Software Engineering

Spring 2016

Soo Dong Kim, Ph.D.
Hyun Jung La, Ph.D.
Department of Computer Science
Soongsil University
Office 02-820-0909 Laboratory 02-824-0909
{sdkim777, hjla80}@gmail.com http://soft.ssu.ac.kr

Unit 1.
Introduction to Software Engineering

Software Engineering

- **Meaning of Software and Engineering**
  - Software
    - Source codes, Object codes, Development documents, User manuals, etc.
  - Engineering
    - Application of a systematic approach, based on science and mathematics, toward the production of a structure, machine, product, process or system.
- **By Its Goals**
  - producing a High "Quality" software system
  - in a "Cost-effective" manner

The Only Course with 'Engineering' within C.S.

Software Crisis

- **40-Year-Old Software Productivity Problem**
  - Software has failed to keep up with hardware evolution.
  - No significant advances for the last few decades
  - Difficulty of writing correct, understandable, and verifiable software
- **Software Crisis manifested in Several Ways**
  - Projects running over-budget and over-time
  - Software with Inefficiency
  - Software with Low Quality
    - Usability, Performance, Maintainability, etc.
  - Software not Meeting requirements
  - Unmanageable Projects

Causes to Software Crisis

- **Increased Software "Complexity"**
  - User Demands on Richer Functionality of Software
  - Heterogeneous Hardware Environment
  - Distributed Computing
  - Web-based and Mobile Applications
  - Software embedded in Hardware Systems
  - Context-aware Computing
- **Increased Software "Cost"**
  - Development Cost
  - Ownership Cost
    - Operation Cost + Maintenance Cost

Software Complexity

- **Building Large-scaled Software**
  - is *not* a simple scaled-up problem.

- **Analogy**
  - *Foot* bridge over stream vs. *Road* bridge over river

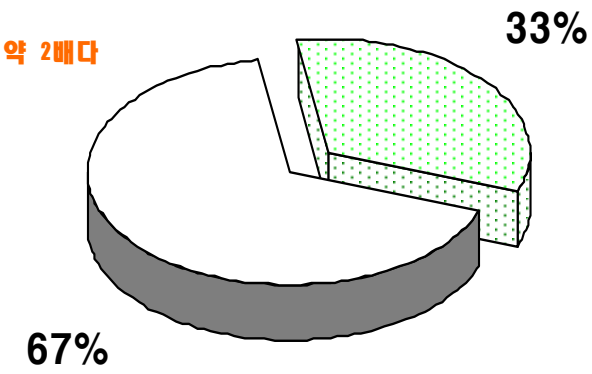
한강을 건널때는 돌다리를 만들거나 나무를 가져와서 가로지른다는 방법이 통하지 않음
〉이와 마찬가지로 소프트웨어개발도 enormous한 문제임

- **Nature of the Problem**
  - *Complexity* of the Software

Software Cost (1)

- **Development Cost vs. Ownership Cost**
  - Software ownership is generally twice as expensive as development.
    - Primarily, the cost of maintenance

유지보수비용이 개발비용의 약 2배다



# of SW Maintenance Staffs → Quite High

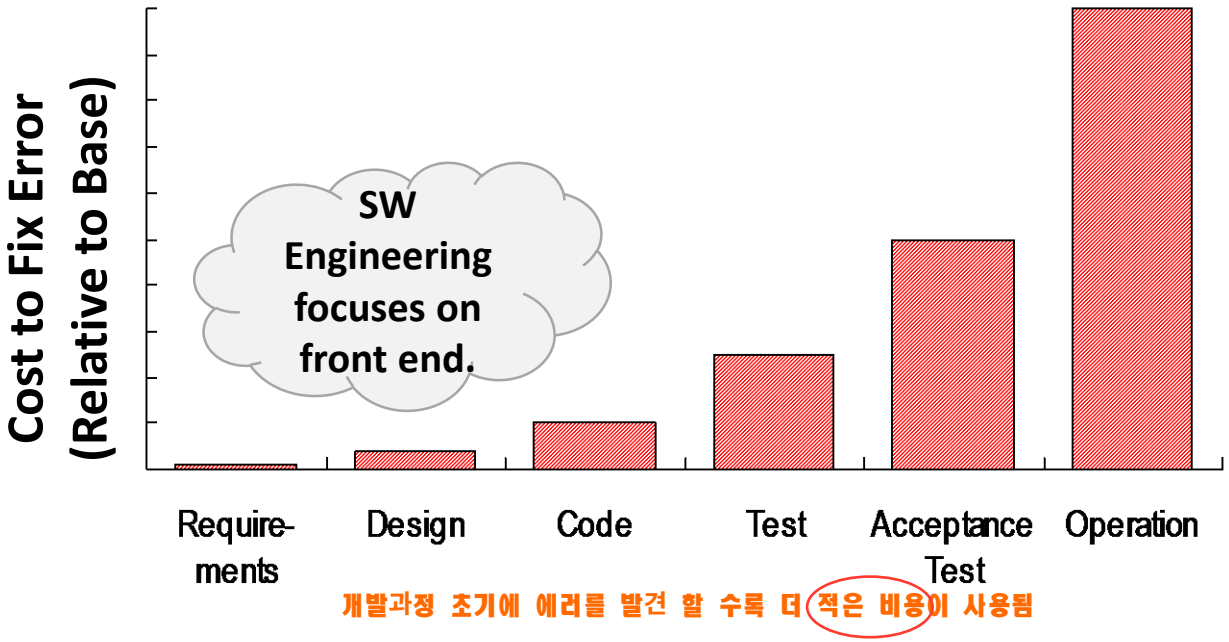
개발단계에서 유지보수단계를 신경쓰지 않지만 신경쓰도록 유지해야한다.
〉약간의 노력으로 유지보수과정에서 엄청난 차이를 나타냄

