Rekayasa Perangkat Lunak

S2 Teknik Informatika dan Komputer
PENS

Content

Week 1-7

Part 1 – Overview of Software Engineering

1. Introduction to Software Engineering
2. Software Process
3. System Modelling: DFD & UML

Part 2 – Agile DSDM

1. Introduction to Agile DSDM
2. Agile Requirements and User Stories
3. Modelling and Timeboxing

Week 8 UTS

Content

Week 9-15

Part 3 – Agile Scrum

1. Introduction to Agile Scrum
2. Scrum Roles, Events, Artifacts

Part 4 – End Product (sesuai topik thesis)

1. Prototyping
2. Software Testing and Evolution

Week 16 UAS

Introduction to Software Engineering

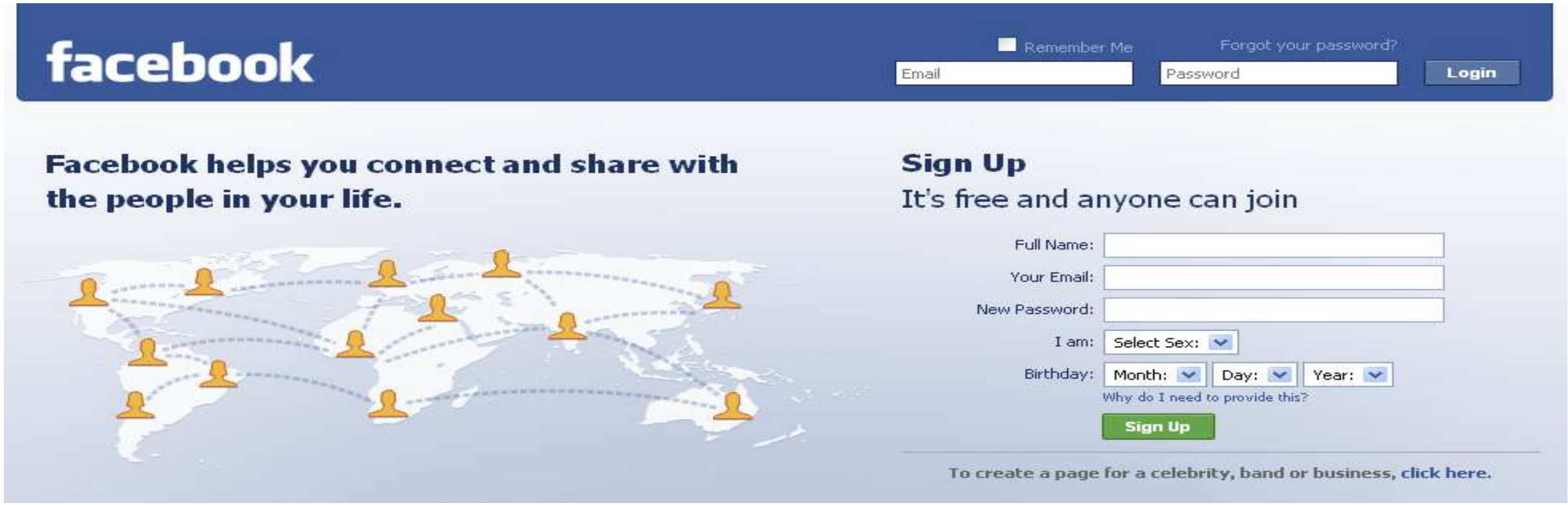
Importance of software

Software can have a huge impact in any aspect of society.

Where can you find software?

