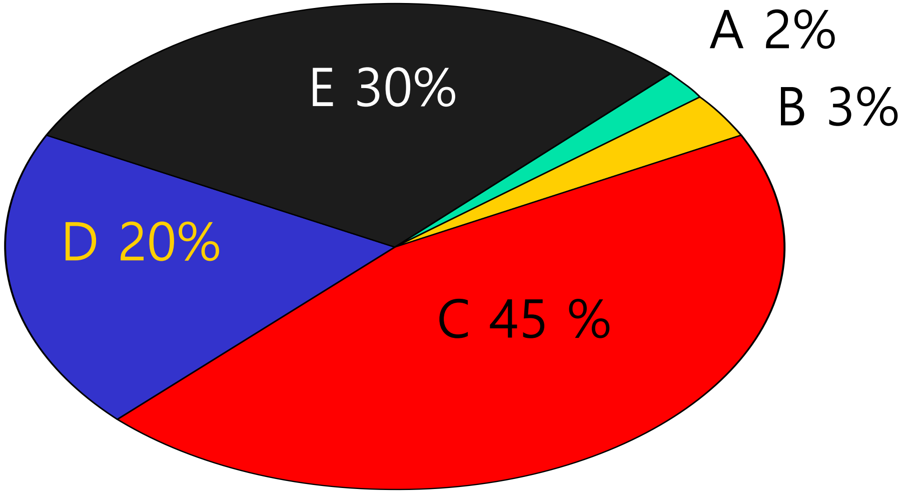
Introduction to Software Engineering

Success or Failure of Software Projects

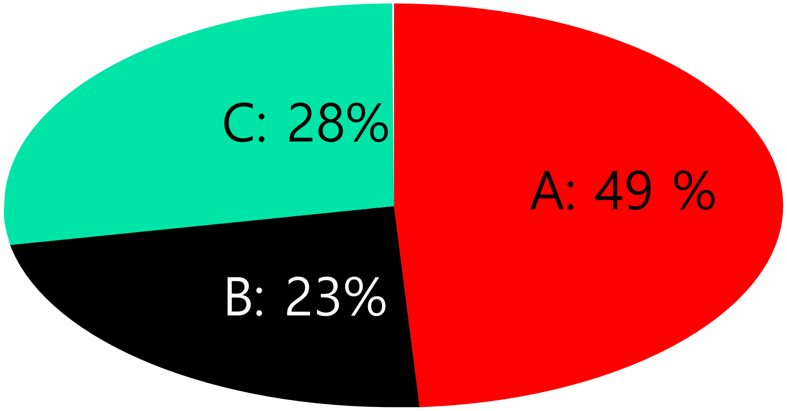
- DoD Announcement on Software Projects(1980s early)



A : worked on delivery
B : worked after some corrections
C : delivered but never successfully used
D : used but either extensively reworked or abandoned
E : paid for, but never delivered