- **Message**
  - Software product is not the final goal.
  - Maintenance becomes a significant issue.

Maintenance at Cosmetic Surgery → Costly

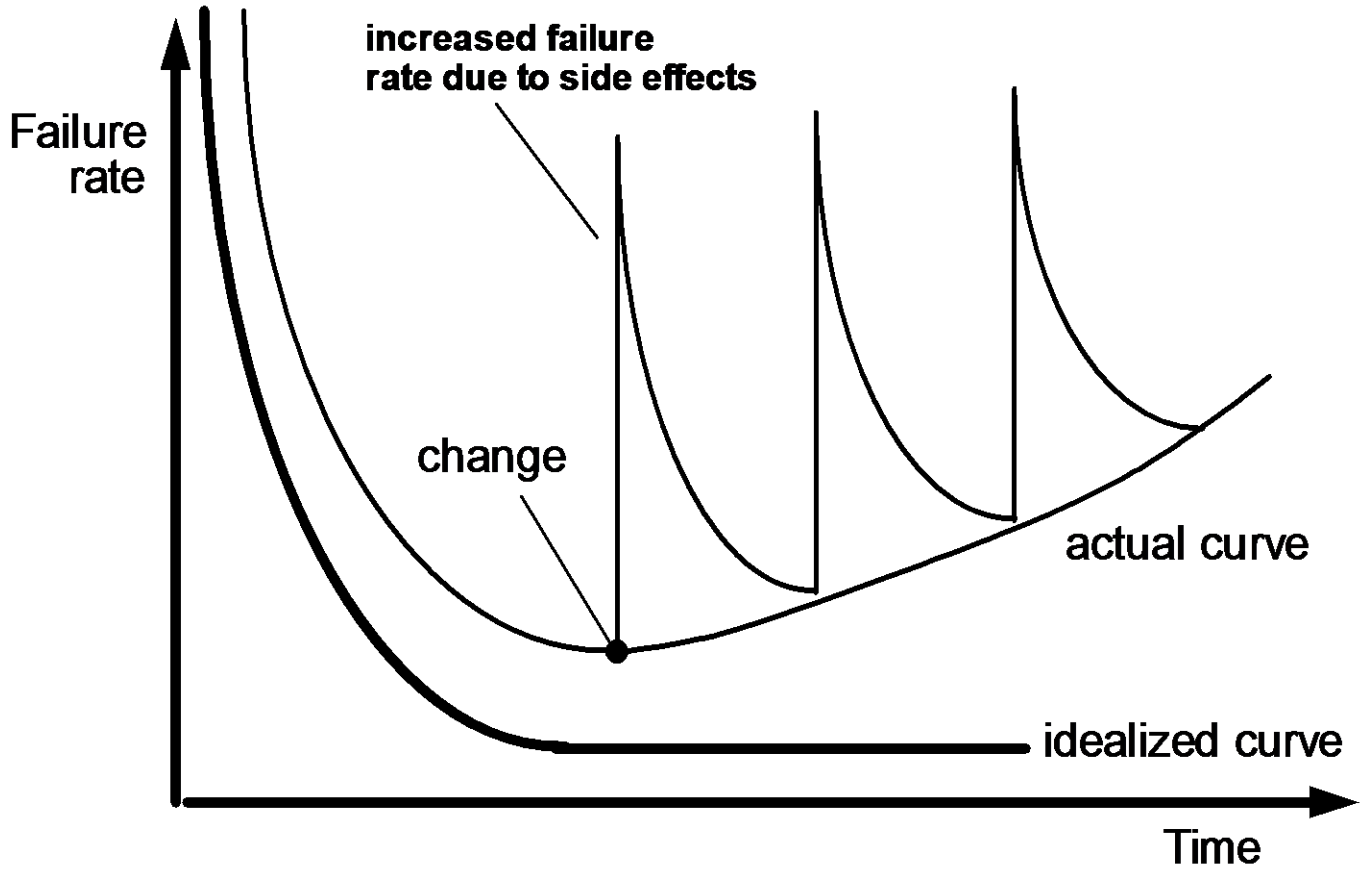
Software Cost (2)

- **Costs to Fix Errors**
  - The sooner an error is discovered, the better.



