

Software Engineering: CSC1233

Advanced Software Engineering Concepts: COM3212

Introduction Lecture 01



Dr. S.M. Vidanagamachchi

CSC1233/COM3212 : Course Outline

- Introduction to Software Engineering
 - Activities and Characteristics Software Processes
 - Software Process Models
 - Requirement Analysis and Specification
 - System Modeling
- 30 hours
- Software Design (Design Principles and Architectural Design)
 - Software Testing
 - CASE tools
 - Software Maintenance and Metrics
- 15 hours

Learning Outcomes

At the end of this **module**, you should be able to:

- Explain Software Development Lifecycle activities
- Describe features of different software process models
- Apply the most suitable process model according to the given project
- Design software products with more flexible efficient principles and good coding standards
- Explain required testing methodologies for different types of projects

Outline

- Introduction to Software Engineering
- Status of Software Engineering
- Development Failures: Case Studies
- Importance of Software Engineering
- Essential Attributes of Software Engineering
- Software Process Activities
- Application Types

Learning Outcomes

At the end of this **lecture**, you should be able to:

- Define the term ‘Software Engineering’
- Identify software quality attributes and their classification
- Understanding the need for a managed approach to software development
- Explain the problems associated with developing a software

Lecture Schedule & Grading Policy

- Lecture hours
COM3112: 30 (2 credits)
CSC1233 : 45 (3 credits)
- Semester End Examination – 100%

References

- Ian Sommerville, Software Engineering, 10th Edition, Addison Wesley, 2016.
- R Pressman, Software Engineering - A Practitioners Approach, 5th Edition, McGraw Hill.

Software

- A set of instructions given to a computer
- Two types:
 - Systems Software
 - Application Software

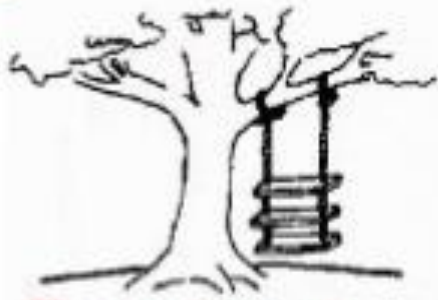
System Software

- A collection of programs written to service the other programs
 - eg. Operating systems, drivers, compilers

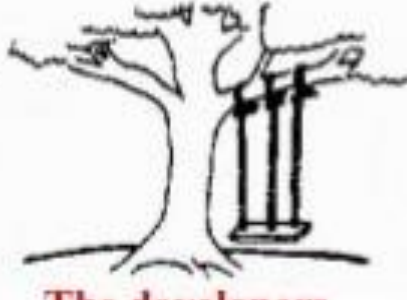
Application Software

- Personal computer software
- Business software
- Engineering software
- Web-based software
- Embedded software
- Artificial Intelligent software

Status of Software Engineering



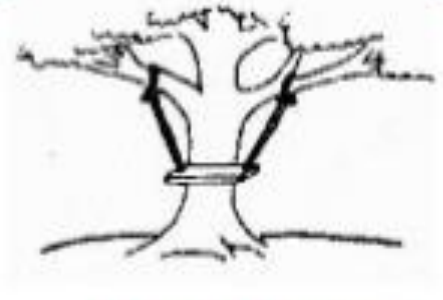
The requirements specification was defined like this



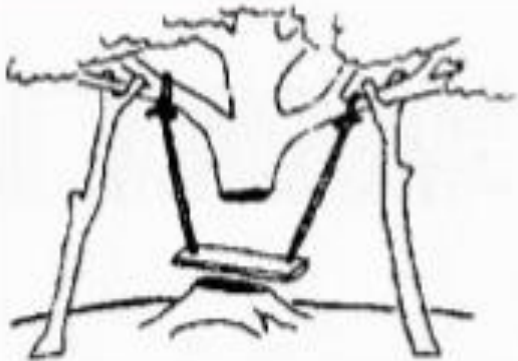
The developers understood it in that way



This is how the problem was solved before.



