CS350 Lecture 1 Overview of SE

Fall 2018

Doo-Hwan Bae
School of Computing
bae@se.kaist.ac.kr

Contents

- What?
  - Science/Engineering, Software, Software Engineering,
- Why Software Engineering?
  - Why?
  - Nature of Software
- 4 C's in Software Engineering
  - Complexity
  - Change
  - Cost
  - Communication
- Summary

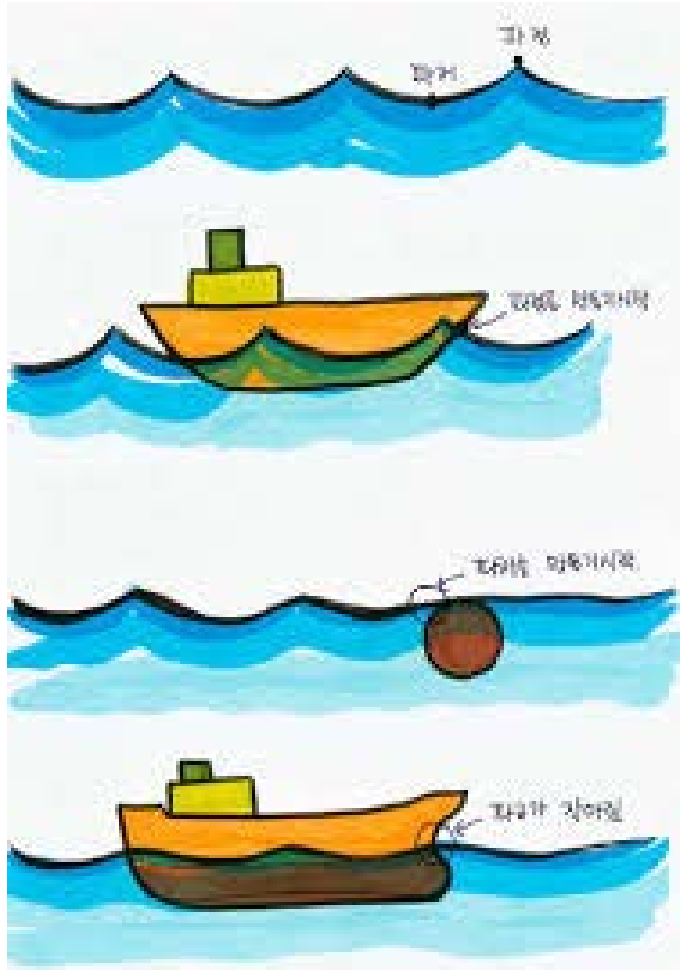
What is ‘Science’?

- Science vs. Engineering

- Chemistry vs. Chemical Engineering
- Example: Transforming Coal into Gas(fuel for vehicles)
  - Scientist 1: 1000kg coal, 100 litters of fuel, 6 months
  - Scientist 2: 1000kg coal, 10 litters of fuel, 1 month
- Chemist finds a new method for transforming ➔ contribution!
- Chemical engineer chooses an option to satisfy a specific purpose, & builds a plant for it.
- Q: Which option do you, as a chemical engineer, choose?
  - Do you have a right option for you?
  - If not, what to do?

What is ‘Engineering’? (1/2)

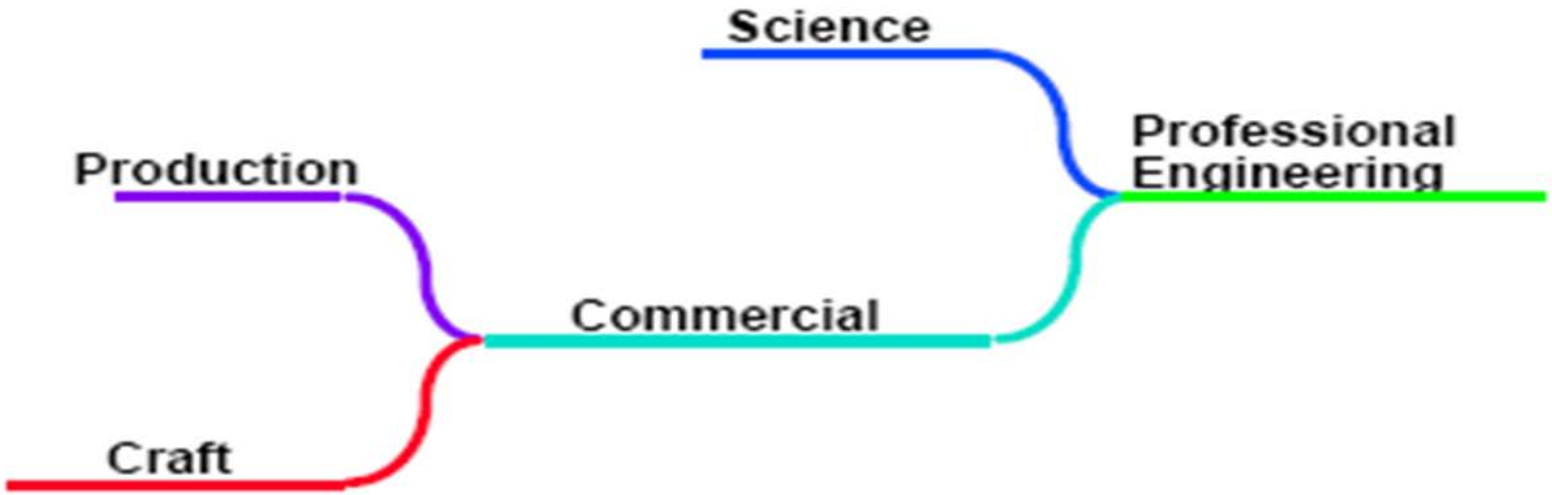
- Do you find anything strange here?



- A professor’s engineering insight
  - Wave in prow(이물)
    +
  - Wave in bulb underneath
    -

