



# Object Oriented Software Engineering

1

## LECTURE 2



# Today's Outline:

2

## **Introduction to Software Engineering**

- The Evolving Role of Software
- Changing nature of software



# What is Software?

3

- *The product that software professionals **build** and then **support** over the long term.*
- *Software encompasses: (1) **instructions** (computer programs) that when executed provide desired features, function, and performance; (2) **data structures** that enable the programs to adequately store and manipulate information and (3) **documentation** that describes the operation and use of the programs.*



# Software products

4

- **Generic products**

- Stand-alone systems that are marketed and sold to **any customer** who wishes to buy them.
- Examples – PC software such as editing, 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.



# Why Software is Important?

5

- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled (transportation, medical, telecommunications, military, industrial, entertainment,)
- Software engineering is concerned with theories, methods and tools for professional software development.
- Expenditure on software represents a significant fraction of GNP in all developed countries.



# Software costs

6

- Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.
- Software costs **more to maintain** than it does to develop. For systems with a long life, maintenance costs may be several times development costs.
- Software engineering is concerned with cost-effective software development.



# Introduction to Software Engineering:

7

## Software + Engineering

- **Software** is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called **software product**.
- **Engineering** on the other hand, is all about developing products, using well-defined, scientific principles and methods.

# Software engineering

8

- **Software engineering** is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.







# Definitions

9

- IEEE defines software engineering as:
  - (1) The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software.
  - (2) The study of approaches as in the above statement.



# Fritz Bauer, a German computer scientist, defines software engineering as:

10

- Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and work efficiently on real machines.



# Software Engineering: A Working Definition

11

Software Engineering is a collection of techniques, methodologies and tools that help with the production of

*A high quality software* system developed with a given *budget* before a given *deadline* while *change* occurs

**Challenge: Dealing with complexity and change**



# Importance of Software Engineering

12

- More and more, individuals and society rely on advanced software systems. We need to be able to produce **reliable and trustworthy systems economically and quickly.**
- It is usually **cheaper, in the long run**, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of system, the majority of costs are the **costs of changing** the software after it has gone into use.

# Software Evolution

13

- The process of developing a software product using software engineering principles and methods is referred to as **software evolution**. This includes the initial development of software and its maintenance and updates, till desired software product is developed, which satisfies the expected requirements.



# Software Engineering

14

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



# Characteristics of software in software engineering:

- **Software is developed or engineered; it is not manufactured in the classical sense:**
  - Although some similarities exist between software development and hardware manufacturing, few activities are fundamentally different.
- **The software doesn't “wear out.”:**
  - There are no software spare parts.
  - When a hardware component wears out, it is replaced by a spare part.
- **The software continues to be custom-built:**
  - A software part should be planned and carried out with the goal that it tends to be reused in various projects.



# THE EVOLVING ROLE OF SOFTWARE

16

- Today, software takes on a dual role.
- It is a **product** and, at the same time, the **vehicle** for delivering a product.
- As a **product**, it delivers the computing potential embodied by computer hardware. Whether it resides within a mobile phone or operates inside a mainframe computer, software is an information transformer.
- As the **vehicle** used to deliver the product, software acts as the basis for the control of the computer (operating systems), the communication of information (networks), and the creation and control other programs (software tools and environments).





# Software Evolution

17

- Evolution starts from the requirement gathering process. After which developers create a prototype of the intended software and show it to the users to get their feedback at the early stage of software product development. The users suggest changes, on which several consecutive updates and maintenance keep on changing too. This process changes to the original software, till the desired software is accomplished.
- Even after the user has desired software in hand, the advancing technology and the changing requirements force the software product to change accordingly. Re-creating software from scratch and to go one-on-one with requirement is not feasible. The only feasible and economical solution is to update the existing software so that it matches the latest requirements.