Success or Failure of Software Projects

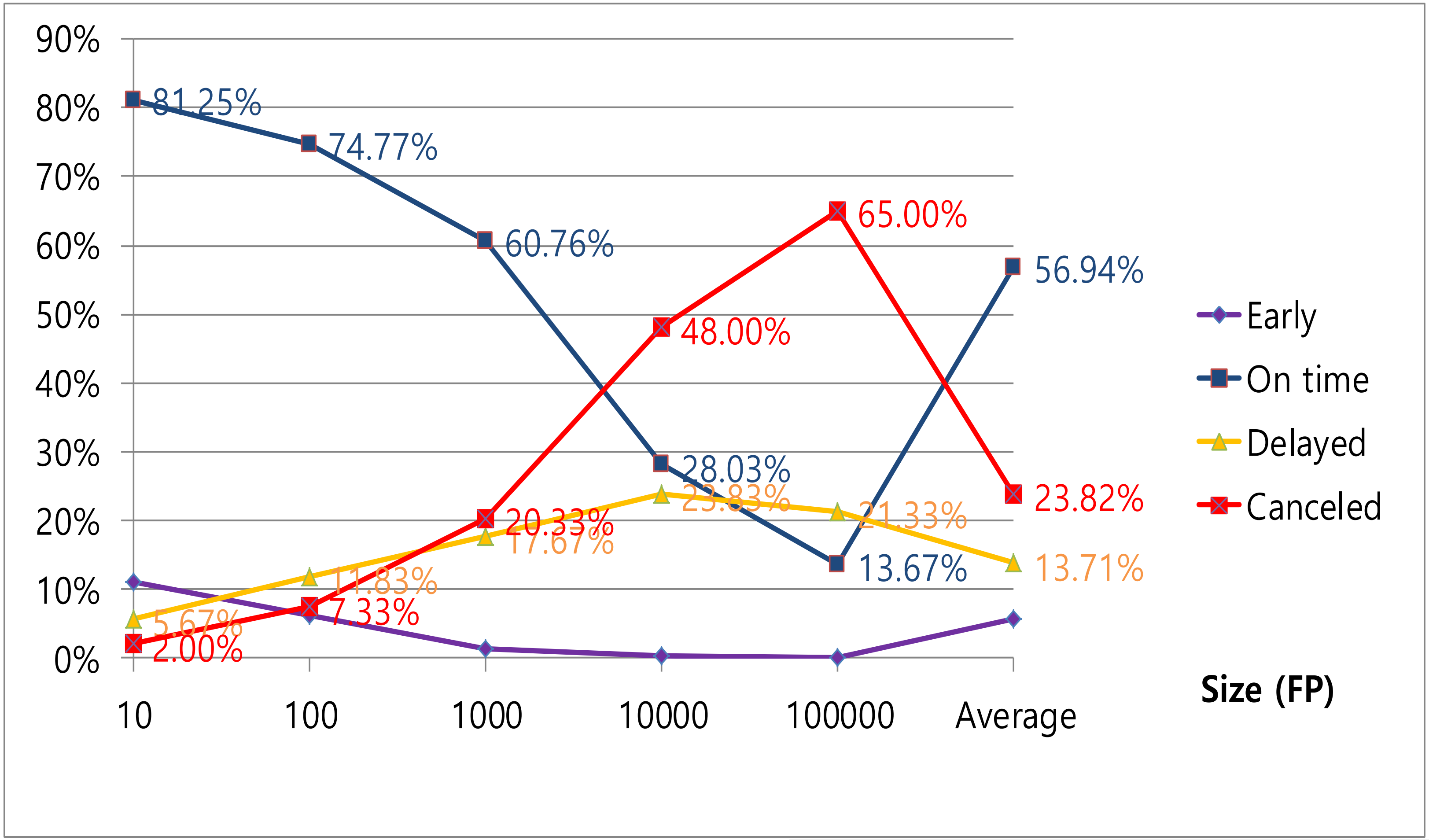
- Data from 1,500 Software Projects (2001)



A: Project late and over budget (49%)
B: Project cancelled before completion (23%)
C: Project completed on time and on budget (28%)

Success or Failure of Software Projects

- Software Quality and Software Economics, Software Tech News April 2010; About 13,000 projects between 1983 and 2009

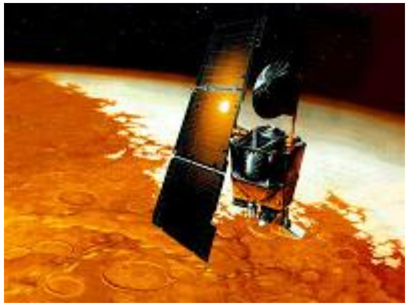


Quality

- Accidents caused by software bugs

Explosion of Ariane 5, 1996 due to "…conversion of a 64 bit integer into a 16 bit signed integer lead to an overflow… "

Loss of Mars Climate Orbiter, 1999 due to "…mix-up between pounds and kilogram…. "



http://discussworldthing.blogspot.kr/2012/01/software-bugs-causing-catastrophic.html

Quality

USS Yorktown dead in the water, 1998 due to "….input and Division by '0'. „ X / 0 = undefined…"

MIM-104 Patriot Missile Failure, 1991 due to "…rounding error"

Software Crisis and Software Engineering

- 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

Goal of Software Engineering

- To overcome software crisis
- That is, developing, operating and maintaining
  - a **quality** software system that satisfies user's need
  - at a low **cost** (within a budget), i.e., cost-effective
  - on **time**

Software Engineering: Definitions

- The disciplined application of engineering, scientific, and mathematical principles and methods to the economical production of quality software. - Watts Humphrey

- The systematic approach to the development, operation, maintenance, and retirement of software. - IEEE Computer Society

- Multi-person construction of multi-version software. - D.L. Parnas

Software Engineering becomes more important

- Increased demands for software
  - More and more systems are <u>software controlled</u>
- The demand for larger and more complex S/W systems ➔
  - Size and Complexity were exponentially increasing
- Software costs often dominate system costs
  - Changes in the ratio of hardware to S/W costs

