

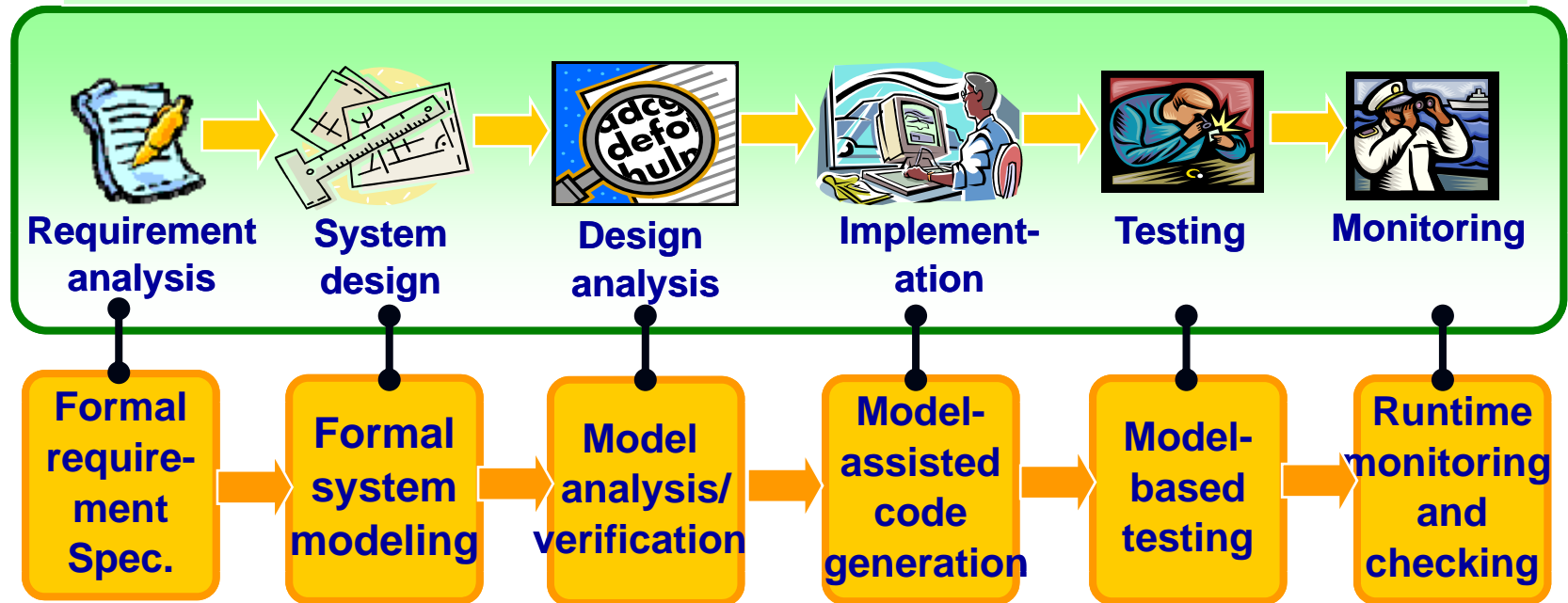
Introduction to Software Engineering (2/2)

