Welcome To

Software Engineering

Contents

1) Course Objectives
2) Software Engineering concepts (Terminology)
3) Software Types
4) What is Software Engineering (SE)?
5) Why Software Engineering?
6) Software Engineering and other related fields

Course Objectives

- How to use software Engineering concept in order to
  - Produce software product that satisfy the user’s needs and meets its goal [budget, deadline, quality]
  - Improve the quality of a software development effort
- **How ?** By
  - Addressing the problems in information systems development
  - Studying and understanding the software engineering process and different process models (Software life cycle)
  - Studying the fundamental theories and techniques of software development methodologies
  - Develop a system design using UML notation
  - Studying the tools and techniques for System Analysis and Design, requirement specifications, software metrics and planning, etc.
  - How to elicit requirements from a client and specify them

Basic Terminology

What is software? Computer program codes and associated libraries and documentations.

What is engineering? Is all about developing products, using well-defined, scientific principles, tools and methods to find **cost effective** solution to problems

What is software engineering? It is an engineering discipline which is concerned with all aspects of software products

What is Information system (IS)? it is an umbrella term for the systems, people and processes designed to create, store, manipulate, distribute and disseminate information.

- All software need not be information system software while information system is always implemented as software.
- For example a company writing anti-virus software is not dealing with any information systems. Information systems let you access data for example your company employee payroll system, employee master file etc.

Basic Terminology

What is software? Computer program codes and associated libraries and documentations.

What is engineering? Is all about developing products, using well-defined, scientific principles and methods.

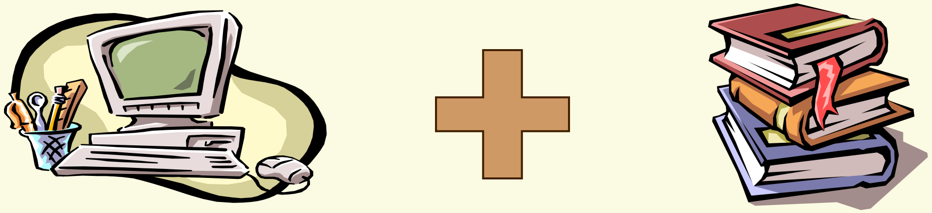
What is software engineering? It is an engineering discipline which is concerned with all aspects of software products

What is Information system (IS)? IS an umbrella term for the systems, people and processes designed to create, store, manipulate, distribute and disseminate information.

- All software need not be information system software while information system is always implemented as software.
  - For example a company writing anti-virus software is not dealing with any information systems. Information systems let you access data for example your company employee payroll system, employee master file etc.

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
- **Software products** may be
  - **Generic** - developed to be sold to a range of different customers (e.g. Word processing, drawing packages, programs of anti-viruses
  - **Customized** - developed for a particular customer according to their specification (e.g. Air traffic software of control system for an airport)

Software Types

- System Software : is the software that acts as tools to help constructs or support applications software (e.g. compilers, operating systems)
- Application software : is software that helps perform some directly useful task (games, ATM software, the control software in an airplane, Word processors, etc.)

Applications Software Categories

- Games
- Information Systems
  - Airline seat reservation system
- Real-time systems
  - Mobile Phone software
- Office Software (Spreadsheets, E-mail)
- Scientific Software (Calculations, prediction, weather forecasting)

Cont.

What is software process?

Set of activities whose goal is the development or evaluation of software

What is software process model?

A simplified representation of a software process.

What are the costs of software engineering?

In most cases: 60% of costs are for development, 40% of costs are for testing

What is SE?

- Software Engineering (SE) is an engineering branch related to the evolution of software product using systematic well-defined scientific principles, techniques, and procedures.
- The result of software engineering is an effective and reliable software product.

What is Software Engineering (SE)?

The Methods, tools and techniques used for developing software are called **Software Engineering** (SE),

Software engineers are concerned with developing software products.

Software Engineering (SE) : It is an engineering discipline whose focus is the development of cheaper, better and faster software product (i.e. it is concerned with all aspects of software production)

Software engineers are concerned with developing software products.

