CS 3300-A Introduction to Software Engineering

Lecture 01: Introduction & Overview

Dr. Nimisha Roy ▸ nroy9@gatech.edu

These slides are adapted from Alessandro Orso and Mahdi Roozbahani

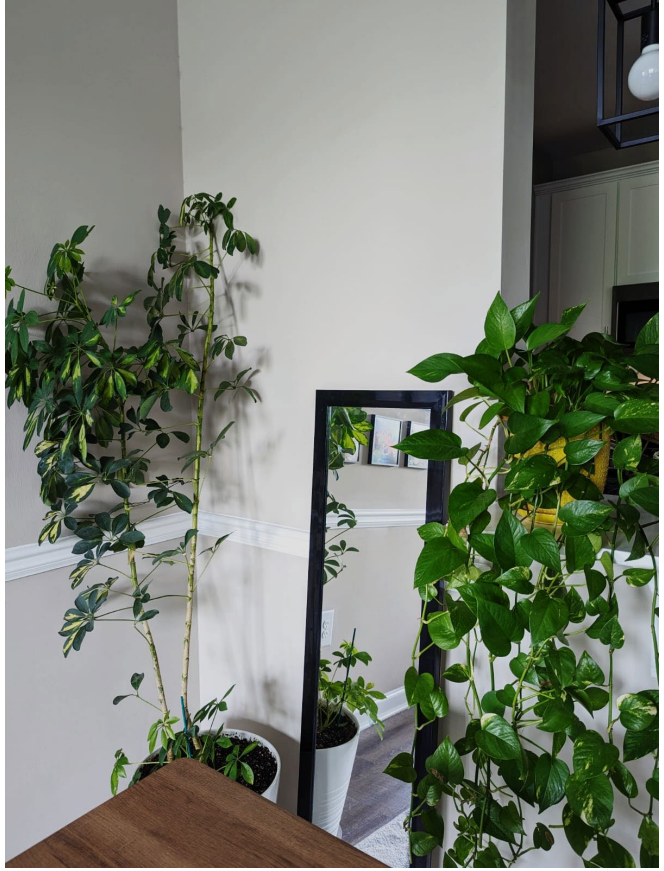
Contents

- Course Overview
  - Introduction and Big Picture
- Course Logistics
  - Schedule, Class Organization, Projects, Team Formation, Policies

Your Instruction Team

Instructor: Dr. Nimisha Roy

- 4th year teaching this course
- Lecturer in the College of Computing (SCI/OMSA/OMSCS)
- PhD in Computational Science & Engineering from Georgia Tech



Research Interests: Computing Education, Gen AI in education, Sustainability in CS education, Intelligent infrastructure, Software Development

Hobbies: singing (jamming), interior designing, caring for my plants

TA Team

- Neha Raikar: Head TA
- Avinash Atluru: GTA
- Krish Ranglani: UTA

Course Overview

What is Software Engineering?
What do software engineers do?

- Software Engineering – The discipline allows systematic application of methods to build and manage high-quality programs/softwares.

- Software engineers are engineering or computer science professionals who combine engineering principles and programming.
  - End-to-end software life cycle planning and execution
  - Maintaining Disciplined Product Control
  - Use modern programming practices
  - Perform Continuous Validation and Improvement of the process

Industry Roles

- **Front-End Engineer/Developer:** Creating user interfaces and focus on the visual elements, fixing bugs and user experience (UX).
- **Back-End Engineer/Developer:** Focus on the core logic and performance of the application while taking scalability and integration into account.
- **Full Stack Engineer/Developer**: Handle both front-end and back-end, which means that they create a full application on their own.
- **QA Engineer/Analyst:** Writes the software that tests for quality. QA ensures the product or processes run as expected.
- **DevOps Engineer:** Build and maintain back-end software and distributed systems, such as database servers and application infrastructure.
- **Security Engineer:** Use their ethical hacking skills to test the security of software systems for vulnerabilities that need to be fixed.

