CS 4530: Fundamentals of Software Engineering
Module 1, Lesson 1
Course Introduction

Rob Simmons

Khoury College of Computer Sciences

Course staff

Me!
Rob Simmons

- Course Adminstration
  - Elizabeth O’Reilly
- TAs
  - Apurva Sani
  - Chaman Kumar
  - Manas Aggrawal
  - Tanya Shukla
  - Anish Hedge (not here today)
  - Vihar Reddy (joining next week)

Learning Objectives for this Module

- By the end of this module, you should be able to:
  - Explain in general terms what software engineering is
  - List your weekly obligations as a student
  - List the requirements for completing the course

What does it mean to make software... engineering?

Wikipedia on...

Engineering

"Engineering is the practice of using natural science, mathematics, and the engineering design process to solve problems within technology, increase efficiency and productivity, and improve systems."

Wikipedia on...

Engineering design process

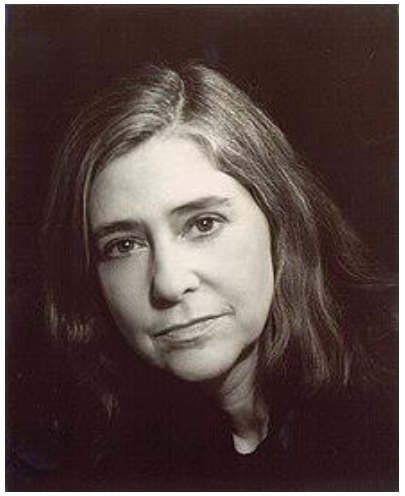
"Engineering is formulating a problem that can be solved through ***design***."

Common stages:

- Research
- Design requirements
- Feasibility
- Concept generation
- Preliminary design
- Detailed design
- Production planning

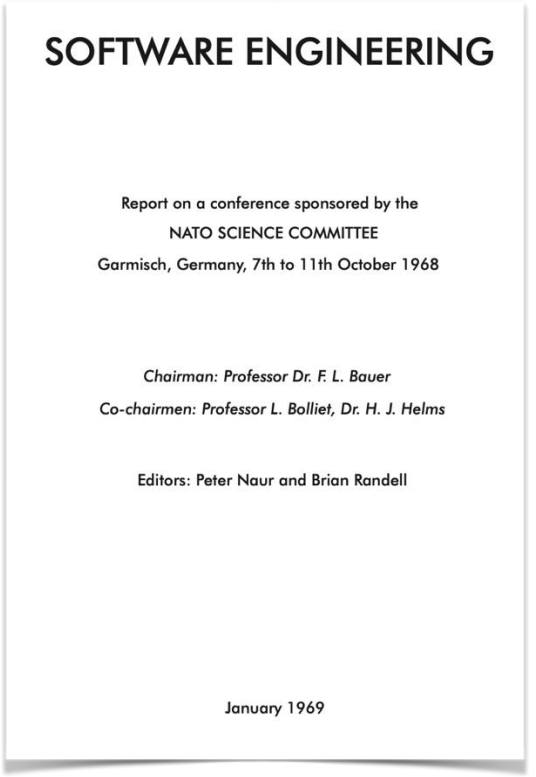
Origins of "software engineering"

Margaret Hamilton @ NASA, around 1963



The Apollo Guidance Computer's software, basically

1968 NATO conference on Software Engineering + Outcomes


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

Friedrich Bauer

Barry Boehm

Anthony Oettinger, ACM President

Comm. of the ACM, August 1966 (page 546)

ing nature. We must recognize ourselves—not necessarily all of us, and not necessarily any one of us all the time—as members of an *engineering* profession, be it hardware engineering or software engineering, a profession without artificial and irrelevant boundaries like that between "scientific" and "business" applications.

(all pictures from Wikipedia)

One definition of "software engineering"

Software engineering concerns the

- design
- construction,
- and maintenance
- ➢ of large programs
- ➢ over time.