SE =Modelling activity

= Problem-solving activity

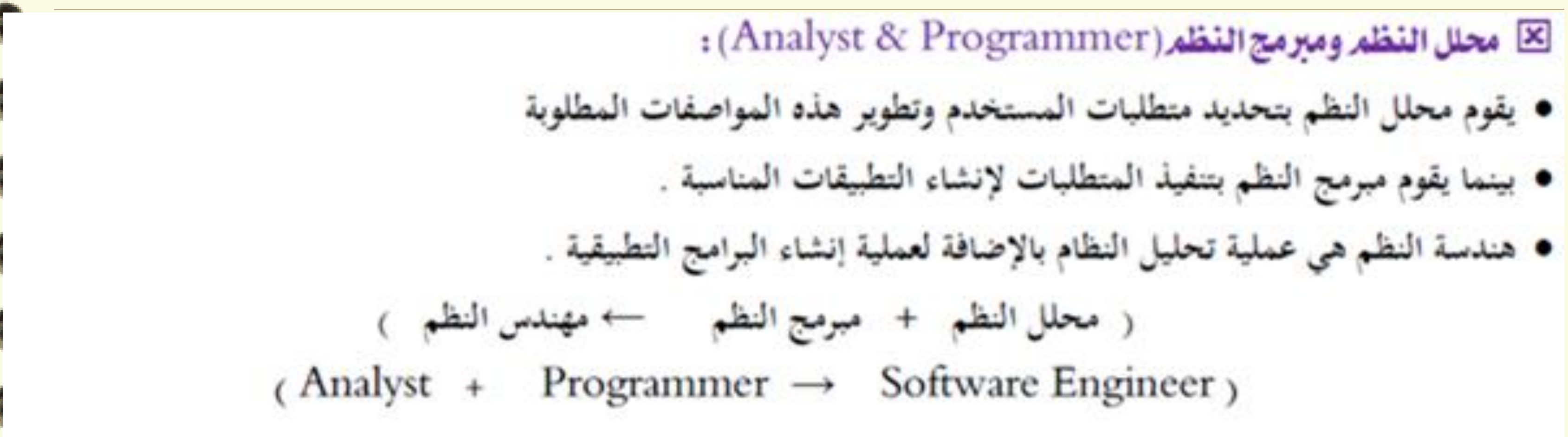
- (formulate the problem and analyze it search for solutions, choose the appropriate one then specify it)

= Knowledge acquisition activity

- (collect data, organize it into information then formalize it into knowledge)

Note:- An activity is a set of tasks that is performed toward a specific purpose

Software Engineers should



☒ محلل النظم ومبرمج النظم(Analyst & Programmer):

- يقوم محلل النظم بتحديد متطلبات المستخدم وتطوير هذه المواصفات المطلوبة
- بينما يقوم مبرمج النظم بتنفيذ المتطلبات لإنشاء التطبيقات المناسبة .
- هندسة النظم هي عملية تحليل النظام بالإضافة لعملية إنشاء البرامج التطبيقية .

(محلل النظم + مبرمج النظم ← مهندس النظم)

(Analyst + Programmer → Software Engineer)

- adopt a systematic and organised approach to their work
- use appropriate tools and techniques depending on
  - the problem to be solved,
  - the development constraints and
  - the resources available

Why is Software Engineering required?

- Software Engineering is required due to the following reasons:
  - To manage Large software
  - For more Scalability
  - Cost Management
  - To manage the dynamic nature of software
  - For better quality Management

Why to use Software Engineering?

- Complex and large-scale software are made by teams not individuals.
- Software engineering is about teams who participate in solving complex or large problems that a single developer cannot solve them anymore.
- Teams do not consist only of developers, but also of testers, architects, system engineers, customer, project managers, etc.
- Software projects can be so large that we have to do careful planning.

Why to use Software Engineering?

- Implementation is no longer just writing code, but it is also following guidelines, writing documentation and also writing unit tests.
- Once we are finished coding, that does not mean that we are finished with the project:
  - for large projects maintaining software can keep many people busy for a long time.
- We also need to learn a little about project management and its drawbacks.

Why to use Software Engineering?