What do software engineers do? Current landscape - **What has changed?**

- Introduction of AI tools like GPT and Copilot into software development processes.
  - AI tools are being integrated into coding, design, and testing phases, automating tasks that were traditionally manual.

- AI is reshaping roles and tasks within the software industry
  - emergence of new roles such as AI Integration Specialist, AI-assisted testing engineer, Prompt engineer, and Ethical AI Advisor, reflecting the need for expertise in navigating the complexities introduced by AI technologies.

- Sustainability and its integration into software engineering to meet global needs.
  - growing importance of sustainable software practices that aim to reduce environmental impact and promote social responsibility within software projects.

The course CS 3300 this semester is a partial CURE (Course Undergraduate Research Experience)

We will be integrating Gen AI into the Software Engineering process throughout the semester.

How do you benefit from it?

- CURES are great to have an experience with within your curriculum
- Can help you explore/reinforce your research interests
- Helps you gain practical skills gained from using AI in real-world software projects.
- Helps you get a taste of software engineering industry relevant skills of the future – prepares you for current and future industry demands.

What is Software Engineering?
Why is it Important?

What is this?

[ ] 4th of July fireworks

[ ] Flare gun in action

[✓] Explosion of Ariane 5 rocket due to Software errors

https://www.bugsnag.com/blog/bug-day-ariane-5-disaster

What is Software Engineering? Why is it Important?

Why is it so difficult to build good software??

Topic of this course and why Software Engineering is an important course in Computer Science

What are attributes of a good software?

- Functionality
  - The software should deliver the required functionality and performance to the user
- Usability
  - Should be easy to use. Not unnecessarily complex
- Maintainability
  - Software must be evolvable to meet changing needs
- Dependability
  - Software must be trustworthy (reliability, security, and safety)
- Efficiency/Productivity
  - Time & Cost effective

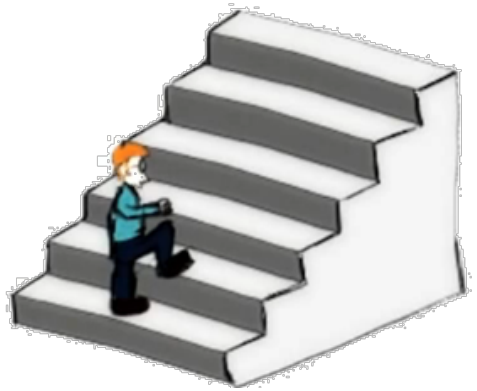
Some questions

- What is the largest software system on which you have worked?
- How many LOCs/day were you producing?
- How many LOCs/day professional software engineers produce?

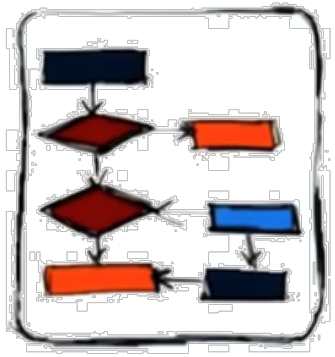
  < 25? 25-50? 50-100? 100-1000? > 1000?

- But what are they doing with the rest of their time?
- How do large systems get built?
- What process should be followed?
  - No one size fits all
  - We will see several