Moonzoo Kim
CS Division, EECS Dept.
KAIST

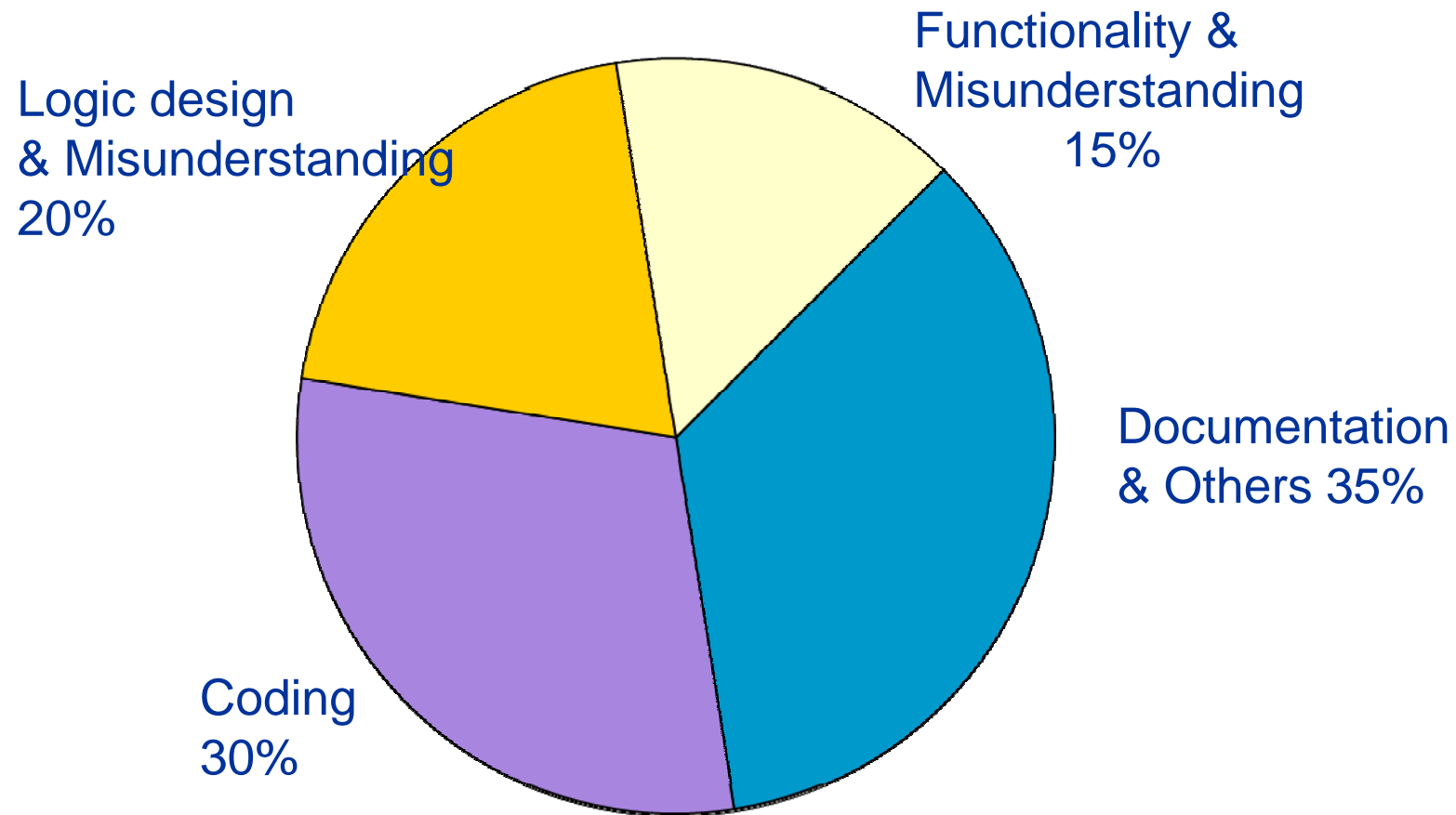
(slides from CS550 '06 taught by prof. D. Bae)

Software Development Process

A SW Development Framework for SW with High Assurance



Sources of Errors in S/W Developments



Ex. Requirement on Retail Chain Management Software

- Find ambiguous points in the following requirement
 - If the sales for the current month are below the target sales, then a report is to be printed,
 - unless the difference between target sales and actual sales is less than half of the difference between target sales and actual sales in the previous month
 - or if the difference between target sales and actual sales for the current month is under 5 percent.