Automatic Parking System

- Audi's automated driving for parking
- http://www.youtube.com/watch?v=b_m8DqTlOLE

Automatic Driving in Traffic Jams

- Audi's automatic driving in traffic jams
- https://www.youtube.com/watch?v=Qa_ZSRj0WM0

Software Size: Militaries

System	LOC	Language
Mars Reconnaissance Orbiter(2005)	545K	C
F-22 Raptor(2005)	2.5M	Ada (90%)
Seawolf Submarine Combat System AN/BSY-2(1997)	3.6M	Ada
Boeing 777(1995)	4M	Ada
Boeing 787(2011)	7M	Ada (largely)
F-35 Joint Strike Fighter(2015)	19M	C and C++
Typical GM car in 2010	100M	MISRA-C

Size and Cost: Example

- Today, premium cars
  - More than 2000 functions are controlled by software with more than 10M LOC
  - Electronics and software: up to 40 % of the production cost of a car
  - Software cost: 50-70% of the development costs of the software/hardware systems

Challenges in Automotive Software Engineering, ICSE 2006

Factors Influencing Software Engineering

- Characteristics of Software
- Characteristics of S/W Development Project
- Types of Software
- Software Size
- Production and Acquisition Mode
  - COTS(Commercial off-the-shelf) vs. Custom-built

1) Software Characteristics: Why ?

- Schedule
  - Why does it take so long to get programs finished?
- Cost
  - Why are costs so high?
- Quality
  - Why can’t we find all errors before we deliver the software?
- Management
  - Why we have difficulty in measuring progress as software is being developed?

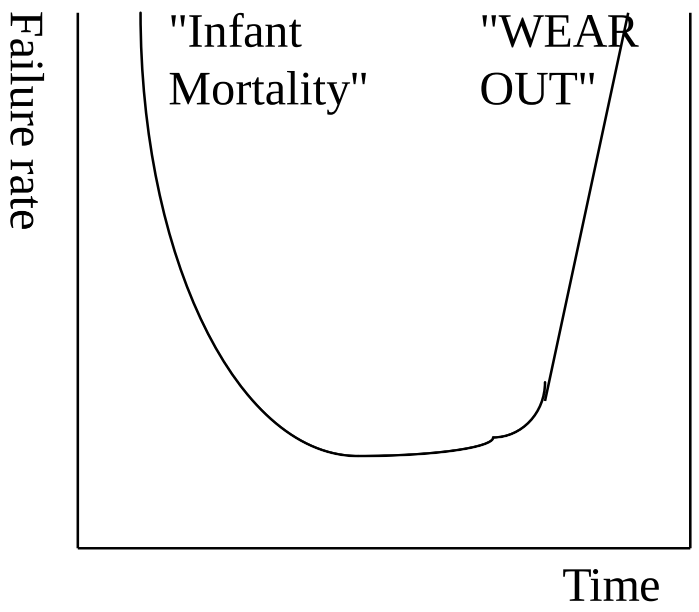
1) Software Characteristics: Because

- Concept:
  - Software is a set of items or objects that form a "configuration" that includes program, documents(deliverables), data, etc.

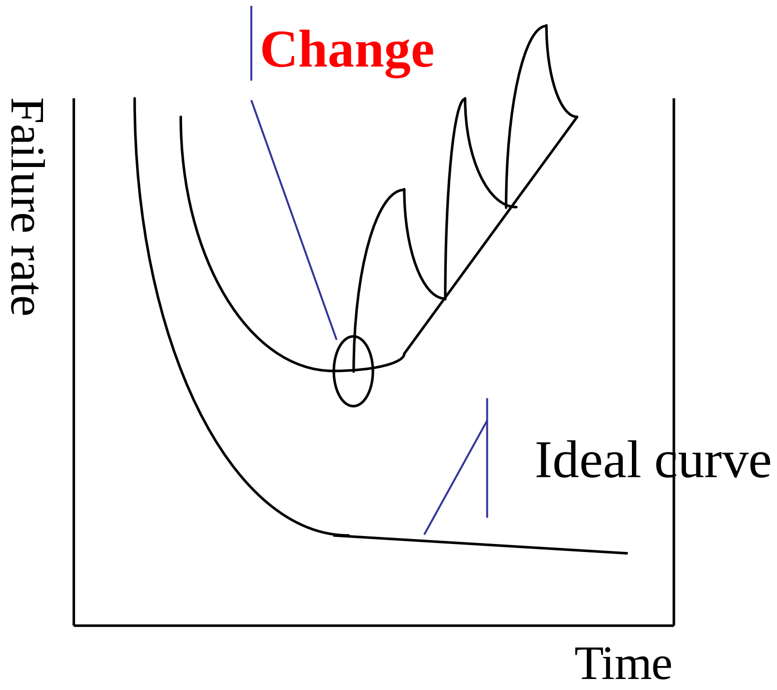
- Software is invisible.
  - Hard to understand, analyze, and evaluate software

1) Software Characteristics (cont.)

