CS 3300-A Introduction to Software Engineering

Lecture 01: Introduction & Overview

Nimisha Roy ▸ nroy9@gatech.edu

These slides are adapted from Alessandro Orso and Mahdi Roozbahani

Welcome to CS 3300A ☺

Instruction Team

Instructor:
Nimisha Roy
(nroy9@gatech.edu)

GTA:
Tanuj Bohra
(tanuj.bohra@gatech.edu)

Refer to:

Class Website

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

Ed Discussion

For any communication with the instruction team

Introduce yourselves on Ed !

Contents

- Software engineering- Introduction
- History
- Software Process, Phases, Tools of the Trade
- Course overview and structure
- Class organization
- Information about projects

What is Software Engineering?
Why is it Important?

- What is a computer? – A programmable device. Most essential function is program development

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

- Also includes understanding the requirements, working with stakeholders, trying to find a solution that balances all the stakeholders' needs in order to deliver the software in a way that's tested and rigorous

What is Software Engineering?
Why is it Important?

What are the 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
  - Time & Cost effective

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?

CRASH

Why is it so difficult to build good software??

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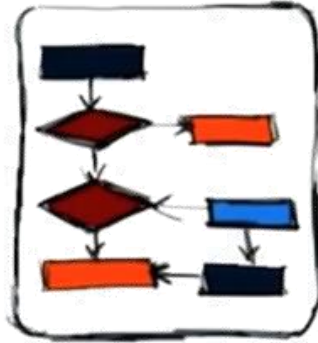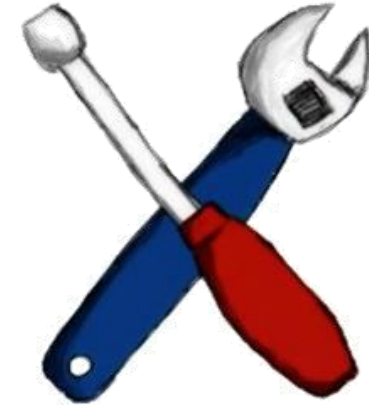
Software Engineering