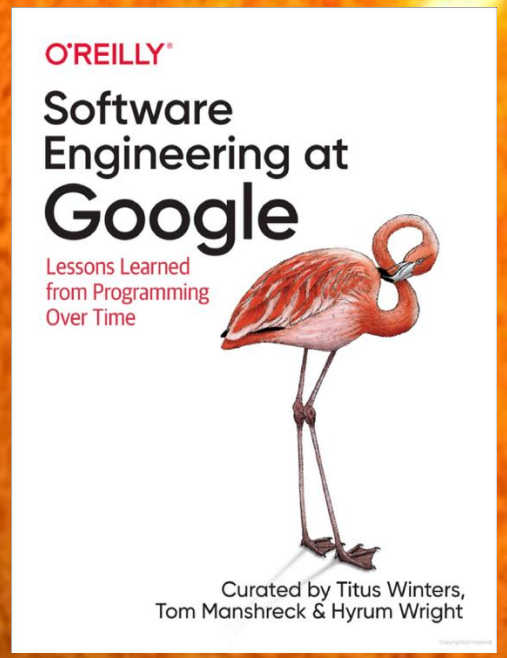
Compare that with Wikipedia on…

Engineering design process

Common stages:

- Research
- Design requirements
- Feasibility
- Concept generation
- Preliminary design
- Detailed design
- Production planning

Okay, what do you mean by "large"



The Apollo Guidance Computer's software

Almost any pre-series-B startup

Your 4-person project in this class

(image from "The Scale of Space" on KWIT, March 2018)

Problem 1: Programs need to be read by people

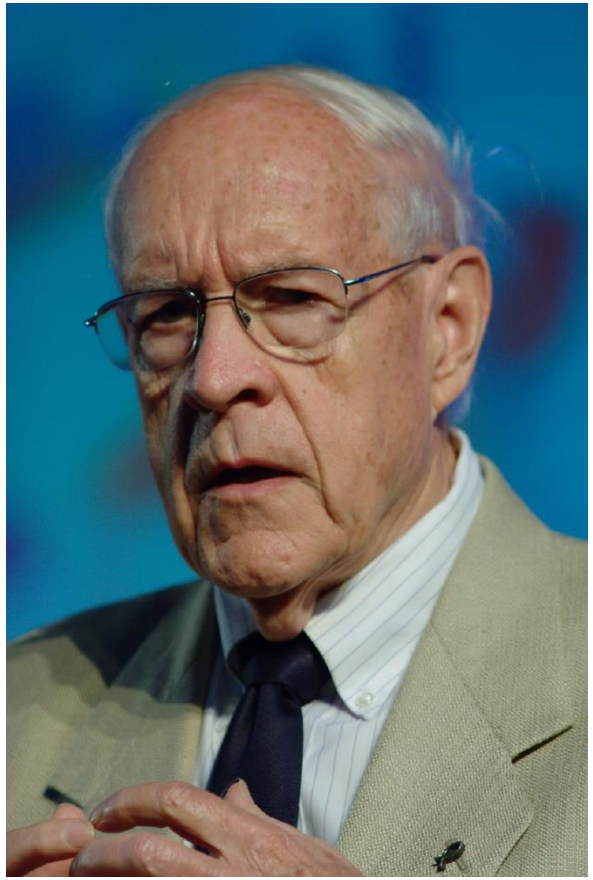
"Any fool can write code that a computer can understand. Good programmers write code that humans can understand"

Martin Fowler

Problem #2: People need to talk to each other

"Adding manpower to a late software project makes it later"

Fred Brooks, 1975



So, software engineering must encompass:

PEOPLE,

PROCESSES,

& PROGRAMS

The course will cover

- People
  - how to organize software teams and make them function effectively
  - how software engineering teams work in larger organizations
- Processes
  - how to divide a large project into engineering tasks
  - how to coordinate the tasks to form a coherent whole
- Programs
  - how to write programs that people can understand and maintain
  - …focusing on a particular domain (medium-sized web applications)

Three Scales of Design

The Planning Scale

- key questions: How do we make software artifacts “good”? What does that mean? Who decides?

The Organizational Scale

- key questions: What are people’s needs? How do we design software artifacts that meets those needs?

The Implementation Scale

- key question: how to design software artifacts that are easy to test, understand, and modify?

