Georgia Tech

CS 3300
Intro to Software Engineering

Mahdi Roozbahani

Lecturer, College of Computing, CSE, Georgia Tech

Founder of Filio, a visual asset management platform through Create-X

Slides are from Alex Orso lectures

Refer to:

Class Website

For anything (updates, lectures, logistics, and so on) related to this class

https://mahdi-roozbahani.github.io/CS3300-spring2020/

Software Engineering

Introduction and Overview

Introductory remarks

- Introductions
- Course overview and structure
- Requirements
- Class organization
- Software engineering intro
- Information about projects
- Assignments

Introductions

- Industrial experience
- Languages, IDEs, OSs, …
- How important is your background and experience?
- How many will/can bring a laptop to class?

Before we start, what is a…

Algorithm

a set of steps or procedure for solving a recurrent problem

Error

a human mistake made during the construction of a system

Fault

the manifestation of an error in a process deliverable; a flaw

Defect

synonym for fault

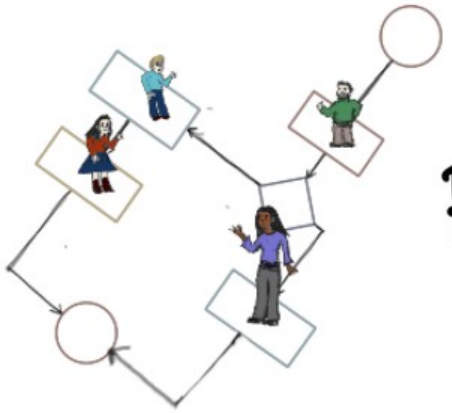
Bug

synonym for fault, usually referring to faults in code

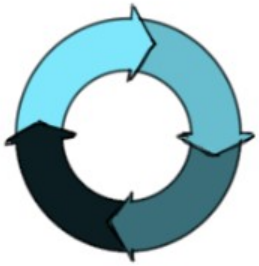
Failure

a program execution that results in incorrect behavior, such as incorrect results or failure to terminate