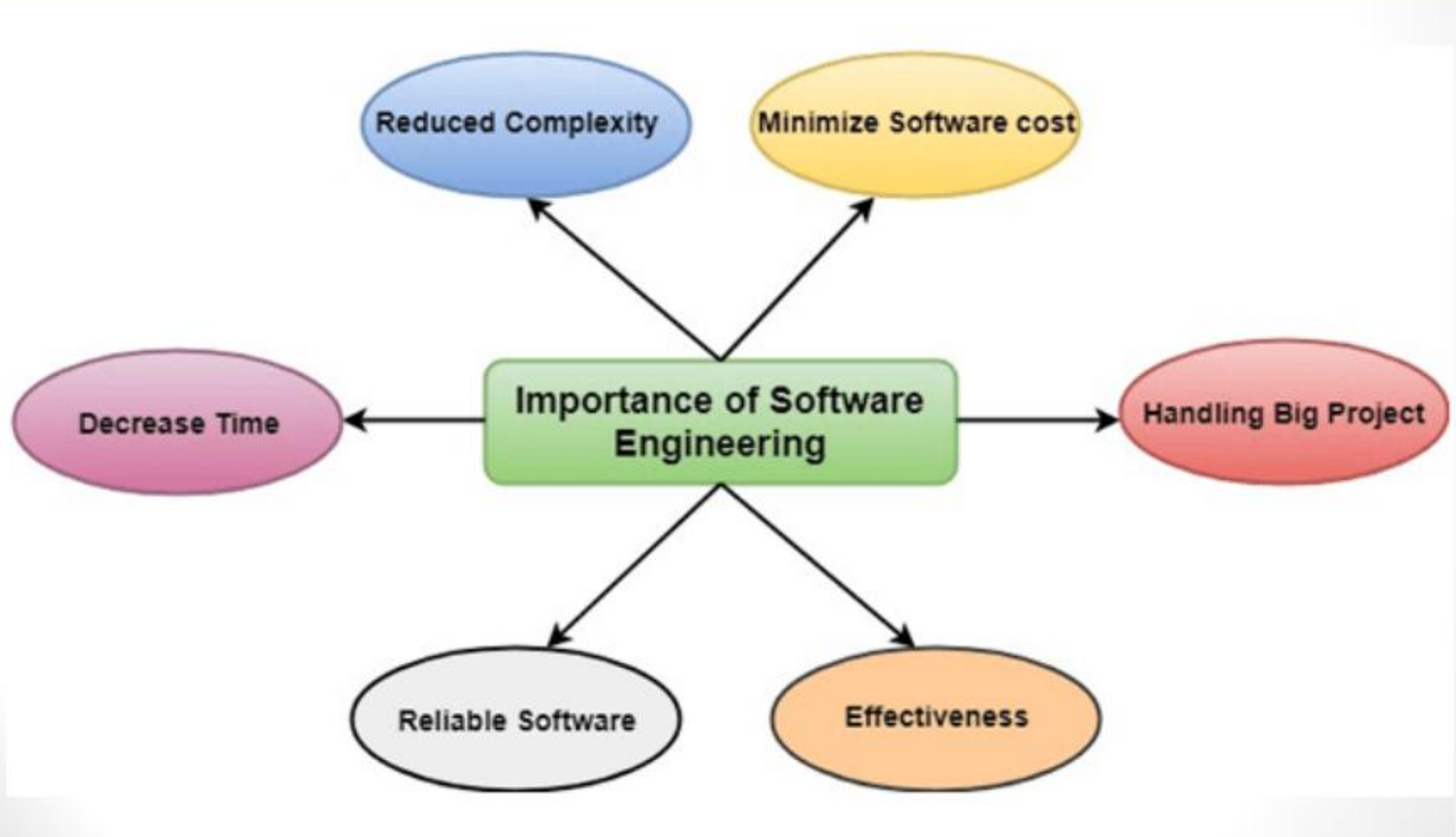
- Teach us to see the big picture (patterns). This gives us yet one higher level of abstraction.
- Also, it is a fantastic way to learn from our seniors. This is essential for designing large software systems.
- And last but not least, a good software engineer, like any engineer, needs tools and we need to know about them.

