Introduction

Tim RPL
Program Studi Teknik Informatika

What is Software ?

- Software adalah sekumpulan item-item atau objek yang membentuk konfigurasai yang melibatkan program, dokumen, data dan lain-lain.

What is Software ?

- Definisi Software menurut IEEE :

  Computer programs, procedures, and possibly associated, documentation and data partaining to the operation of a computer system (**IEEE Standard Glossary of Software Engineering Therminology, 1990** *)*

What is Software ?

- Software dirancang dan dibangun oleh software engineer
- Software digunakan oleh siapapun dalam masyarakat
- Software engineer mempunyai kewajiban moral untuk membangun software yang dapat diandalkan yang tidak merugikan orang lain.
- Pengguna perangkat lunak hanya fokus pada produk software apa yang mereka butuhkan dan membuat tugas mereka lengkap

What is Software ?

- Software adalah dua hal, produk dan kendaraan untuk menyampaikan sebuah produk (informasi)

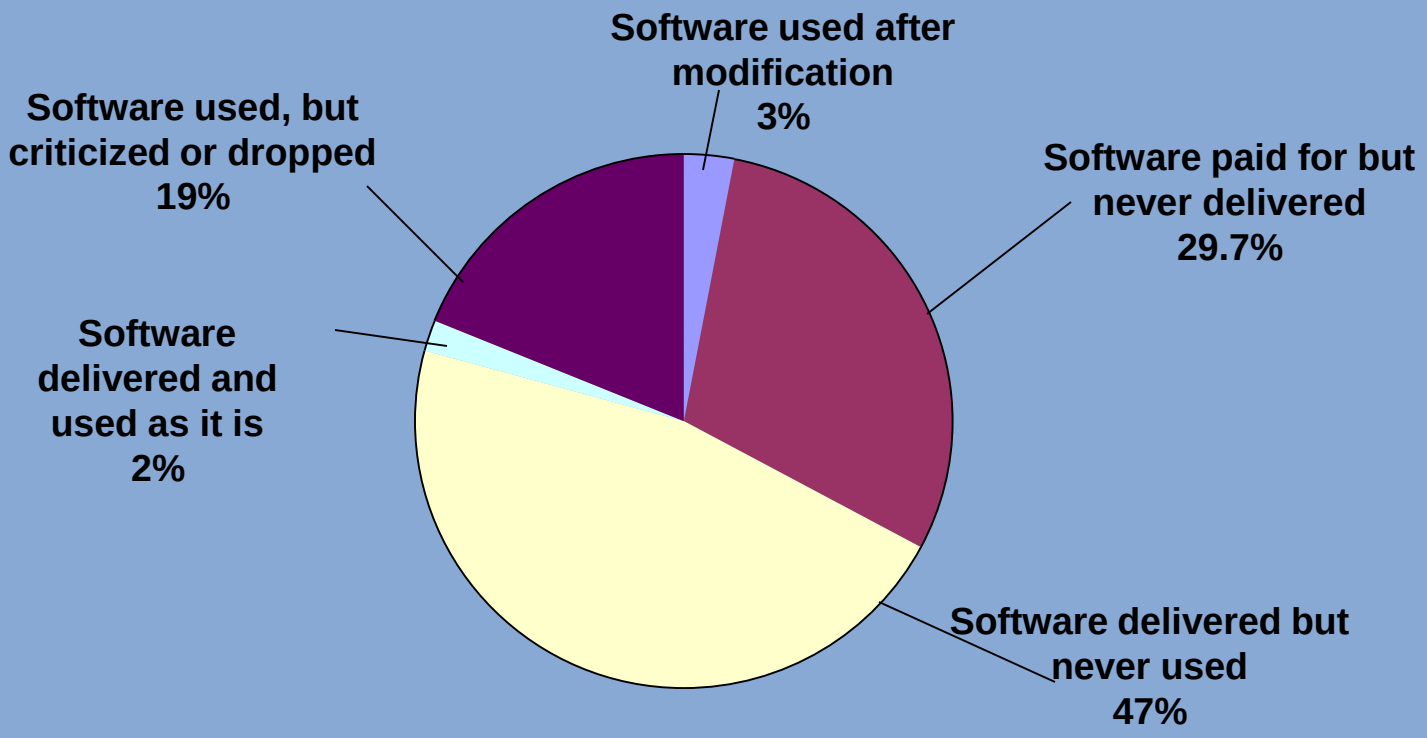
Software Evolution

Unified theory untuk evolusi software (Lehman) :

