INTRODUCTION TO SOFTWARE ENGINEERING

Part I

CICT – CTU

INTRODUCTION TO SOFTWARE ENGINEERING

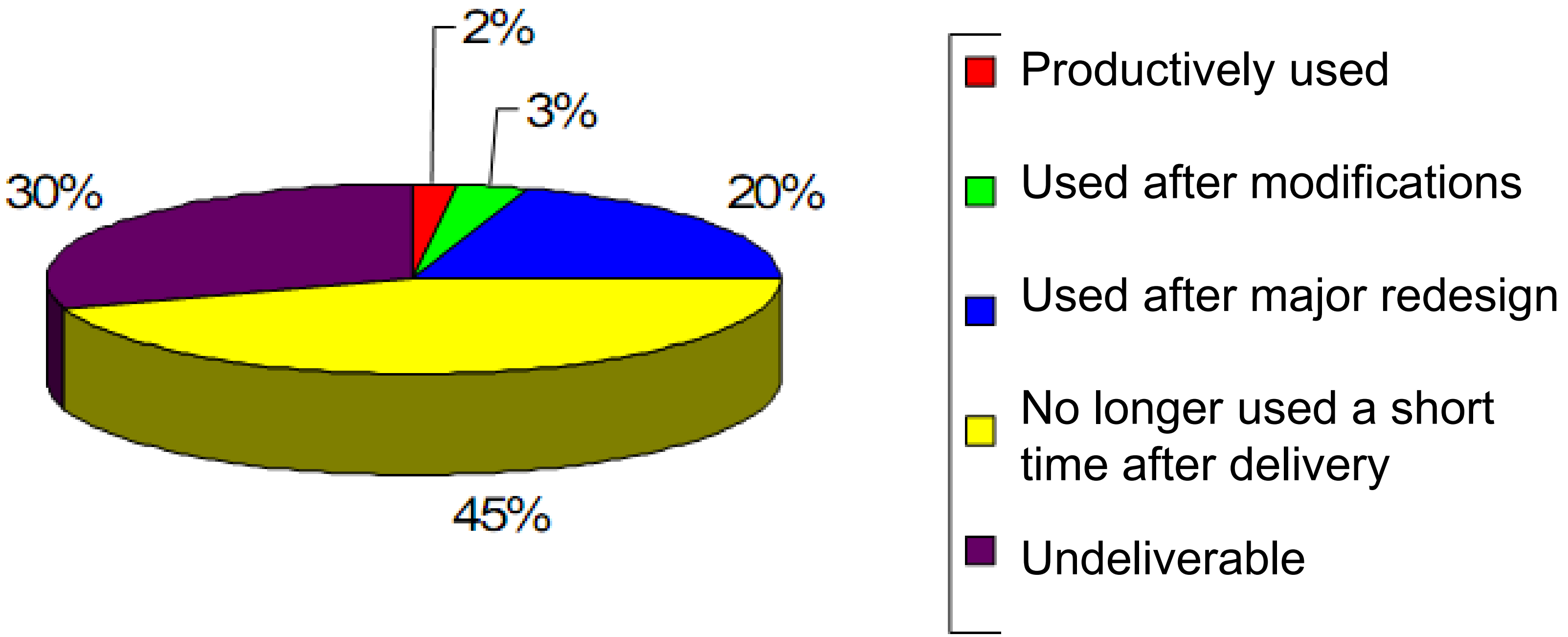
Part I.1 – Overview of Software Engineering

Content

- Why study software engineering?
  - Software crisis
  - Software project success rate
  - Examples of software development failures
  - Causes of software failures
  - How has software engineering changed?
- What is software engineering?
- Steps of software development
- Who does software engineering?
  - Members of the development team

Why study software engineering?

- Software crisis (of the 1960s, 1970s, and 1980s)
  - During the 1970s,



Why study software engineering?