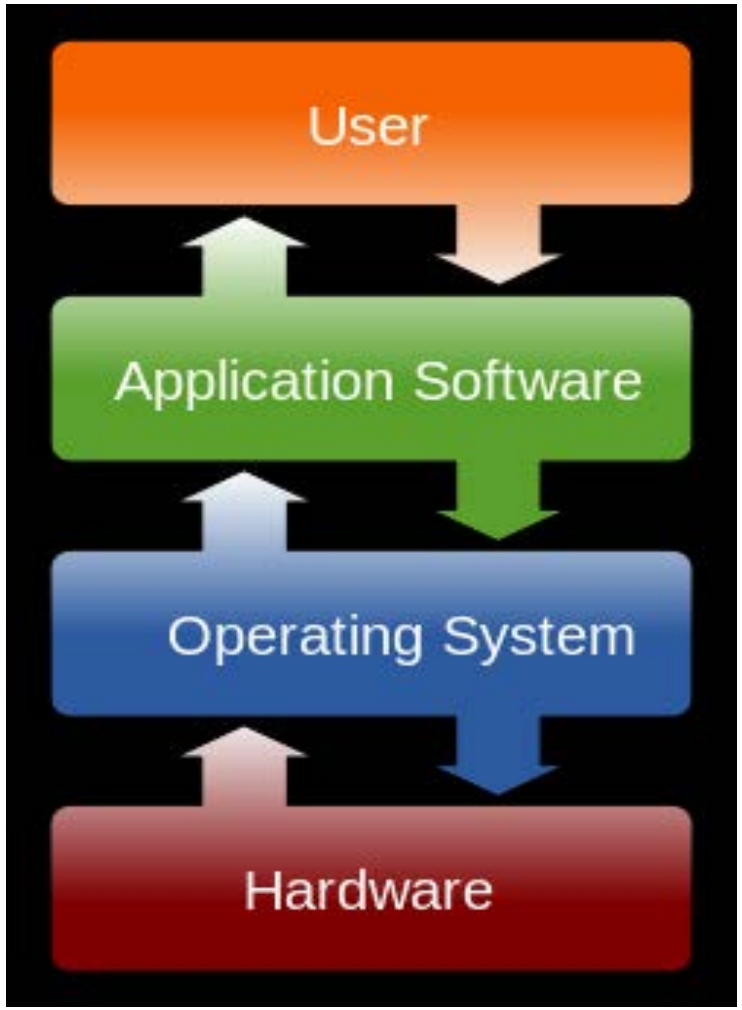
What is Engineering? (2/2)

- Evolution of Engineering Discipline with Science together



What is ‘Software’? (1/3)

- **Computer software**, or simply **software**, is that part of a computer system that consists of encoded information or computer instructions, in contrast to the physical hardware from which the system is built. (Wiki)



What is Software? (2/3)

In comparison to 'Program'

Programs vs Software Products

Characteristics	Program	SW product
Users	self	Others
Number of user	Self/few	Large number
Size	small	Large
Functionality	limited	Large
Interfaces	Ok	Well designed
Environment	One	Several
System	Used by itself	Works with other systems
User background	Similar	Varied
Presence of bugs	Not a major concern	Major concern
Documentation	Minimal	Exhaustive
Testing	Minimal	Exhaustive
Cost/user	High	low
Developers	One /few	Many
Use of standards, etc	Not essential	essential

What is Software? (3/3)

In comparison to 'Program'

- Software = Program + Document (?)
  - Does this equation make sense to you?
- Software Engineering (=, >, <) Programming + Documenting
  - Q: What else needs to be added on the right-hand side?

- Programming: personal activity, technical,
- Software Engineering: team activity, technical & managerial

- Q: In SE, what needs to address(or manage) in addition to programming?

Nature of Software (1/2)

- Software transforms data using computers
- Software is invisible/intangible
- Software is malleable: Easier to modify than hardware
- Software development is human-intensive; no real cost of materials
  - Physical Engineering (Manufacturing) vs. Software Engineering
  - But, there are similarities between the two.
    - Principles borrowed from hardware engineering
    - Avoid known mistakes

Nature of Software(2/2)

- Software itself is a large/complex system or a component of larger complex system (Complexity)
  - Interacting with other software
  - Interacting with other hardware, mechanical devices
- Software requires working with various people (Communication)
- Software is very expensive to develop, but more expensive to maintain due to inevitable changes. (Cost, Change)

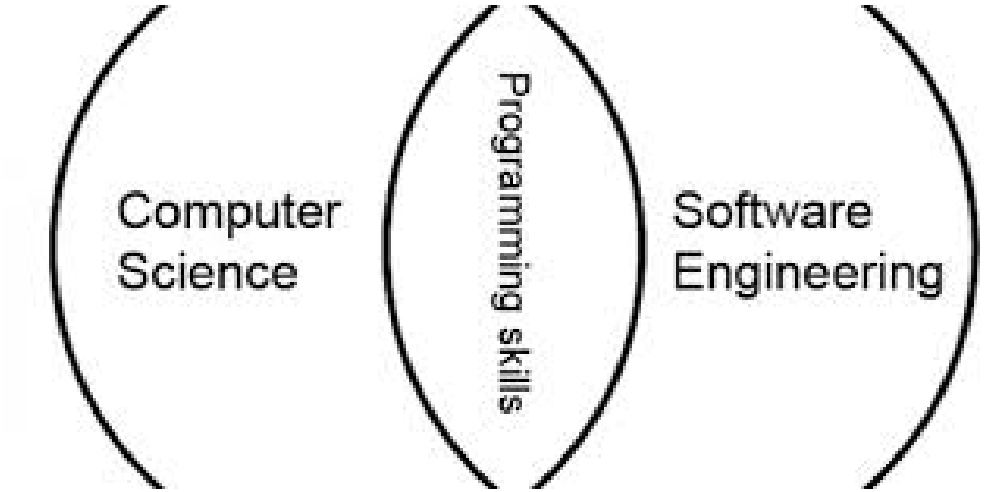
Environmental Changes for Software

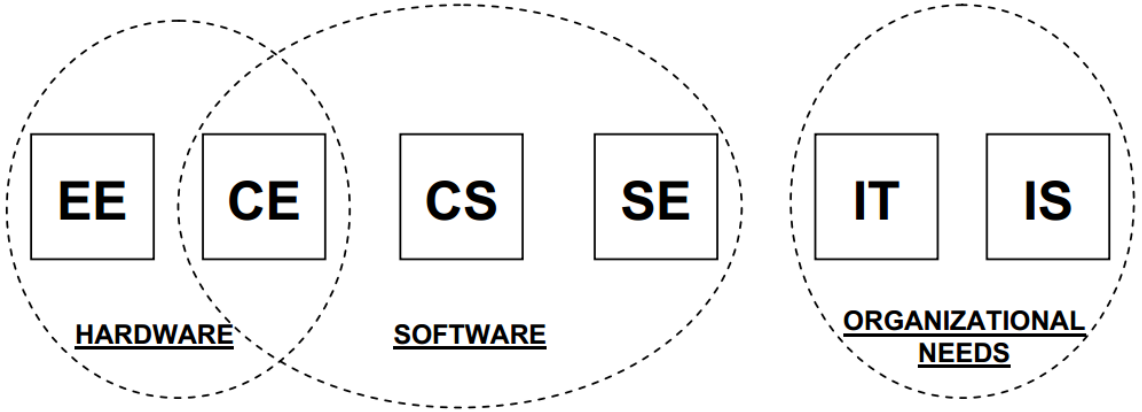
- Size gets larger and complex; application domains increased
- More distributed/concurrent execution, real-time
- More users involved
- Virtually, software is eating the world
- Various layers* available
  - Commodity layer: every service/product provider has its own
  - Value-added layer: Value-added on top on the commodity part
  - Innovative layer: creating something new, open up a new market

