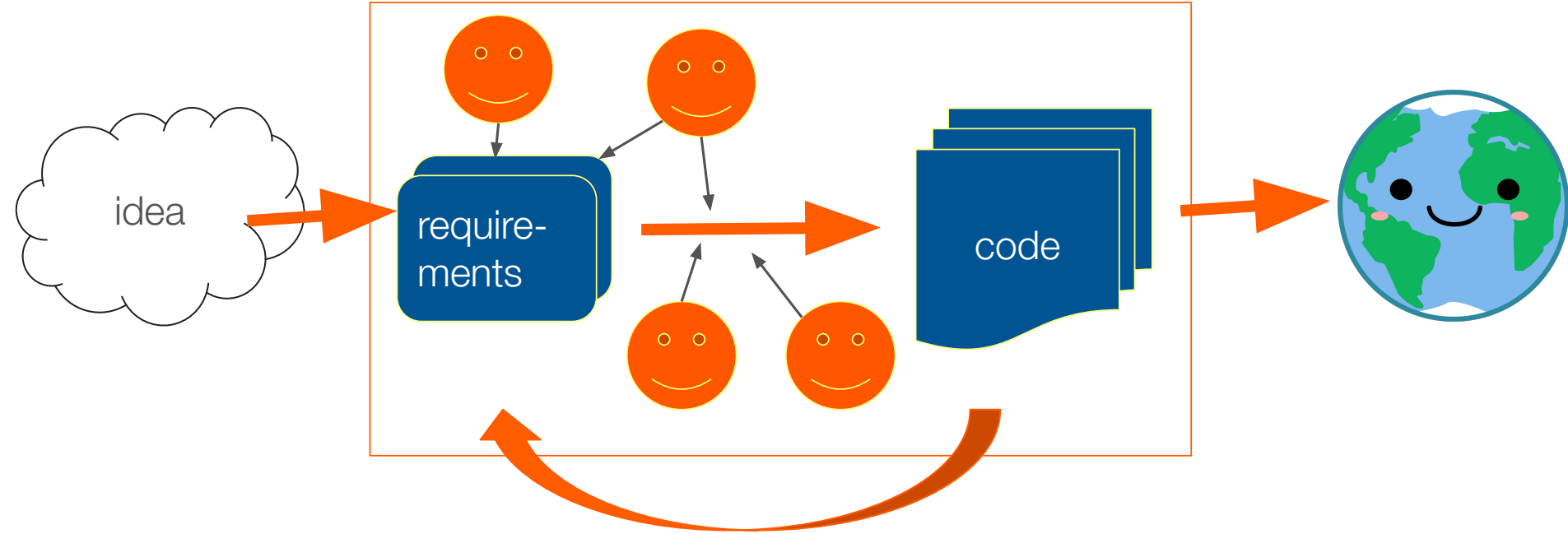


CPSC 210



Software Engineering

CPSC 310



Software engineering is the discipline of managing what a change costs as a system keeps growing.