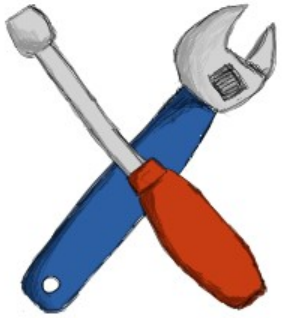
Course Overview



Process



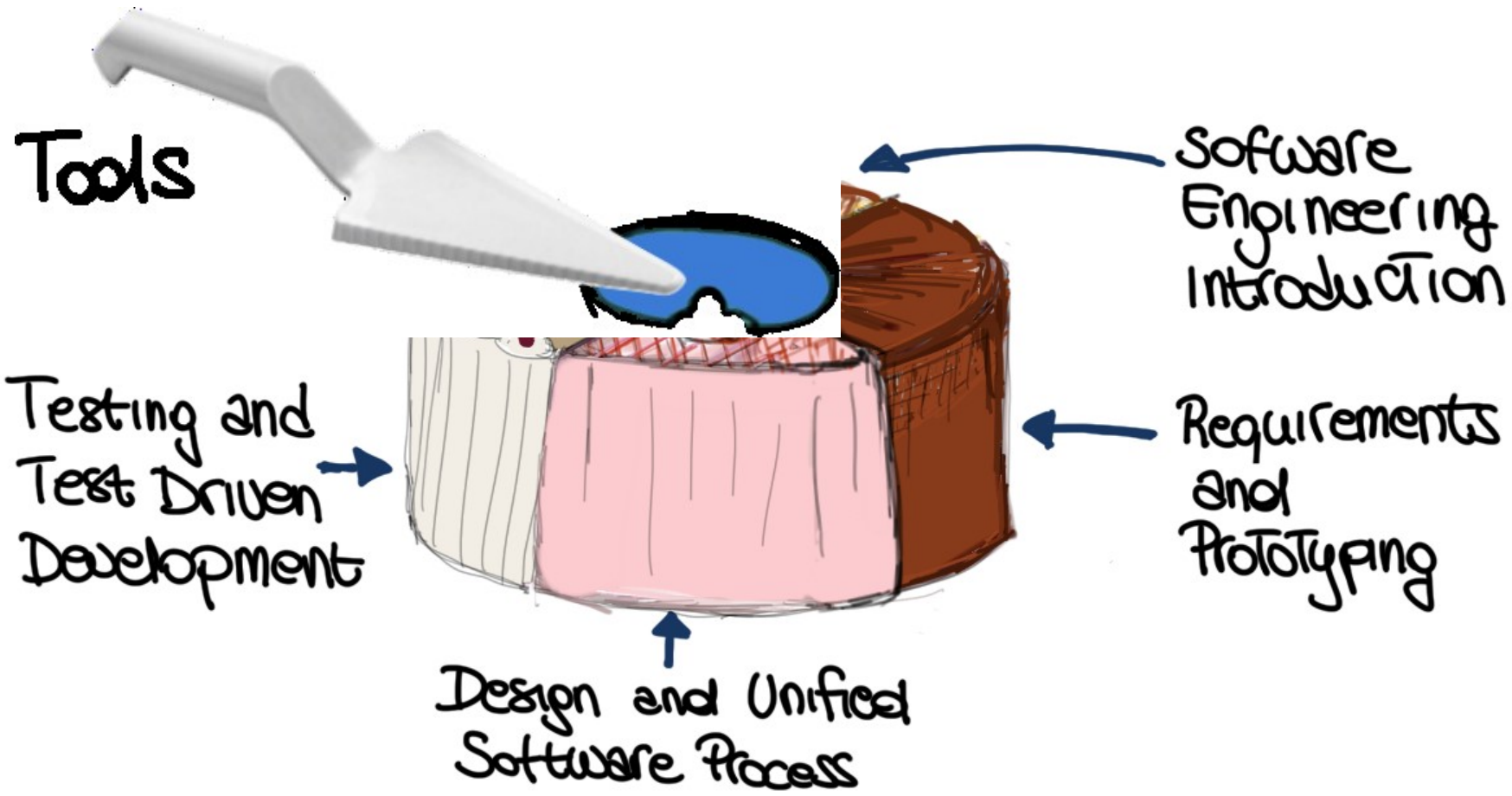
Phases



Tools

Projects

Course Structure



REQUIREMENTS

Install Tools

Teamwork

Readings

Online submission

Class organization

- Website and Canvas
- Piazza
- Attendance is required
- Lectures
- Online lectures (possibly…)
- In-class exercises
- Team work
- Tool/technology days
- Invited lectures
- Discussion, discussion, discussion

An Introduction to Software Engineering

WHAT IS SOFTWARE ENGINEERING?
WHY DO WE NEED IT?

What is the difference between SE and CS?

CS is concerned with theory and fundamentals; SE is concerned with the practicalities of developing and delivering useful software

WHAT IS SOFTWARE ENGINEERING?
WHY DO WE NEED IT?

What is this?

[ ] 4th of July fireworks

[ ] Flare gun in action

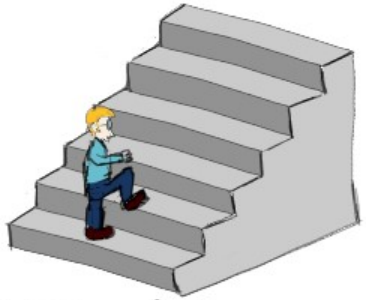
[ ] Explosion of Ariane 5 rocket due to Software errors

Why is it so hard to build good sofware?

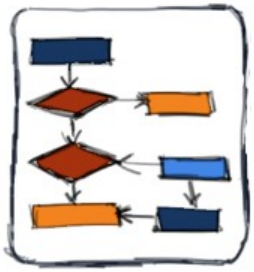
What are the attributes of a good SW?