Some popular ones…

Some popular ones…



Some popular ones…

And even in…


Sony Ericsson
WALKMAN
UNOFFICIAL CLUB SONY ERICSSON


AT&T
9:42 AM
SMS
Text
Calendar
Photos
Camera
YouTube
Stocks
Maps
Weather
Clock
Calculator
Notes
Settings
iTunes
App Store
Phone
Mail
Safari
iPod

Conclusion

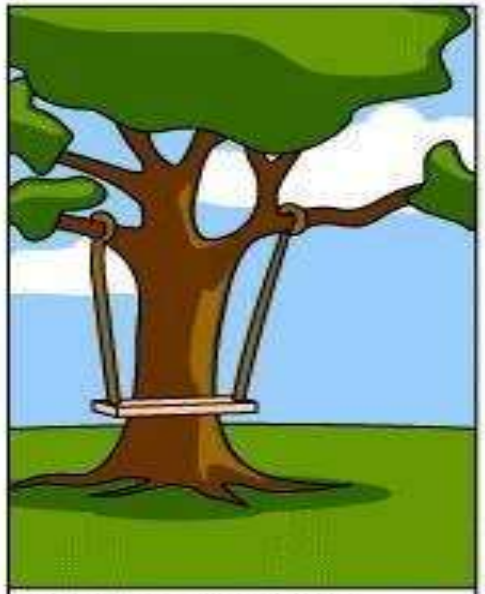
Software is Almost Everywhere.

Problems in software development

Common issues

- ✓The final Software doesn´t fulfill the needs of the customer.
- ✓Hard to extend and improve: if you want to add a functionality later is mission impossible.
- ✓Bad documentation.
- ✓Bad quality: frequent errors, hard to use, ...
- ✓More time and costs than expected