- Software project success rate

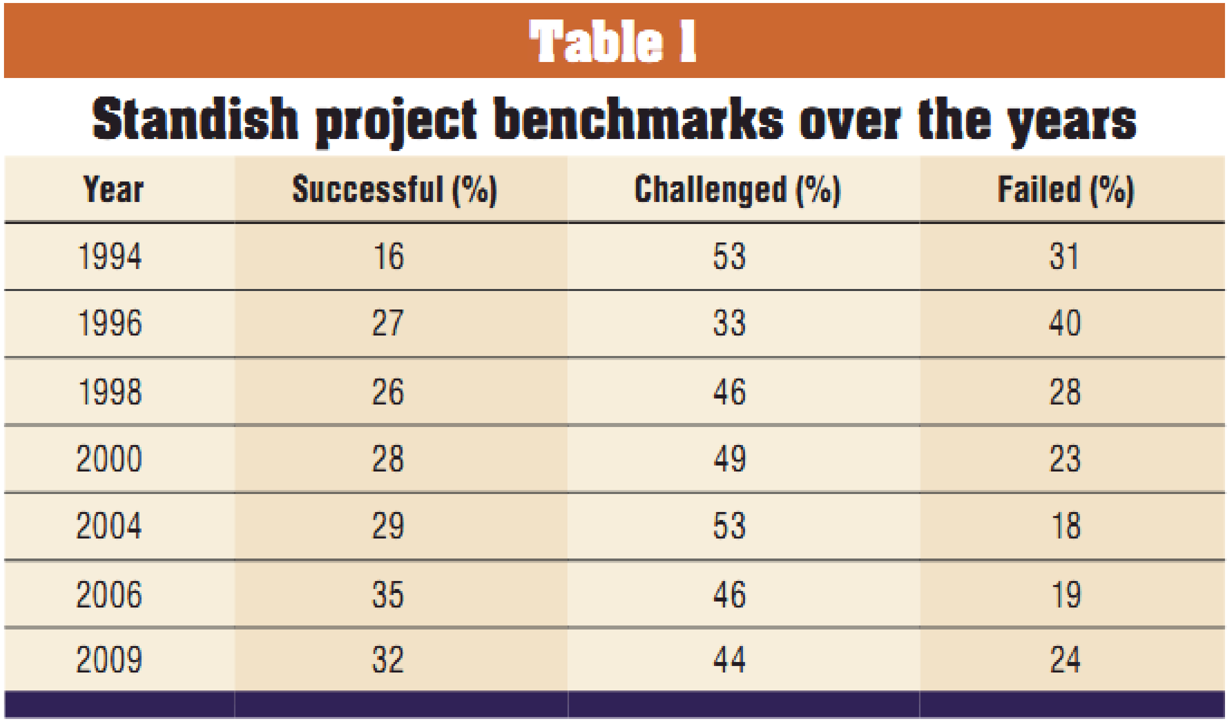


Table I

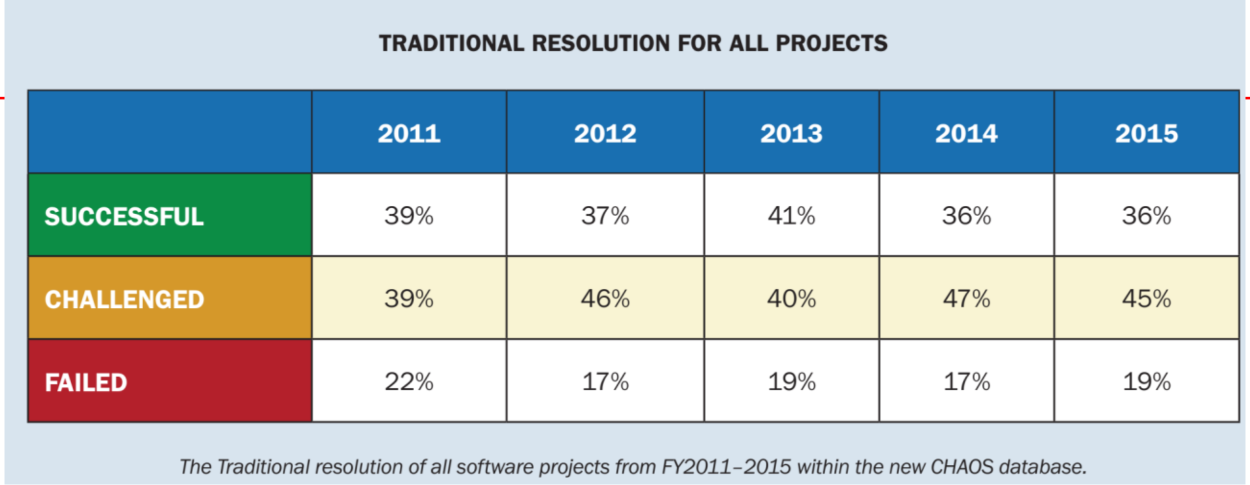
Standish project benchmarks over the years

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

Standish Group, CHAOS Report, 2010

- *Successful projects are on time, on budget, and have a satisfactory implementation.*
- *Challenged projects are over budget, late, and/or have an unsatisfactory implementation.*
- *Failed projects are projects that were either canceled prior to completion or not used after implementation.*

Why study software engineering?



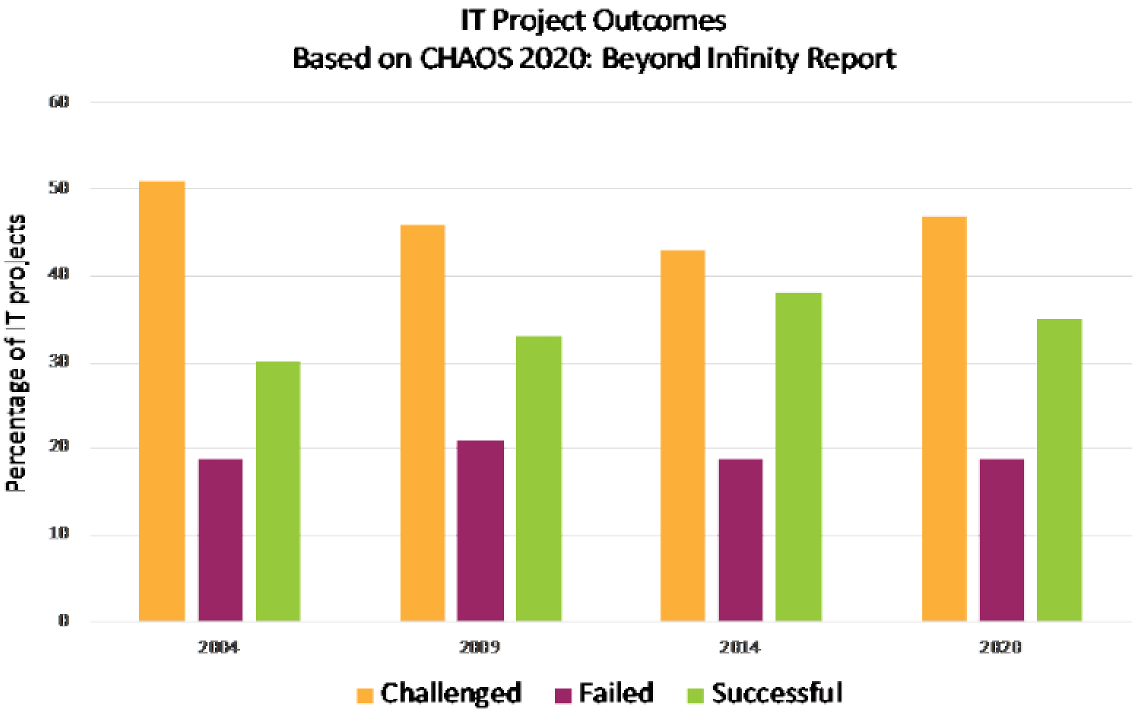
TRADITIONAL RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

The Traditional resolution of all software projects from FY2011–2015 within the new CHAOS database.