- The software should deliver the required functionality and performance to the user and should have several qualities:
- Maintainability
  - Software must be evolvable to meet changing needs
- Dependability
  - Software must be trustworthy (reliability, security, and safety)
- Reliability
  - Software must behave as expected under all circumstances
- Other -ilities

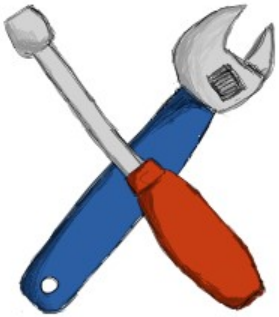
DISCIPLINE OF SOFWARE ENGINEERING



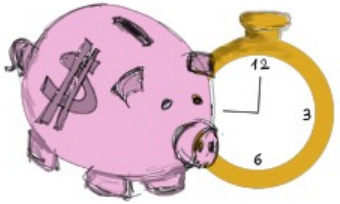
Methodologies



Techniques



Tools



High Quality sofware that Works and Fits Budget

THE 60's

Man on the Moon.

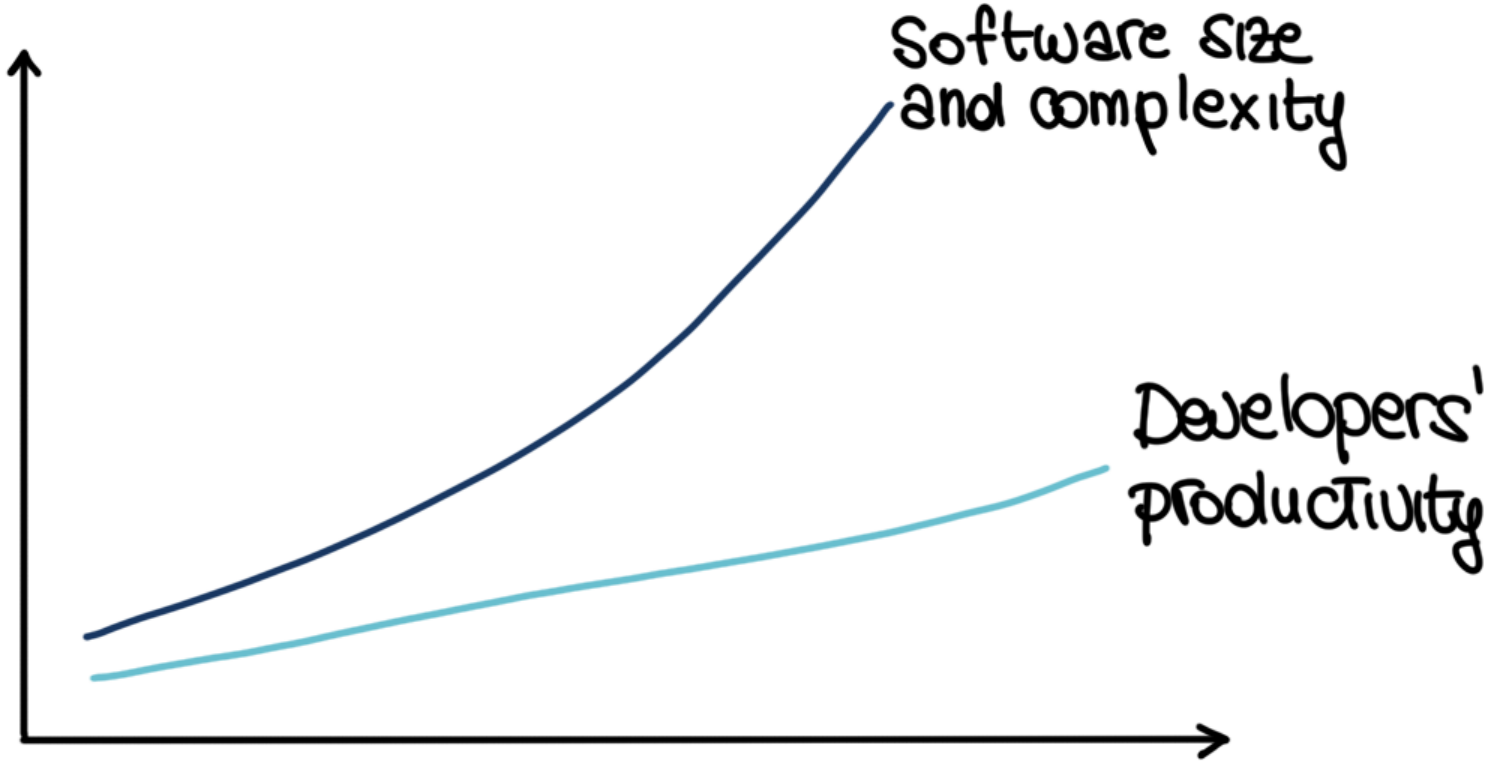
Polaroid


THE SOFTWARE
CRISIS

Increasing product complexity

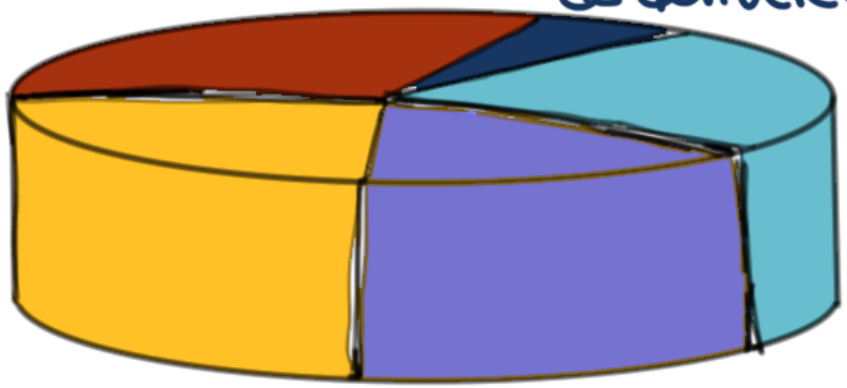
SIZE	EXAMPLE	
10^2 LOC	Class exercise	Programming effort
10^3 LOC	Small project	
10^4 LOC	Term project	
10^5 LOC	Word processor	Software engineering effort
10^6 LOC	Operating system	
10^7 LOC	Distributed system	
...	...	