* Jan Bosch at the ICSE2017

What is Software Engineering (1/5)

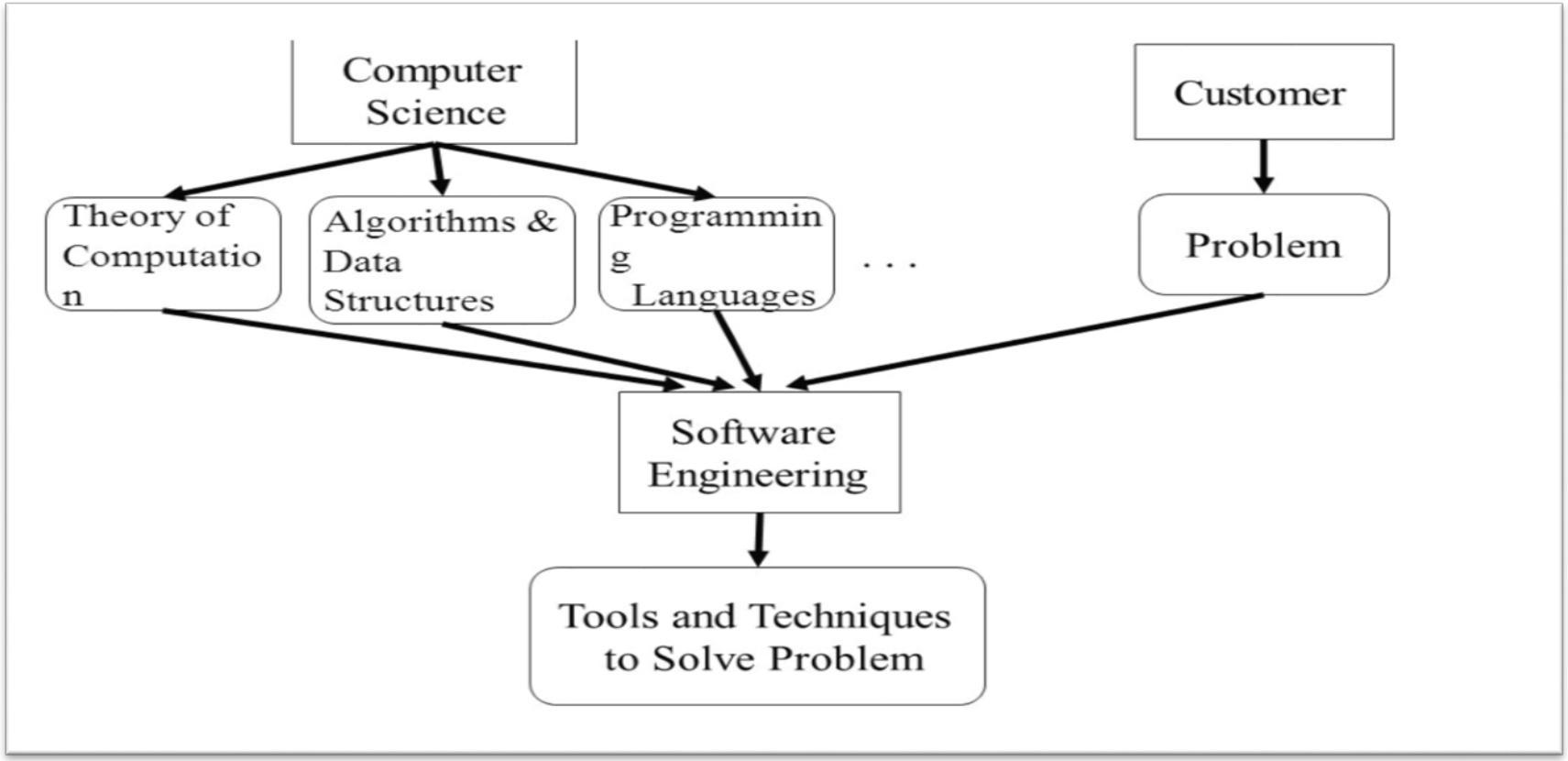
Computer Science vs. Software Engineering by ACM





What is Software Engineering (2/5)

- Relation btw. CS and SE



What is Software Engineering (3/5)

- Definitions for Software Engineering
  - (IEEE): The application of a systematic, disciplined, quantifiable approach to the development, maintenance and operation of software
  - What you do when you have two or more people working on a project (NRC Canada)
  - What software engineers do

What is Software Engineering (4/5)

Goal of Software Engineering

- Produce quality software on time within budget that satisfies user's needs.

- Produce quality system and service

- Eventually, make business profit through software

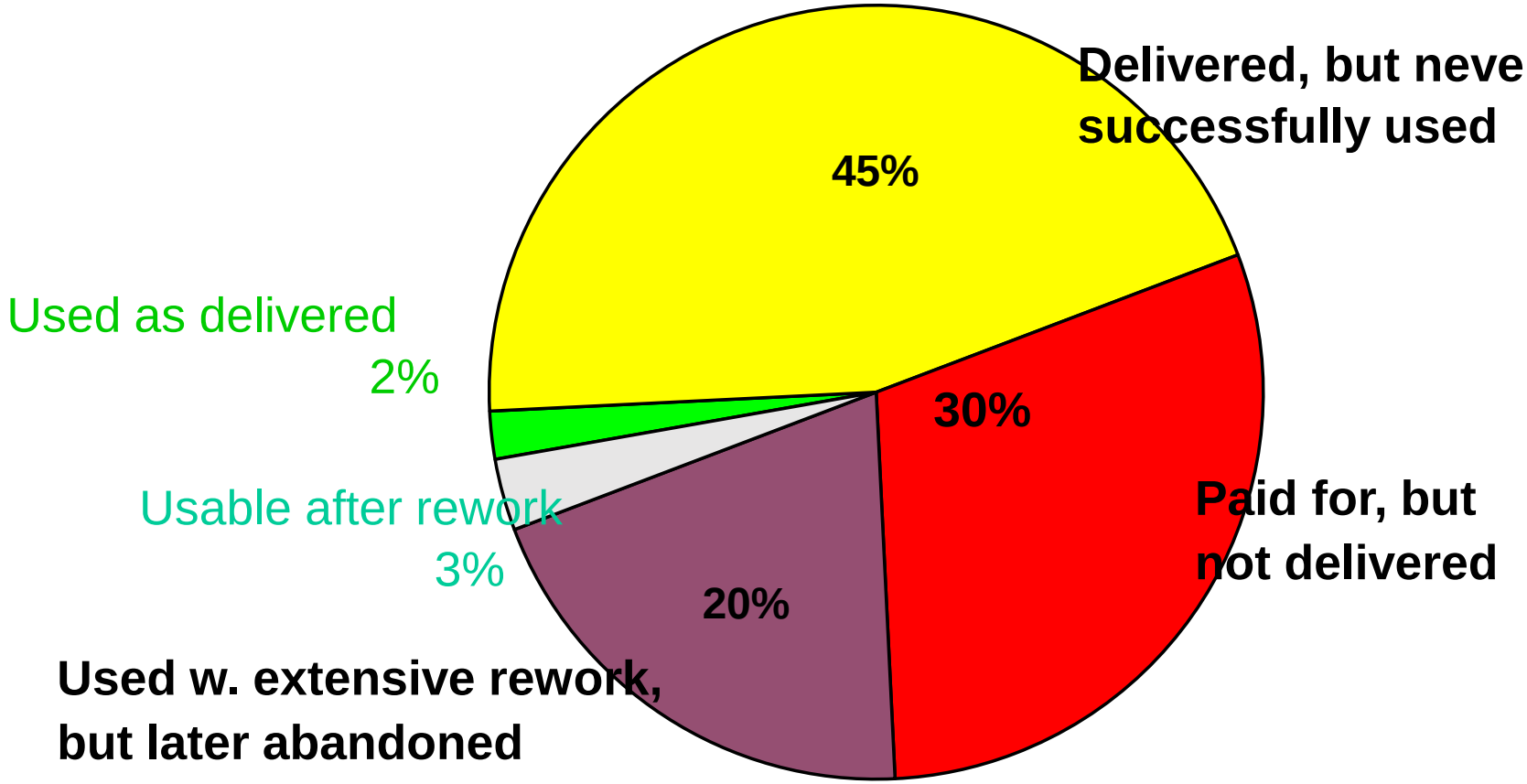
What is Software Engineering (5/5)