Standish Group, CHAOS Report, 2015

- Software project success rate



The Cost of Poor Software Quality in the US: A 2020 Report

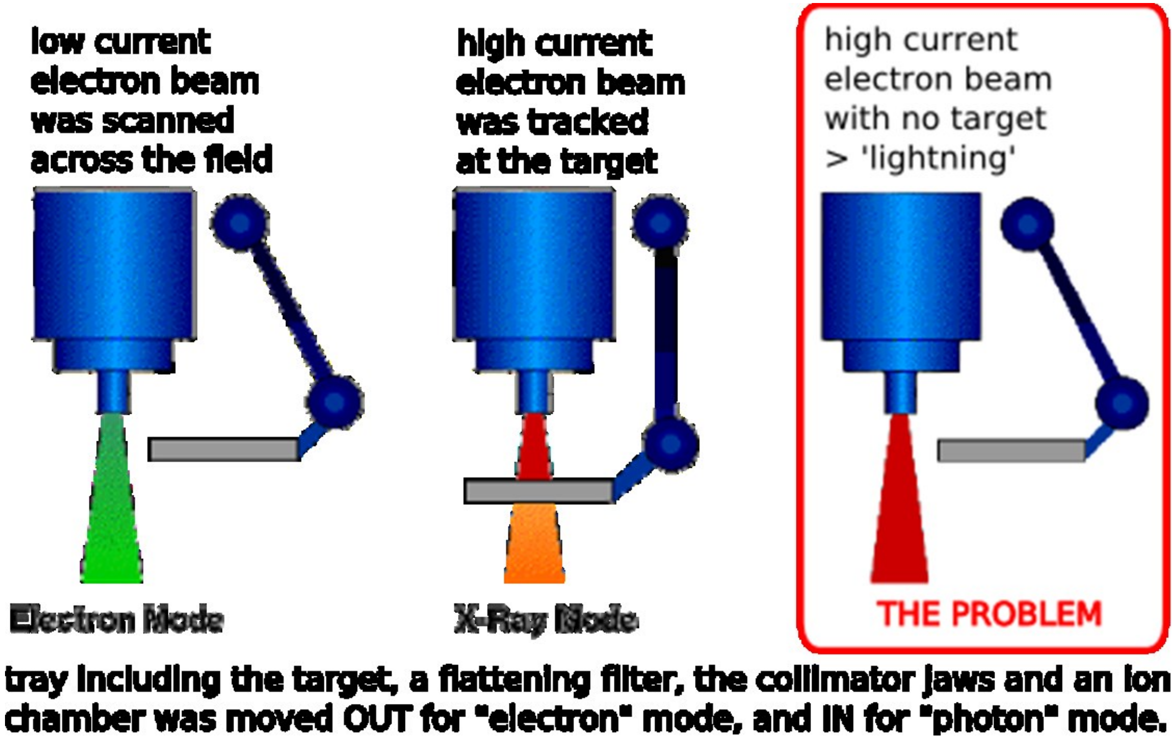
Why study software engineering?

- Examples of software failures
  - **Ariane 5 Flight 501** functioned well for 40 seconds, then veered off course and was destroyed (June 4, 1996); contained four satellites: cost was $500 million

https://www.youtube.com/watch?v=N6PWATvLQCY

Why study software engineering?

- **Therac-25,** a computer-controlled radiation therapy machine was involved in at least six accidents


low current electron beam was scanned across the field
high current electron beam was tracked at the target
high current electron beam with no target > 'lightning'
Electron Mode
X-Ray Mode
THE PROBLEM

tray including the target, a flattening filter, the collimator jaws and an ion chamber was moved OUT for "electron" mode, and IN for "photon" mode.

https://www.crcpress.com/

Why study software engineering?

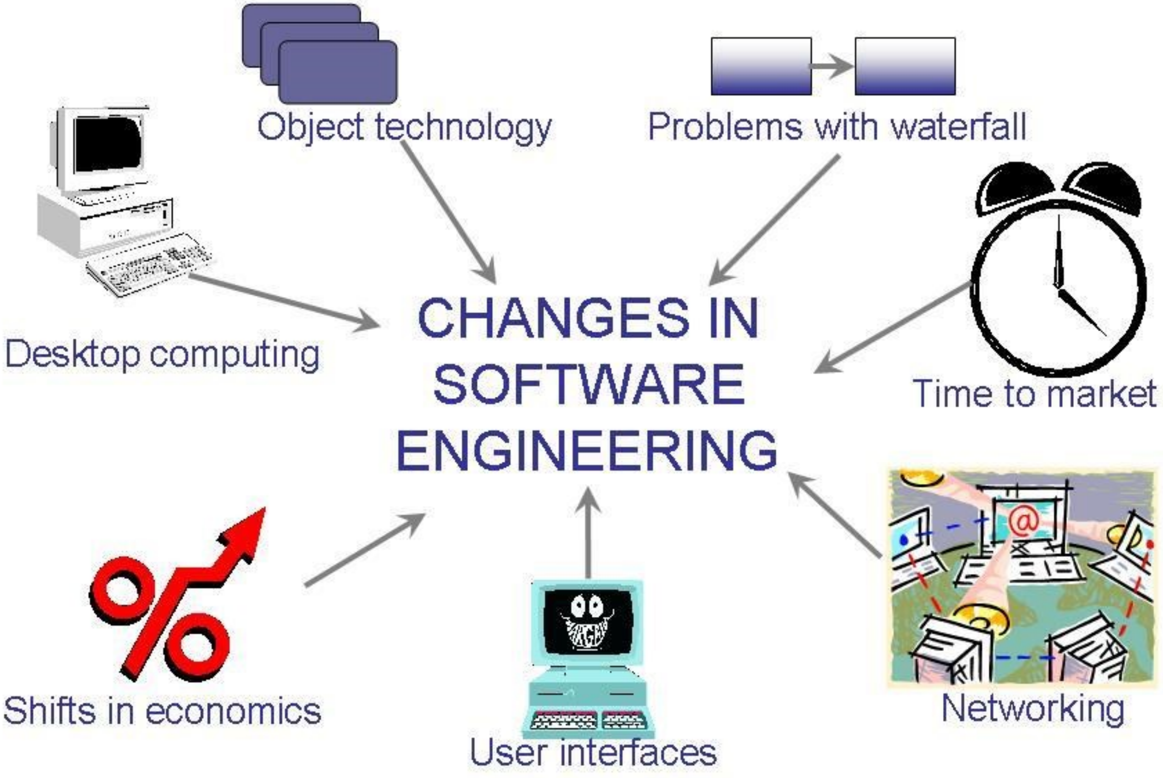
- Causes of software failures
  - More complicated problems
  - Lack of appropriate training in software engineering
  - Unclear goal
  - Incomplete/wrong specification and requirements
  - Change of specification and requirements
  - Design errors, programming errors
  - Lack of planning
  - ...

