

Software Engineering

CSE 3102

smtareeq@cse.du.ac.bd

Objective of the Course

- **Appreciate Software Engineering:**
 - Build complex software systems in the context of frequent change
- **Understand how to**
 - produce a high quality software system within time
 - while dealing with complexity and change
- **Acquire technical knowledge**
- **Acquire basic managerial knowledge**

Course Contents

| Topic | Lectures |
|-------------------------|--|
| Introduction | 1, 2, 3, 4 |
| Requirement Engineering | 6, 7, 8, 9 |
| System Design | 10, 11, 12, 13 |
| Object Design | 14, 15, 16, 17 |
| Implementation | 18, 19 |
| Testing | 20, 21, 22, 23 |
| Software Life Cycle | 24, 25 |
| Project Management | 5 |
| Book | Object Oriented Software Engineering, B Bruegge and Software Engineering: A Practitioner' s Approach, 7/e by Roger S. Pressman |

Course evaluation

Theory

| Topic | Marks |
|-----------------|-------|
| Attendance | 5 |
| Quiz/Assignment | 5 |
| Mid Term | 20 |
| Final | 70 |

Lab

| Topic | Marks |
|--------------|-------|
| Proposal | A/R |
| RAD | 20 |
| SDD | 20 |
| STD | 20 |
| Presentation | 40 |

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 is Software?

Software Engineering is a collection of

- techniques,
- methodologies and
- tools

that help with the production of

a *high quality software* system

- with a given *budget*
- before a given *deadline*
- while *accepting changes*

What is Software Engineering

ISO/IEC/IEEE Systems and Software Engineering Vocabulary (SEVOCAB) defines **software engineering** as

“the application of a **systematic, disciplined, quantifiable approach** to the **development, operation, and maintenance** of software;

that is, **the application of engineering to software**”

What is the connection between software engineering and computer science?

Computer science focuses on theory and fundamentals (e.g. , programming languages, data structures, algorithms, operating systems, databases, etc.).

Software engineering is concerned with the practicalities of developing and delivering useful software.

What are the fundamental software engineering activities?

Specification – Requirements, architecture.

Development – Design, implementation/coding.

Verification & validation – Reviews and testing.

Evolution – Maintenance and growth.

What are the costs of software engineering?

Roughly

- 60% are development costs,
- 40% are testing costs.

For custom software, evolution costs often exceed development costs.

What are the key challenges facing software engineering?

Coping with increasing **diversity**, demands for **reduced delivery times** and **developing trustworthy software**.

Why software Fail?

- Lack of commitment – management and user
- Misunderstanding requirements
- End user expectations
- Scope change, new technology
- Expertise
- Requirements froze/change
- Etc.

