Lesson 1: Course Overview

CS 438
Ali Aburas PhD

Today’s Goals

- Course overview and expectations
- What is meant by software engineering.
- Join the class site:
- https://canvas.instructure.com/enroll/

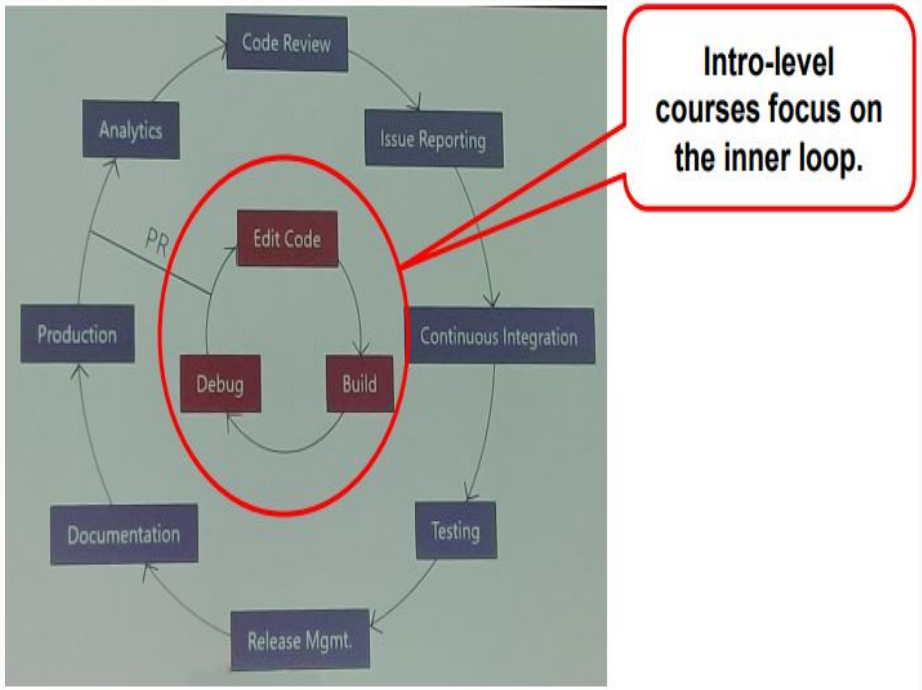
What is Software Engineering?

Software Engineering is the complete process of

- **specifying**, requirements engineering, specifications, documentation
- **designing**, software architecture and design, UI
- **developing**, programming (just one of many important tasks)
- **analyzing**, testing, debugging, linting, verification, performance engineering
- **deploying**, DevOps, CI, packaging, operation, remote diagnostics
- **maintaining** refactoring, extensions, adaptation, issue tracking
- **Writing** (design) docs?

a software system.

The Role of Software Engineering in Practice



CS438/CS338 largely focuses on the outer loop.

Why is Software Engineering important?

Software engineering importance stems from the need for systematic methods to develop and maintain software systems that are reliable, efficient, and secure.

- **Quality Assurance**: Software engineering incorporates rigorous testing and validation processes to guarantee that the software meets high-quality standards. This focus on quality assurance helps minimize defects, thereby reducing the risk of software failures, which can result in costly downtime and harm to reputation.
- **Cost Efficiency**: Structured methodologies in software engineering optimize development processes, reducing both time and costs associated with projects.
- **Scalability and Flexibility**: Software engineering principles prioritize the creation of scalable applications that can adapt to changing business needs.
- **Collaboration and Communication**: The software engineering process encourages collaboration among diverse teams, including developers, designers, and stakeholders.
- **Risk Management**: Software engineering employs risk management strategies to identify, assess, and mitigate potential issues throughout the development lifecycle.
- **User-Centric Design**: Software engineering emphasizes the importance of understanding user needs and incorporating feedback into the development process.
- **Continuous Improvement**: The iterative nature of software engineering allows for continuous improvement based on user feedback and performance metrics.

Why is software development so hard?