- 'Software Engineering at Google', by Fergus Henderson
  - Software development
  - Project management
    - 20% time: spend 20% od time working on any project without approval.
    - .....
  - People management
    - Roles: engineering manager, software engineer, research scientist, site reliability engineer, product manager, technical program manager

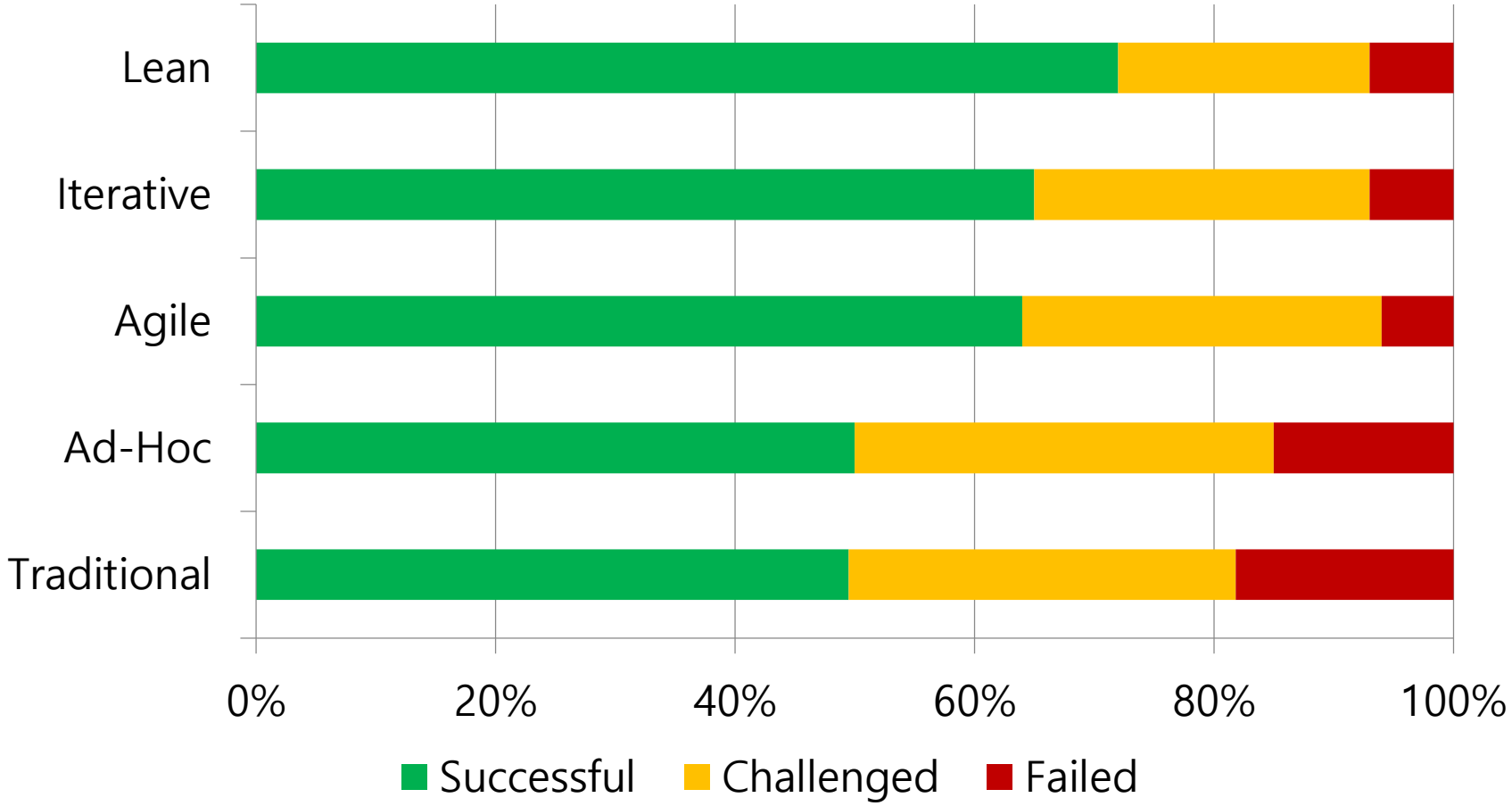
Why Software Engineering? (1/2)

9 software projects totaling $96.7 million: Where The Money Went

[Report to Congress, Comptroller General, 1979]