- Software doesn’t “wear out”, but change will cause …



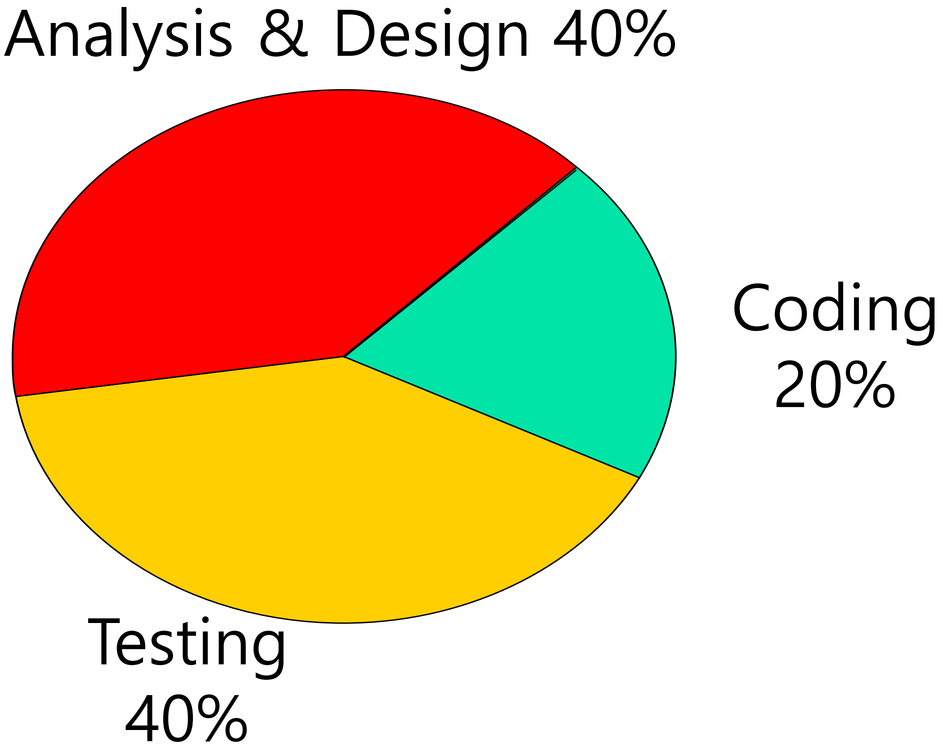
Hardware Failure



Software Failure

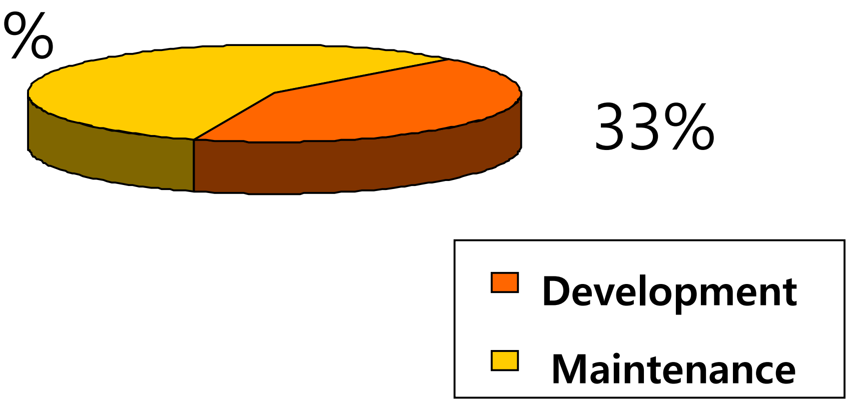
2) S/W Development Project: Custom-Built