- The Law of Contuining Change (1974)
- The Law of Increasing Complexity (1974)
- The Law of Self-Regulation (1974)
- The Law of Conservation of Organizational Stability (1980)
- The Law of Conservation of familiarity (1980)

**SEPA 6th ed.Roger S.Pressman*

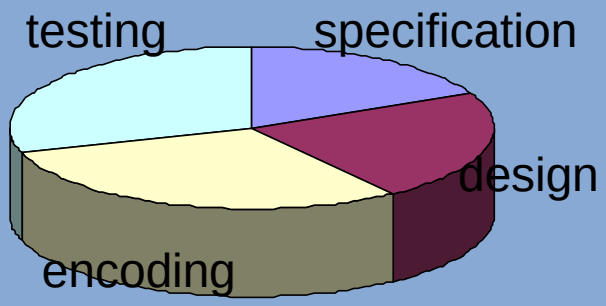
Software Problems (1)



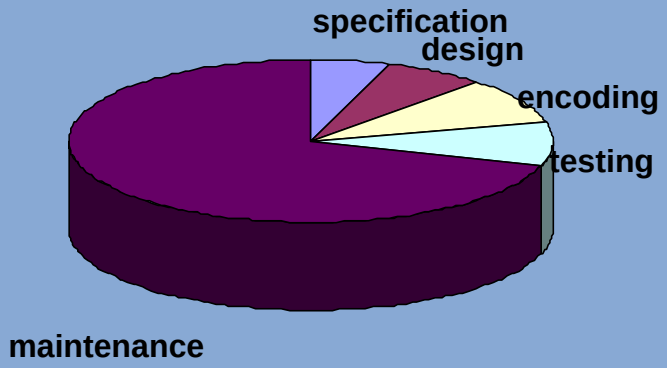
1982: Nine DOD contracts amounting to $6.8 million
(source: GAO, quoted in CMU/SEI-**93**-EM-8)

Software Problems (2)

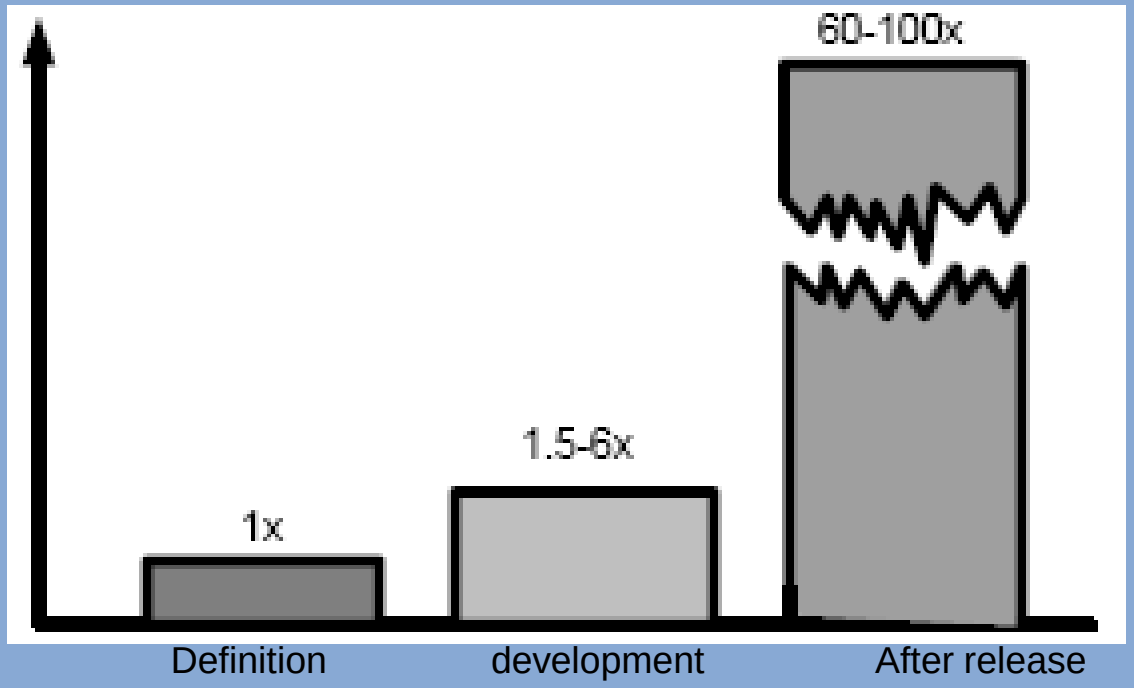
Distribution of effort : what is believed