Why to use Software Engineering ?

- Producing high quality software system
- Complexity
- Many sources, but ***size*** is key:
  - UNIX contains 4 million lines of code
  - Windows 2000 contains 10^8 lines of code

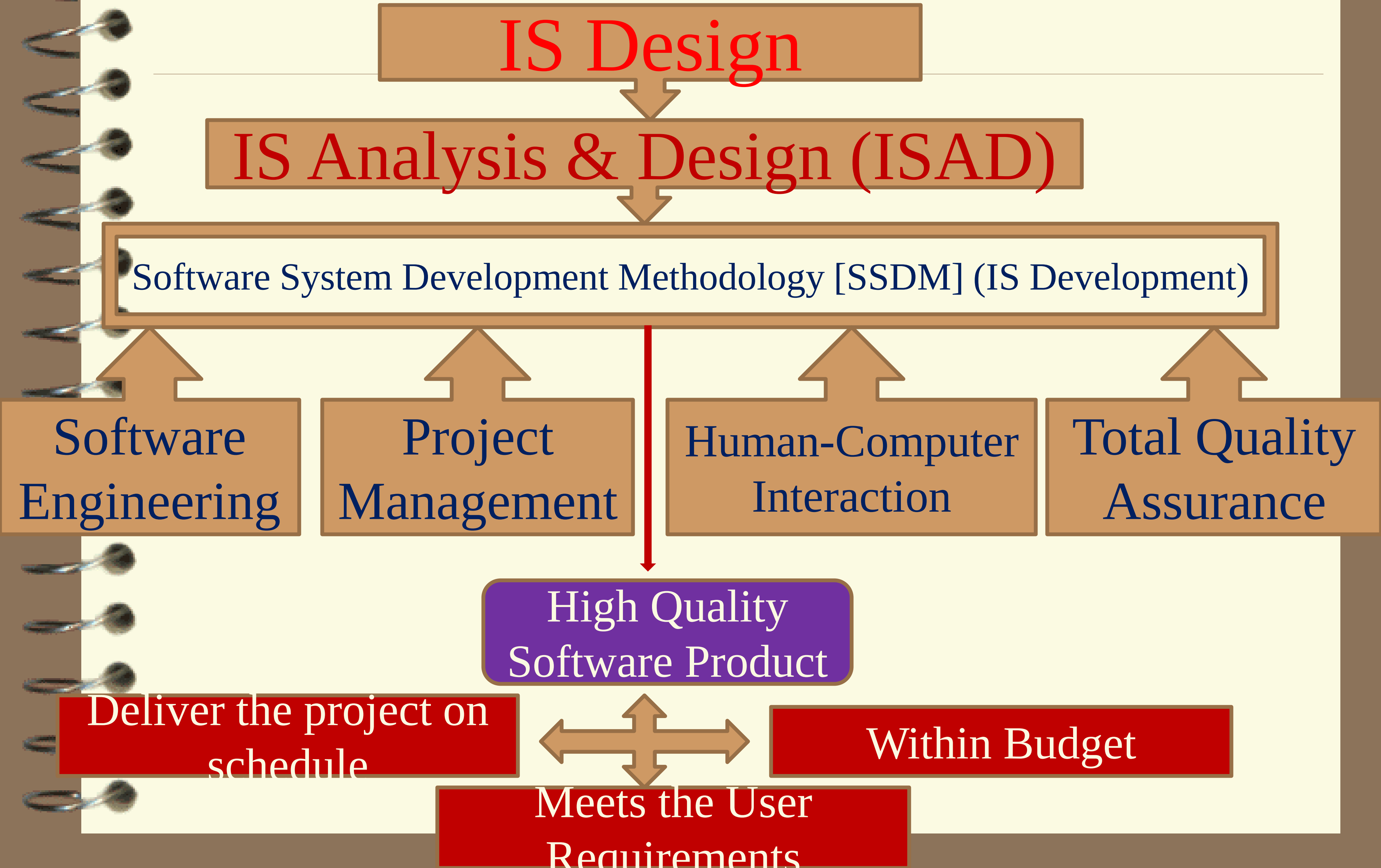
Software engineering is about managing this complexity.