How the customer explained it

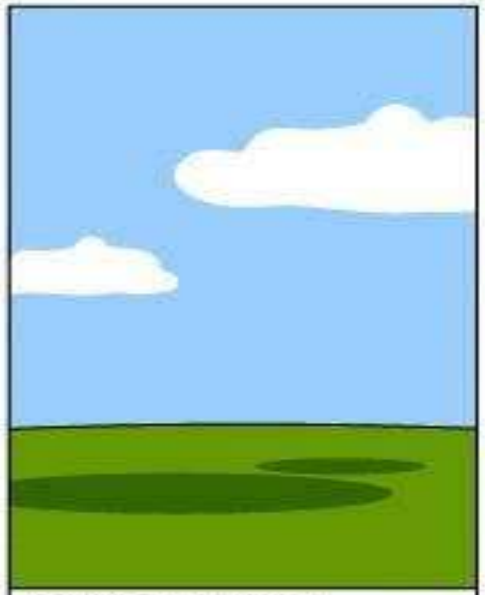


How the Project Leader understood it

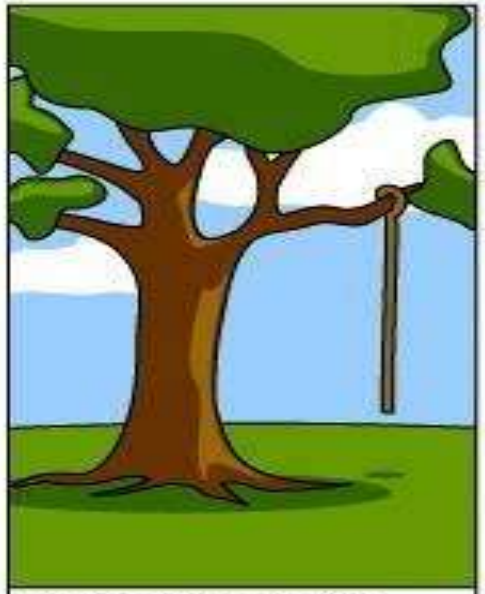
How the Analyst designed it

How the Programmer wrote it

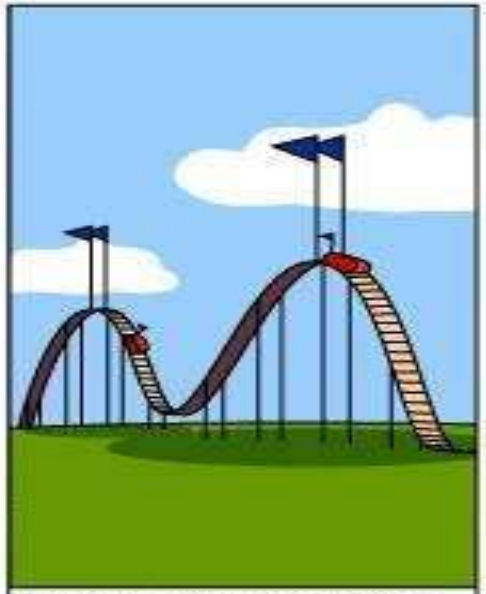
How the Business Consultant described it



