

# Week 1 – Intro to SE and SDLC

**CSC 317 / INF 307 – SOFTWARE ENGINEERING**  
2022/2023 Academic Year  
Semester 1

# Recommended Reading

- **Chapter 1** of “Software Engineering at Google” by Winters, T., Wright, H. and Manshreck, T.
- **Chapter 1 & 2** of “Software Engineering: a practitioner’s approach” by Pressman and Maxim

# Today

- Important Questions
  - What is Software Engineering
  - Programming vs SE
  - Why is SE difficult?
- The Software Development Lifecycle (SDLC)

# In-Class Discussion

- What is Software Engineering?

# In-Class Discussion

- What is the difference between Software Engineering and Programming?

# What is software engineering?

## Definition 1

“Software Engineering is the systematic approach to development , maintenance, organization of software systems” – Definition by IEEE

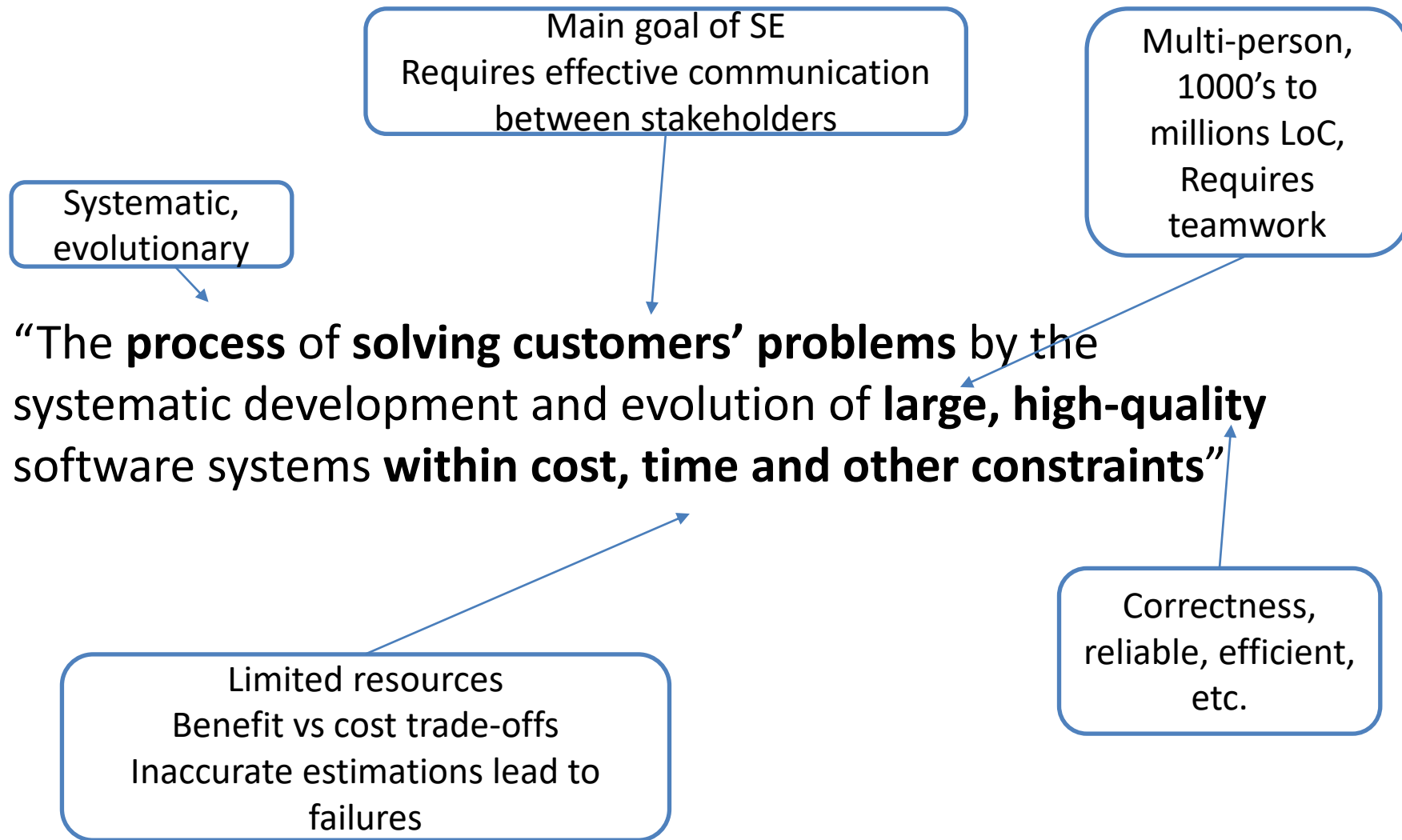
## Definition 2

“The process of solving customers’ problems by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints”

## Definition 3

“Software engineering is about understanding business problems, inventing solutions, evaluating alternatives, and making design tradeoffs and choices.” – Marsic I. (2012)

# What is software engineering?



# SE vs. Programming

Programming is just a small part of SE!