How Has SE Changed?
Wasserman's Seven Key Factors

1. Critically of time-to-market
2. Shifts in the economics of computing
3. Availability of powerful desktop computing
4. Extensive local- and wide-area networking
5. Availability and adoption of object-oriented technology
6. Graphical user interfaces
7. Unpredictability of the waterfall model of software development

How Has SE Changed?
Wasserman's Seven Key Factors (continued)

- The key factors that have changed the software development



How Has SE Changed?
Wasserman's Discipline of Software Engineering

- Abstractions
- Analysis and design methods and notations
- User interface prototyping
- Software architecture
- Software process
- Reuse
- Measurement
- Tools and integrated environments

Content

- Why study software engineering?
  - Software crisis
  - Software project success rate
  - Examples of software development failures
  - Causes of software failures
  - How has software engineering changed?
- What is software engineering?
- Steps of software development
- Who does software engineering?
  - Members of the development team

What is Software Engineering

- Software
  - Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
  - Software includes:
    - The source code and binary/object code;
    - Documents such as requirements analysis, specification, design.
    - The procedure is used to setup and operate software system.

What is Software Engineering

Software

Software components		Example
Program	1. Source codes 2. Object codes	
Documents	1. Analysis/Specification	a. Formal specification b. Context diagram c. Data flow diagram
	2. Design	a. E-R diagram
	3. Implementation	a. Code list
	4. Testing	a. Test data b. Test results
Operation procedures	1. Installation and user guide 2. Troubleshooting guide	

What is Software Engineering

- Types of software
  - *System software* is the software that acts as tools to help construct or support applications software.
    - Operating systems
    - Databases
    - Networking software
    - Compilers
    - …
  - *Applications software* is software that helps perform some directly useful or enjoyable task.
    - Games
    - Information systems
    - Real-time systems
    - Embedded systems
    - Office software
    - Scientific software
    - …

Software can either be off-the-shelf or tailor-made for a particular application.

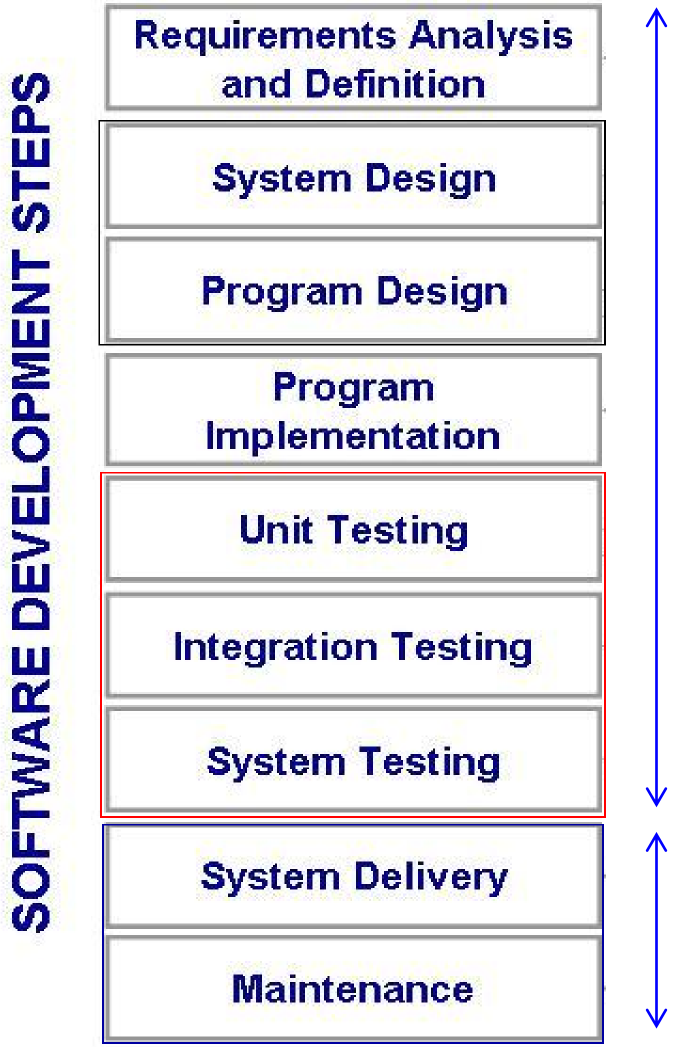
What is Software Engineering

- The first NATO conference (1968): Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.
- IEEE Standard 61012-1990 Software engineering is:
  - (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, that is, the application of engineering to software, and
  - (2) The study of approaches as in (1).