An Example



How the customer
explained it

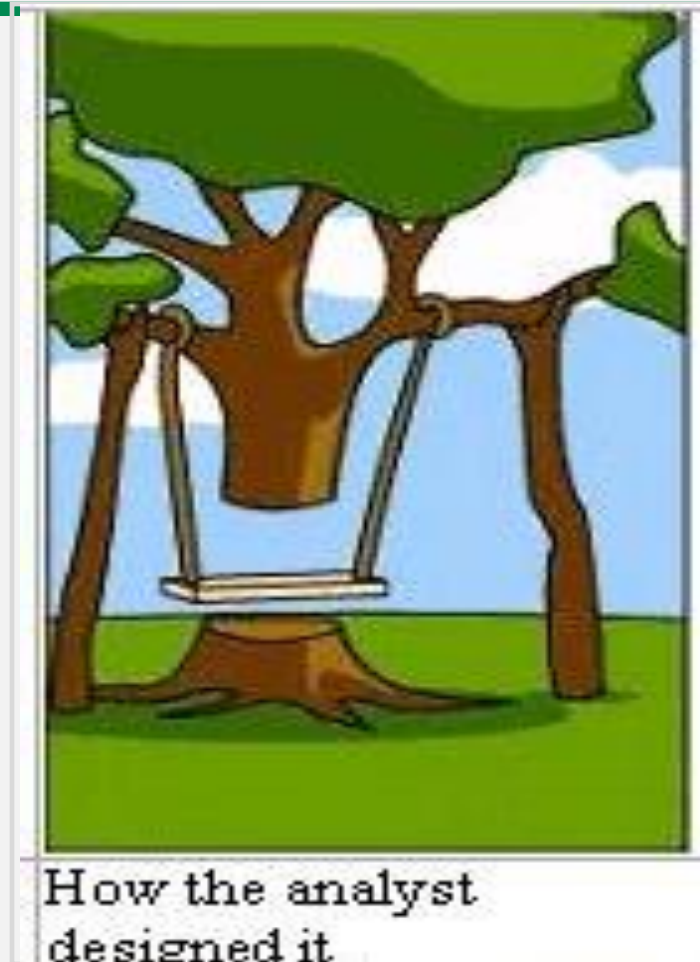
An Example



How the project leader
understood it

Poor Communication

An Example



Poor Planning and Modeling

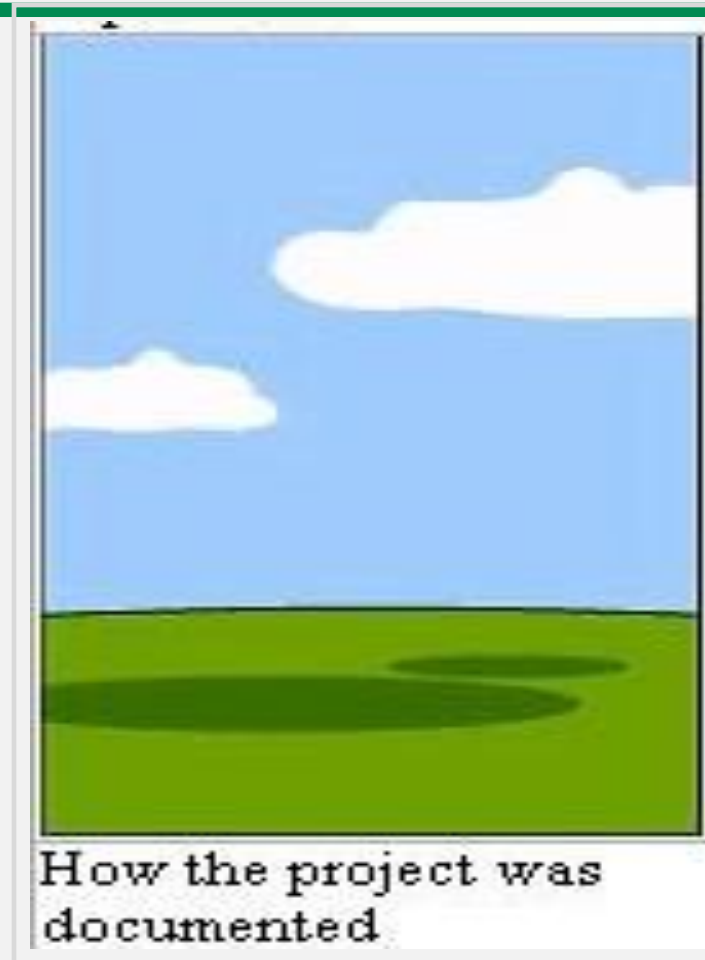
An Example



How the programmer
wrote it

Poor construction

An Example



Poor documentation

An Example



What the customer
really needed

Failure to deploy

What are the properties of good software?

Good software should deliver the **required functionality** to the user and should be **efficient, maintainable, dependable** and **usable**.

What are the key challenges facing 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.

What are the key challenges facing software engineering?

Maintainability

Maintainability as the **capability** of the **software product** to be **modified**.

Modifications may include **corrections**, **improvements**, or **adaptation** of the software to changes in environment as well as changes in **requirements** and functional specifications.

What are the key challenges facing software engineering?

Dependability

Software dependability includes a range of characteristics including **reliability, availability, 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.

What are the key challenges facing software engineering?

Usability

Software must be **acceptable** to the type of **users** for which it is designed.

It must be **understandable**, **usable** and **compatible** with other systems that they use..

What are the expected outcomes?

The **study** and **application** of **systematic processes, methods** and **techniques** for software **design, implementation** and **testing**.

Learn how to:

Specify **requirements** for a large scale software system.

Specify the **architecture** of the system based on the requirements specification.

Design and **implement** the subsystems of the system's architecture.

Test the system in a principled way that guarantees the quality of the result.

Organize the **delivery** of the **system** and the user's **training**.

How to Deliver?

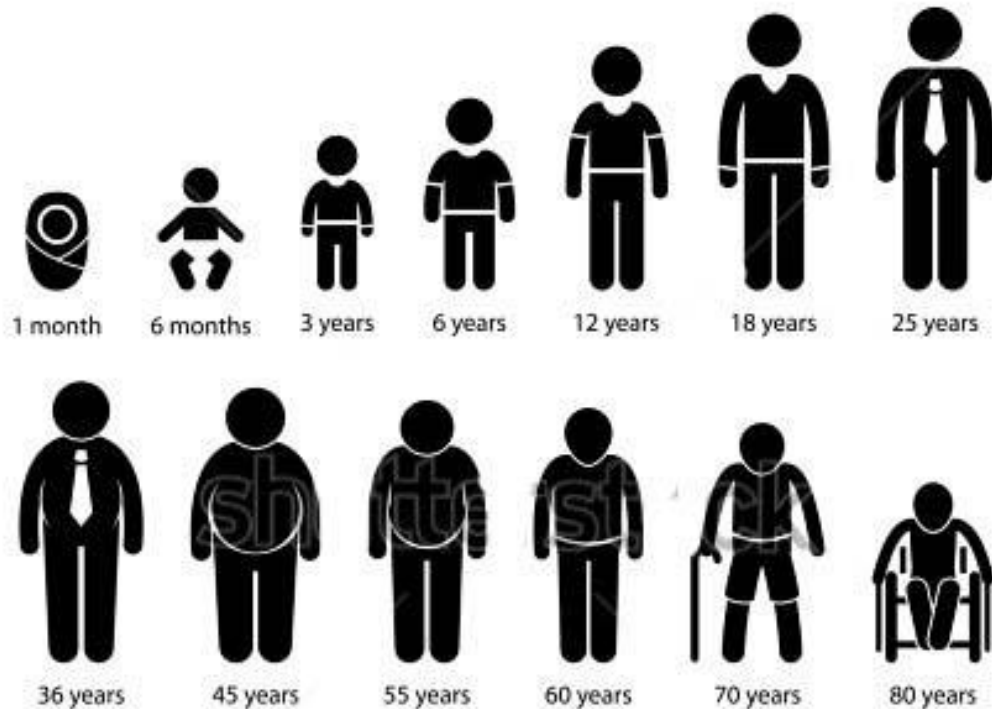
An Eskimo!



A Face!



Extreme Programming?



dead



RIP

Thank You