Previous
Practice

Traditional
Development
Practice

Users/
Customers



<needs,
feedback>



Technical
Teams

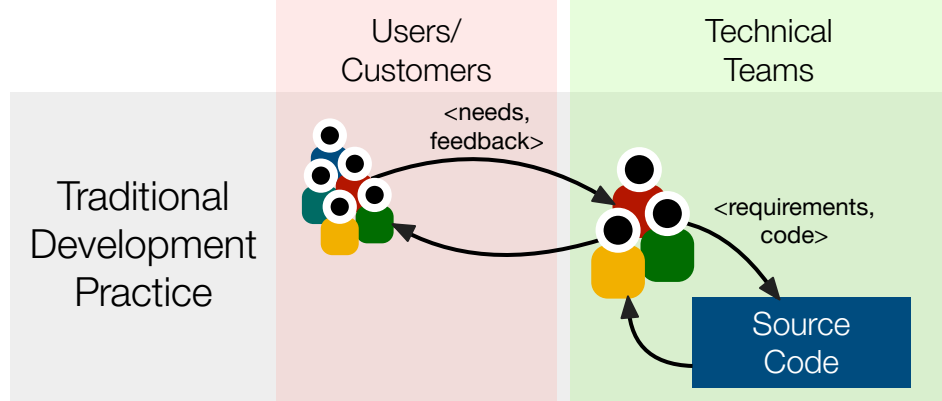
<requirements,
code>

Source
Code

Engineers need to know
how to write and read
source code

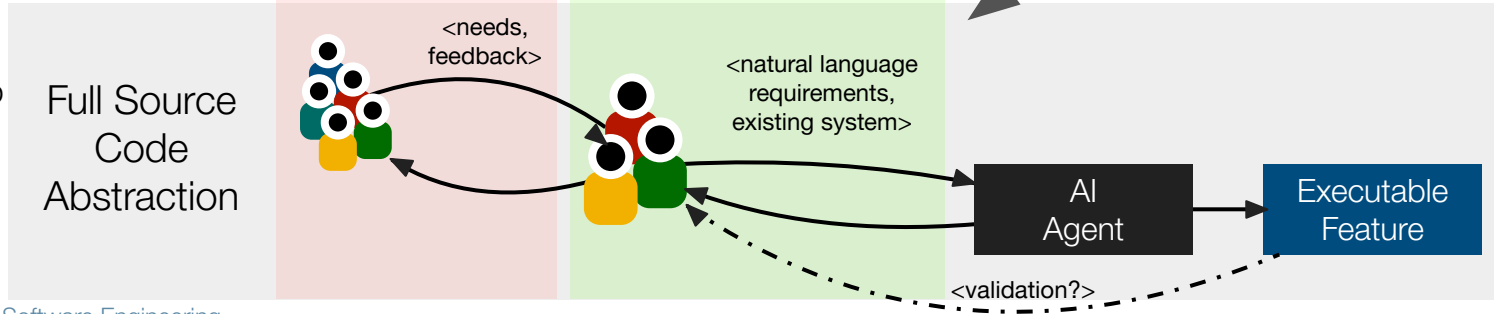
Increasing
Abstraction

Previous
Practice



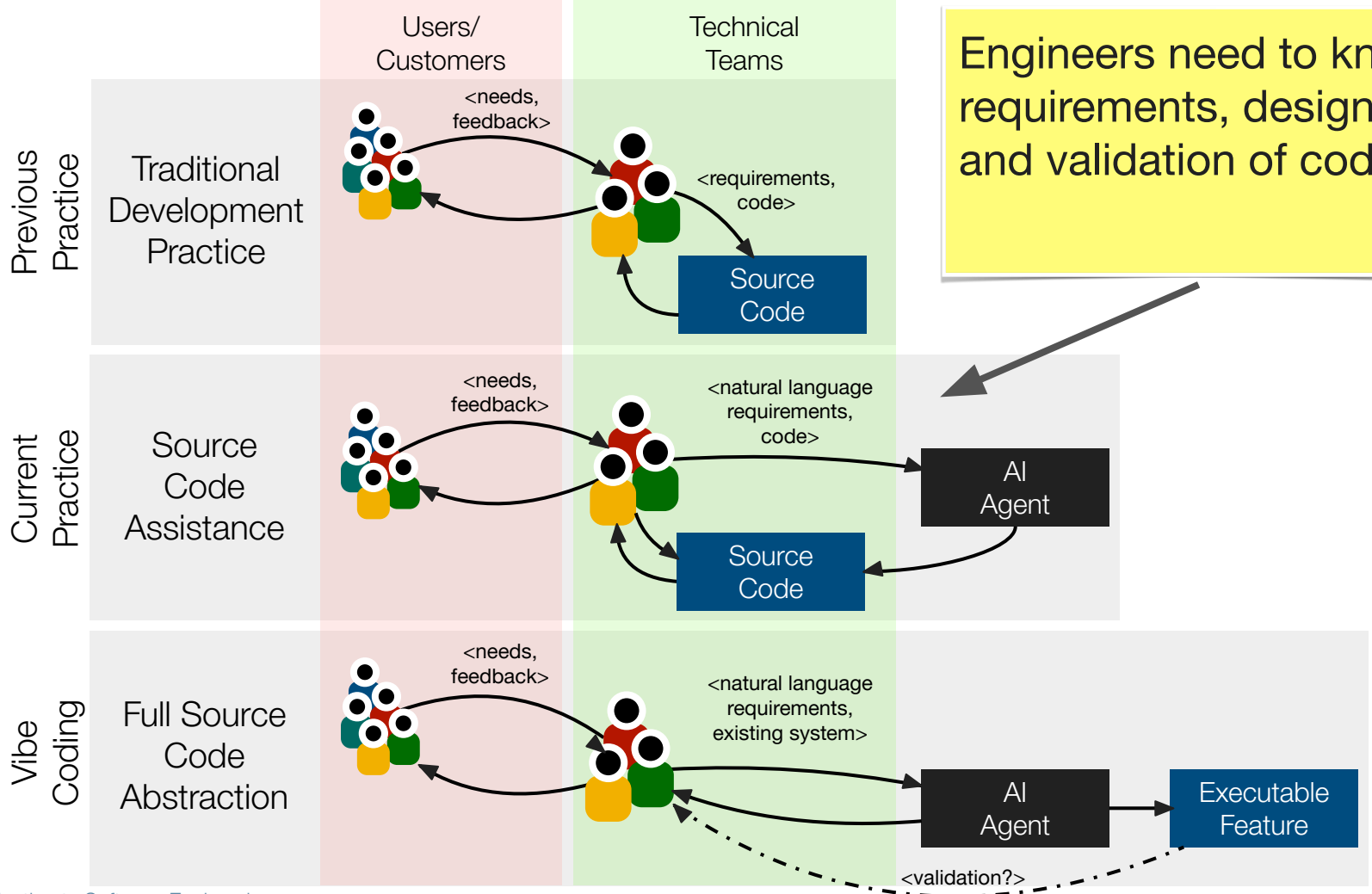
Engineers need to know
... ???????