What is Software Engineering

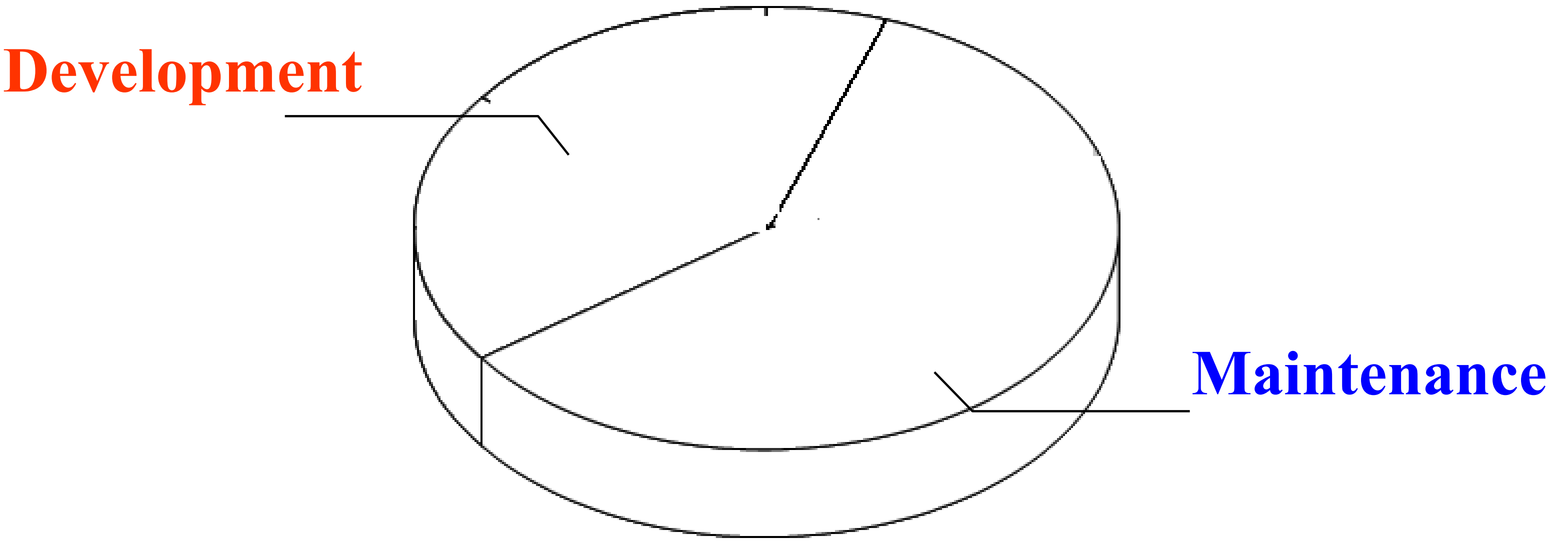
- The goal of software engineering is to create software:
  - High quality
    - Satisfying customer requirements
    - Deploying and operating easily
    - Maintaining easily
  - On time
  - On budget
  - The price is getting lower and lower

Steps of Software Development



Software development steps

- Maintenance step and various development steps
  - Maintenance alone consumes about 50-70% the total cost of a software system over its lifetime.



Steps of Software Development

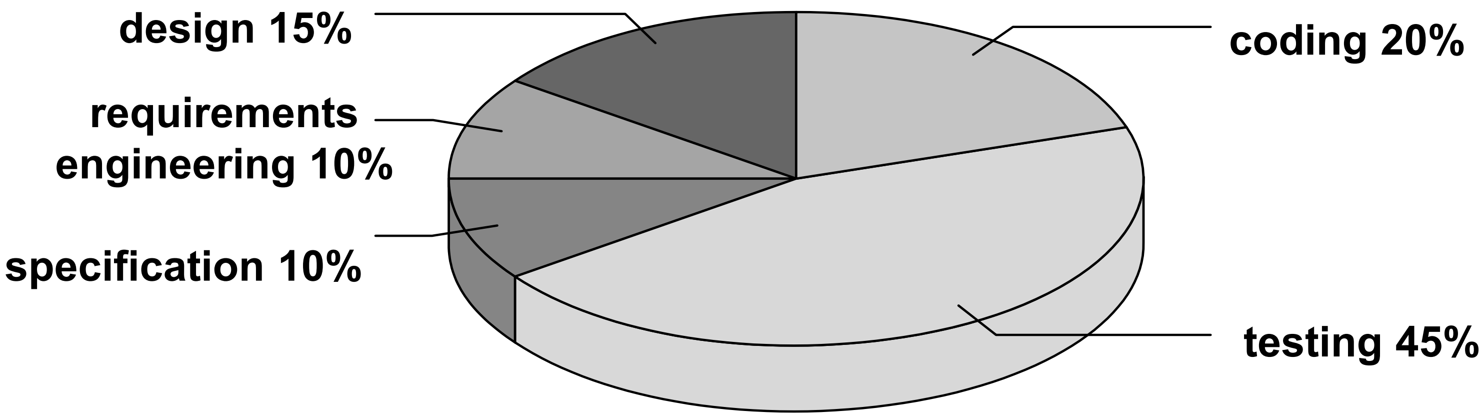
- Requirements Analysis & Definition: collecting customers' requirements and modeling the requirements.
- Design: modeling the system and detailing each module (system design and program design).
- Implementation: using a detailed design and choose a programming tools for coding modules.

Steps of Software Development

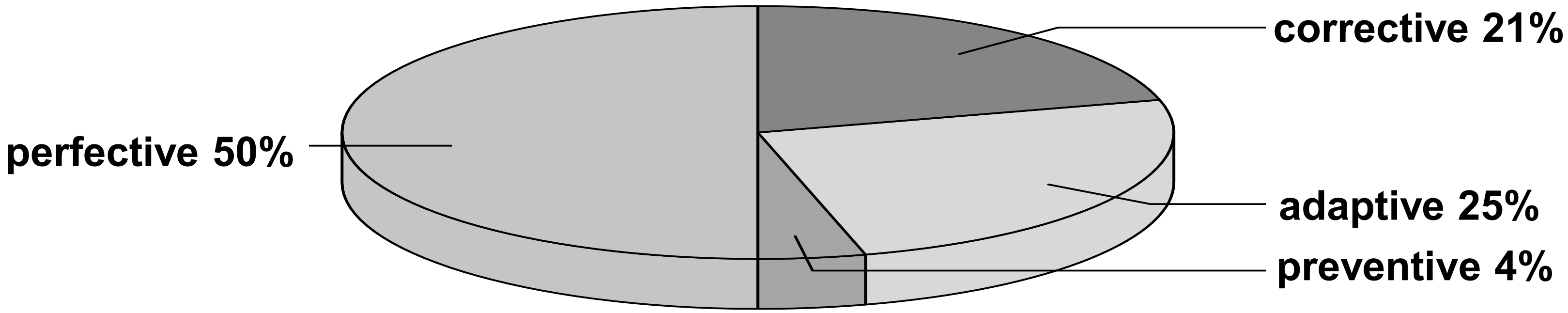
- Testing: detecting errors via program testing and system testing.
- System delivery: deploying the system at the customer side; training and supporting clients.
- Maintenance: correcting errors; improving functions; modifying software to adapt to environmental change.