Importance of Software Engineering



Software reliability & Effectiveness

- Software reliability is the "probability that the software will execute for a particular period of time without failure"
- Effectiveness means getting the desired results/ doing the right thing!



IS Design

IS Analysis & Design (ISAD)

Software System Development Methodology [SSDM] (IS Development)

Software Engineering

Project Management

Human-Computer Interaction

Total Quality Assurance

High Quality Software Product

Deliver the project on schedule

Within Budget

Meets the User Requirements

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
- **System engineers** are involved in

  system specification,

  architectural design,

  integration and **deployment**

Software engineering is part of this process concerned with developing the software infrastructure, control, applications and databases in the system.

Questions?

What are the attributes of good software?

The software should deliver the required functionality and performance to the user and should be

- Maintainable
- Dependable
- Usable

What are the key challenges facing software engineering?

- Coping with increasing diversityالتعامل مع التنوع المتزايد
- Demands for reduced delivery times
- Developing trustworthy softwareتطوير برامج جديرة بالثقة

What are the attributes of good software?

The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable

- Maintainability
  - Software must evolve to meet changing needs
- Dependability
  - Software must be trustworthy
- Efficiency
  - Software should not make wasteful use of system resources
- Usability
  - Software must be usable by the users for which it was designed (i.e. the product is easy to use)

Accessibility, Usability and User experience

- **The user experience** is the degree of "satisfaction" that the end user has with the system or service after using it
- **Accessibility** refers to the system is available and can be easily accessed by a normal person or even disabled person
- **Usability** means that the product is easy to use
- **Usability testing** focuses on measuring the ease at which a system can be used

Accessibility

- Accessibility is the condition that environments, services, processes, and objects (everything that involves an interaction) must meet, which must be understandable and usable by the broadest range of people, regardless of their capabilities.
- إمكانية الوصول هي الشرط الذي يجب أن تفي به البيئات والخدمات والعمليات والأشياء (كل ما ينطوي على تفاعل) ، والتي يجب أن تكون مفهومة وقابلة للاستخدام من قبل مجموعة واسعة من الناس ، بغض النظر عن قدراتهم.

User Experience

User experience is meaningful and valuable when a product, service or system is

- useful (that is, its content is original and satisfies a need),
- usable (the product is easy to use),
- desirable (the image, identity, brand, and other design elements produce positive emotions towards the product),
- locatable (the content is accessible to people with disabilities),
- credible (users have confidence in the product), and
- valuable (an added value is generated from the product).