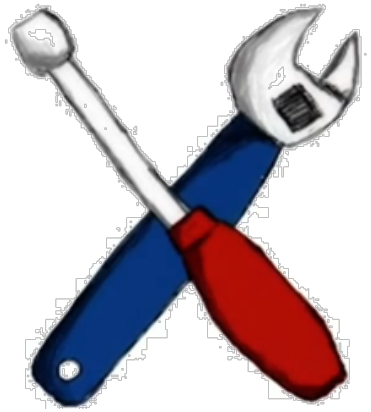
Software Engineering



Methodologies



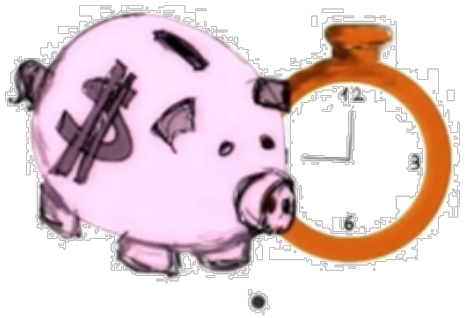
Techniques



Tools

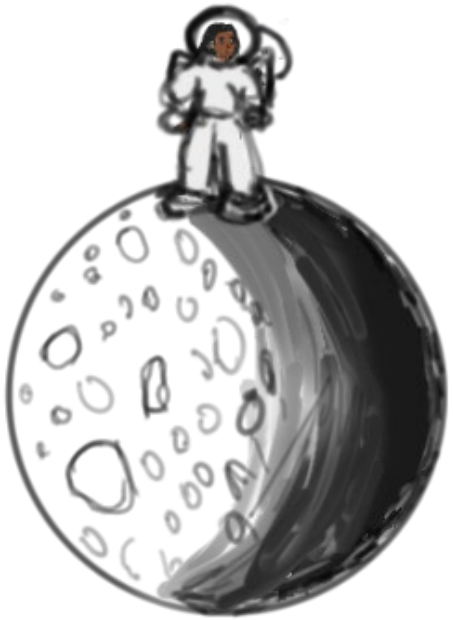
To build software of high quality

That works



And fits into Budget

The 60s



Man on Moon

Polaroid

History

https://en.wikipedia.org/wiki/Software_crisis

History

Reason 1: Rising Demand for Software

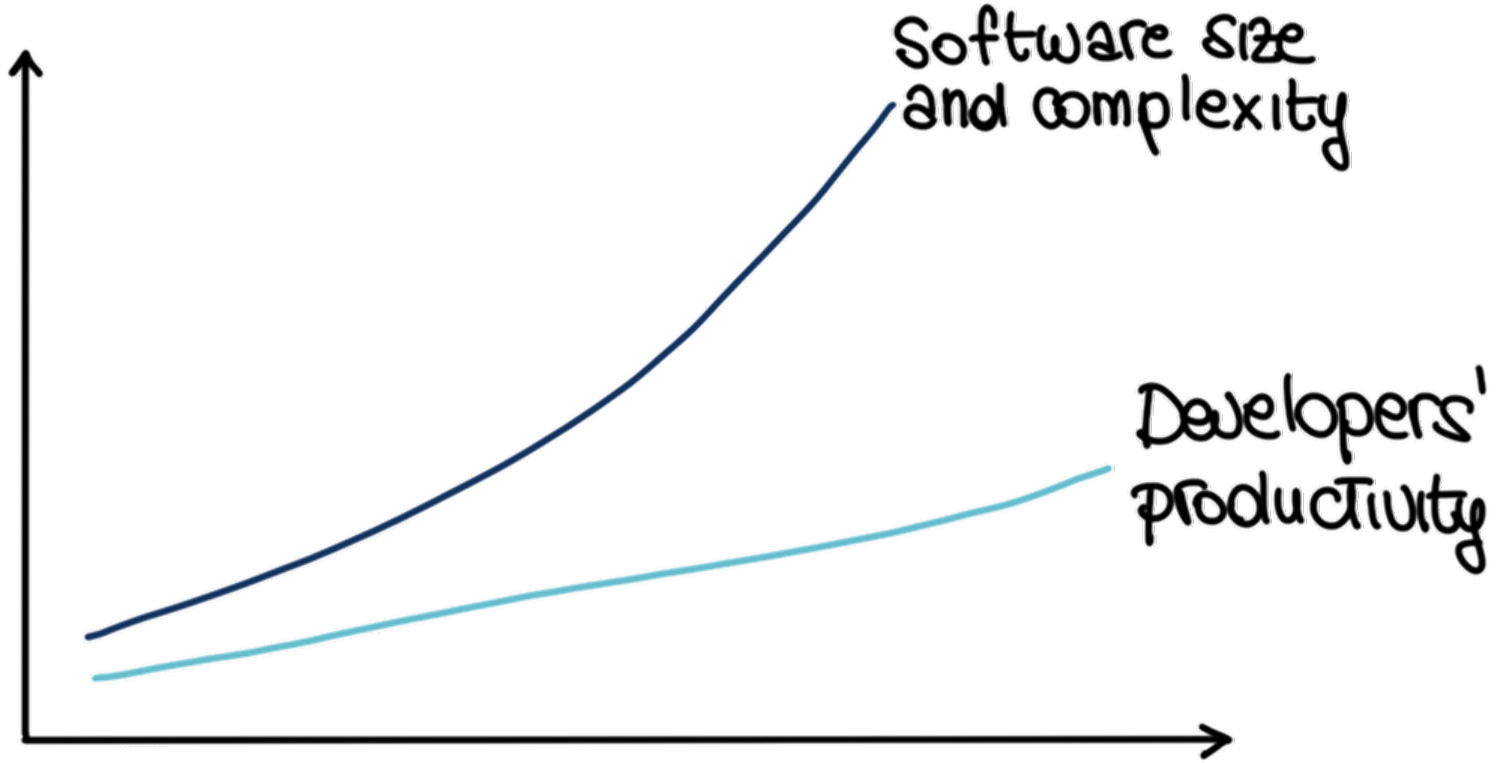
History

Reason 2: Increasing product complexity

SIZE	EXAMPLE	
10^2 LOC	Class exercise	Programming effort
10^3 LOC	Small project	
10^4 LOC	Term project	
10^5 LOC	Word processor	Software engineering effort
10^6 LOC	Operating system	
10^7 LOC	Distributed system	
...	...	