Why Software Engineering? (2/2)
Project Success Rate after more than 30 yrs



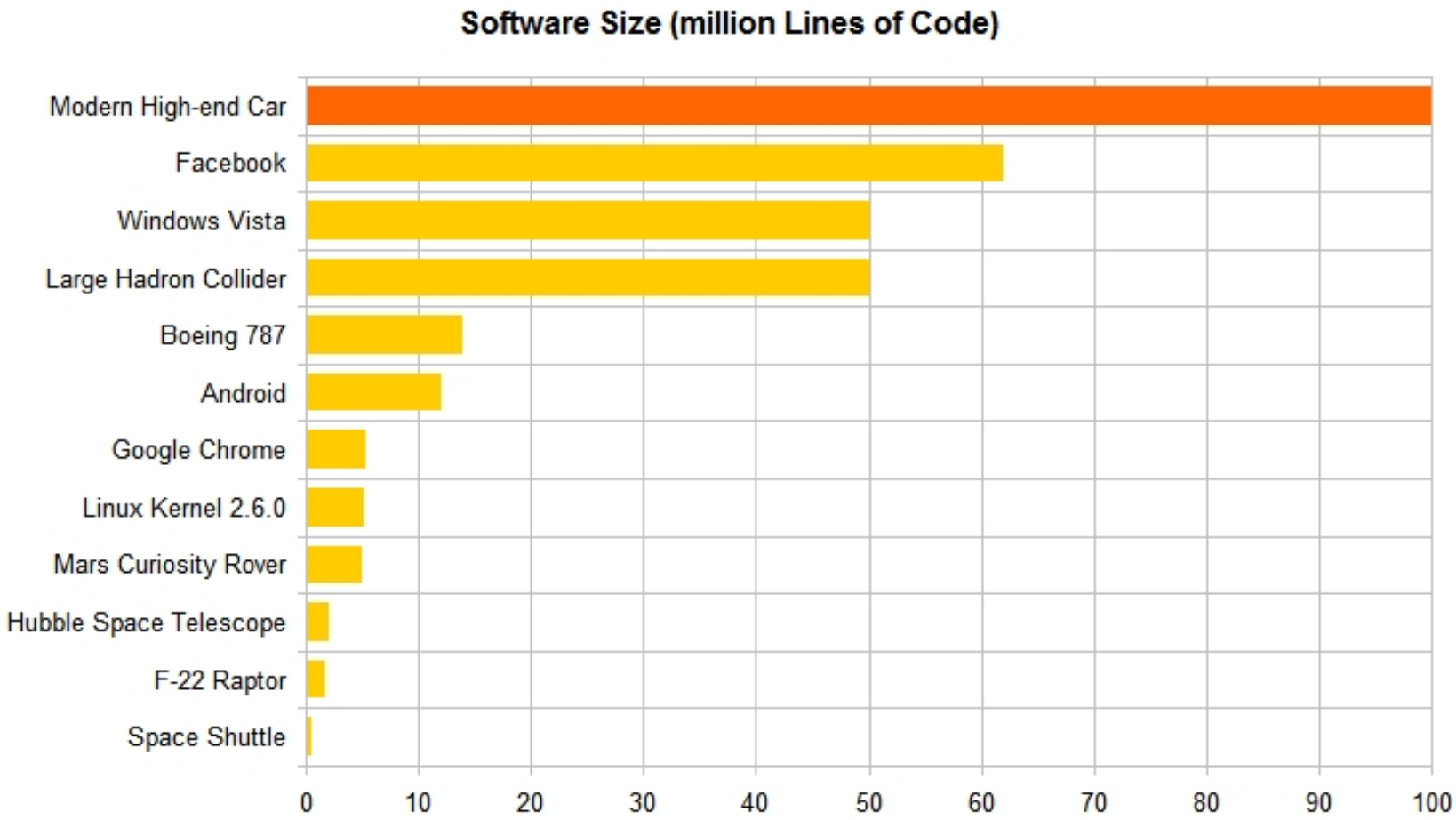
4 C's in Software Engineering?

What to address (or manage) in Software Engineering:

- Complexity
- Change
- Communication
- Cost

Complexity (1/3)

- Software size



Complexity (2/3)

- A complex program example: https://dzone.com/articles/dependency-analysis-and-1

Complexity (3/3)

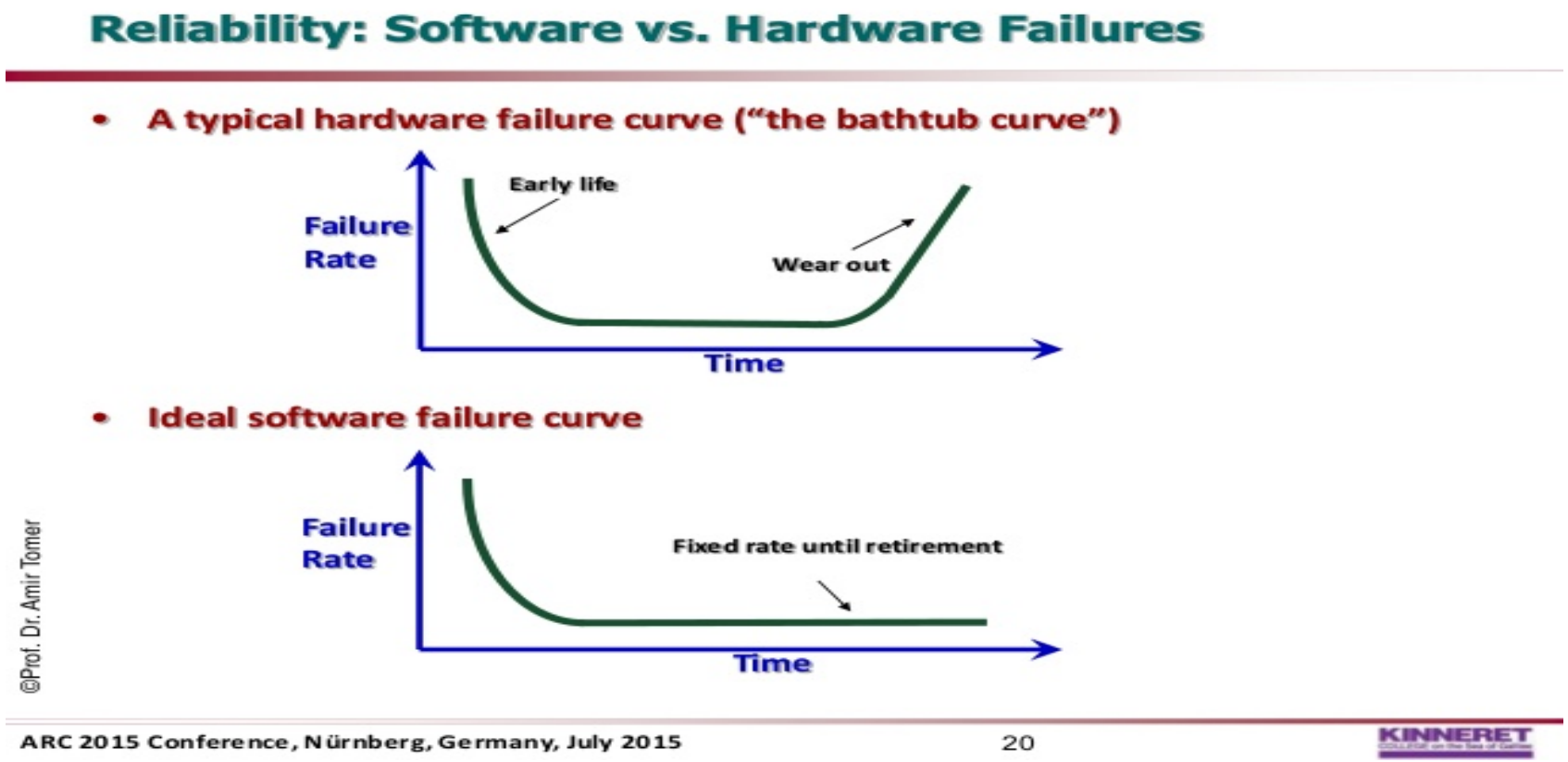
- How complex software can be?
- Why is software becoming more complex each day?
  - Study 4TH Industrial Evolution(4차 산업혁명), and find out by yourself!
- How to handle complexity involved in software development?
  - Can we reduce it? How?
  - If not, can we manage (keep it under control) it?
  - What are the methods or techniques to manage it?