Scope of S/W Engineering

- Historical Aspects
- Economic Aspects
- Maintenance Aspects
- Specification & Design Aspects
- Team Programming Aspects

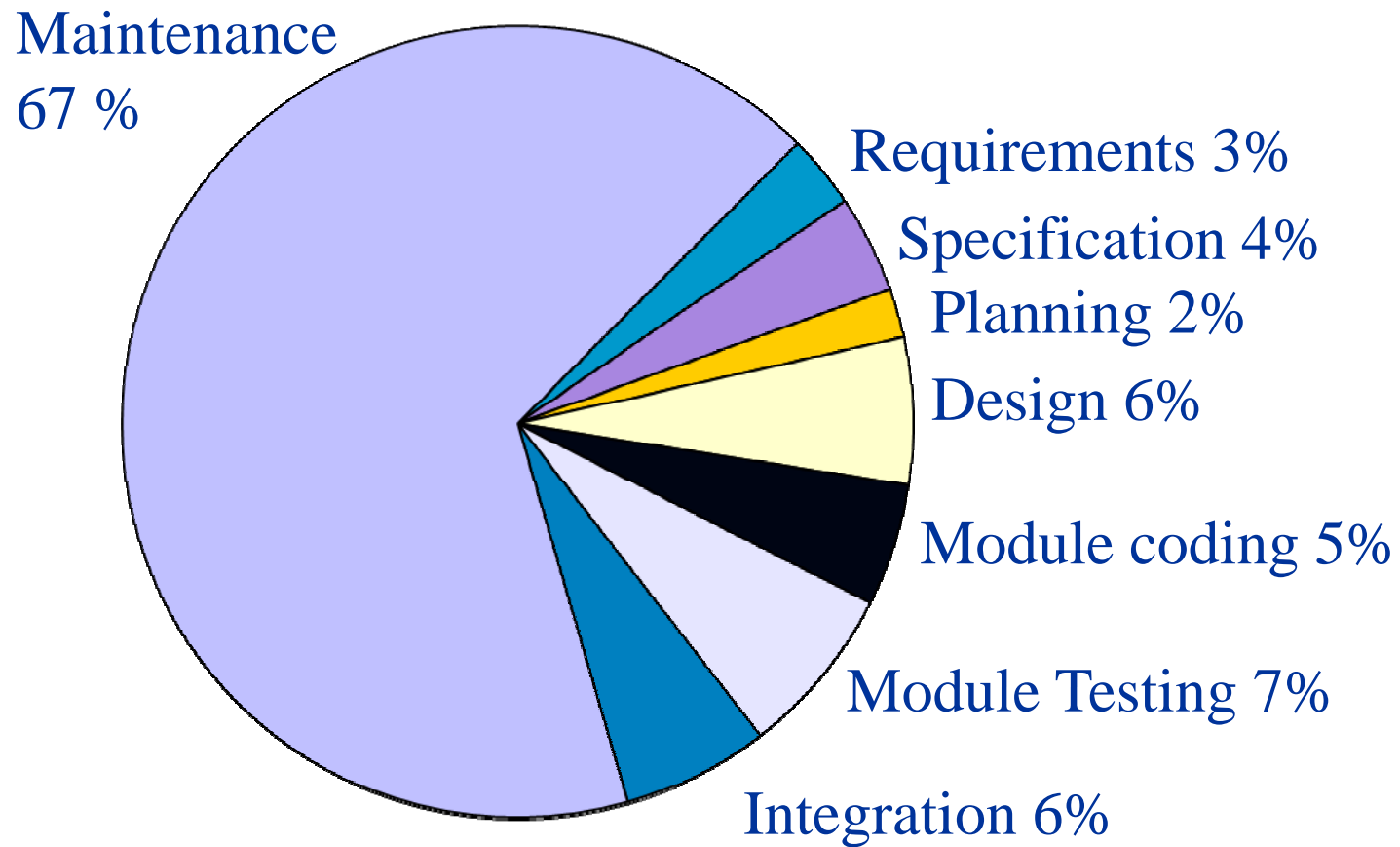
Historical Aspects

- 1967, A NATO group coined the term " Software Engineering"
- 1968, NATO conference concluded that software engineering should use the philosophies and paradigms of established **engineering disciplines**, to solve the problem of software crisis