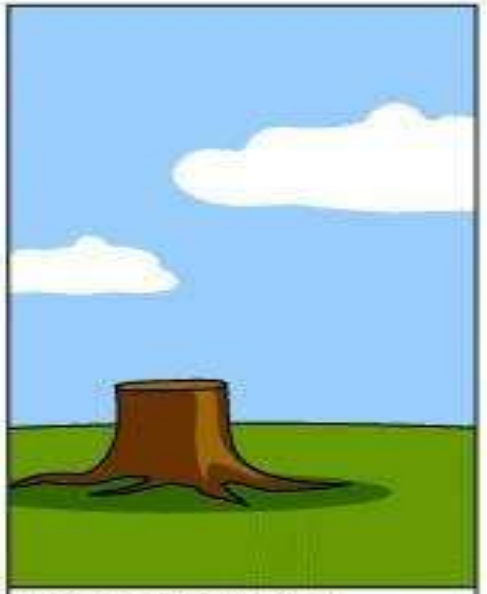
How the project was documented



What operations installed



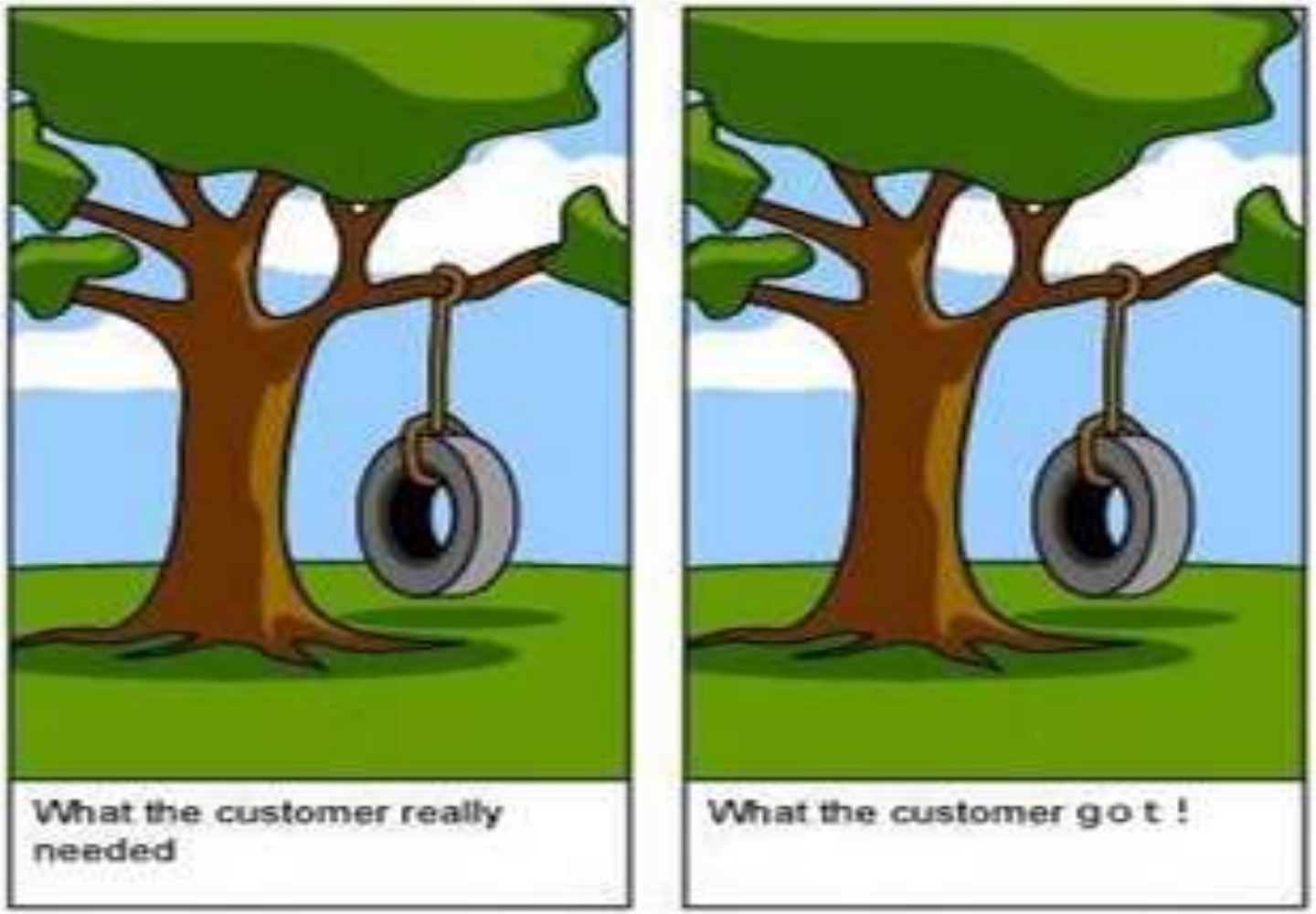
How the customer was billed



How it was supported

What the customer really needed

Is it Possible?



Ariane 5 Flight 501

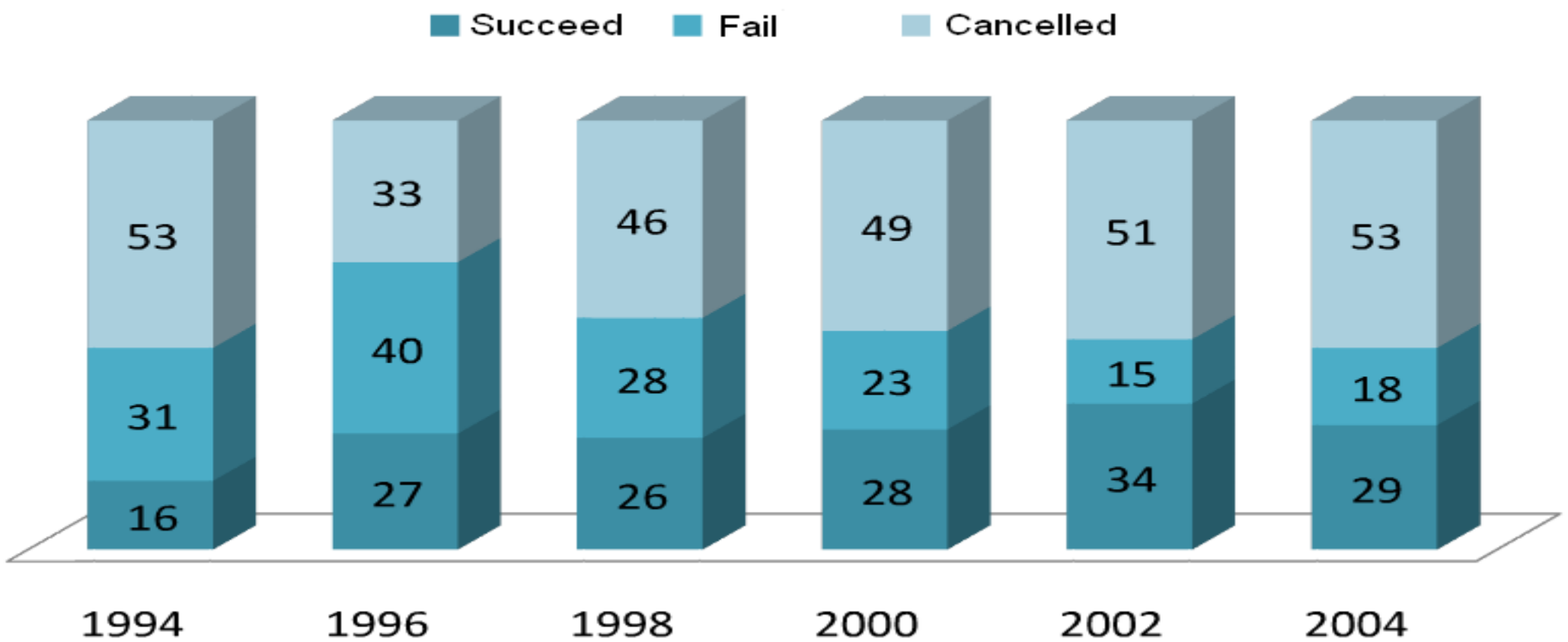
https://www.youtube.com/watch?v=gp_D8r-2hwk