- Most S/W is custom built, rather than being assembled from existing components



2) S/W Development Project: Development vs. Maintenance

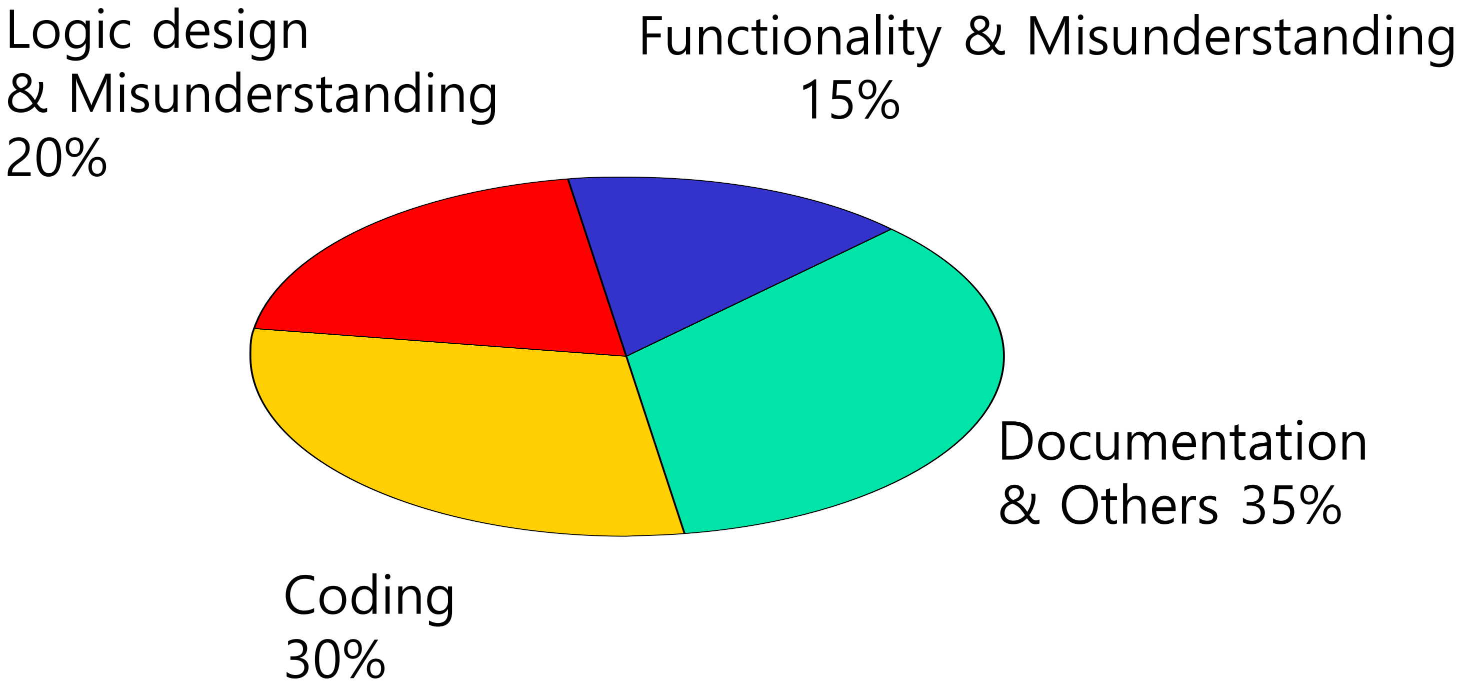
- Software maintenance cost is generally *twice* as expensive as development cost
  - Why ?