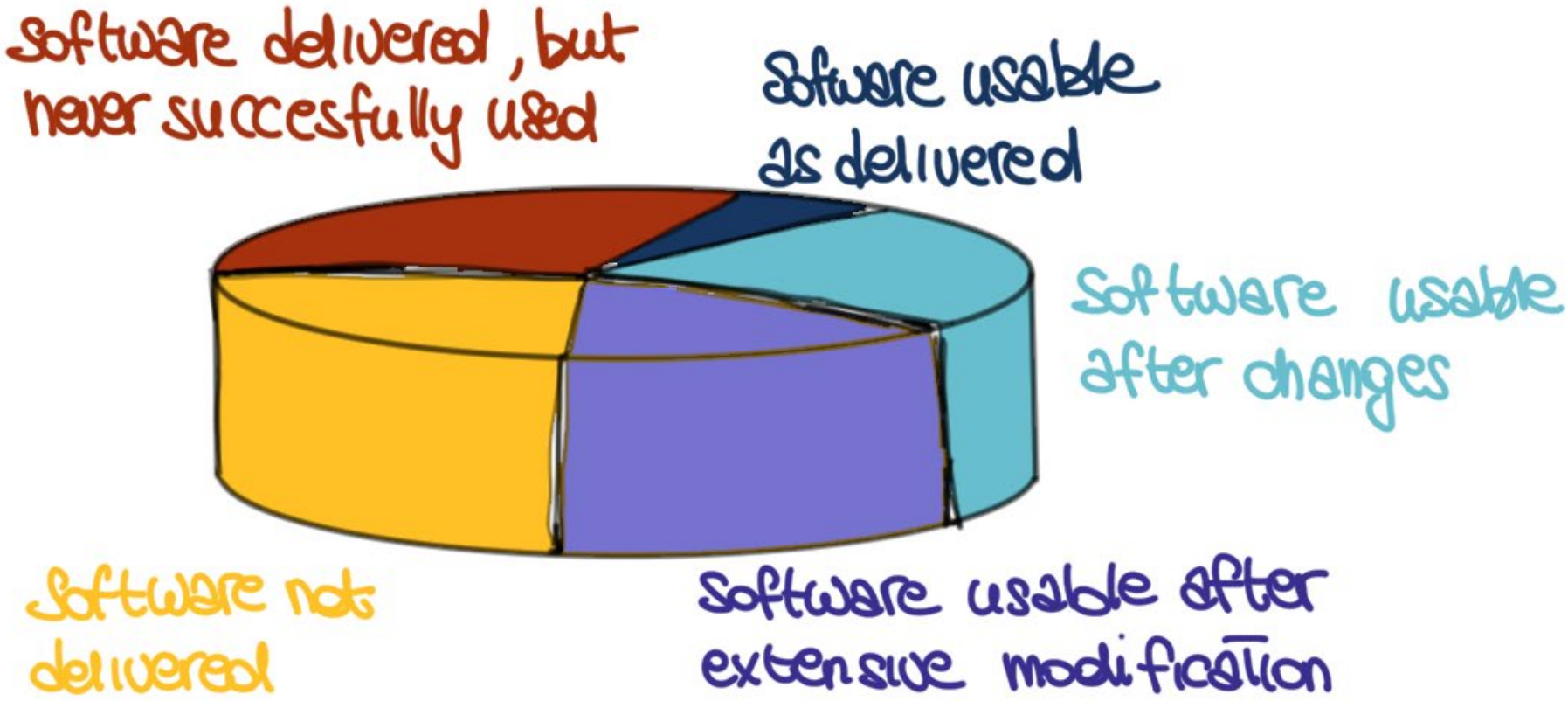
History

Reason 3: Developers Productivity Growth



History

Study of 9 Development Contracts (Davis, 1990)

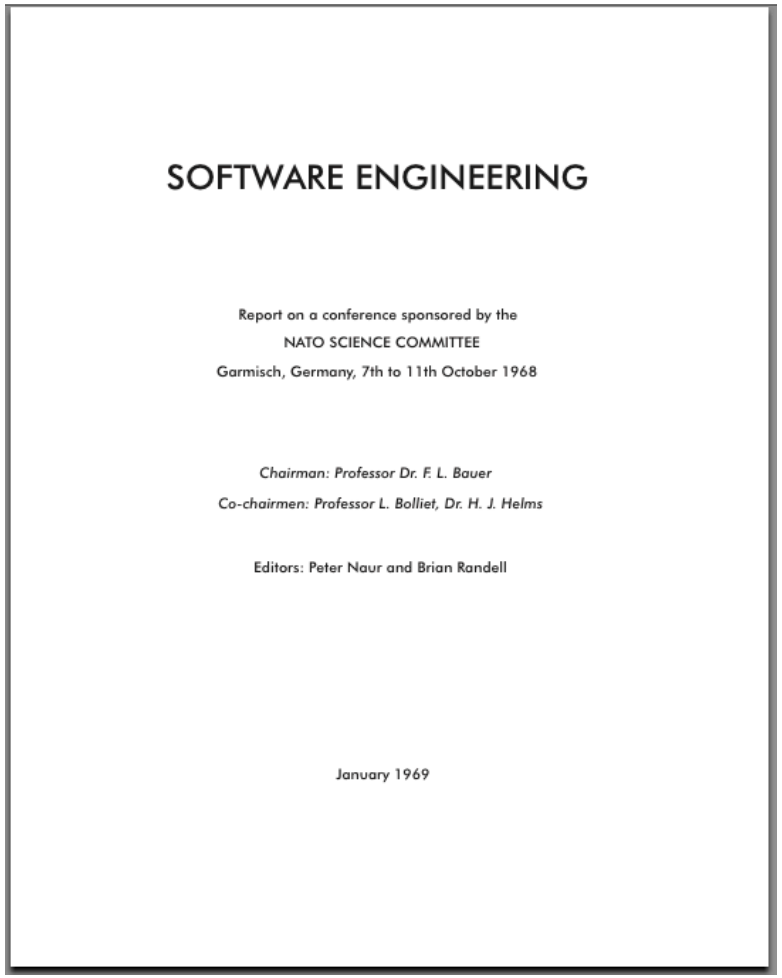


History

Birth of Software Engineering

History

Birth of Software Engineering



SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

January 1969

Link to proceedings:
http://homepages.cs.ncl.ac.uk/brian.randell/NATO/

What are common causes of SW failures?