Methodologies
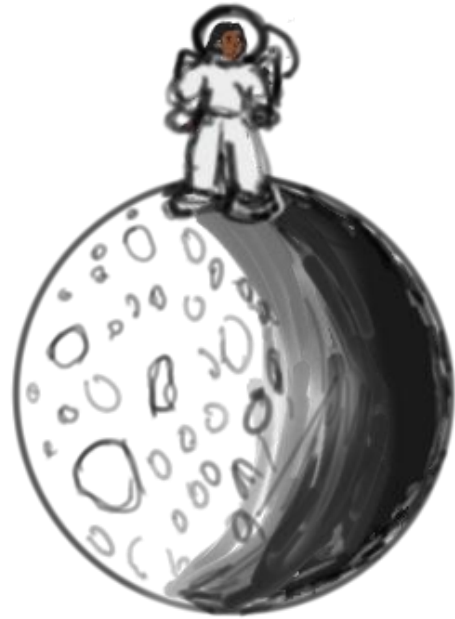

Techniques

Tools

To build software
of high quality

That works

And fits into
Budget

History

The 60s

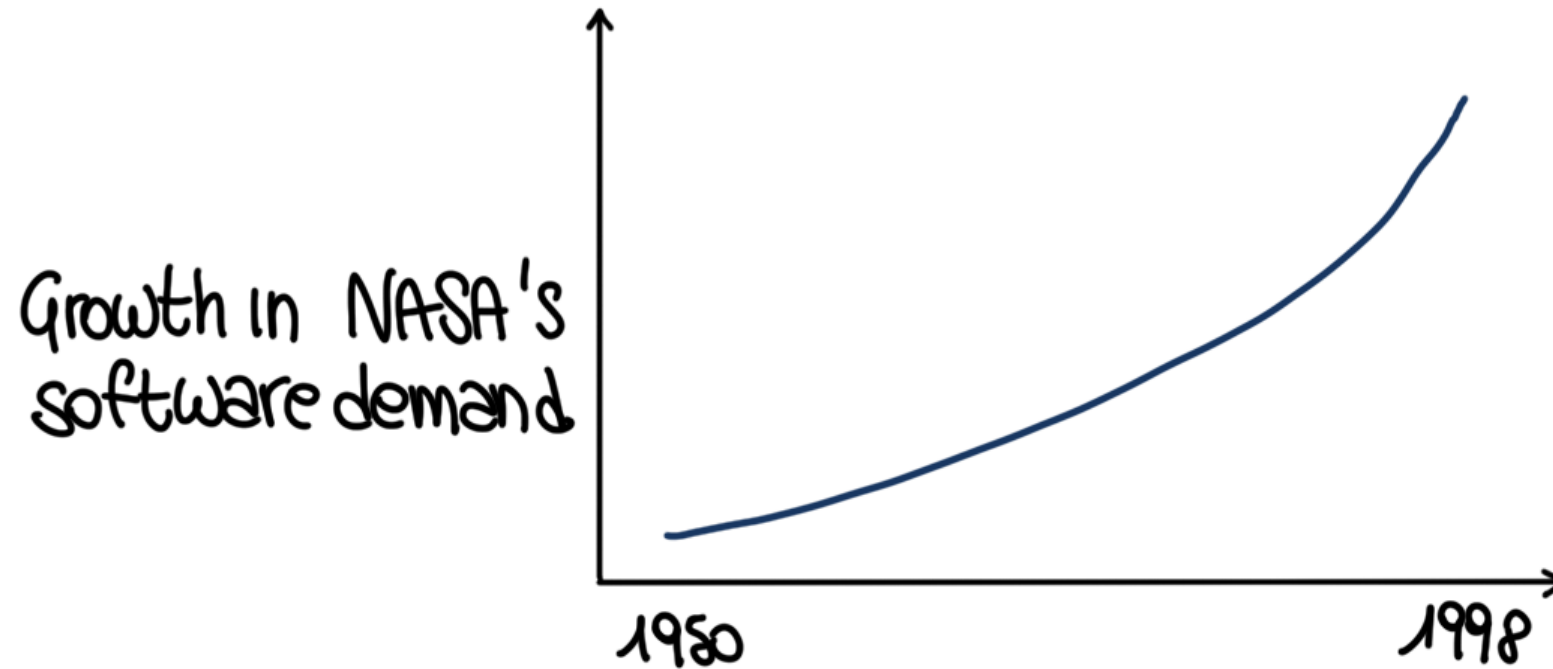Man on Moon   Polaroid

History

History

Reason 1: Rising Demand for Software

