



*University of mustansiriyah /College of Education*

*Computer Science Department*

*Software Engineering 3<sup>rd</sup> Class*

*Lecturers maha ali hussain & muntaha abbod*

## **Chapter 2: Software Development Life cycle ( دورة حياة اعداد المنظومة )**

Topics:

2.1 Software Lifecycle

2.2 Software Development

2.3 Software Process Models

## **2.1 Software Lifecycle:**

Each software product proceeds to a number of distinct stages, these are:

- Requirements engineering
- Software design
- Software construction
- Validation and verification
- Software testing
- Software deployment
- Software maintenance

The software process used for the development of the software Product, these stages may occur in different orders, or frequency.

### **2.1.1 Requirements Engineering** (requirement analysis and definition by using engineering approach)

Requirements engineering is the interface between customers and developers on a software project. Requirements should make explicit the ideas of the customer about the prospective system.

### **2.1.2 Software Design**

The designers converts the logical software requirements from stage 1 into a technical software design by describe the software in such a way that programmers can write line of code that implement what the requirements specify.

### **2.1.3 Software Construction**

Software construction is concerned with implementing the software design by means of programs in one or more programming languages and setting up a build management system for compiling and linking the programs.

### **This stage content several steps, these are:**

#### **a. Software reuse**

1. Component based software engineering
2. Software product lines

#### **b. Security and reliability**

#### **c. Software documentation**

#### **d. Coding standards**

## **a. Software Reuse**

The goal of software engineering is to achieve many features with little effort and few defects. Software reuse is believed to play an important role in achieving this goal by encapsulating effort in units of source code, which can be reused in other projects.

However, the effort needed to make something reusable may not be worth it, if it is only reused few times, or needs extensive adaptation for each reuse.

### **a.1. Component Based Software Engineering**

Building software systems from prefab software components is an old dream of software engineering.

### **a.2. Software Product Lines**

Software systems are often part of a family of similar systems. The goal of a software product line is to maintain a set of reusable core artifacts that are common to all systems in the product line. Thus, code for a specific product can focus on the specifics of that product, reusing the common functionality.

## **b. Security and Reliability**

Software must be dependable by making it reliable ( software should work very well under any environments ), secure and safety (by verifying from user authentication to using any system).

## **c. Software Documentation**

- User documentation?
- Technical documentation?
- Documentation generation?

## **d. Coding Standards**

Coding standards are important to ensure portability and make code maintainable by others than the original developer.

### **2.1.4 Validation and Verification**

- Software inspection
- Software testing

### 2.1.4.A) Software Inspection

Software inspections are reviews of the code with the purpose of detecting defects. In an inspection someone other than the programmer reads a program unit of limited size to determine whether it satisfies the requirements and specification. A formal process and checklist are used to ensure that no aspects are forgotten.

### 2.1.4.B) Software Testing

Testing each unit founded in this software, follow by testing software integration.

### 2.1.5 Software Deployment

After development, software should be put to use. That is, it should be released and made available to users, who can then download, install, and activate it. These activities are captured under the common term *software deployment*. Richard S. Hall in the 'Software Deployment Information Clearinghouse' defines software deployment as follows: "The term software deployment refers to all of the activities that occur after a software system has been developed and made available for release. As such, software deployment includes activities such as packaging, releasing, installing, configuring, updating, and uninstalling a software system." and "Software deployment is the assembly and maintenance of the resources necessary to use a version of a system at a particular site".

#### **The following deployment activities make up the software deployment process:**

- \_ Release
- \_ Packaging
- \_ Transfer
- \_ Installation
- \_ Configuration
- \_ Activation
- \_ De-activation
- \_ Update
- \_ Adapt
- \_ De-installation
- \_ De-release

These activities are not necessarily performed sequentially. Many phases of the deployment process are often performed manually. For example, downloading, building and installing a source distribution of a software package, requires a number of commands to be formulated and executed. Each such command requires knowledge of some sort about the activity.

## **Manual deployment does not scale when deploying:**

- \_ many applications
- \_ applications composed from separately deployed components
- \_ on multiple machines
- \_ on different types of machines

### **2.1.6 Software Maintenance**

As software evolves after its first release, software maintenance is needed to improve it, i.e., repair defects, and to extend it, i.e., add new functionality.

## **2.2 Software development**

### **Three phases to develop the software**

- 1- Definition
- 2- Design
- 3- Maintenance

#### **1- Definition**

- 1- What information to be processed
- 2- What design constraints exist.
- 3- What function and performance desired.
- 4- What interfaces are desired.
- 5- What validation criteria are required?
- 6- What is modeling?

#### **2- Design**

- 1- How data structures to be designed.
- 2- How procedural details to be implemented.
- 3- How design to be translated into language.
- 4- How testing is performed.

#### **3- Maintenance**

- 1- Error
- 2- Adaptation.
- 3- Modification