جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قسم الانظمة الطبية الذكية

المحاضرة السادسة

Software engineering

المادة : Software engineering
المرحلة : الثالثة
اسم الاستاذ: م.د أحمد عدنان المحنا

Software Development Life Cycle (SDLC):

SDLC is a series of **phases** that provide a common understanding of the software building process. How the software will be realized and developed from the business understanding and requirements elicitation phase to convert these business ideas and requirements into functions and features until achieve the customer needs.

The good software engineer should have enough knowledge on **how** to choose the SDLC model based on the project context and the business requirements.

Therefore, it may be required to choose the right SDLC model according to the specific concerns and requirements of the project to ensure its success.

Software Process Models

What is a process model?

A process model is a simplified representation of a software process. It is a set of ordered tasks, involving activities, constraints; and Resources.

- The process of building a software product, called a **software life-cycle**.

- ***Life cycle -*** describes the life of the software from conception through its implementation, delivery, use and maintenance.

Why we need Software process?

1- Common understanding of the activities, resources and constraints involved in software development.

2- Creating processes helps to find inconsistencies, Redundancies; and Omissions.

Software Process Models :

A. Waterfall model
B. V- Shaped model
C. Evolutionary Prototyping Model
D. Iterative and Incremental Model
E. Spiral Model (e.g. Reuse-based development)
F. Agile development

A- The Waterfall Model

The waterfall model is the classic lifecycle model – it is widely known, understood and commonly used. Is the first process model to be introduced and followed widely in Software Engineering to ensure success of the project.

- The whole **SDLC** is divided into separate process phases, e.g., "***requirement phase***", "***design phase***", etc. (see Figure 6.1). These phases correspond to the four stages of the fundamental software process activities (as explained in ***Lecture-1***).

- A phase takes place in sequence to another.
- Each activity is completed before the next starts.

In theory:

- Each phase produces documents that are: ***Verified*** and ***validated***; Assumed to be ***complete***.

- Each phase depends on the document of the previous stage to proceed → it has to wait for the completion of previous stage.

In practice:

- The phases overlap and feedback to each other (see the feedback arrow in the diagram).

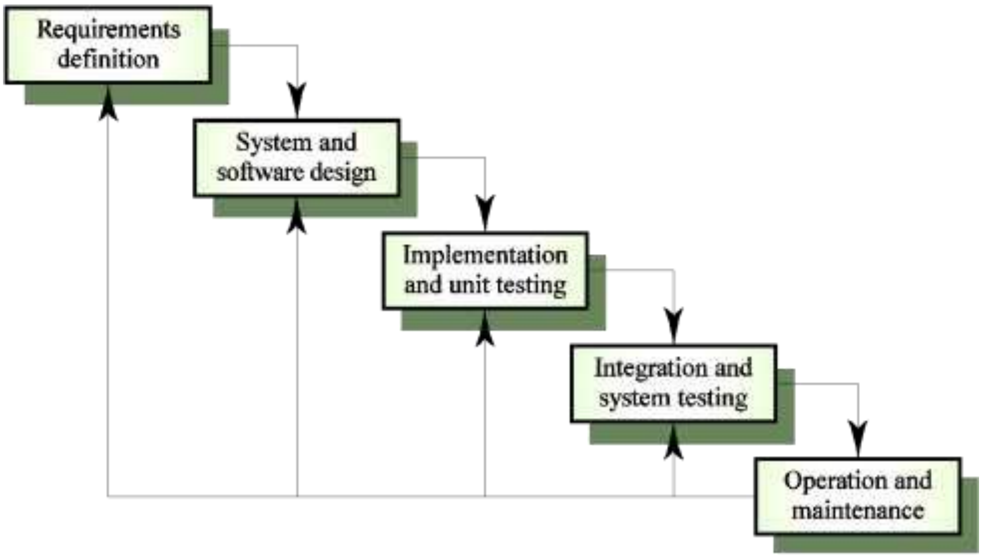

Figure 6.1: The waterfall model lifecycle.

Waterfall Model Phases:

1- Requirements analysis and definition

- All possible requirements of the system to be developed are captured in this phase.
- Requirements then are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.
- Finally, a ***Requirement Specification*** document is created which serves the purpose of guideline for the next phase of the model.

2- System and software design

- System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- The system design specifications serve as input for the next phase of the model.

3- Implementation and unit testing

- On receiving system design documents, the work is divided in modules/units and actual coding is started.
- The system is first developed in small programs called units, which are integrated in the next phase.
- Each unit is developed and tested for its functionality referred to as Unit Testing.
- Unit testing mainly verifies if the modules/units meet their specifications.

4- Integration and system testing

- The units are integrated into a complete system during Integration phase and tested to check if all units/modules coordinate between each other and the system as a whole behaves as per the specifications.
- After successfully testing the software, it is delivered to the customer.

5- Operation and maintenance

- This phase of "The Waterfall Model" is virtually never ending phase (Very long).
- Generally, problems with the system developed (which are not found during the development life cycle) come up after its practical use starts, so the issues related to the system are solved after deployment of the system.
- Not all the problems come in picture directly but they arise time to time and needs to be solved- referred as Maintenance.

Features of a Waterfall Model

1. A waterfall model is easy to follow. It can be implemented for any size project.
2. Process stages can be iterative. Every stage has to be done separately at the right time so you cannot jump stages.
3. Early and frequent validation of software system. Testing is done at every stage.
4. Documentation is produced at every stage of a waterfall model allowing people to understand what has been done.
5. Works well when quality is more important than cost or schedule.

Waterfall Model Advantages:

1. Easy to explain to the users.
2. Structures approach.
3. Stages and activities are well defined.
4. Helps to plan and schedule the project.
5. Verification at each stage ensures early detection of errors/misunderstanding.
6. Each phase has specific deliverables.

Waterfall Model Disadvantages:

1. If requirements may change, the Waterfall model may not work.
2. Very difficult to go back to any stage after it finished.
3. Is only appropriate when the requirements are well-understood; and changes will be fairly limited during the design process.
4. Difficult to estimate time and cost for each stage of the development process.
5. Constant testing of the design is needed.

When to use the Waterfall Model

- Requirements are very well known.
- Product definition is stable.
- Technology is understood.
- New version of an existing product.
- Porting an existing product to a new platform.