Change (1/3)

- What are the changes encountered during software development?
  - Requirements
  - Design/Code,..
  - People
  - What else?
  - Software is evolving like a life form. Thus, changes are inevitable!
- How to deal with such changes during software engineering practice?
  - ….

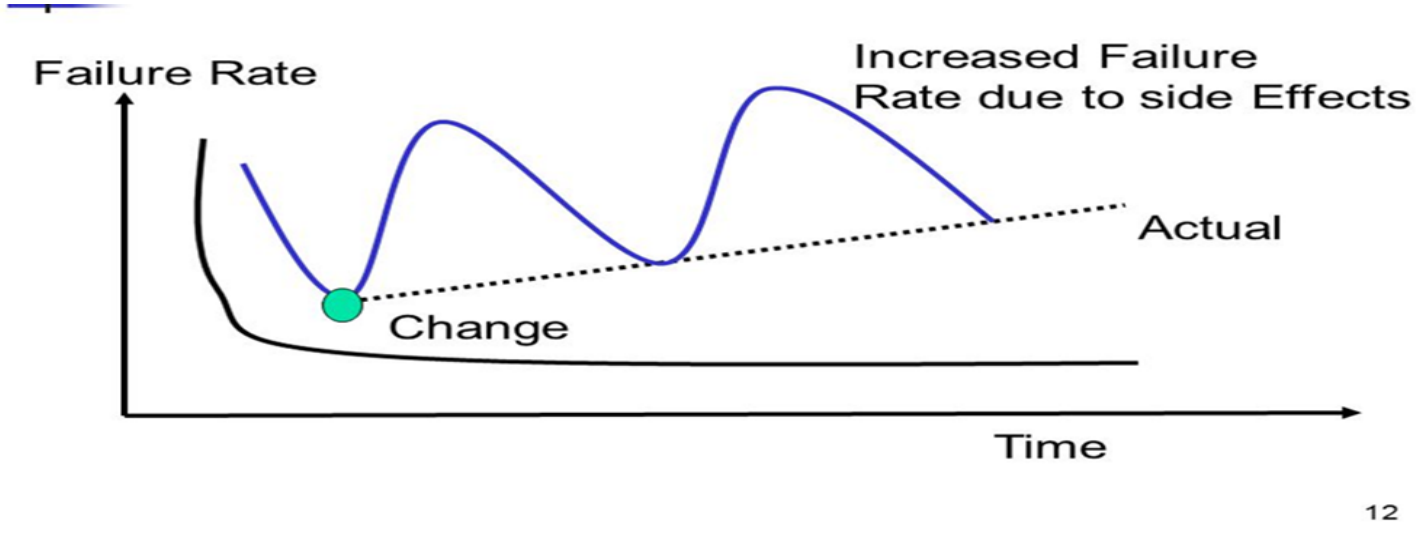
Change (2/3)

- Hardware vs. Software



Change (3/3)

- Real-world Software Change&Failure
- What to do when it gets worse?

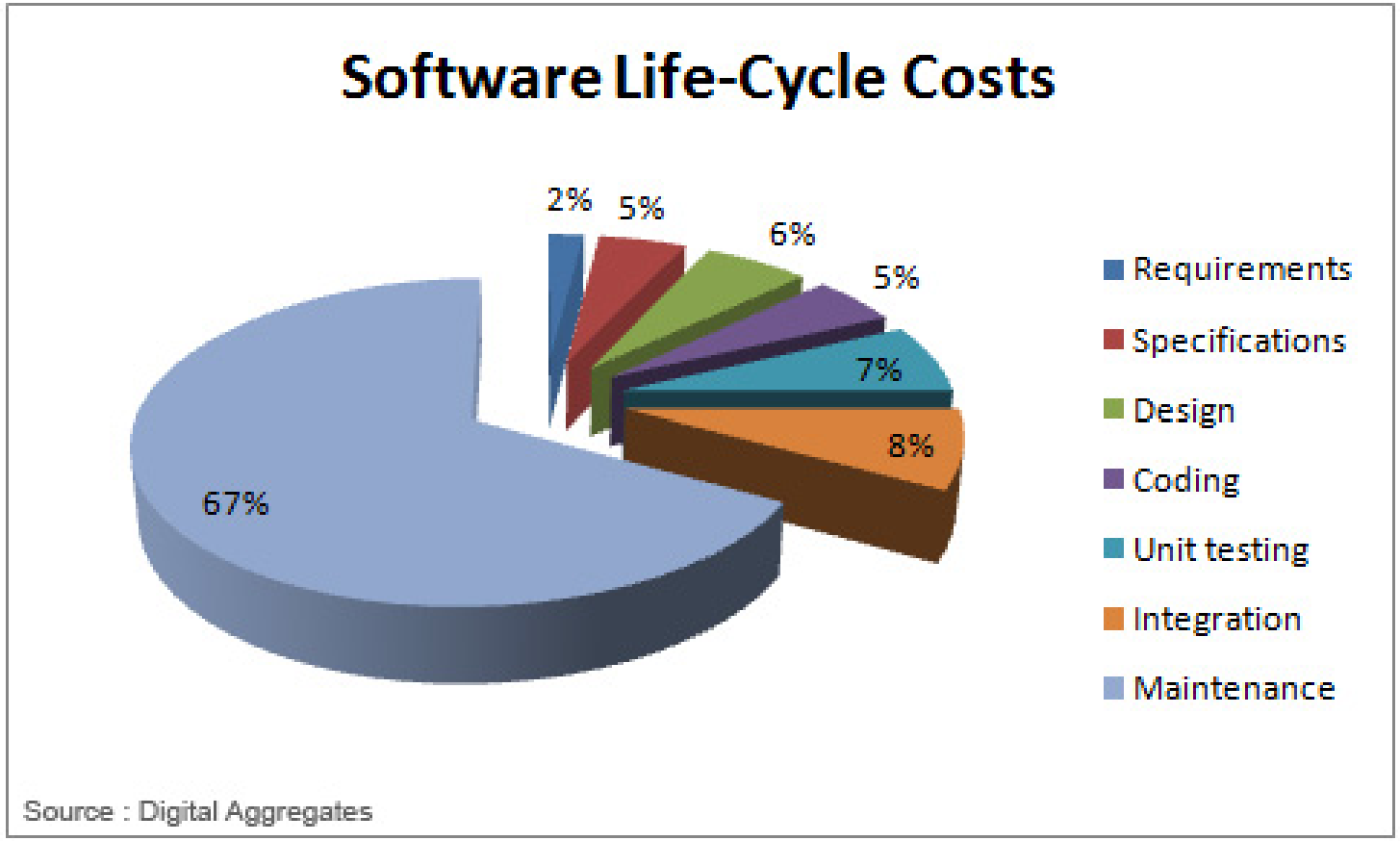


Cost (1/6)