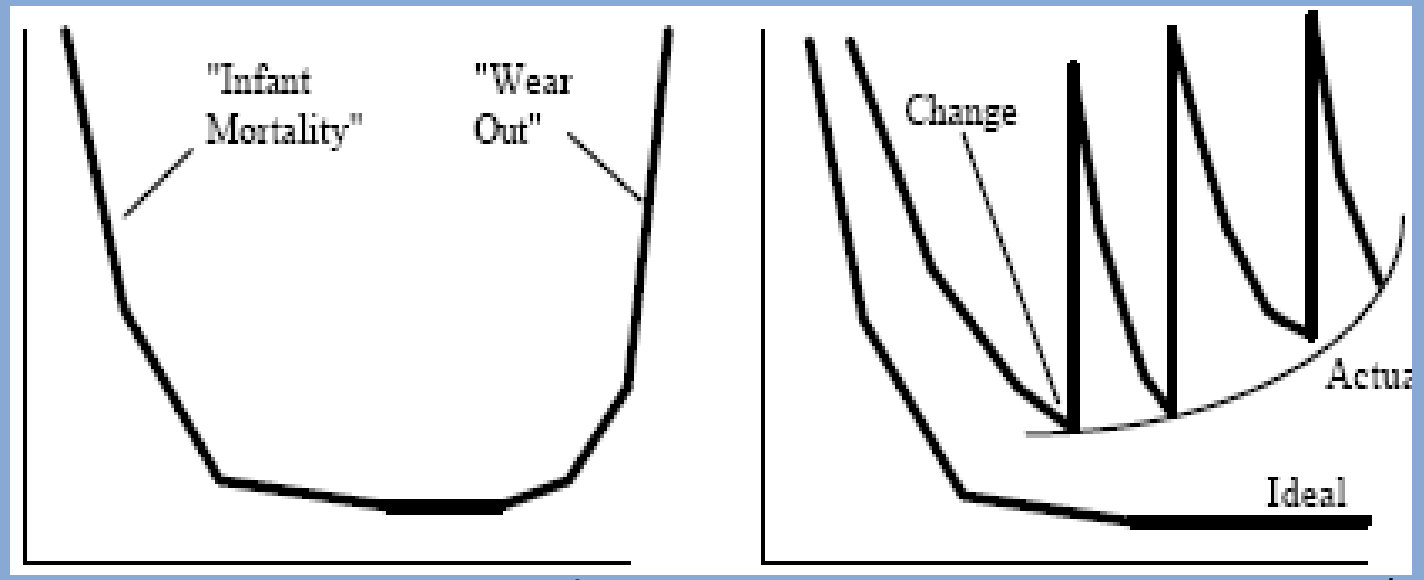
Distribution of effort : what happens



The Cost of Change



Kurva Kesalahan



Failure curve
For hardware

Failure curve
For software

**Software Engineering. Module 3. Richard Conn. University of Cincinnati, May 1993*

Software Application Type

- System Software
- Application Software
- Embedded Software
- Engineering/ Scientific Software
- Product Software
- Web Application Software
- Artificial Intelligence Software

**SEPA 6th ed.Roger S.Pressman*

New Software Challenges

- Ubiquitous computing (ada dimana-mana)
  - Creating software to allow machines of all sizes to communicate with each other across vast networks
- Netsourcing
  - Sederhana arsitekturnya dan aplikasinya canggih yang diperuntukan pasar end user di dunia.
- Open Source
  - Distributing source code for computing applications so customers can make local modifications easily and reliably
- New economy
  - Pembangunan aplikasi yang memfasilitasi pendistribusian komunikasi masa dan produk masa dengan mengunakan konsep perubahan.