DEVELOPER'S PRODUCTIVITY GROW

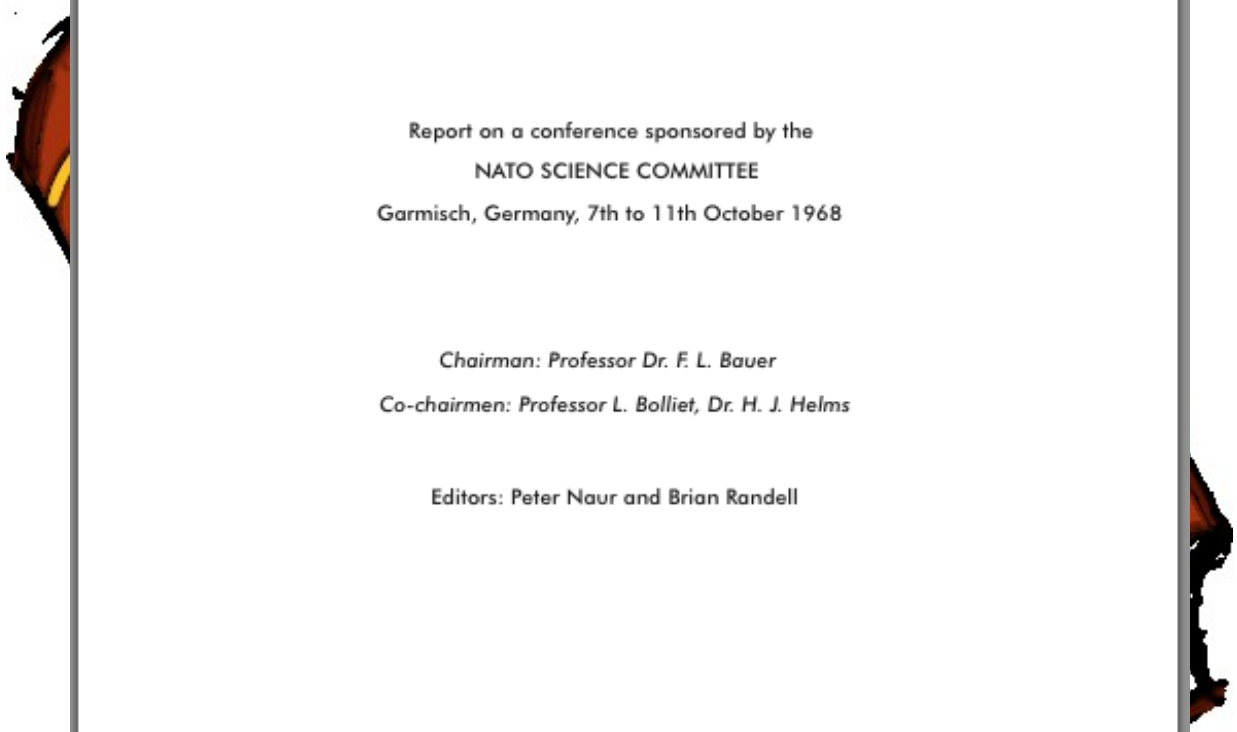


STUDY OF 9 SOFTWARE DEVELOPMENT CONTRACTS

(Davis, 1990)




NATO
SOFTWARE
ENGINEERING
Jan 1969



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

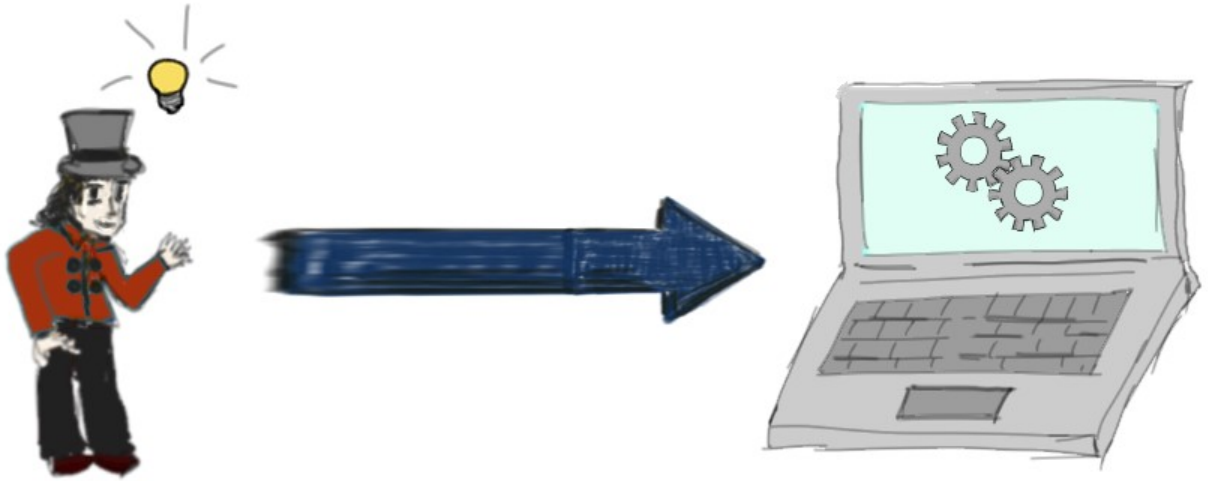
Software importance today

- 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

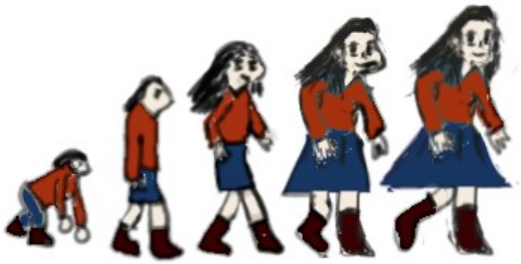
What are common causes of SW failures?