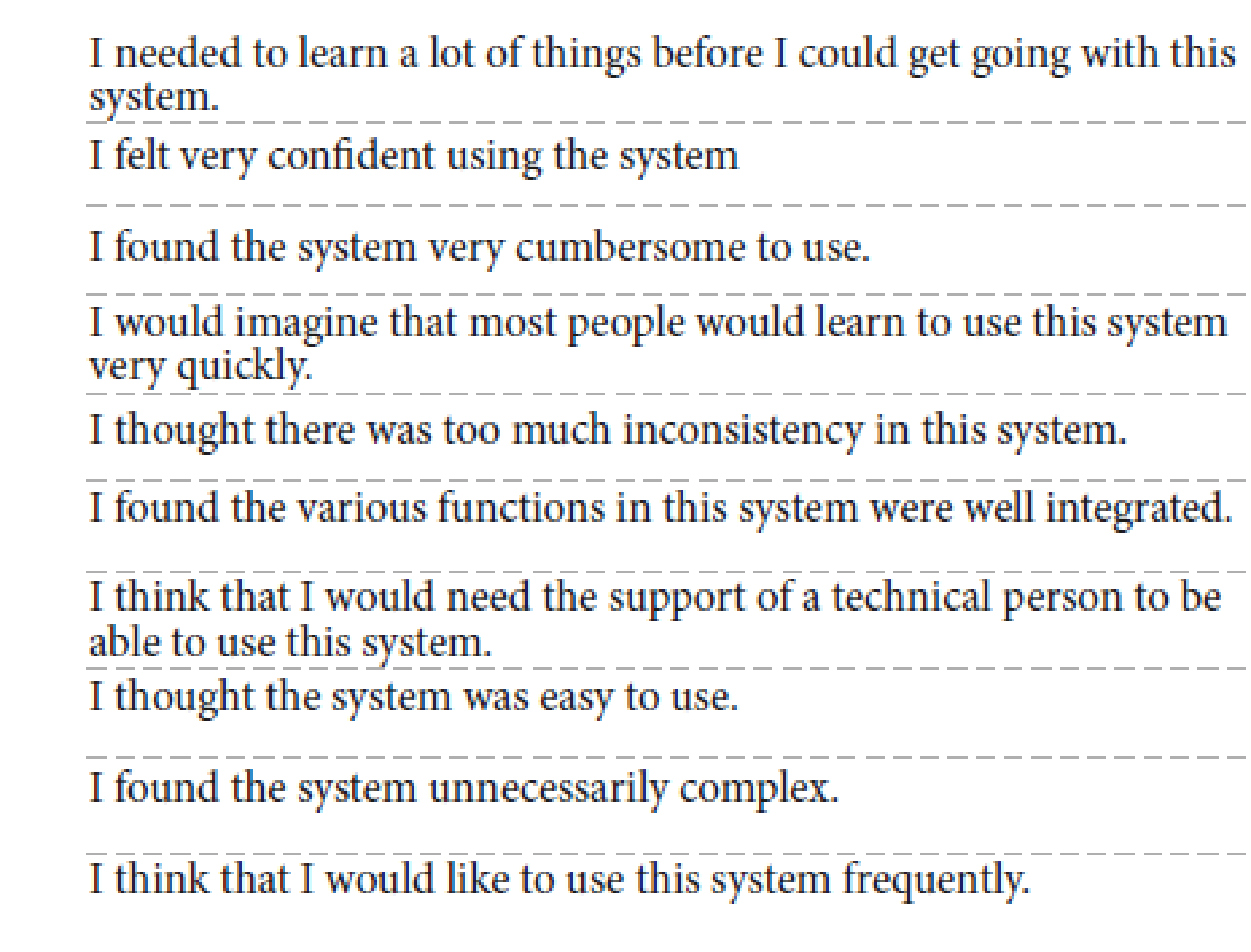
System Usability Scale (SUS)

- The SUS questionnaire contains 10 questions about:
- **ease of learning**,
- **efficiency**,
- **ease of memorization**,
- **occurrence of execution errors, and**
- **level of satisfaction**.
- Each question has a five-point scale varies from one (totally disagree) to five (totally agree). The SUS provides a score from 0 to 100, where a score of 85 or higher represents exceptional usability and a score below 70 represents unacceptable usability.

SUS

- **efficiency**: how fast the user can accomplish
- a task after learning to use the interface;
- **ease of memorization**: when the user returns to the interface after a period of no use, how easily can it establish the same competence already achieved;
- **minimization of errors**: level of seriousness of the mistakes made by the user and how easily the system can recover from them;
- satisfaction: how pleasant is the use of the interface;
- **ease of learning**: how easy it is for the user to perform simple tasks using the interface for the first time.

SUS Questions



I needed to learn a lot of things before I could get going with this system.

I felt very confident using the system

I found the system very cumbersome to use.

I would imagine that most people would learn to use this system very quickly.

I thought there was too much inconsistency in this system.

I found the various functions in this system were well integrated.

I think that I would need the support of a technical person to be able to use this system.

I thought the system was easy to use.

I found the system unnecessarily complex.

I think that I would like to use this system frequently.

What are the key challenges facing software engineering?

Software engineering in the 21st century faces three key challenges:

- **Legacy systems**
  - Old, valuable systems must be maintained and updated
- **Heterogeneity**
  - Systems are distributed and include a mix of hardware and software
- **Delivery**
  - There is increasing pressure for faster delivery of software

Why Projects Fail ?