Lientz and Swanson '1980

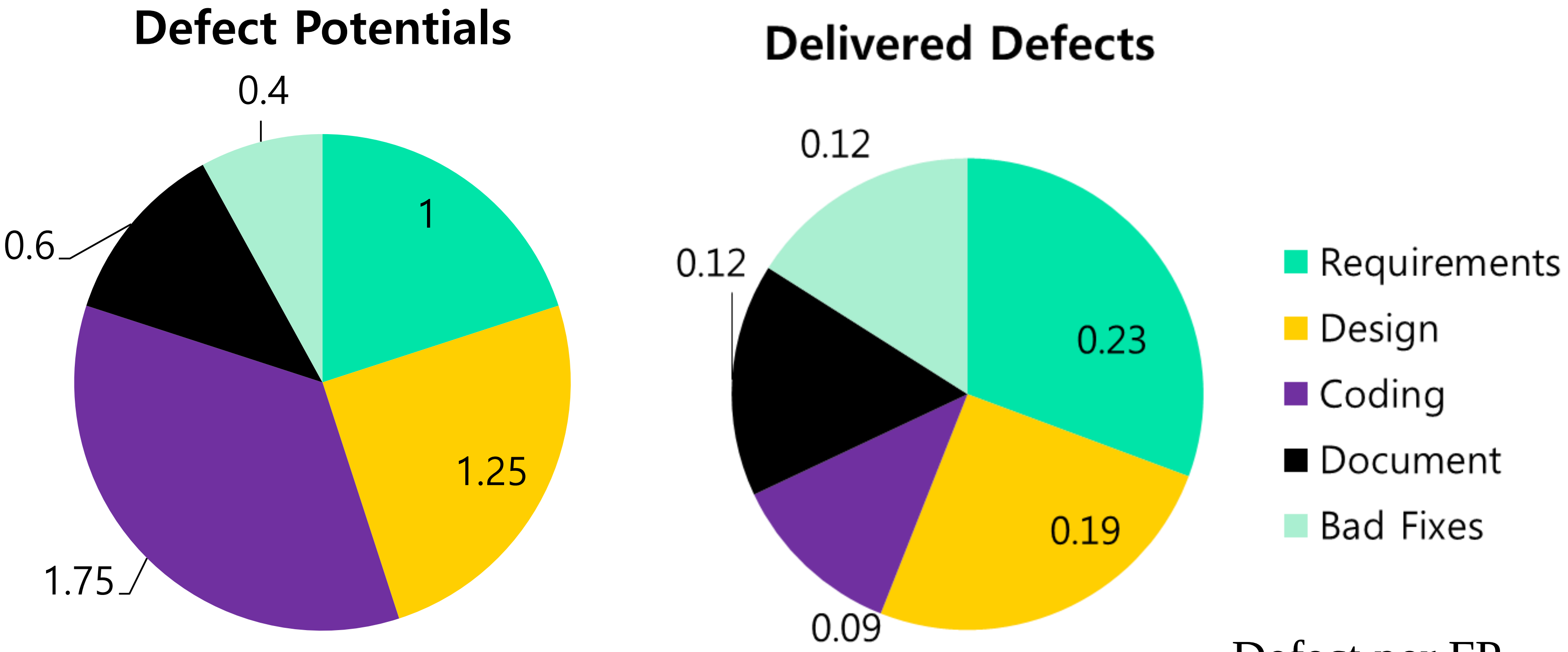
⇒ Software development is not the final goal.
Maintenance becomes a significant issue.