- No standard procedures for development
- Inadequate understanding of requirements
- Sheer complexity of software (e.g., concurrency, distribution)
- Size of project (too large for a single manager)
- Difficult to match technical knowledge of staff with project needs
- Poor design/implementation/testing methodology
- Requirements change during project
- Poor documentation
- Force fitting software components to applications
- Changing/reusing code without understanding it
- Poor management: lack of communication, poor cost/schedule estimates
- Unrealistic expectations
- Lack of measurement
- Lack of teamwork
- Performance differences among staff
- …

Software importance today

- More and more systems are software-controlled
- The economies of ALL developed nations are dependent on software
- Expenditure on software represents a significant fraction of GNP in all developed countries
- AI technologies integrated into every stage of the software development lifecycle.
- AI-Driven Innovations: Examples include AI-assisted code completion, code generation, test case generation, design diagram generation, predictive analytics in software design, etc.
- Impact on Productivity and Efficiency: AI tools streamline processes, reduce development time, and enhance software quality and accuracy.

What are the key challenges facing SE?

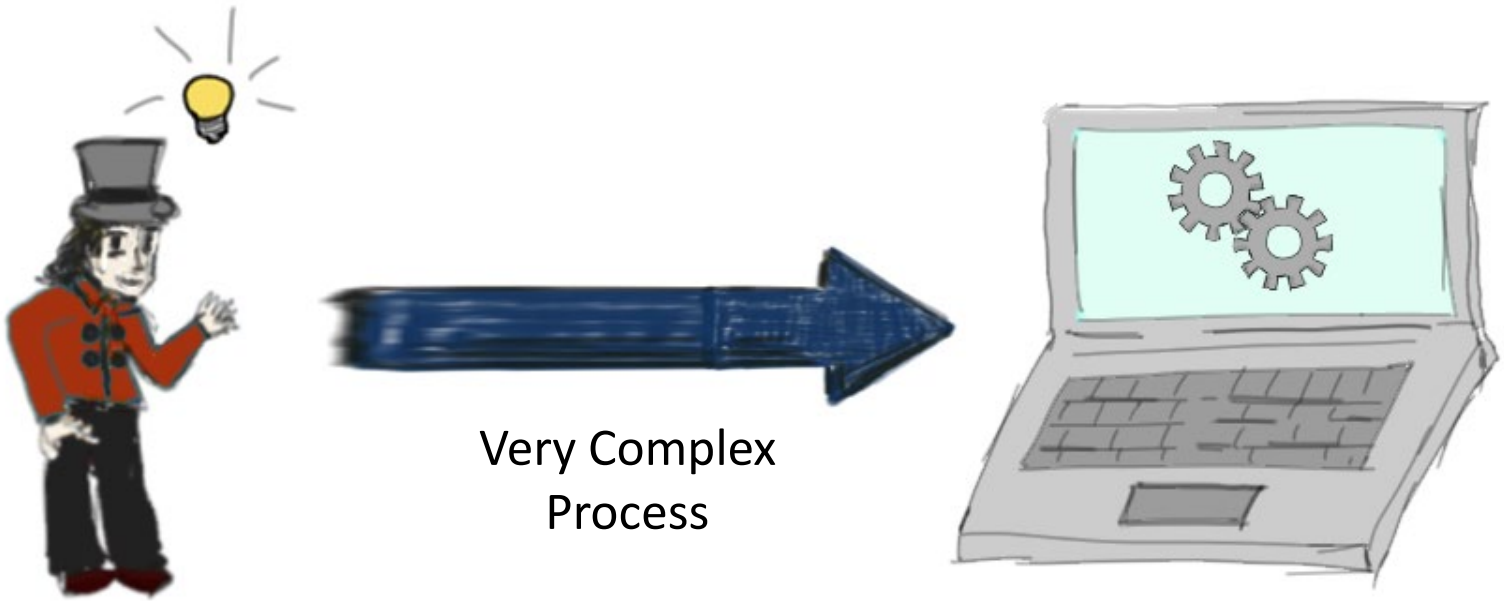
- How can we build high-quality systems?
- How can we do it in a reasonable time?
- How can we do it at a reasonable cost?