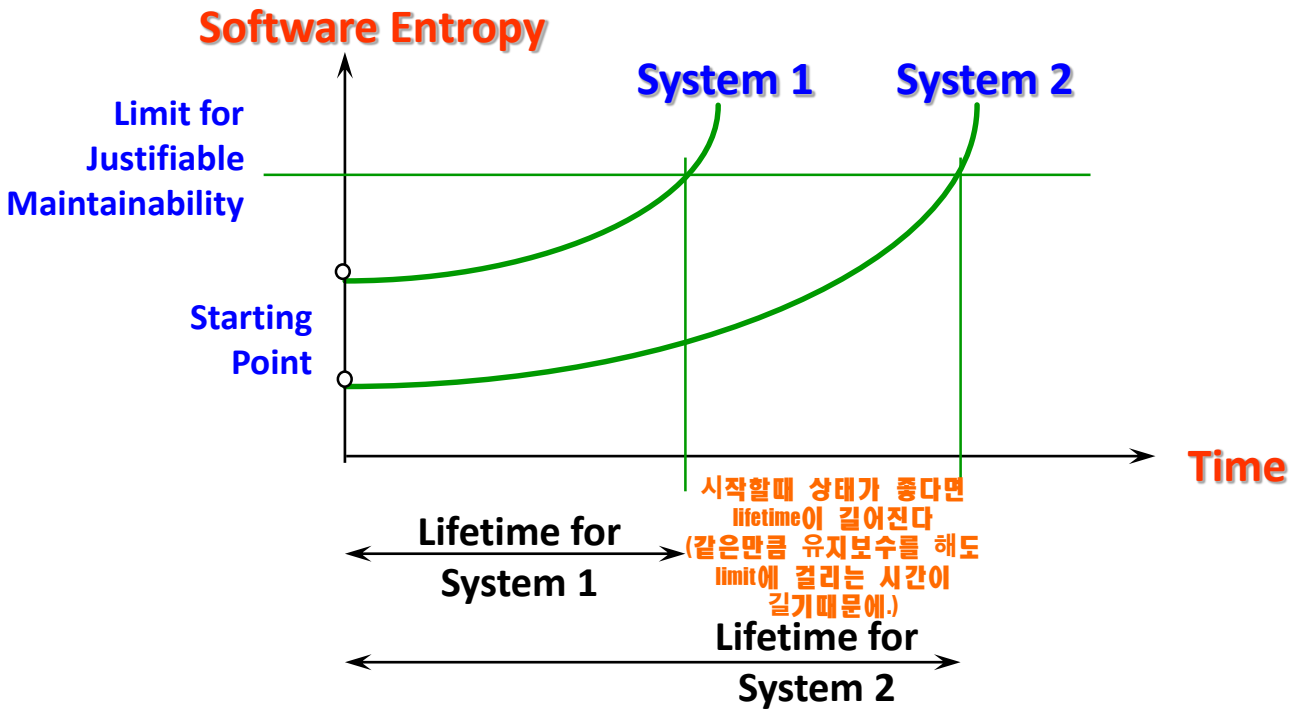
  - More errors are found by outside testers and users than by developers.
  - More errors are found in the two latest stages.

Wear vs. Deterioration



Software Quality & Lifetime

- **Software Entropy**
  - A program that is used will be modified.
  - Increased failure rate and complexity due to side effects of maintenance.



Overcoming Software Crisis

- **Engineering Approach to Software Development**
  - Software Process, Design Methods, Cleanroom Engineering, etc.
- **Reusing Software Assets**
  - Assembling with Components
  - Subscribing Cloud Services
  - Applying Design Patterns
  - Reusing Architectural Styles and Tactics
  - Adopting Frameworks

ComponentSource .com

검증된 라이브러리를 사용(재사용)한다는 것은 코스트를 줄이고 퀄리티를 보장한다.

Benefits of Reuse
- Cost Saving
- Quality Enhancement

- **Focus more on Modeling and Design**
  - Good design is a prerequisite to good implementation.

코드를 사용하기 전에 좋은 디자인을 가지고 시작해야한다