So, software engineering must encompass:

PLANNING,

ORGANIZING,

& IMPLEMENTING

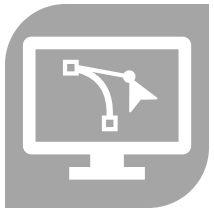
So, software engineering must encompass:

	 PEOPLE	 PROCESSES	 PROGRAMS
 PLANNING			
 ORGANIZING			
 IMPLEMENTING			

So, software engineering must encompass:

PEOPLE

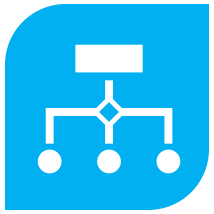
PROCESSES



PROGRAMS

PLANNING

ORGANIZING



IMPLEMENTING

mostly out of scope

This class is here-ish

previous classes (OOD)

Not unique to *software* engineering!

PLANNING

ORGANIZING

IMPLEMENTING

howstuffworks.com

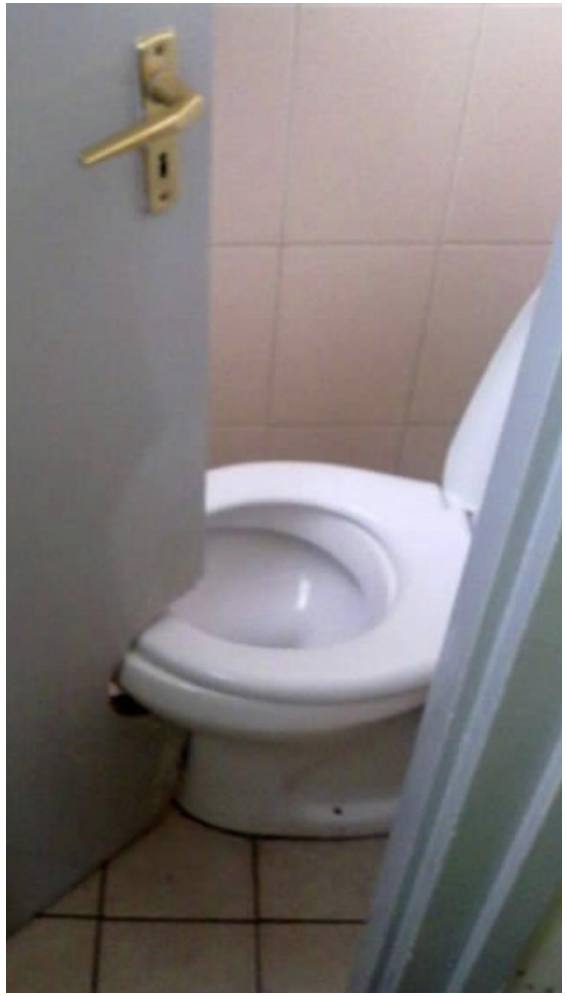
Half as Interesting on YouTube

Not unique to *software* engineering!

PLANNING

ORGANIZING

IMPLEMENTING



www.oddee.com/item_97852.aspx

Not unique to *software* engineering!

PLANNING

Photo: Carola Ripamonti; design: Andrea Marcante, Adelaide Testa; from Take a Bath (copyright Gestalten 2017)

ORGANIZING

SMALL FOOTPRINT, BIG IDEAS

43 Small Bathroom Ideas to Make Your Bathroom Feel Bigger

www.architecturaldigest.com

IMPLEMENTING

Not unique to *software* engineering!

PEOPLE

PROCESSES

PROGRAMS

Course outcomes

- By the end of this course, you will…
  - …be able to define and describe the phases of the software engineering lifecycle.
  - …demonstrate an ability to use key processes and technologies in modern software development and be able to explain the purpose of those processes and technologies.
  - …be able to apply instances of major tools used for elementary software engineering tasks in the context of web applications.
  - …design and implement a portfolio-worthy software engineering project in a small team environment that can be showcased to recruiters.

Course outcomes

- By the end of this course, you will…
  - …be able to define and describe the phases of the software engineering lifecycle.
  - …demonstrate an ability to use key processes and technologies in modern software development and be able to explain the purpose of those processes and technologies.
  - …be able to apply instances of major tools used for elementary software engineering tasks in the context of web applications.
  - …design and implement a portfolio-worthy software engineering project in a small team environment that can be showcased to recruiters.




These are normal learning goals for a course at a university!

This is maybe a little different