Steps of Software Development

- Global distribution of effort: 40 - 20 - 40



- Distribution of maintenance activities

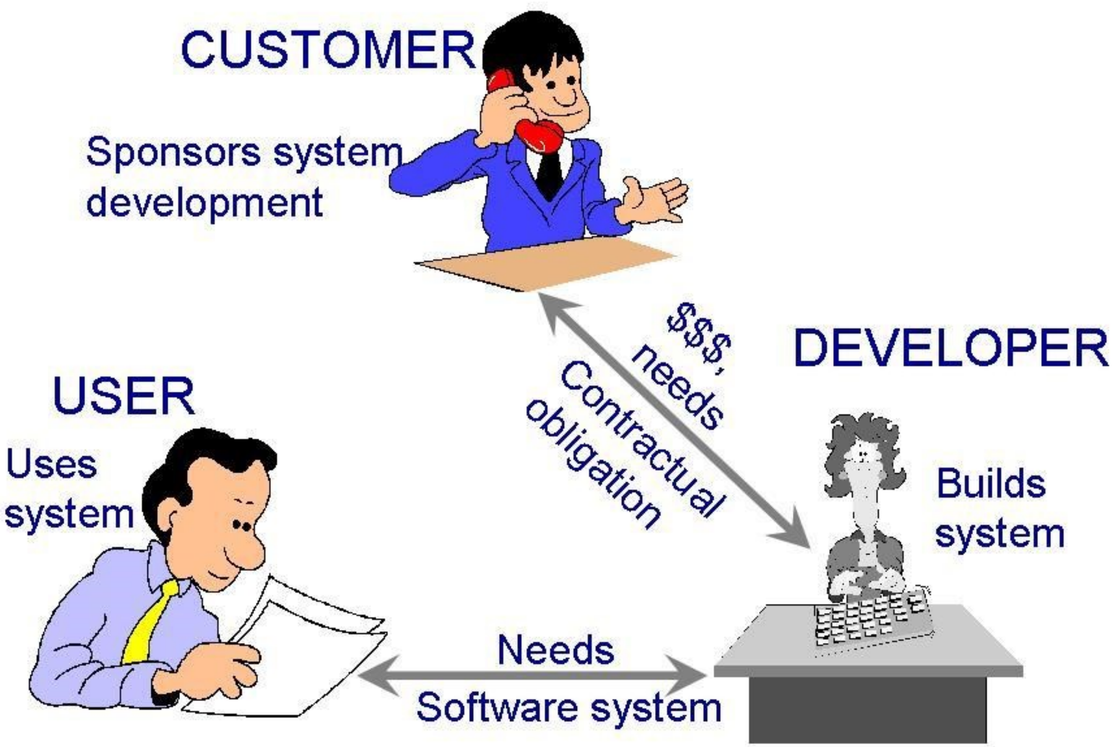


Who Does Software Engineering?

- **Customer**: the company, organization, or person who pays for the software system.
- **Developer**: the company, organization, or person who is building the software system.
- **User**: the person or people who will actually use the system.

Who Does Software Engineering? (continued)

- Participants (stakeholders) in a software development project

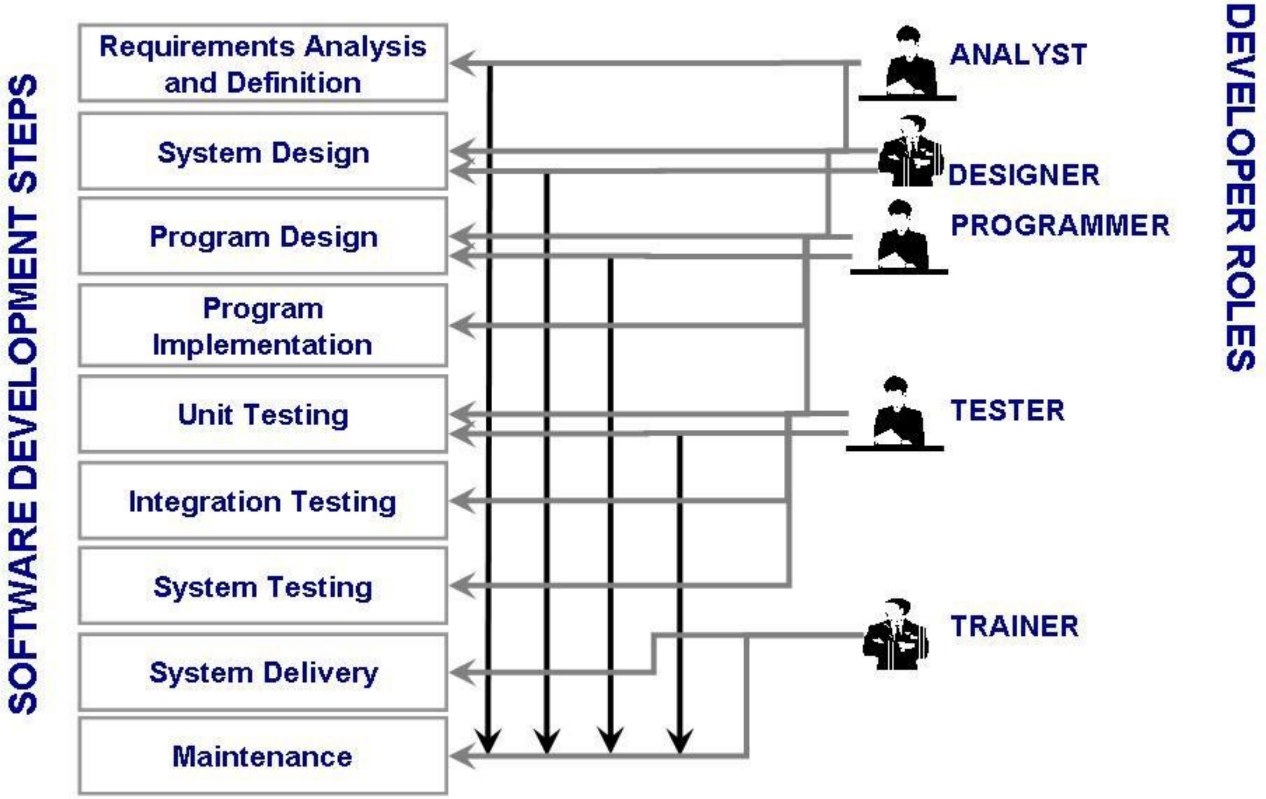


Who Does Software Engineering?
Members of the Development Team

- **Requirement analysts**: work with the customers to identify and document the requirements.
- **Designers**: generate a system-level description of what the system is to do.
- **Programmers**: write lines of code to implement the design.
- **Testers**: catch faults.
- **Trainers**: show users how to use the system.
- **Maintenance team**: fix faults or make changes to the system after the system has been accepted.
- **Librarians**: prepare and store documents such as requirements specification, design description, etc.
- **Configuration management team**: maintain a correspondence among various the requirement, the design, the implementation and the tests.