Cause: design errors in the software

CAUSE OF THE FAILURE

- The failure of the Ariane 501 was caused by the complete loss of guidance and attitude information 37 seconds after start of the main engine ignition sequence (30 seconds after lift- off). This loss of information was due to **specification and design errors in the software** of the inertial reference system.
- The extensive reviews and tests carried out during the Ariane 5 Development Programme **did not include adequate analysis and testing** of the inertial reference system or of the complete flight control system, which could have detected the potential failure.

Chaos Report



The Uniqueness of the Software Development

Frame 1.1 **The uniqueness of the software development process**

- **High complexity**, as compared to other industrial products
- **Invisibility of the product**
- **Opportunities to detect defects ("bugs")** are limited to the product development phase

Characteristic	Software products	Other industrial products
Complexity	Usually, very complex product allowing for very large number of operational options	Degree of complexity much lower, allowing at most a few thousand operational options
Visibility of product	Invisible product, impossible to detect defects or omissions by sight (e.g. of a diskette or CD storing the software)	Visible product, allowing effective detection of defects by sight
Nature of development and production process	Opportunities to detect defects arise in only one phase, namely product development	Opportunities to detect defects arise in all phases of development and production: - Product development - Product production planning - Manufacturing

LIMITED WARRANTY

Dagal Features provides no warranty, either expressed or implied, with respect to **AMGAL**'s performance, reliability or fitness for any specified purpose. **Dagal Features** does not warrant that the software or its documentation will fulfil your requirements. although **Dagal Features** has performed thorough tests of the software and reviewed the documentation, **Dagal Features** does not provide any warranty that the software and its documentation are free of errors. **Dagal Features** will in no case be liable for any damages, incidental, direct, indirect or consequential, incurred as a result of impaired data, recovery costs, profit loss and third party claims. the software is licensed "as is". the purchaser assumes the complete risk stemming from application of the **AMGAL** program, its quality and performance.

If physical defects are discovered in the documentation or the CD on which **AMGAL** is distributed, **Dagal Features** will replace, at no charge, the documentation or the CD within 180 days of purchase, provided proof of purchase is presented.

Conclusion

Programming is NOT enough!

It is not enough to do your best: you must
Know what to do, and THEN do your best.
-- *W. Edwards Deming*

And Since...

A clever person solves a problem.
A wise person avoids it.
- *Albert Einstein*

Solution

1. What is Software

The Definition of Software

Frame 2.1 **Software – IEEE definition**

Software is:
Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system.

The IEEE definition of software, which is almost identical to the ISO definition (ISO, 1997, Sec. 3.11 and ISO/IEC 9000-3 Sec. 3.14), lists the following four components of software:

- Computer programs (the "code")
- Procedures
- Documentation
- Data necessary for operating the software system.