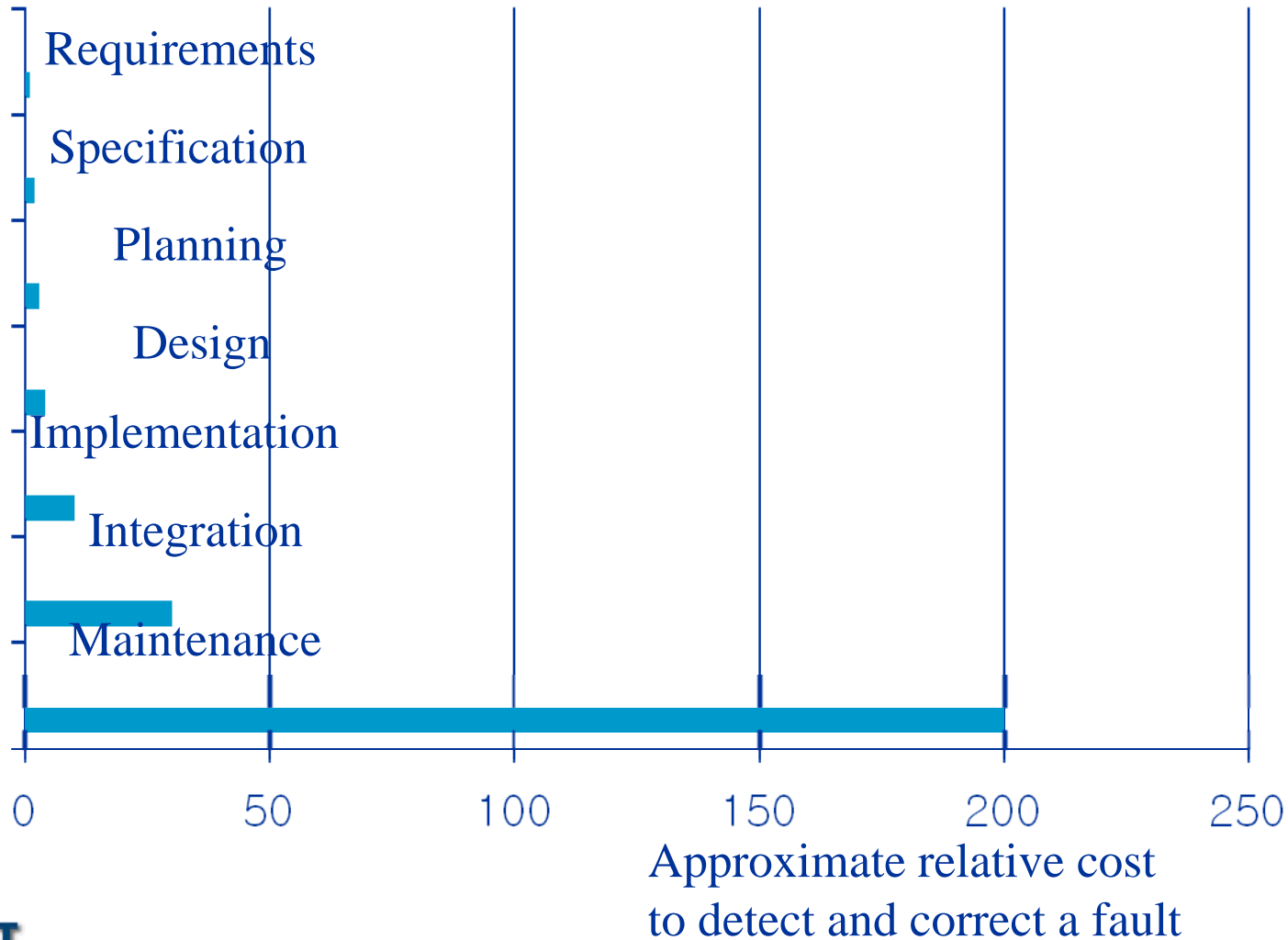
Economic Aspects

- Relationship between computer science and software engineering
 - cf: chemistry and chemical engineering
- Software engineer is intended in only those techniques which make sound **economic sense**, while computer scientists investigate a variety of ways of producing software, some good and some bad

Maintenance Aspects



Specification and Design Aspects



Team Programming Aspect

- Parnas, "Multi-person construction of multiversion software."
 - Programming : personal activity
 - S/W engineering : team activity

Team Programming Aspect (Cont.)