- What is most dominant cost factor involved in software business?
  - Software business includes
    - Software development
    - Software maintenance
    - Software acquisition
    - Software operation
  - Past: Computing power, most expensive
  - Now: Software engineers' labor cost
  - Future: What would be?
- How to deal with Software cost?

Cost (2/6)

- Effort Distribution
- Which phase?
  - Dev.
  - Or Maintenance



Cost (3/6)

- NASA Space Shuttle control software: The best SW ever built!
  – 420,000 lines of code
  – 1 error each in last three versions
  – 17 errors total in 11 versions
  - Rumor saying that there was no failure while operating in air!
- Commercial projects of the same size
  – 5000 errors
- But…
  – The highest $/loc of any project
  - Q: Can you guess how much?

Cost (4/6)

- What happens when…
  - you don’t have NASA’s budget
  - bugs don’t kill people
  - you must offer lots of features
  - users will tolerate bugs
  - it helps being first to market

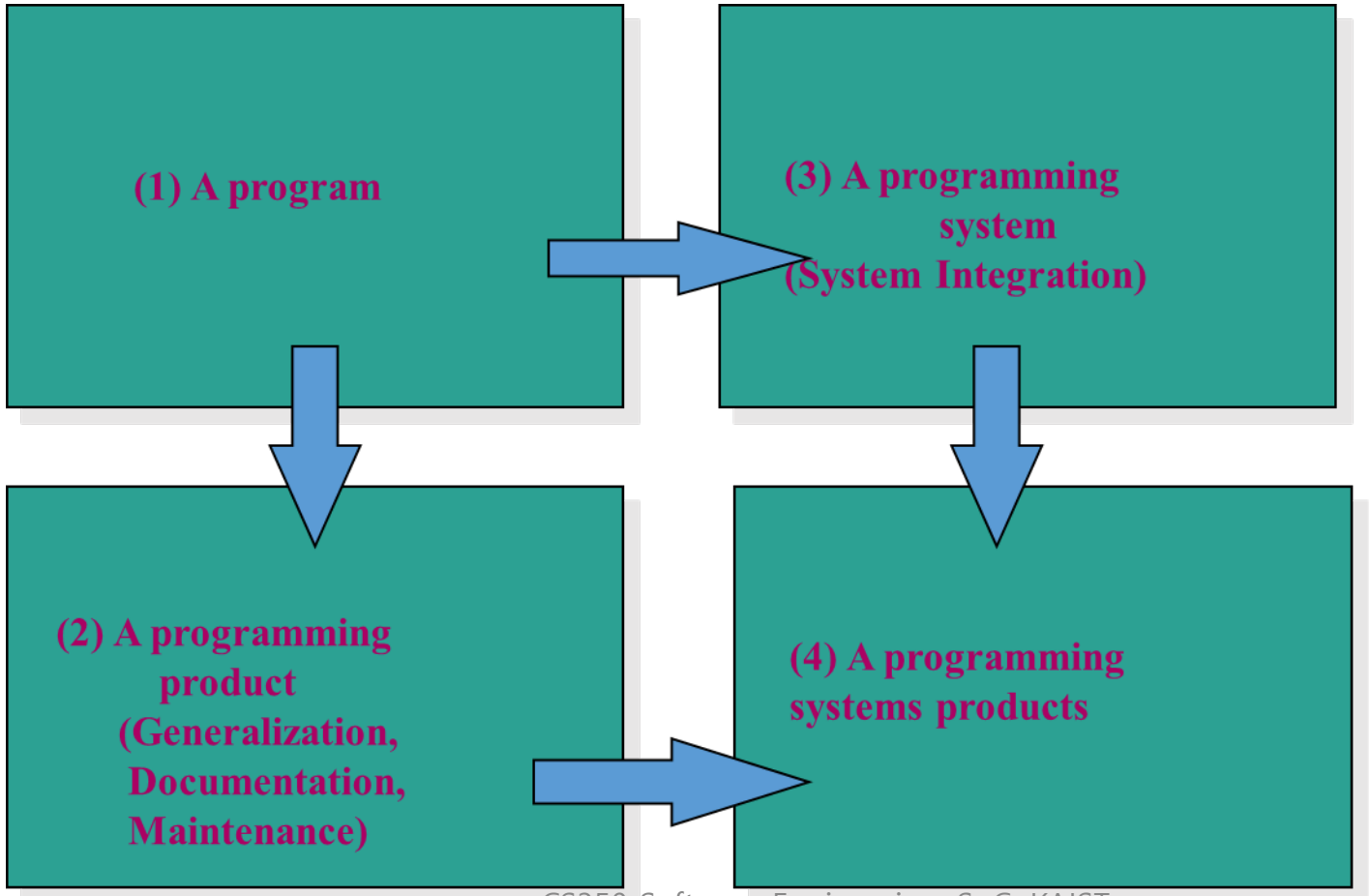
- Software Crisis?
  - Low quality, budget & time overrun
  - Programmers working days and nights

Cost (5/6)