2) S/W Development Project: Sources of Errors



2) S/W Development Project: Sources of Errors

- U.S. average about defects



Defect Potentials

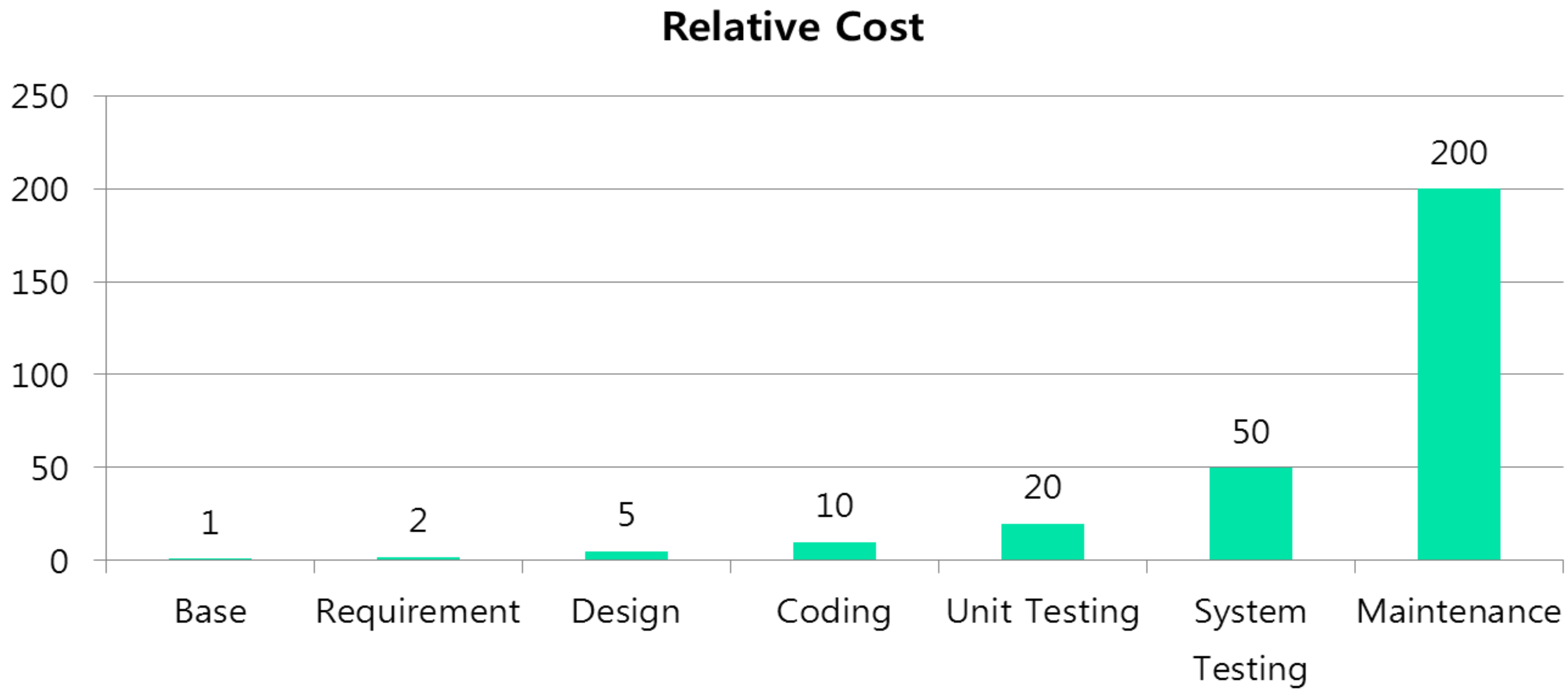
Delivered Defects

Defect per FP

Software Quality and Software Economics, Software Tech News April 2010
the results of about 13,000 software projects between 1983 and 2009

3) S/W Development Project: Costs to Fix Errors

- There is difference between the introduction time and detection time of errors.
  - Errors found later require more correction cost.
  - Thus, the sooner errors are discovered, the better.



Davis, Alan M. Software Requirements: Objects, Functions, and States. Prentice-Hall, 1993

3) Type of Software(1/2)

- System Software
  - A collection of programs to provide services for application programs
  - Operating Systems, Device Drivers, File Managers, etc.
  - c.f.) Application Software
- Engineering and Scientific Software
  - Characterized by Number Crunching
  - CAD, Automobile Street Test, etc.
- Business Software
  - Facilitates business operations or management decision making
  - Often referred to as Information System.
  - MIS(Management Information Systems), etc.

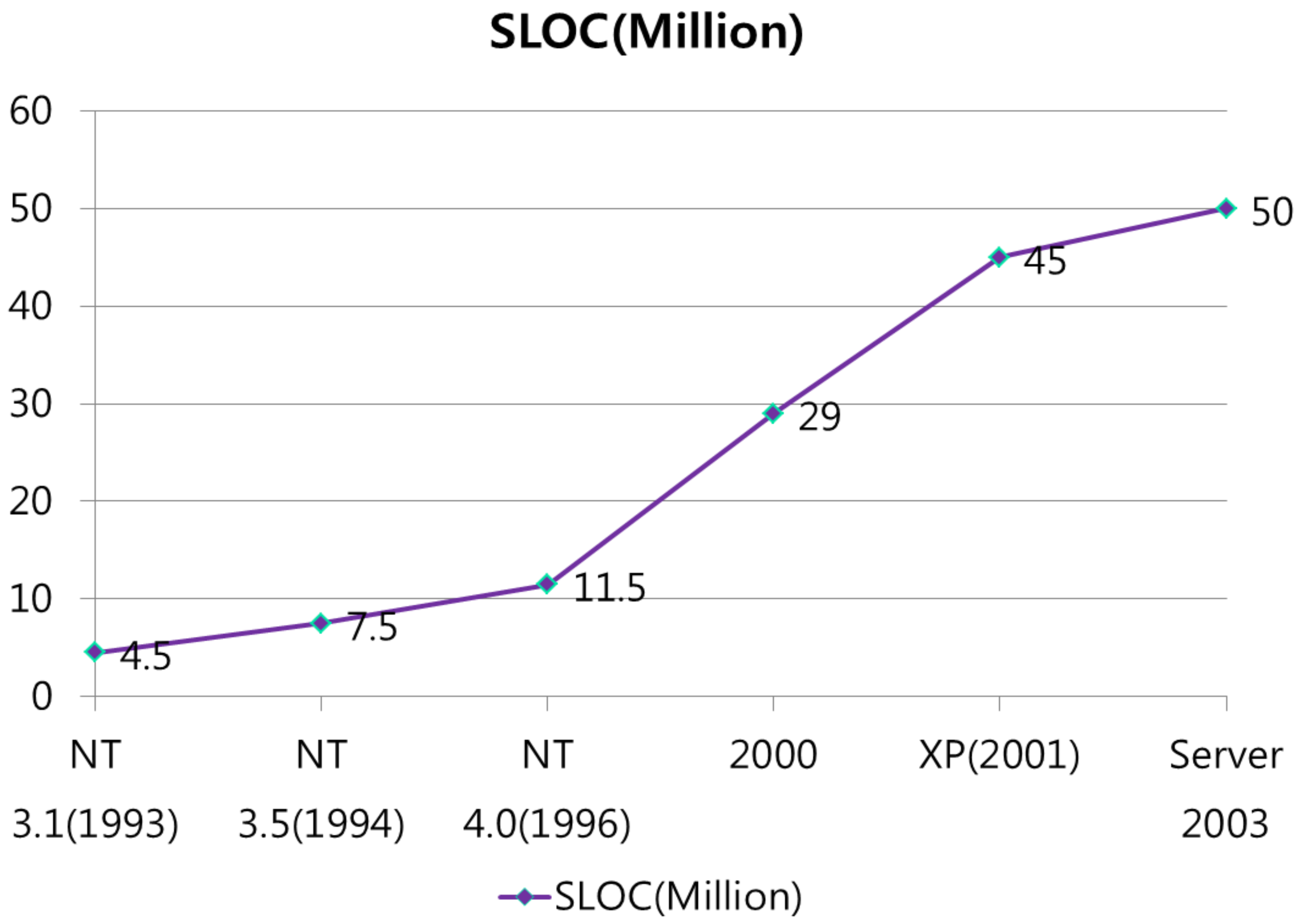
3) Type of Software(2/2)