SEPA 8th ed.Roger S.Pressman

Legacy Software

- The software must be adapted to meet the needs of new computing environments or technology.
- The software must be enhanced to implement new business requirements. (*software harus ditingkatkan*)
- The software must be extended to make it interoperable with more modern systems or databases. (*Software harus dpt diperluas*)
- The software must be re-architectures to make it variable within a network environment.

***SEPA 8th ed.Roger S.Pressman**


©2000jkin
Miccosoft

The essence of Software Engineering

- Understand the problems (communication and analysis)
- Plan a solution (modeling n software design)
- Carry out the plan (code generation)
- Examine the result for accuracy (testing n quality assurance)

1. Understand the problems

"I understand, let's get on with solving this thing"

- Unfortunately,understanding isn't always that easy.
  - Who has a stake in the solution to the problem ? (who are the stakeholder ?)
  - What are the unknowns ? What data, functions, and features are required to properly solve the problem ?
  - Can the problem be compartmentalized ?

2. Plan the Solution

- Have you seen similar problems before ?
- Has a similar problem been solved ?
- Can subproblems be defined ?
- Can you represent a solution in a manner that lead to effective implementation ? Can a design model be created ?

3. Carry out the plan

- Does the solution conform to the plan ? Is source code traceable of the design mode ?
- Is each component part of the solution provably correct ? Has design and code been reviewed, or better, have correctness proofs been applied to the algorithem ?

4. Examine the Result

- Is it posible to test each component ?
- Does the soltion produce result that conform to the data, functions and features that are required ?

7 Principles SE (David Hooker)

1. *The Reason It All Exists*
2. *KISS, (Keep It Simple, Stupid !)*
3. *Maintain the Vision*
4. *What You Produce, Others will Consume*
5. *Be Open to The Future*
6. *Plan Ahead for Reuse*
7. *Think !*

Software Myths

- Masih dipercaya oleh banyak manager dan praktisi
- Berbahaya karena mereka dipercaya.
- Setiap praktisi dan manajer seharusnya memahami realitas dari bisnis proses

SEPA 6th ed.Roger S.Pressman

Software Myths: Customer Myths

Myths :

- A general statement of objective is sufficient to begin writing programs, fill in the details later
- Project requirements continually change, but change can be easily accomodated because software is flexible

Reality :

- Poor up-front definition of the requirements is THE major cause of poor and late software.
- Cost of the change to software in order to fix an error increases dramatically in later phases of the life of the software

Software Myths : Practitioner's myths

Myths :

- Once a program is written and works, the practitioner's job is done
- Until a program is running, there is no way to assess its quality
- The only deliverable work product for

Reality :

- 60%-80% of effort expended on a program occurs after it is delivered to the customer.
- Software is reviews can be more effective in finding errors than testing for certain classes of errors

Software Mhyts
Management myhts

Myths :

- Books of standards exist inhouse so software will be developed satisfactorily.
- Computers and software tools that are available inhouse are sufficient.
- We can always add more programmers if the project gets behind.

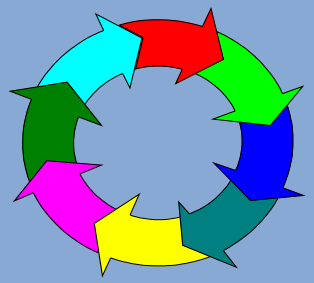
Reality :

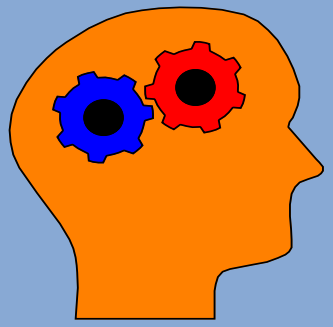
- Books may exist, but they are usually not up to date and not used.
- CASE(**) tools are needed but are not usually obtained or used.
- "Adding people to a late software project makes it later." -- *Brooks*

“ You said I should spend more time with our children, so I turned their faces into icons “

Apakah Software Engineering ?