HW => SW

Growth in NASA's software demand

1950 1998

History

Reason 2: Increasing product complexity

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

History

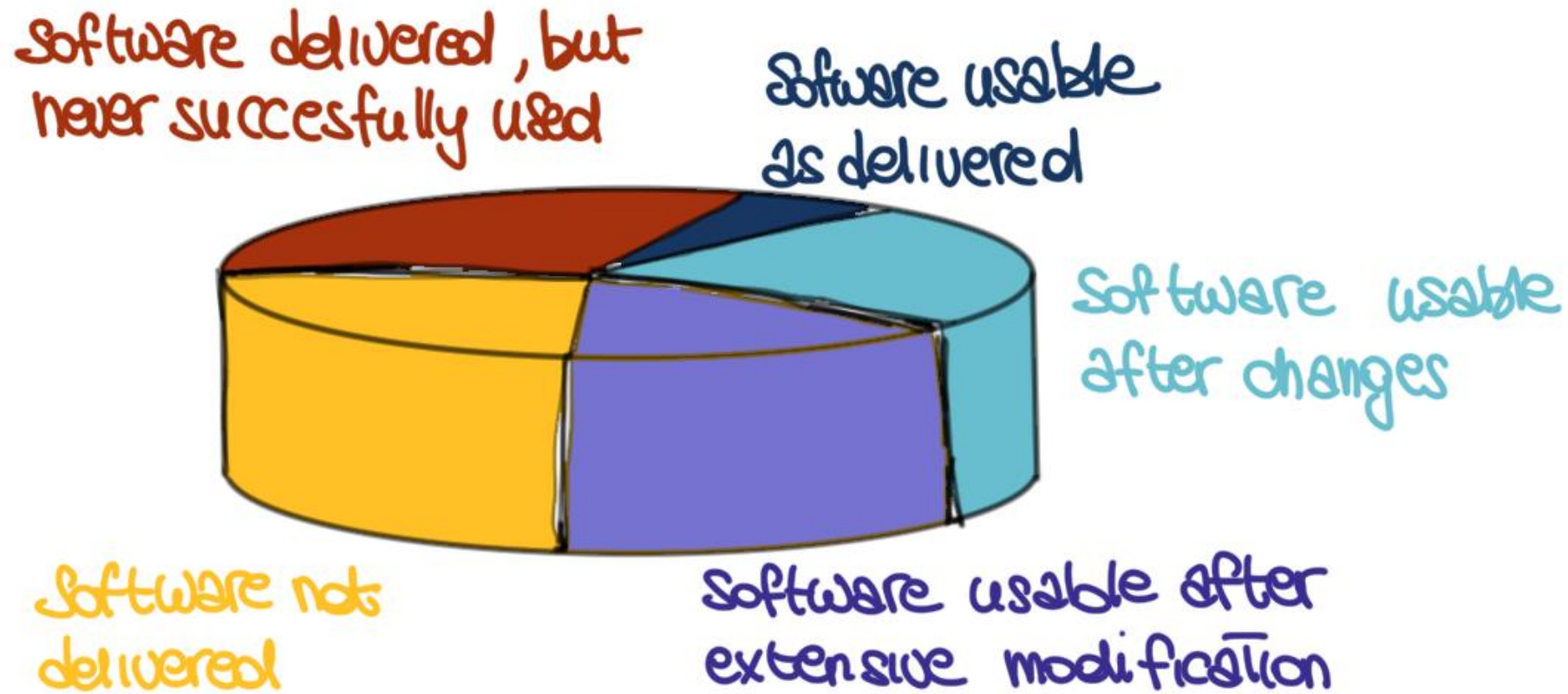## Reason 3: Developers Productivity Growth