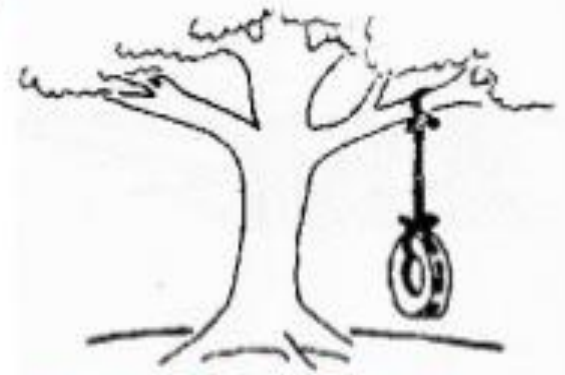
This is how the problem is solved now



That is the program after debugging



This is how the program is described by marketing department



This, in fact, is what the customer wanted ...

“Before software
can be reusable,
it has first to
be usable.”

- Ralph Johnson

Development Failures

- **IBM Survey, 2000**

- 55% of systems cost more than expected
- 68% overran the schedules
- 88% had to be substantially redesigned

- **Bureau of Labour Statistics (2001)**

- for every 6 new systems put into operation, 2 cancelled
- probability of cancellation is about 50% for large systems
- average project overshoots schedule by 50%

Development Failures

- **Over budget: The Boeing Dreamliner programme**
- This was announced in 2003, was supposed to cost **\$6 billion** and see the plane take to the air in 2008. The final bill was **\$32 billion** and the 787 Dreamliner arrived three years late. The result of a combination of various technical failures and supply-chain chaotic mess. With engineers, designers and other resources diverted into saving the Dreamliner production plans, the rest of its production were delayed.

Development Failures

- **Over schedule: Nats Project**
- National Air Traffic Services (Nats) was looking at replacing the systems at its new control center at Swanwick in Hampshire. This project is **six years late and 180m pounds over budget.**

This was originally meant to be operational by 1997, but problems with the development of software by Lockheed Martin caused delays, according to Nats.

Development Failures

- **Security failure: Internet Worm**

The **1988** Internet Worm was the first major worldwide computer security incident where malware (software that is malicious) propagated throughout the internet. This worm infected Unix servers, taking advantage of different types of vulnerability in installed code such as **Sendmail and finger**. The lessons from that incident are still valid and, surprisingly perhaps, the vulnerabilities identified that allowed the worm to cause such problems are still present in some modern software.

Development Failures

- **Security failure: Ariane 5**

In June 1996, the new Ariane 5 rocket was launched on its maiden flight. It carried a payload of scientific satellites. Ariane 5 was commercially very significant for the European Space Agency as it could carry a much heavier payload than the Ariane 4 series of launchers. Thirty seven seconds into the flight, software in the inertial navigation system, whose software was reused from Ariane 4, shut down causing incorrect signals to be sent to the engines. These swivelled in such a way that uncontrollable stresses were placed on the rocket and it started to break up. Ground controllers initiated self-destruct and the rocket and payload was destroyed.

What is Software Engineering?

Software Engineering

- **Definitions**

Simple definition : Designing, building and maintaining large software systems

- **Others**

‘Software engineering is concerned with the theories, methods and tools for developing, managing and evolving software products’

- I Sommerville

- ‘The practical application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate and maintain them’ - B.W.Boehm
- ‘The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines’ - F.L. Bauer

Importance of Software Engineering

1. It is difficult for a customer to specify complete set of requirements
2. It is difficult for the developer to understand fully the customer needs.
3. Software requirements change regularly.
4. Software is primarily intangible; much of the process of creating software is also intangible, involving experience, thought and imagination.
5. It is difficult to test software exhaustively.

Essential Attributes of Software Engineering

- 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 : 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.

Essential Attributes of Software Engineering

- Efficiency : Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes **responsiveness, processing time, memory utilization**, 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.**

Software process activities

- **Software specification:** customers and engineers define the software that is to be produced and the constraints on its operation
- **Software development:** software is designed and programmed
- **Software validation:** software is checked to ensure that it is what the customer requires
- **Software evolution:** software is modified to reflect changing customer and market requirements

Application Types

- **Stand-alone applications:** run as a separate computer process on PC without connecting to the internet
- **Interactive transaction-based applications:** execute on a remote computer and are accessed by users from their own PCs or terminals
- **Embedded control systems:** These are software control systems that control and manage hardware devices

Application Types

- **Batch processing systems:** business systems that are designed to process data in large batches
- **Entertainment systems:** personal use and intended to entertain the user
- **Systems for modeling and simulation:** model physical processes or situations, which include many, separate, interacting objects

Exercise

- Define the term “Software Engineering”.
- Discuss the major reasons for software failures.

Thank you