- Effective Integration of AI: Overcoming the complexity of integrating AI tools into established software engineering workflows.
- Educational Adaptation: Updating curricula to include AI-focused software engineering practices to prepare graduates for the evolving industry demands.
- Quality Assurance with AI: Ensuring that AI-generated code or designs maintain high quality and adhere to industry standards.

Software Process, Phases, Tools of the Trade

Abstract Idea

System Implementation



Very Complex Process

Software Process

- Systematic
- Formal

Software Process, Phases, Tools of the Trade

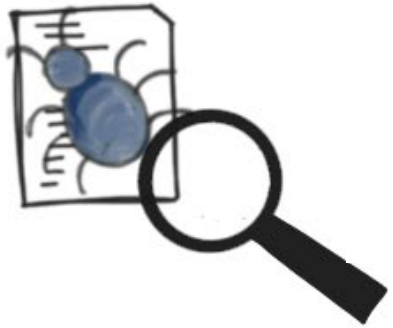
Software Phases

Requirements Engineering

Design

Implementation

Verification & Validation

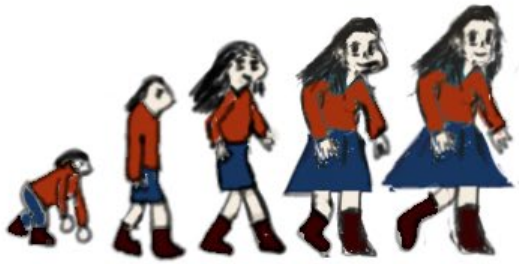


Maintenance

Software Process, Phases, Tools of the Trade

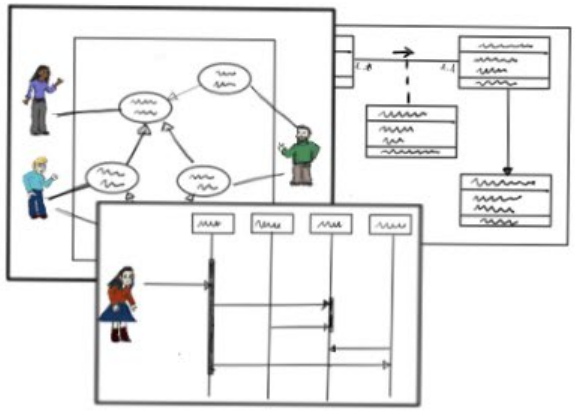
Software Process

Waterfall



Evolutionary Prototyping

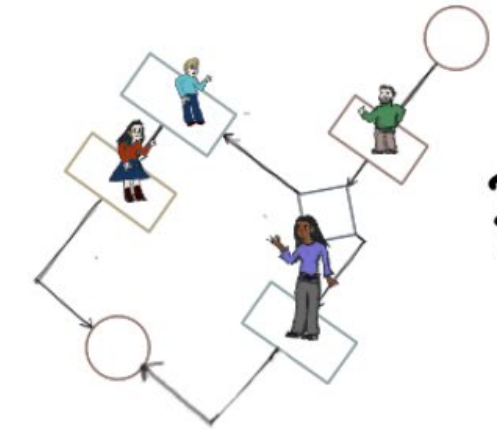
RUP/USP



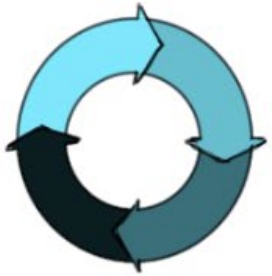
Agile

Course Logistics

Course Overview



Process



Phases

Tools

Projects

Class organization

- Class Website and Canvas
- Ed Discussion
- In-class Lectures
- Attendance is required
- Tools of the Trade Demonstrations – Bring Laptops to class.
- Team-work
- Some research

Refer to:

Canvas

For anything (syllabus, lectures, helpful links, logistics etc.) related to this class

Ed Discussion

For any communication with the instruction team. No Emails please!

Introduce yourselves on Ed ☺

Grading Distribution

- Assignments (Individual) 20%
- Project 1 (group) 35%
- Project 2 (group) 30%
- Team Management/ CATME Peer Review (Individual) 8%
- Quizzes/Lecture Activities (Individual) 8%
- Survey Submissions (Individual) 3%

Tools of the Trade Lectures

Demonstration in Lectures (lets check the living schedule) – Bring your Laptops

- Version Control System (Look at course prerequisite knowledge: You should know basic Git and code review from 2340)
- IDEs
- Front-End Development
- Back-End Development
- Google Cloud Platform
- Basics of how to use AI tools

Information about projects

- Two projects
- **Project 1:** WEB-APP deployed on Google Cloud Platform. All teams will do the same project. You will use AI tools for implementation and design.
- **Project 2:** You will choose the project. It can be web- or mobile-app. You will use AI tools for implementation, design, and testing.
- Team based
  - Different teams