(From programming to sw engineering)

- Programming in early days
 - The problem is well understood.
 - Mostly scientific applications.
 - By a person, who is an expert in that area.
 - User = programmer = maintainer
- User and programmer separation
 - User: specify the problem(tasks)
 - Programmer: interpret and translate into code

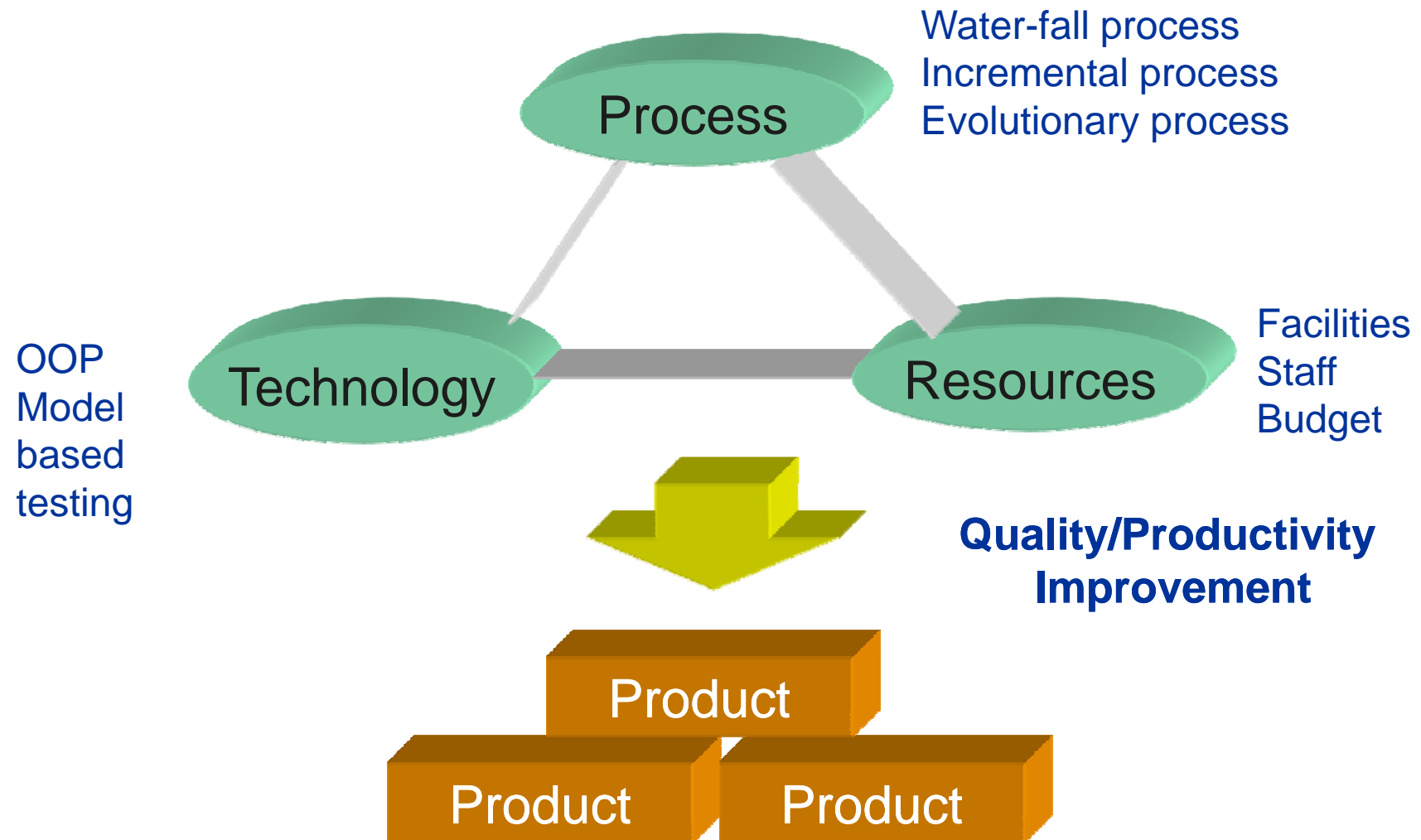
Team Programming Aspect (Cont.)