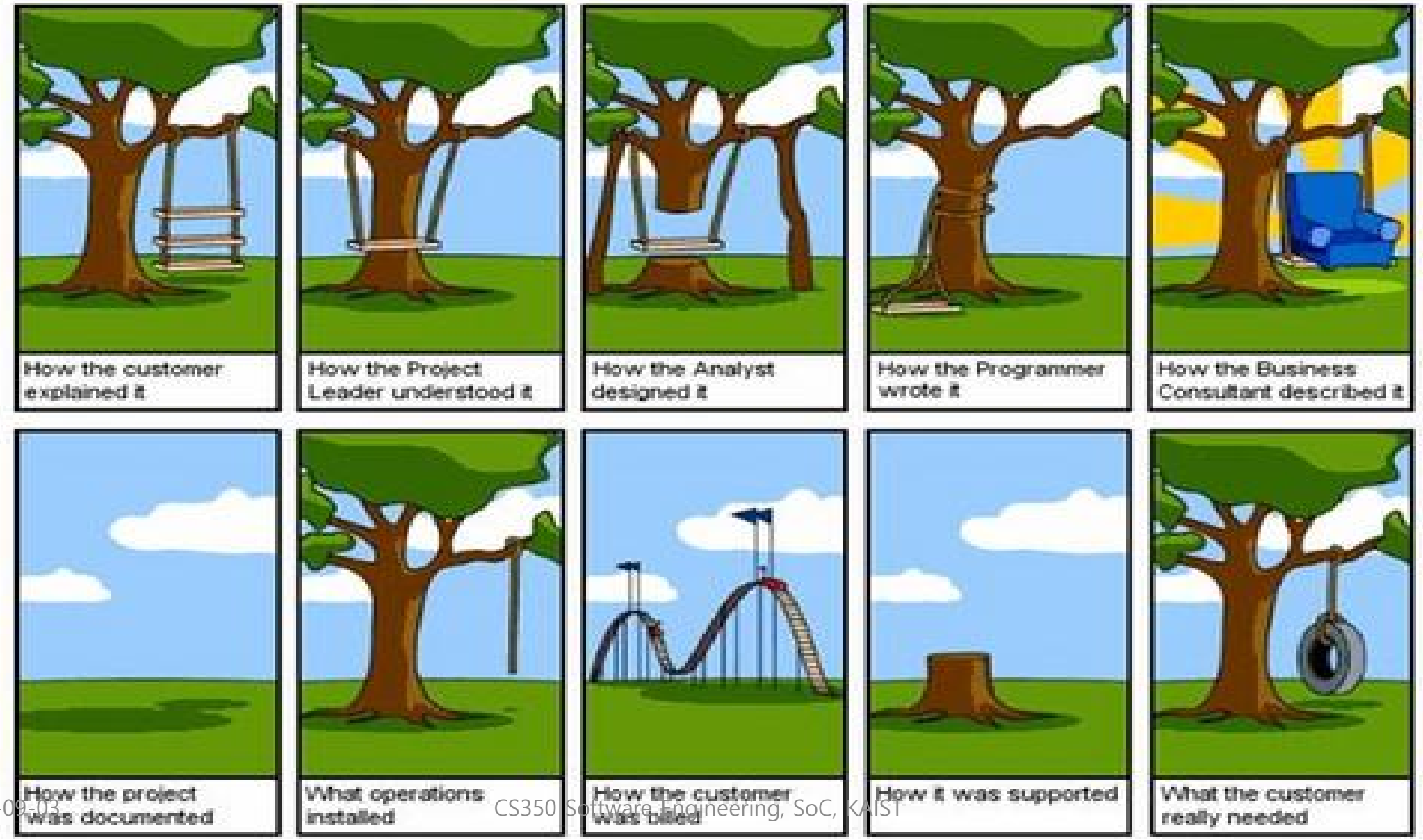
- What cost can we reduce/manage?
- How can we do that?
- Q: What is the goal of Software Engineering?
  - Developing a quality software system on time and within budget,
  - Or, making profit from software?

Cost (6/6)

- Cost by types of Software(X3) (Source: Brooks' Mythical Man Month)



Communication (1/2)



Communication (2/2)

- Who does a software developer need to communicate with?
  - Other software developers with the same role as well as different roles
    - What roles are involved in software development?
  - Stakeholders
    - Business managers, Software project managers
    - Users/customers, maintenance people,

- How to facilitate communication during software development without ambiguity?
  - Compare with natural languages.

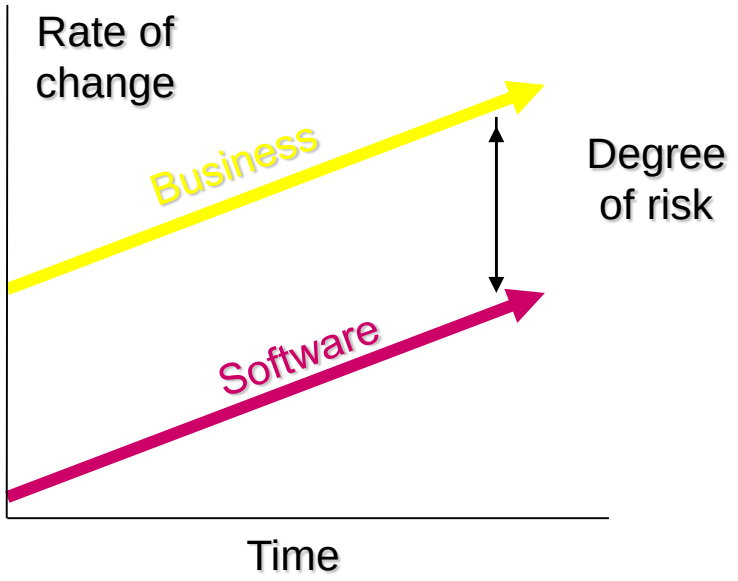
Discussion: Predicting the Future?

- Will a ‘programming position’ be good for you?
- Will a ‘software engineering position’ good for you?
- In order to answer these questions, you need to know the future changes in various aspects

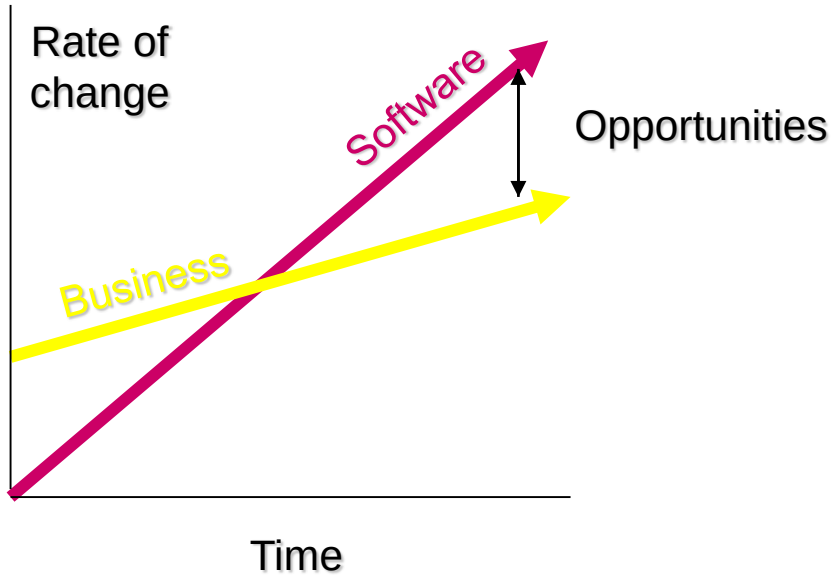
- In order to predict the future, what can you do now?

Discussion: "Can we, as a Computer Scientist, do better in future than now?"

Understanding Business implication of Software Role



When it takes longer to change the software than to change the business, the business is at risk



When software changes faster than the business, the business creates strategic opportunities