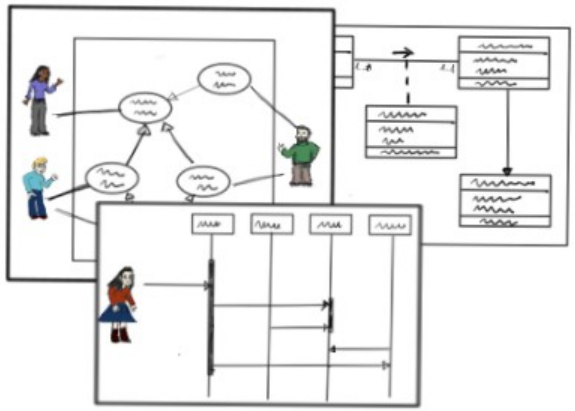
- No standard procedures for development
- Inadequate understanding of requirements
- Sheer complexity of software (e.g., concurrency, distribution)
- Size of project (too large for a single manager)
- Difficult to match technical knowledge of staff with project needs
- Poor design/implementation/testing 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 DEVELOPMENT



SOFTWARE PROCESS





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'll see several

SOFTWARE PHASES

Requirements Engineering

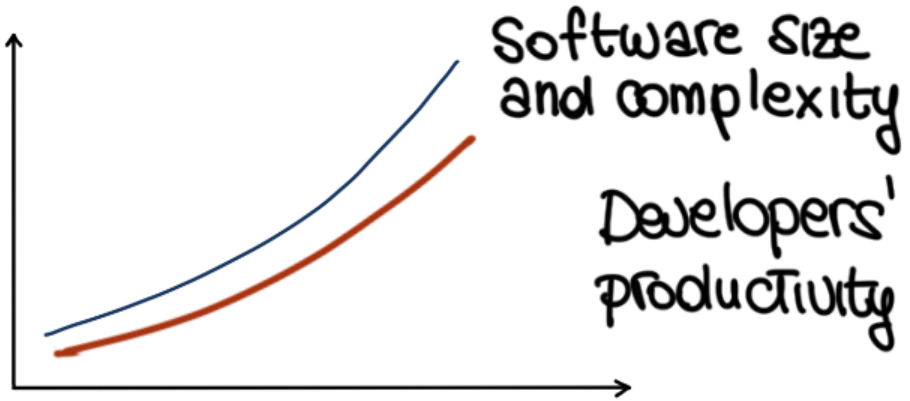
Design

Implementation

Verification and Validation

Maintenance

Tools of the Trade

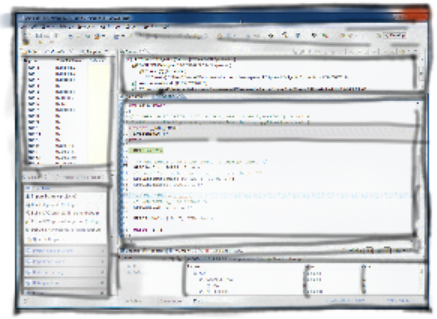


Development: punch cards => IDEs

Languages: machine code => High-level languages

Debugging: print statements => Symbolic debuggers

Tools of the Trade



IDE

VCS

Coverage and verification Tools

Two projects:

Project 1: WEB-APP using Google Cloud Computing. All teams will do a same project

Project 2: You will choose the project. It can be web- and mobile-app.

In 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

In summary…

- SE im
  - Con
    deve
  - Bas
    appr
    proc
    cons

  s
  a
  elopment

- Goal o
  that pr
  projec
  reason

  oducts
  , meet