- *Software Engineering* adalah teknologi yang harus digunakan oleh setiap orang yang akan membangun software, dengan melalui serangkaian proses, menggunakan sekumpulan metode dan alat bantu (*tools*) (Pressman, 1997)

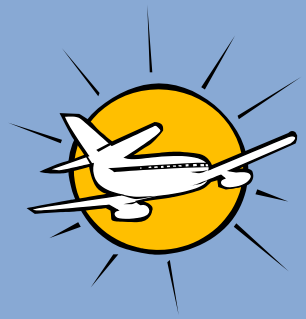




Why Software Engineering ?

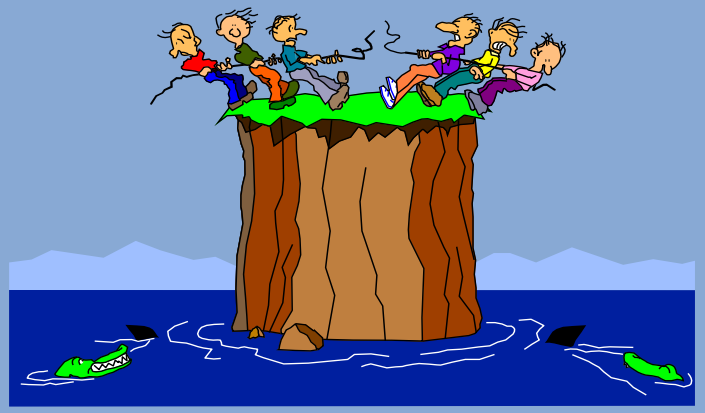
- Untuk mendapatkan software yang benar dan untuk membuat software menjadi benar

- Software adalah sesuatu yang kompleks dalam hal:
  - *Domain problem: Business Rule*
  - *Data size: Digital and Non Digital*
  - *Solution: Algorithm*
  - *Place or Sites*



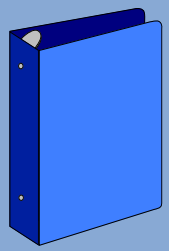
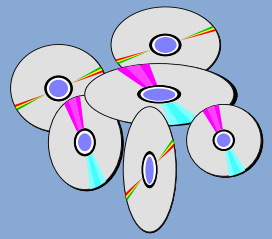
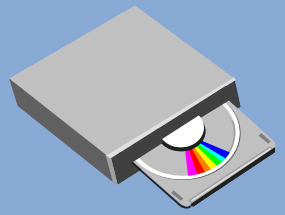
Why Software Engineering ?

- Software harus benar (***correct)***:
  - Berdasarkan *business rule*
  - Sejalan dengan segala sesuatu dan semua pihak yang terkait
- Pembangunan software harus dikelola dengan baik untuk memelihara kebenarannya (*correctness*)



Bagaimana seharusnya SE dijalankan ?

- There are 2 things to be considered in SE:
  - Product = Software:
    - Programs
    - Documents
    - Data

  - Process of how the software is build:
    - Management process
    - Technical process


Perbedaan Software Engineering dan Computer Science

- Computer science fokus pada teori dan dasar-dasar; software engineering fokus pada praktek dan pembangunan dan pengiriman penggunaan software.
- Teori Computer science masih belum cukup untuk menetapkan sebagai sebuah tiang fondasi untuk software engineering.

**** Software Engineering 7th ed, Ian Sommerville***

Perbedaan Software Engineering dan System Engineering

- System engineering fokus pada semua aspek pembangunan sistem dasar komputer meliputi **hardware**, **software** and **process engineering**.
- Software engineering adalah bagian dari proses ini yang berfokus pada pembangunan prasarana perangkat lunak, kontrol, aplikasi dan database pada sistem.
- System engineers terlibat dalam spesifikasi sistem, perancangan arsitektur, integrasi dan penyebaran.

- **** Software Engineering 7th ed, Ian Sommerville***

Important Questions of Software Engineer

- Why does it *take so long* to get software finished ?
- Why are development *cost so high* ?
- Why can't we *find all errors* before we give the software to our customers ?
- Why do we *spend so much time* and effort maintaining existing programs ?
- Why do we continue to have difficulty in *measuring progress* as software is being developed and maintained ?

SEPA 6th ed.Roger S.Pressman