- Real-Time Software
  - Program that monitors, analyzes and controls real world events as they occur
  - Must respond within **strict time constraints**
  - e.g.) Military Systems
- Embedded Software
  - Resides in read-only memory, and is used to control products and systems for consumer and industrial markets
  - Performs very limited and esoteric functions
- Transformational vs. Reactive Systems

4) Software Size

Category	Number of programmers	Duration	Product size (source lines)
Trivial	1	1-4 wks	500
Small	1	1-6 mos	1K-2K
Medium	2 - 5	1-2 yrs	5K-50K
Large	5 - 20	2-3 yrs	50K-100K
Very large	100 - 1000	4-5 yrs	1M
Extremely large	2000 - 5000	5-10 yrs	1M-10M

4) Software Size



SLOC(Source Lines of Code)

- Software metric
  - used to measure the size of a software program
  - by counting the number of lines in the text of the program's source code
- Two types of SLOC

Type	Definition	Characteristics
Physical	The number of lines including comment lines	sensitive to formatting and style conventions
Logical	The number of executable "statements"	less sensitive to formatting and style conventions

SLOC(Source Lines of Code)

- Example #1

```
for (i = 0; i < 100; i += 1) printf("hello"); /* How many lines of code is this? */
```

  - 1 Physical LOC
  - 2 Logical LOC

- Example #2

```
/* Now how many lines of code is this? */
for (i = 0; i < 100; i += 1)
{
        printf("hello");
}
```

  - 5 Physical LOC
  - 2 Logical LOC

Aspects of Software Engineering

- Software engineering covers various issues:
  - Requirement, analysis, design and coding techniques
  - Verification and Validation(V&V) techniques including testing
  - Quality assurance
  - Project management
  - Configuration management
  - CASE(Computer-Aided Software Engineering) tools: IDE, Modeling tools, Testing tools, Source code version management tools(Git, SVN, …)

- Other knowledge
  - human factors (i.e., GUI) ➔ Usability
  - Hardware characteristics ➔ embedded SW
  - technical communication and documentation

Software Myths …

- Practitioners
  - Once the program is written and run, our job is done.
  - Until a program is running, there is no way of assessing its quality.
  - The only deliverable for a software project is a working program.

- Clients
  - A general statement of objectives is sufficient to begin writing programs - - we can fill in the details later.
  - Requirements changes are easy to accommodate because software is flexible.

- Managers
  - We have standards and procedures for building software, so developer have everything they need to know.
  - We have state-of-the-art software development tools.
  - If we are behind schedule, we can add more programmers to catch up.
  - A good manager can manage any project.

Q&A