- Why is it so difficult to build good software?
  - Software systems are very varied
    - Applications: web, smartphone
    - System software: operating systems, compilers
    - Communications: routers, telephone switches
    - Real time: air traffic control
    - Embedded software: device drivers, controllers
    - Mobile devices: digital camera, GPS, sensors
    - Information systems: databases, digital libraries
    - Offices: word processing, video conferences
  - Client/Customer requirements are very different.
  - There is no best language, operating system, platform, database system, development environment, etc.
  - A skilled software engineer/developer
    - knows about a wide variety of approaches, methods, tools, and,
    - selects appropriate methods for each project and apply them effectively.

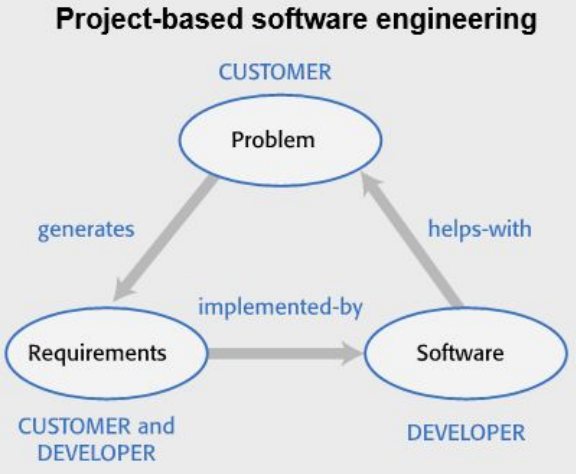
What is Good software?

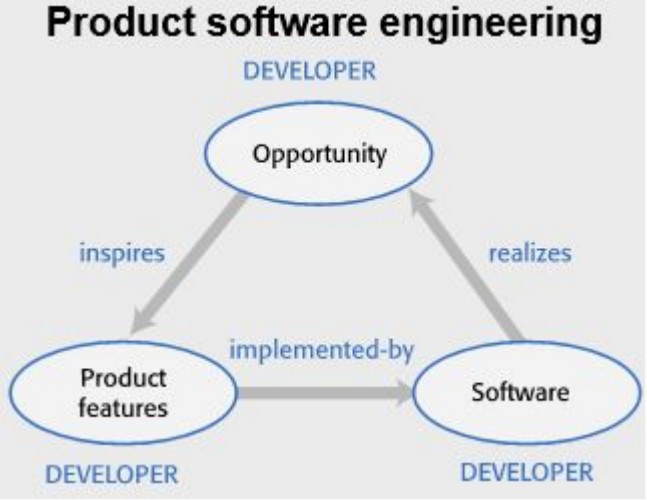
The quality attributes of good software:

- Reliability
- Testability
- Efficiency
- Usability
- Maintainability
- Integrity
- Re-usability
- Dependability
- Portability
- Availability

Project-based vs. Product-based

- **Product software engineering (Generic products)**
  - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
- **Project-based software engineering (Customized products)**
  - The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.





Frequently asked questions about software engineering

Frequently asked questions about software engineering

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

Frequently asked questions about software engineering

Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

What can you learn in CS 438

- Learn best software development best practices
- Understand how software is produced – from conception to continuous development and release
- Develop skills to effectively collaborate with others towards a common delivery goal
- Experience the responsibilities, issues and tradeoffs involved in making decisions as software engineers

Much of the above is grounded by working with a team to incrementally deliver a real software product/service

Expectations

- Ability to program (in any programming language such as C++, Python, Java, or PHP).
- Active participation in discussions.
- Teamwork and communication.
- Reflect on and improve your submissions.
- Go beyond adequate.

CS438: challenges for students

- **Team work**
  - Effective communication and coordination
  - Different backgrounds, skills, and incentives/motivations
- **Complexity**
  - Tooling and technology stacks
  - Scale of code base
- **Uncertainty**
  - No simple check-box grading
  - Trade-offs, decisions, and justifications

What’s next?

- Work on project requirements engineering.
- Software process models (SDLC)