2. What is Software Engineering

Software Engineering

What is it?

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software.

-Wikipedia

Software Engineering

What is it?

The study and application of methodologies to develop quality software that fulfill customer needs.

Software Engineering

Objective

To produce software that is:

- ✓ On time: is deliver at the established date.
- ✓ Reliable: doesn´t crash.
- ✓ Complete: good documentation, fulfill customer needs.

Software Engineering

- Rekayasa Perangkat Lunak / RPL (in Indonesia)
- Suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal *requirement capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), desain, *coding*, *testing* sampai pemeliharaan sistem setelah digunakan.

  ~ *Romi Satria Wahono (berdasar pendapat Ian Sommerville)*

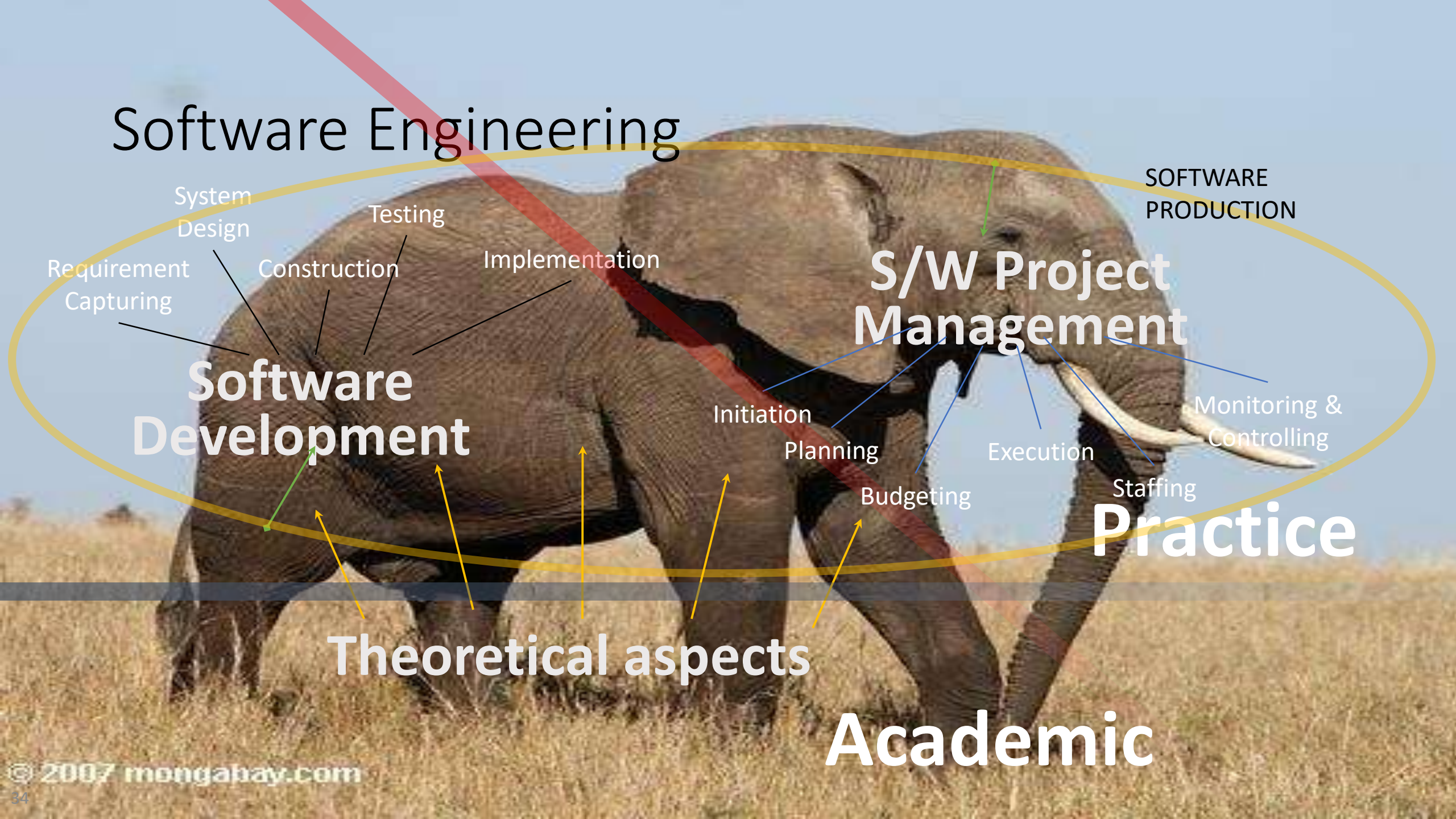
Mengapa Software Engineering?