- It is reported that
  - 42% of all corporate IS projects were abandoned before completion
  - 53% of all U.S. government IS projects were abandoned
  - Many of the systems that aren’t abandoned are delivered to the users significantly late, cost far more than planned and have fewer features than originally planned
- Generally over 50% of projects are either canceled or over 100% late
- Even Microsoft has a history of failures and overdue projects (e.g. Windows 1.0, Windows 95)

The causes of Failure

- Unrealistic schedules
- Inappropriate staffing
- Changing requirements
- Poor quality work

Flynn Approach to IS Project Failures

- **Two types**
- **Quality problems**
  - Quality means
    - Fitness for purpose, or
    - Meeting all user requirements
- **Productivity problems**
  - Will the product be delivered?
  - Will it be delivered in time to be useful?
  - Will it be affordable
- Flynn suggestion

Quality Problem Reasons

- The wrong problem is addressed (no clear requirements)
- Wider influences are neglected (organization culture)
- Analysis is carried out incorrectly (poor skilled team)
- Project undertaken for wrong reason

Productivity Problem Reasons

- Users change their minds (requirements)
- External events change the environment
- Implementation is not feasible (too late)
- Poor project control (poor managers)

Software Engineering (SE) Goals

- meeting user’s needs (Requirements analysis or requirements engineering)

[Find out what is the customer or user needs]

[establishing clearly what the user wants]

- low cost of production
- high performance (efficiency)

Software Engineering (SE) Goals Cont.

- portability
  - (Transferring the software product from one type of computer to another with minimum effort)
- low cost of maintenance
  - (is the term for any effort that is put into a piece of software after it has been written and put into operation)
  - maintenance types
    - Remedial maintenance (Fixing bugs) [Correcting faults in the software]
    - Adaptive maintenance (is altering software either because the users needs have changed or the computer, OS or programming language has changed)
  - Maintenance is a consequence of inadequate testing

Software Engineering (SE) Goals Cont.

- delivery on time
- high reliability
  - A piece of software is said to be reliable if it works and continues to work without crashing and without doing something undesirable
  - Bug Types
    - Error: a wrong decision made during software development
    - Fault: a problem that may cause software to departs from its intended behavior
    - Failure: an event when software to departs from its intended behavior
  - An error causes one or more faults within the software. A fault can cause one or more failures.

Maintenance Vs Reliability

Reliability α (1/Maintenance)

Software Engineering Blueprints

- Specifying software problems and solutions is like cartoon strip writing
- Unfortunately, most of us are not artists, so we will use something less exciting:
  UML symbols

UML (Unified Modeling Language)

- For teams to work effectively they need a language to communicate (UML).
- In previous courses we learned languages, such as Java or C++, and how to turn ideas into code. But these ideas are independent of the language.
- With UML we will see a way to describe code independently of language, and more importantly, we learn to think in one higher level of abstraction.
- UML can be an invaluable communication and documentation tool..أداة اتصال وتوثيق لا تقدر بثمن

Remark

System to be developed consists of:

- Actors
  - Agents external to the system that interact with it
- Concepts/ Objects
  - Agents working inside the system to make it function
- Use Cases
  - Scenarios for using the system

Self-test Question

Flynn categorizes projects failure into productivity and quality problems. The following are some reasons of projects failure, indicate which one is productive or quality problems