# THE EVOLVING ROLE OF SOFTWARE

18

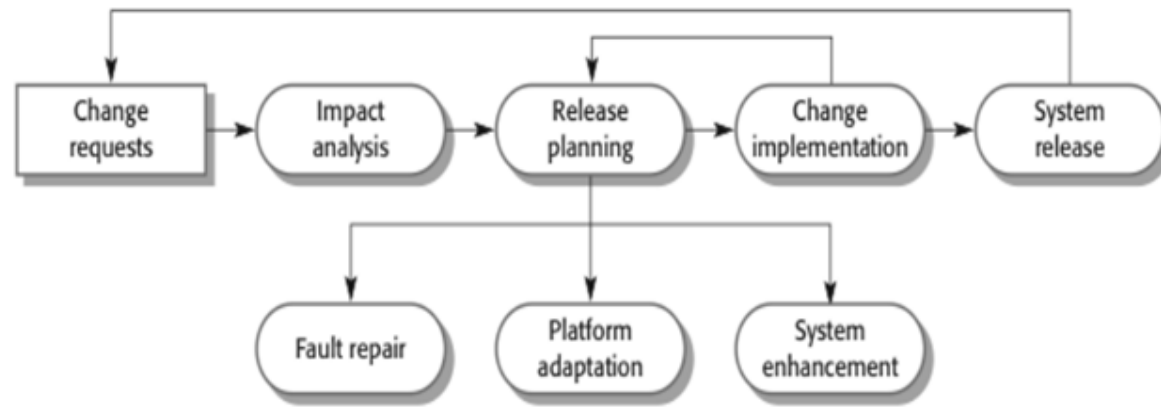
- Today, software takes on a dual role. It is a product and, at the same time, the vehicle for delivering a product. As a product, it delivers the computing potential embodied by computer hardware or, more broadly, a network of computers that are accessible by local hardware. Whether it resides within a cellular phone or operates inside a mainframe computer, software is an information transformer—producing, managing, acquiring, modifying, displaying, or transmitting information that can be as simple as a single bit or as complex as a multimedia presentation. As the vehicle used to deliver the product, software acts as the basis for the control of the computer (operating systems), the communication of information (networks), and the creation and control of other programs (software tools and environments). Software delivers the most important product of our time—information.

# THE EVOLVING ROLE OF SOFTWARE

19

- Software transforms personal data (e.g., an individual's financial transactions) so that the data can be more useful in a local context; it manages business information to enhance competitiveness; it provides a gateway to worldwide information networks (e.g., Internet) and provides the means for acquiring information in all of its forms.

The software Evolution Process:



The software Evolution Process



- **Lehman** has given laws for software evolution. These laws divided the software into *three different categories* are as follows:
- **S-Type (Static-type)** - This is software, which works strictly according to defined specifications and solutions. The s-type software is least subjected to changes hence this is the simplest of all. *For example, Calculator [Program for Mathematical Computation]*
- **P-Type (Practical-type)** - This is software with a collection of procedures. This is defined by exactly what procedures can do. *For example, Gaming Software.*
- **E-Type (Embedded-type)** - This software works closely as the requirement of the real-world environment. This software has a high degree of evolution as there are various changes in laws, taxes, etc. in real-world situations. *For example, Online Trading Software.*



# THE EVOLVING ROLE OF SOFTWARE

21

- The role of computer software has undergone significant change over a time span of little more than 50 years. Dramatic improvements in hardware performance, profound changes in computing architectures, vast increases in memory and storage capacity, and a wide variety of exotic input and output options have all precipitated more sophisticated and complex computer-based systems. The lone programmer of an earlier era has been replaced by a team of software specialists, each focusing on one part of the technology required to deliver a complex application. And yet, the same questions asked of the lone programmer are being asked when modern computer-based systems are built:
  - 1)Why does it take so long to get software finished?
  - 2)Why are development costs so high?
  - 3)Why can't we find all the errors before we give the software to customers?
  - 4)Why do we continue to have difficulty in measuring progress as software is being developed?



# Changing Nature of Software :

22

- The nature of software has changed a lot over the years.
- **1. System software:** Infrastructure software come under this category like compilers, operating systems, editors, drivers, etc. Basically system software is a collection of programs to provide service to other programs.
- **2. Real time software:** These software are used to monitor, control and analyze real world events as they occur. An example may be software required for weather forecasting. Such software will gather and process the status of temperature, humidity and other environmental parameters to forecast the weather.
- **3. Embedded software:** This type of software is placed in “Read-Only-Memory (ROM)” of the product and control the various functions of the product. The product could be an aircraft, automobile, security system, signalling system, control unit of power plants, etc. The embedded software handles hardware components and is also termed as intelligent software .



# Changing Nature of Software :

23

- **4. Business software :** This is the largest application area. The software designed to process business applications is called business software. Business software could be payroll, file monitoring system, employee management, account management. It may also be a data warehousing tool which helps us to take decisions based on available data. Management information system, enterprise resource planning (ERP) and such other software are popular examples of business software.
- **5. Personal computer software :** The software used in personal computers are covered in this category. Examples are word processors, computer graphics, multimedia and animating tools, database management, computer games etc. This is a very upcoming area and many big organisations are concentrating their effort here due to large customer base.





- **6. Artificial intelligence software:** Artificial Intelligence software makes use of non numerical algorithms to solve complex problems that are not amenable to computation or straight forward analysis. Examples are expert systems, artificial neural network, signal processing software etc.
- **7. Web based software:** The software related to web applications come under this category. Examples are CGI, HTML, Java, Perl, DHTML etc.





# FAQ about software engineering

25

Question	Answer
What is software?	Computer programs, data structures 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 is the <b>difference</b> 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 <b>difference</b> 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.



# Essential attributes of good software

26

## 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.

# THANK YOU