- Terminologi rekayasa perangkat lunak (software engineering) pertama kali digunakan pada conference tentang software crisis tahun 1968
- Krisis perangkat lunak merupakan akibat langsung dari lahirnya komputer generasi ke 3 yang canggih pada waktu itu
- Perangkat lunak yang dihasilkan menjadi menjadi beberapa kali lebih besar dan kompleks
- Pendekatan informal tidak cukup efektif (cost, waktu dan kualitas) dalam pengembangan perangkat lunak
- Biaya hardware jatuh dan biaya perangkat lunak naik cepat

Generasi Komputer

1. Generasi I (1946-1959)
   - Menggunakan tabung hampa
   - ENIAC, EDSAC
2. Generasi II (1959-1964)
   - Menggunakan transistor
   - PDP-1, PDP-8, UNIVAC, IBM 70xx
3. Generasi III (1964-1979)
   - Menggunakan IC
   - IBM S360, NOVA, UNIVAC 1108
4. Generasi IV (1980-sekarang)
   - Menggunakan VLSI

Software Engineering



Software Engineering

- RPL bukan cabang dari Computer Science yang mempelajari *technical coding / programming* [algorithm, data structure, programming language, etc].

SE based on SWEBOK*)

*) *Software Engineering Body of Knowledge, IEEE Computer Society, 2004*

- S/W requirement
- S/W design
- S/W construction
- S/W testing
- S/W configuration management

Software Development

- S/W engineering management
- S/W engineering process
- S/W quality
- S/W maintenance

Project Management

- Software engineering tools and methods
- Knowledge area of the related disciplines

Tools & Basic-Theories

SE in Practice: Its Application

- Aplikasi SE dalam praktek: manajemen kegiatan produksi perangkat lunak.
- Ada 2 bagian besar:
  - *Software Development* (pembangunan perangkat lunak) berhubungan dengan bagaimana perangkat lunak dirancang dan dibangun.
  - *Project Management* (manajemen proyek pembangunan perangkat lunak) berhubungan dengan bagaimana perancangan dan pembangunan perangkat lunak bisa terlaksana dengan baik hingga berhasil diimplementasikan.

Software products

- Generic products
  - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.
- Customized products
  - Software that is commissioned by a specific customer to meet their own needs.
  - Examples – embedded control systems, air traffic control software, traffic monitoring systems.

Product specification

- Generic products
  - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
- Customized products
  - The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

Frequently asked questions about software engineering

Question	Answer
What is software?	Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
What are the attributes of good software?	Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.
What is software engineering?	Software engineering is an engineering discipline that is concerned with all aspects of software production.
What are the fundamental software engineering activities?	Software specification, software development, software validation and software evolution.
What is the difference between software engineering and computer science?	Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
What is the difference between software engineering and system engineering?	System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.

Frequently asked questions about software engineering

Question	Answer
What are the key challenges facing software engineering?	Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software.
What are the costs of software engineering?	Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs.
What are the best software engineering techniques and methods?	While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.
What differences has the web made to software engineering?	The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

Essential attributes of good software

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.