Software size and complexity

Developers' productivity
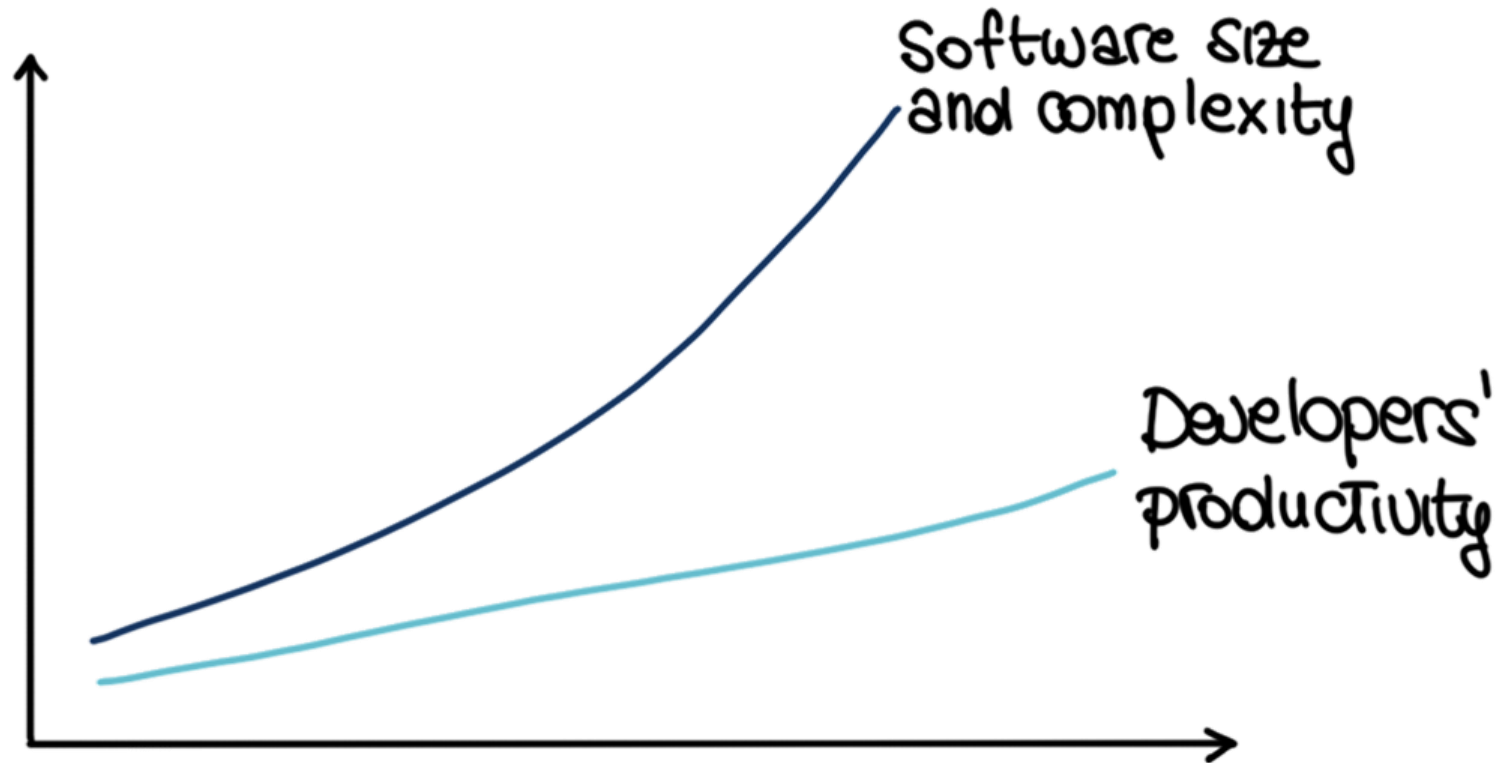
History

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

What are the key challenges facing SE?

- How can we build high-quality systems?

- How can we do it in a reasonable time?
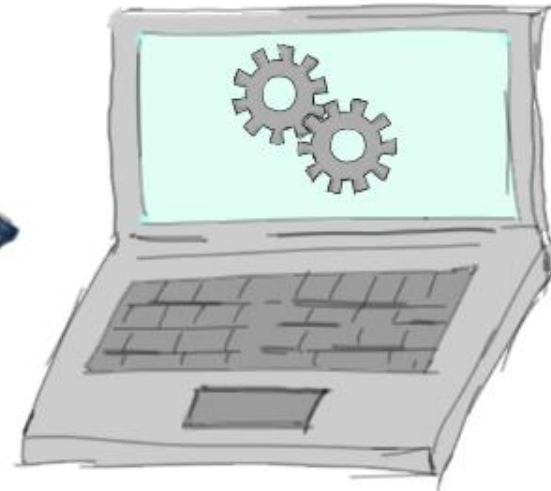
- How can we do it at a reasonable cost?

Software Process, Phases, Tools of the Trade

Abstract Idea

System Implementation

Very Complex
Process

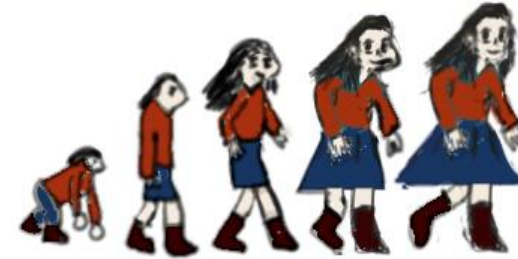Software Process
- Systematic
- Formal

Software Process, Phases, Tools of the Trade
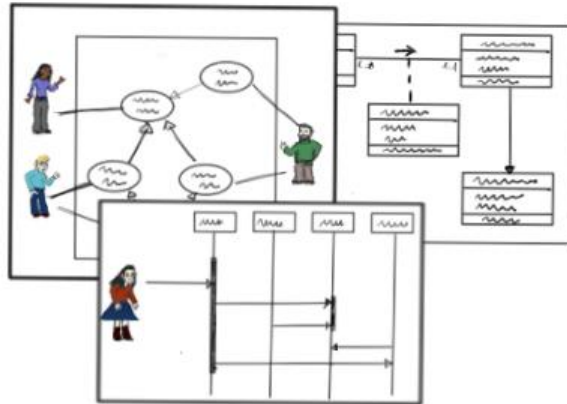
Software Process

Waterfall

Evolutionary Prototyping

RUP/USP

Agile

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 Process, Phases, Tools of the Trade
Software Phases