Vibe
Coding



Engineers need to know requirements, design, and validation of code

Increasing Abstraction



Fluency

Requirements fluency: Tracing between design and implementation decisions and their requirements.

Decomposition fluency: Knowing how to break problems apart, and critically evaluating how others have done the same.

Validation fluency: Confirming that an implementation realizes its design today, and will continue to afford change as needs evolve.

Critical thinking is more important than ever. If software engineering is the accountable stewardship of systems that people depend on, then **the core skill of the profession was never writing code. It was always the judgment to know whether a decision or implementation was right, and the ability to say why.** GenAI can generate code, but people must retain agency over the systems that shape society.

Course Learning Objectives

1. Reason about how a codebase's structure influences the cost of changing it.
2. Use abstraction to make changing a system cheaper at increasingly larger scales.
3. Elicit, deconstruct, and refine requirements, and write specifications precise enough to check.
4. Evaluate the socio-technical trade-offs in collaborative software engineering processes.

Semester Roadmap

Cost of Change at bigger radii:

Scale	Weeks	The Question
Inside a codebase you own	2–5	Why is this change expensive here and cheap there?
At its boundaries	6–7	Where should the boundaries be, and what happens once you publish one?
Across a system and team	8–13	How does a team change one codebase without breaking each other?

How to succeed in this class

- **Show Up & Engage:** Lectures inform the project and exams. You will only learn as much as you choose to.
- **Start Early & Iterate:** Deliverables get progressively more involved. Good solutions rarely happen on the first attempt.
- **Collaborate & Seek Help:** Diversity in thought brings better solutions. Use your classmates & TAs to your advantage.
- **Reflect & Internalize:** Thinking about your learning & development processes solidify them.

Grade Breakdown

Component	Weight
Participation	5%
Lab assignments	10%
Project	20%
Midterm	25%
Final exam	40%

Practice

- Labs & lab assignments!
 - Marked on completion only
- Project deliverables

All will have you identify and articulate trade-offs, implement changes, and reflect on the process and results.

Assessment

- Midterm (Oct 29 @ 7pm)
- Final Exam (TBA)

All will have you identify and articulate trade-offs, implement changes, and reflect on the process and results.

On GenAI

Use at your own peril...

The Project

Getting help

- Multiple TA office hours / week
 - Hours closest to the deadlines will always be busiest.
- In Lab
 - Every lab will be staffed by multiple TAs.
- Piazza
- Instructor office hours



Important Notes

- Office Hours:
 - Will be posted on the course website
 - Office hours will start next week
- Labs:
 - Start tomorrow! The first lab will help onboard you to the project
 - Are required: it is your opportunity to practice course material
 - Over Zoom
 - Assignment due at start of *your* next lab: graded on completion
- Project:
 - First deliverable is posted
 - Will start provisioning repos soon