  - Grades will be adjusted based on peer review
- Tools/environments
  - Project 1 – Springboot, Maven, GCP, GPT 4o API, Copilot, Lucidchart
  - Project 2- GCP but tech stack upto you. AI tools.

AI Tool Integration

- You will choose and experiment with AI tools for implementation, design, and testing.
- You will document your findings according to the metrics provided in the assignments
- Focus is on the basic implementation of software features in projects and more points allocated to your AI integration/reflection aspects
- We are providing primer videos on how to use AI for:
  - Design – lucid chart video on the website (it's free). Many design AI tools are free.
  - Implementation – GPT 4o API credits video on website
  - GPT 4o API credits will be provided to each team - $100 credit
  - Implementation – copilot video on the website (it is free)
  - Testing – video will be published after project 1.
- Each individual will be allocated $50 (+$50) worth of Google Cloud credits
- I will be covering basics of prompt engineering next class

Team Formation

- Team Formation Survey:
  - To be completed by ***everyone***
  - **Released today; Due: 8/25 at 11:59 PM**
- Team Preferences:
  - Groups of 2+ students: Sign up using Canvas Groups
    - Use the ***existing*** available group
    - ***Do NOT create your own group***
  - No preferences:
    - Will be matched via CATME

Teams will be formed and mentors by 8/26.

All Project 1 assignments out 8/22.

Summary

Summary

- SE important/critical discipline
  - Concerned with cost-effective software development (all aspects!)
  - Based on a systematic approach that uses appropriate tools and techniques, follows a process, and operates under specific development constraints

- Goal of SE is to deliver high-quality products that provide the expected functionality, meet projected time estimates, and have a reasonable cost

To do this week

- Introduce yourself on Ed
- Submit team formation survey on CATME
- Go through the class website to learn about the schedule
- Go through Project 1 assignments on Canvas to learn the course expectation
- Go through the 3 AI videos on the class website in the readings column for Day 1
- Ask questions on Ed
- 2 surveys out on the class website. They are for extra credit. 1 is related to sustainability in software engineering, and the other is AI integration in software engineering. Take 10 minutes to complete the survey now and submit the extra credit assignment.