## SE

- Focuses on understanding the business problem
- Identify a solution
- Design the blueprint of the solution

## Programming

- Focuses on implementing the blueprint



# Skills of a Software Engineer

- Programming skills
  - Data structures and algorithms
  - Programming languages
  - Tools – compilers, debuggers, editors
- Communication skills
  - Spoken, written, presentation
  - Teamwork
  - Communicating with external people (customers)
- Design skills
  - Application domain jargon and modeling
  - Architectural designs
  - Shifting between several levels of abstraction

# SE is HARD

- SE = knowledge of software domain + problem domain
  - Discussion: What knowledge is required in the design of an ATM system
- SE aims to model the physical world through abstractions
  - The physical world has many parts
  - Physical world is constantly changing
  - Physical world is not well understood/explained
  - Abstractions are approximations at best – on a few variables of the physical world

# SE is HARD

- We often only see or hear about the successes
- A lot of major issues happen during software engineering:
  - Wrong requirements
  - Cost overruns
  - Late delivery
  - Delivered product not accepted by customers
- Software **FAILS A LOT**
  - Read this article:  
<https://www.bugraptors.com/blog/top-software-failures-due-to-lack-of-testing>

# Software Fails A Lot



Danny Thompson  
@DThompsonDev

No matter how bad your day is going as a software engineer, it can't be as bad as creating a bug that literally stopped all the flights in America lol

Raw Story

## Catastrophic FAA outage traced back to tiny mistake by one engineer: report

Story by Matthew Chapman · Wednesday

23 79 32 Comments

On Wednesday, ABC News' Josh Margolin reported that the cause of today's catastrophic flight system failure, which paralyzed the Federal Aviation Administration and led to the first nationwide ground stop of all aircraft in the United States since the September 11 terrorist attacks, has been identified.



1:12 PM · Jan 14, 2023 · 634.7K Views

## Airbus A400M plane crash linked to software fault

20 May 2015



Ben  
@JustAnother\_Ben

Learning that the Southwest Airlines meltdown is due to an antiquated crew scheduling system that assumes where a crew member is instead of working off live flight data. They've lost track of almost every crew member and the only way to solve it is...

11:58 AM · Dec 27, 2022 · 18.1M Views

7,275 Retweets 1,836 Quote Tweets 92.3K Likes



Tweet your reply

Reply



Ben @JustAnother\_Ben · Dec 27, 2022

Replying to @JustAnother\_Ben

for each one to individually call scheduling to tell them where they are. Their phone lines are swamped and I've seen screenshots of crew on hold for 23+ hours trying to get through to scheduling.

## Tesla to recall 135,000 vehicles over computer memory failure

BY FRENCH PRESS AGENCY - AFP | WASHINGTON DC | FEB 02, 2021 - 8:54 PM GMT+3



## Software bug in New Jersey hospital's vaccine scheduling system causes thousands of duplicate appointments

Staff - Monday, February 15th, 2021



Marlton, N.J.-based Virtua Health last month discovered and fixed a bug in its vaccination self-scheduling system that caused between 10,000 and 11,000 appointments to be duplicates, according to a Feb. 12 [TapIn Camden](#) report.

"In some extreme cases, a single person had 20 or more vaccine appointments," Virtua Health Access Center Director Malik Bahar told the publication. "Clearly, this was a big problem and we needed to find a solution fast."

A team of 200 staff made more than 10,400 phone calls to remedy the issue, and it opened up about 5,000 appointments. Many of the people affected by the duplicate appointments were seniors.

# SE is HARD

## Bottom Line

- Writing code is very easy
- But engineering good software is HARD
- So how do we make software engineering work?
  - By following a **process**

# The SDLC

The phases of a **software development cycle**:

- Feasibility study
- Requirements Analysis & Specification
- Design
- Implementation
- Testing
- Delivery
- Maintenance

# The SDLC

- **Feasibility study: WHY?** Cost benefit analysis (is it worth doing the project)
- **Requirement analysis + specification: WHAT?** What should the software do, produce a document.
- **Design: How?** How should the software do it. Architectural design (overall structures + organization of objects/modules, choice of data structures, etc.
- **Coding/Implementation: Realize** components. Code modules, products: software,
- **Testing: Verify!** Test individual modules, test whether several modules work together, test system as a whole, document test results.
- **Deliver: Release** to production!
- **Maintenance : Evolve!**

# References

- Winters, T., Wright, H. and Manshreck, T. (2020). Software Engineering at Google: Lessons Learned from Programming over Time. 1st Edition. O'Reilly Media
- Pressman, R. S. and Maxim, B. R. (2014). Software Engineering: a practitioner's approach, 8th edition: McGraw-Hill
- Winston R. Royce (1970). "Managing the Development of Large Software Systems", in Proc. of IEEE WESCON, pp. 1-9.