Requirements

Engineering

Design

Implementation

Verification &

Validation

Maintenance

Software Process, Phases, Tools of the Trade

Tools of the Trade

Software size and complexity

Developers' productivity

- Tools and automation are fundamental in increasing efficiency and effectiveness
- Development: punch cards => IDEs
- Languages: machine code => High-level languages
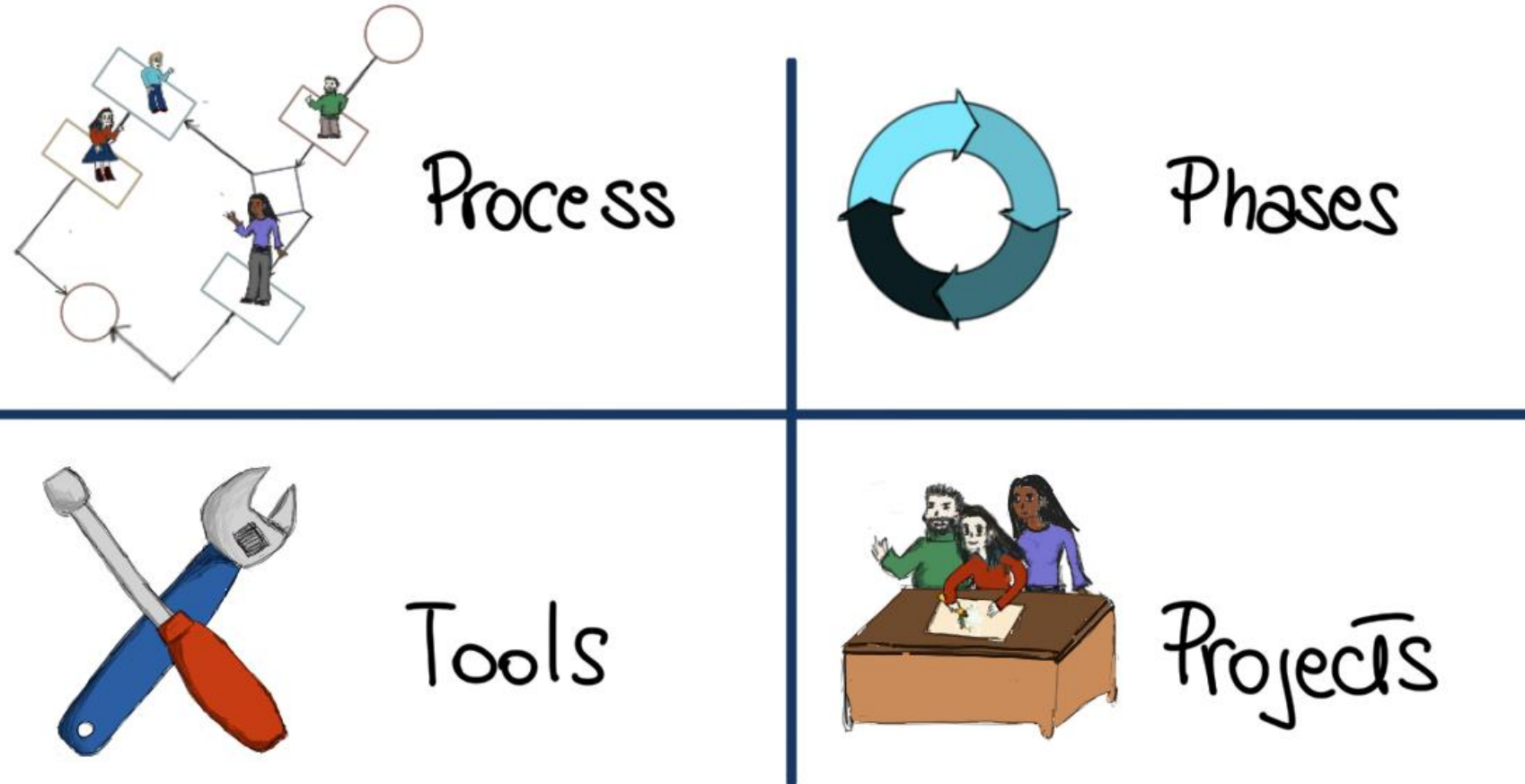- Debugging: print statements => Symbolic debuggers

Software Process, Phases, Tools of the Trade

Tools of the Trade

Demonstration in Lectures
- IDEs
- Front-End Development
- Back-End Development
- Version Control Systems
- Google Cloud Platform

Course Overview

Process

Phases

Tools

Projects

Course Overview

Requirements

Install Tools

Teamwork

Readings

Online submission

Class organization

- Class Website and Canvas

- Ed Discussion

- In-class Lectures

- Attendance is required

- Tools of the Trade Demonstrations - Laptops recommended.

- Invited lectures

- Team-work - Discussion, discussion, discussion

Information about projects

- Two projects
- **Project 1:** WEB-APP using Google Cloud Computing. All teams will do the same project
- **Project 2:** You will choose the project. It can be web- or mobile-app.
- Team based
  - Different teams
  - Grades maybe adjusted based on peer review
- Tools/environments (Java, Eclipse, CVS, GIT, Visual Code …)

Information about projects

Form your Teams and Post on Ed Discussions (Due 30th August)

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

Summary