Who Does Software Engineering?
Members of the Development Team (continued)

- Typical roles played by the members of a development team



Homework

- Content:
  - Find at least 5 of the most costly software failures in 2023? For each failure, name the software and summarize the damage caused by that failure?
  - As a software developer, present your thoughts on reducing the negative effects caused by software failures?
- Format: the answer sheet is presented on 1 page of A4 paper, set the margins (top, bottom, left, right) of 2cm, use Times New Roman 12pt font, line spacing 1, justified text.
- Note: write down the references used for this homework on page 2.
- Deadline: 17h00, 02/01/2024 (submit: elearning.ctu.edu.vn)

References

- Shari Lawrence Pfleeger, Joanne M.Atlee, Software Engineering theory and practice, Pearson Prentice Hall, 4th edition, 2010.
- Hans Van Vliet, Software Engineering principles and practice, John Wiley, 2007.
- Hiep Xuan Huynh, Lan Phuong Phan, Introduction to Software Engineering, Can Tho university publishing house, 2010.

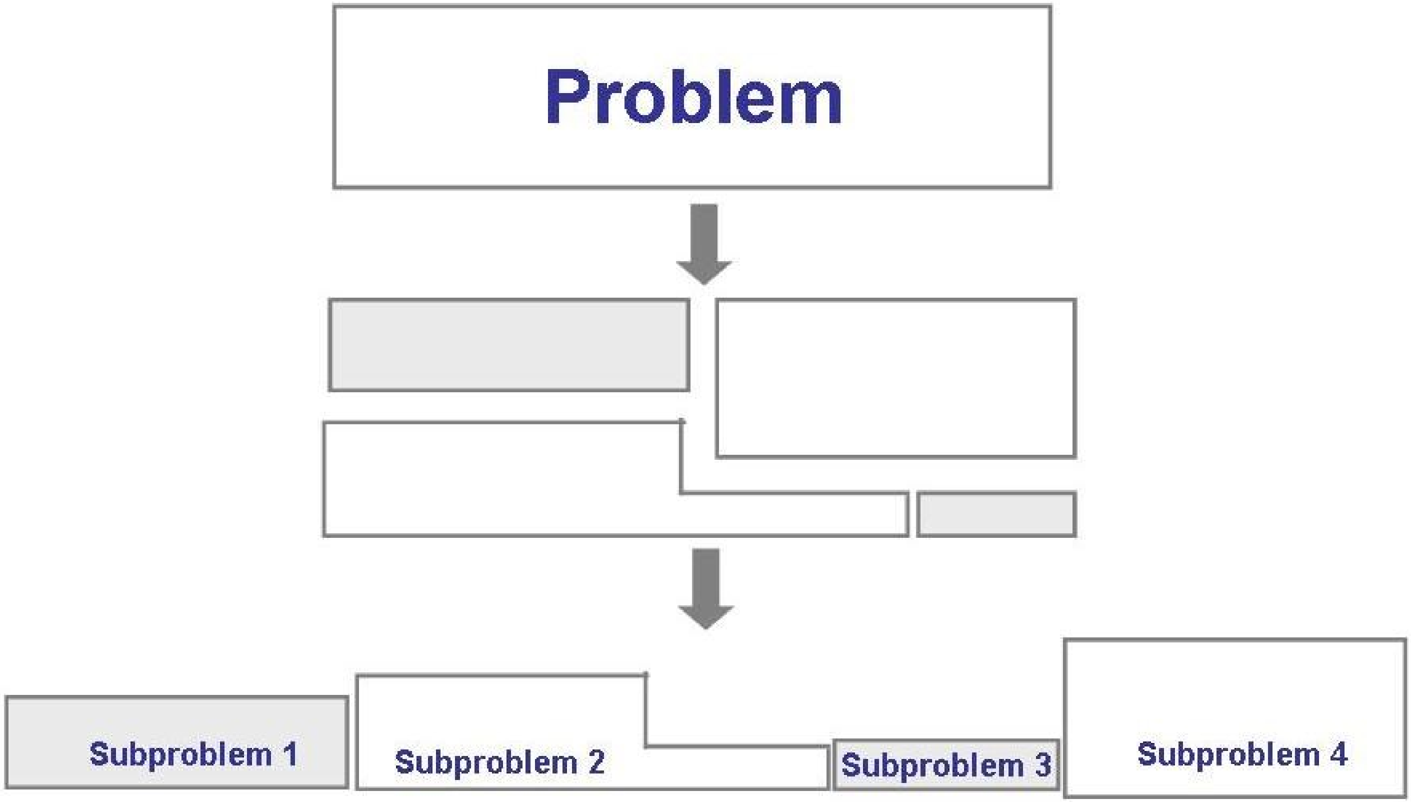
Q&A

What is Software Engineering
Solving Problems

- Software products are large and complex
- Development requires analysis and synthesis
  - *Analysis*: decompose a large problem into smaller, understandable pieces
    - abstraction is the key
  - *Synthesis*: build (compose) a software from smaller building blocks
    - composition is challenging