a) The user changes the requirements

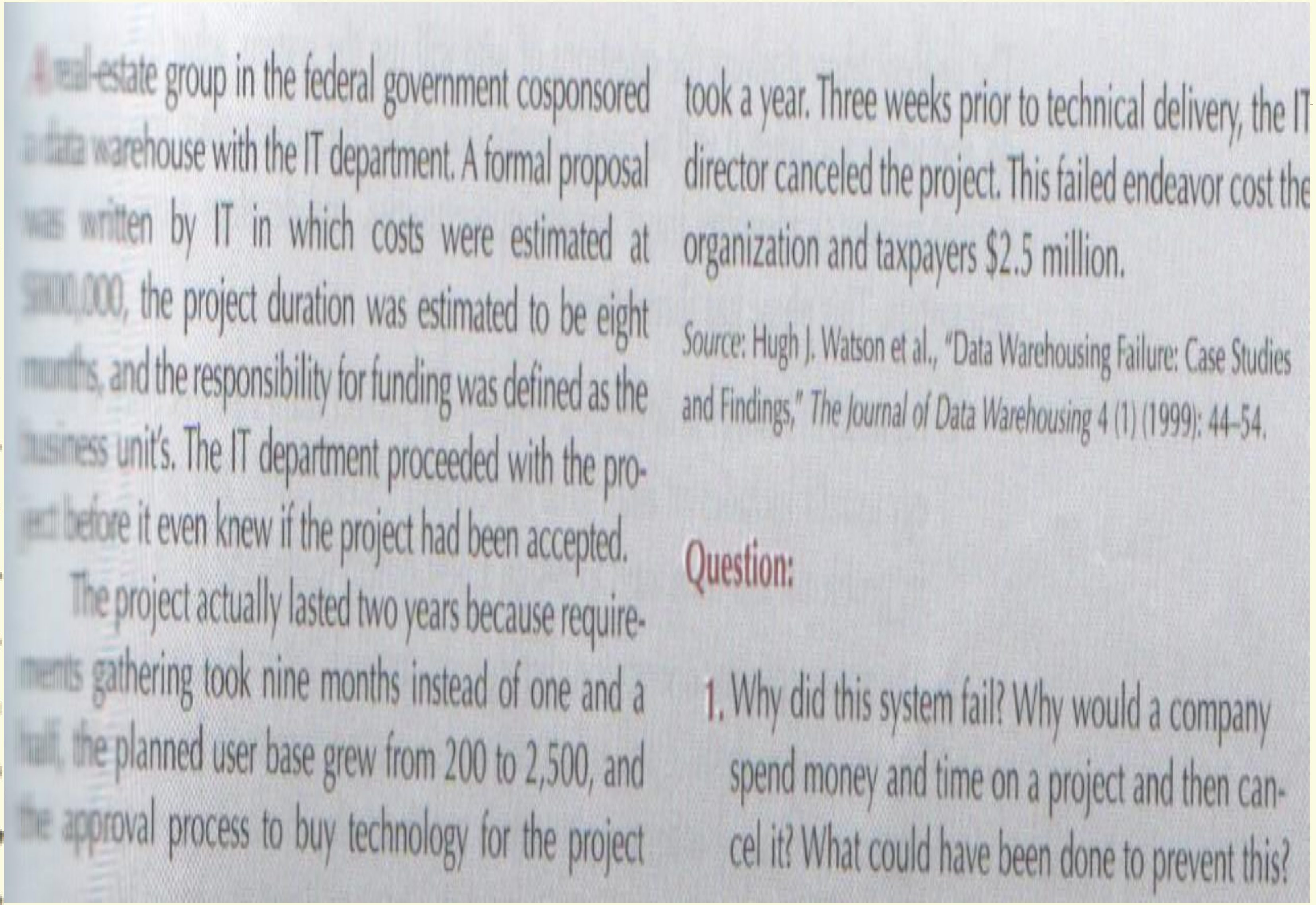
b) The wrong problem is addressed

c) Analysis is carried out incorrectly

d) Poor project control

e) Implementation is not feasible

Activity#1



A real-estate group in the federal government cosponsored a data warehouse with the IT department. A formal proposal was written by IT in which costs were estimated at $800,000, the project duration was estimated to be eight months, and the responsibility for funding was defined as the business unit's. The IT department proceeded with the project before it even knew if the project had been accepted.

The project actually lasted two years because requirements gathering took nine months instead of one and a half, the planned user base grew from 200 to 2,500, and the approval process to buy technology for the project took a year. Three weeks prior to technical delivery, the IT director canceled the project. This failed endeavor cost the organization and taxpayers $2.5 million.

Source: Hugh J. Watson et al., "Data Warehousing Failure: Case Studies and Findings," *The Journal of Data Warehousing* 4 (1) (1999): 44–54.

Question:

1. Why did this system fail? Why would a company spend money and time on a project and then cancel it? What could have been done to prevent this?