- Team project started in late 1960's
 - IBM360 Operating system
 - Software crisis observed
 - ``Software Engineering'' coined
- Solutions to software crisis
 - Management techniques
 - Team organization
 - Chief programmer team
 - Democratic team
 - Hierarchical team
 - Better languages and tools
 - Standards
 - ==> Applying engineering approach

Team Programming Aspect (Cont.)

- Requirements in the programming-in-the-small
 - Good programming skill
 - Skilled in data structures and algorithms
 - Fluent in programming languages
- Requirements in the programming-in-the large
 - Needs communication skills and **interpersonal** skills
 - Be familiar with **design** approaches (i.e. system abstraction)
 - Top-down design
 - Divide and conquer paradigm
 - Component based integration
 - Be able to translate vague requirements and desires into **precise spec.**
 - Be able to build and use a **model** of the application
 - Needs ability to **schedule** work

Three Elements of S/W Development



Special Software
Domain: Commercial
Electronics and
Embedded System

What's Different About Embedded Systems

- Embedded systems have different design constraints than general purpose systems
 - **Cost** may matter more than speed
 - **Long life cycle** may dominate design decision
 - Since ubiquitous computing paradigm occurred, this aspect is changing
 - **Reliability/safety** may constraint design choice
- Because applications are often unique, software development may wait for hardware to become available
 - need for simulator/emulators/etc
- Time to market constraints
 - Rapid redesign for changing form factors
 - Rapid redesign for changing control algorithms

Software Characteristics by Domain

- Ordinary IT Software System(e.g. systems developed by SI organizations)
 - Size : Very Large
 - Domain consistency: Low
 - New technology sensitivity: High
 - Hardware dependency: Low
 - Time-to-market pressure: Low

Software Characteristics by Domain

- Commercial Software(e.g. systems developed by software vendors)
 - Size : Large
 - Domain consistency: High
 - New technology sensitivity: High
 - Hardware dependency: Low
 - Time-to-market pressure: Moderate

Software Characteristics by Domain

- Controller Systems/Automation Systems
 - Size : Medium
 - Domain consistency: High
 - New technology sensitivity: Low
 - Hardware dependency: Moderate
 - Time-to-market pressure: Moderate

Software Characteristics by Domain

- Embedded Systems /Commercial Electronics
 - Size : Small
 - Domain consistency: High (-> Moderate)
 - New technology sensitivity: High
 - Hardware dependency: High
 - Time-to-market pressure: High

Software Engineering Applicability

- In general, Controller Systems/Automation Systems (and Embedded Systems /Commercial Electronics) can give much higher rewards for software engineering activity
 - Domain consistency is high and new technology sensitivity is low
 - Ease of accumulating empirical data
 - High reusability in accumulated developments assets(e.g. requirements specification, domain model, test cases, modules)
 - Ease of continuous improvement

General Obstacles

- Hardware dependency is high
 - Software development may wait for hardware to become available
 - Confident testing environment is **not** supported even until complete hardware is ready
 - May need for effective simulator/emulator for testing
- Time-to-market pressure is high
 - High schedule pressure causes difficulties in software engineering activities
 - Overcome the hardware dependency as much as possible
 - Set up process to reduce redundant time consumption
 - Asset reuse

1st Hw: due Feb 25 23:59

- Read the following paper and summarize it with your opinion in one A4 (around 500 words)
 - “No silver bullet: essence and accidents of software engineering”
Computer, 20(4):10-19, April 1987
- Your summary and opinion must flow **logically**
 - Write a key sentence at the head of a paragraph
 - Then, write/enumerate supportive sentences for the key sentence
 - ex. “These days, technologies to develop reliable software become a key power for industrial competition. **First**, software takes more and more portion in embedded systems. **For example**, F22(a military fighter jet) controls more than 80% of its functionalities with software. ... **Second**, production cost for software takes a significant portion in total product cost. ... **Finally**, product’s time to market must be kept in schedule, which is not possible without proper means of quality control of software... “
- Submit a hard copy to HW Box by the due
- Academic dishonesty will be handled with severe penalty