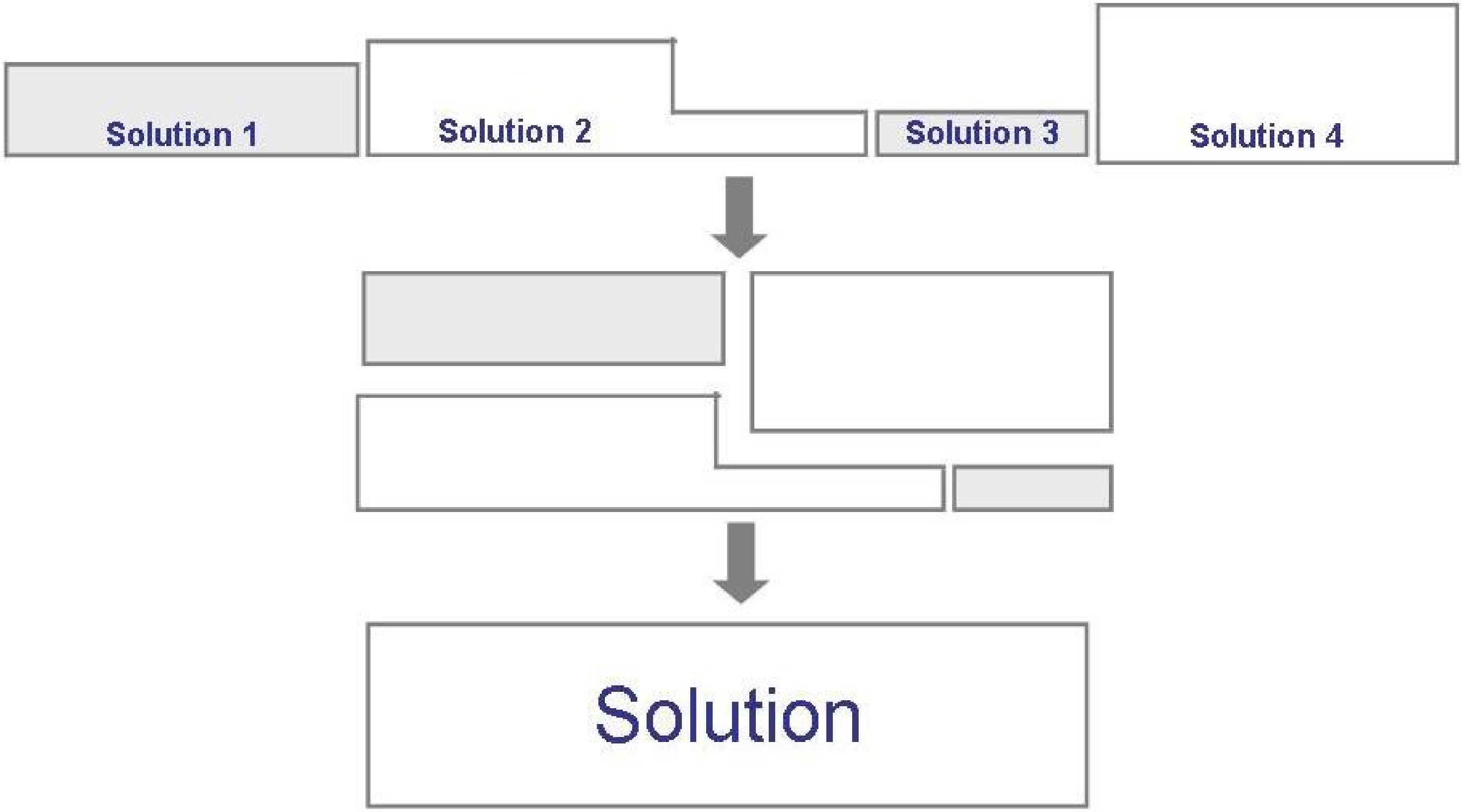
What is Software Engineering
Solving Problems (continued)

- The analysis process



What is Software Engineering
Solving Problems (continued)

- The synthesis process

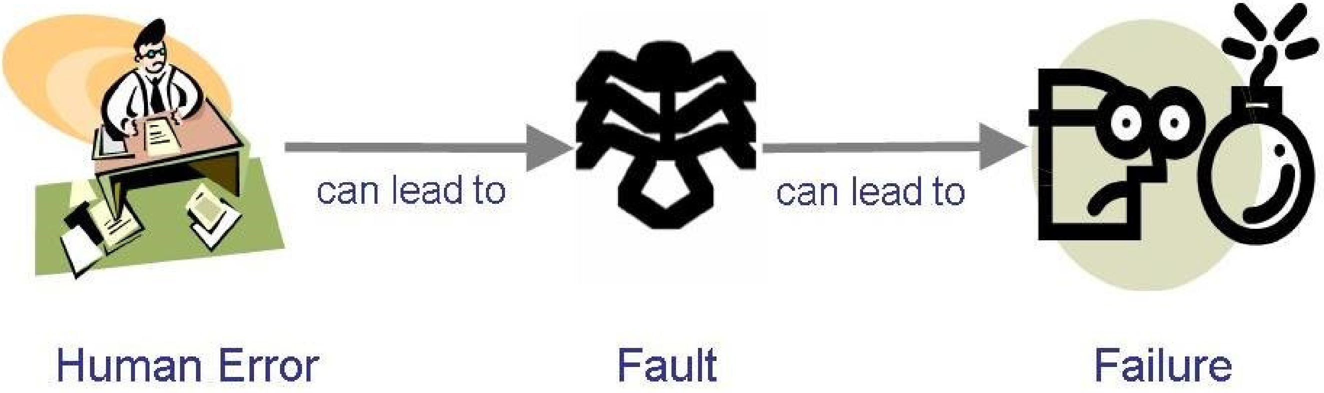


What is Software Engineering
Solving Problems (continued)

- **Method**: refers to a formal procedure; a formal "recipe" for accomplishing a goal that is typically independent of the tools used
- **Tool**: an instrument or automated system for accomplishing something in a better way
- **Procedure**: a combination of tools and techniques to produce a product
- **Paradigm:** philosophy or approach for building a product (e.g., OO vs structured approaches)

How Successful Have We Been?
Terminology for Describing Bugs

- **A fault**: occurs when a human makes a mistake, called **an error**, in performing some software activities
- **A failure**: is a departure from the system's required behaviour


can lead to
can lead to
Human Error
Fault
Failure