



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الانظمة الطبية الذكية كلية العلوم

المحاضرة الخامسة

Software engineering

المادة : Software engineering

المرحلة : الثالثة

اسم الاستاذ: م.د أحمد عدنان المحنا

Software design process

It is the process of implementing software solutions to one or more set of problems. One of the important parts of software design is the software requirements analysis (SRA). It is a part of the software development process that lists specifications used in software engineering. Software design includes both low- level component and algorithm design and high level, architecture design as shown in Figure (5.1).

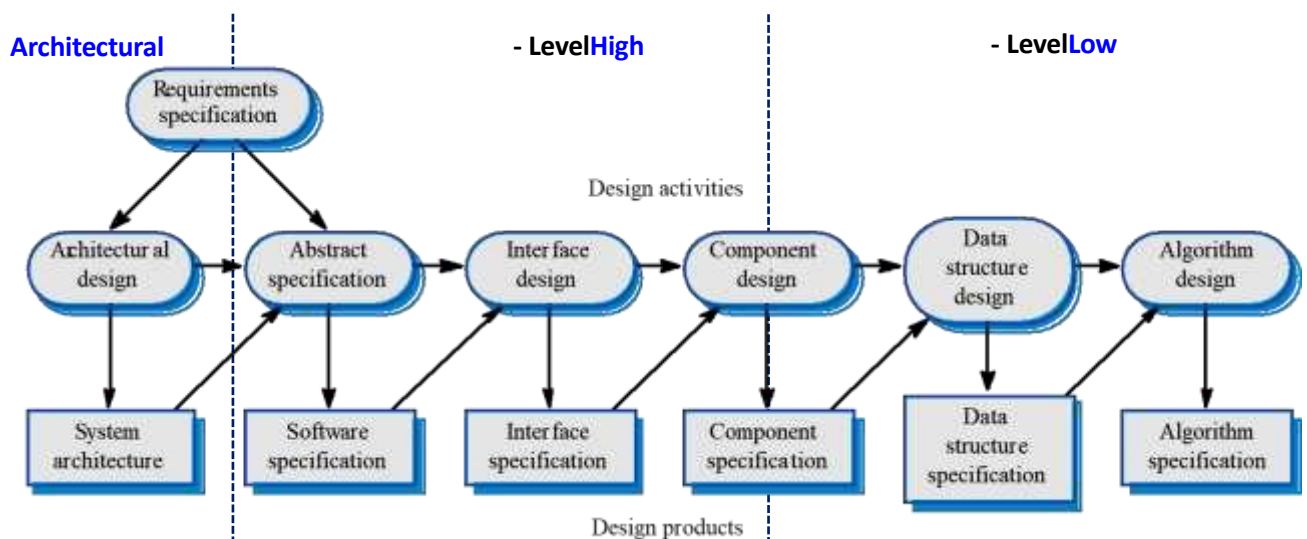


Figure 5.1: The typical Phases of design process.

- **Architectural design:** Identify sub-systems
- **Abstract specification:** Specify sub-systems
- **Interface design:** Describe sub-system interfaces
- **Component design:** Decompose sub-systems into components
- **Data structure design:** Design data structures to hold problem data
- **Algorithm design:** Design algorithms for problem functions

Software design can be considered as creating a solution to a problem in hand with available capabilities. The main difference between **software analysis** and **design** is that the output of a software analysis consists of smaller problems to solve. Also, the analysis should not be very different even if it is designed by different team members or groups.

The design focuses on the capabilities, and there can be multiple designs for the same problem depending on the environment that solution will be hosted. Sometimes the design depends on the environment that it was developed for, whether it is created from reliable frameworks or implemented with suitable design patterns. When designing software, **two important factors to consider are its Security and Usability.**

Requirements and Design

- In principle, requirements should state **what** the system is to do and the design should describe **who** it accomplishes this.
- *In practice, requirements and design are inseparable; because:*
 - 1- System architecture may be designed to structure the requirements.
 - 2- The system may inter-operate with other systems that generate design requirements.
 - 3- The use of a specific design may be a domain requirement.

The Software Requirements Document (SRD)

- SRD is the official statement of what is required of the system developers.
- Should include both the requirements **definition** and **specification**.
- It is not a design document. It should set out what the system is to do rather than how it should accomplish this.

There are six requirements which SRD should satisfy:

- 1- Specify external system behavior.
- 2- Specify implementation constraints.
- 3- Needs to be easy to change.
- 4- Serve as reference tool for maintenance.
- 5- Recode forethought about the life cycle of the system, i.e. predict change.
- 6- Characterize acceptable responses to unexpected events.

There are different users of the SRD ,such as:

1. **System customers:** Specify the requirements and read them to check that they meet their needs. They specify changes to the requirements.
2. **Managers:** Use the requirements document to plan a bid for the system and to plan the System development process.
3. **System engineers:** Use the requirements to understand what system is to be developed.
4. **System test engineers:** Use the requirements to develop validation tests for the system.
5. **System maintenance engineers:** Use the requirements to help understand the system and the relationship between its parts.

The best organization of SRD is as a series of chapters with the detailed specification perhaps presented as an appendix to the document.

A generic structure for SRD is shown in the following Table 5.1.



Al-Mustaqbal University
College of Science
Intelligent Medical System Department

Chapter	Description
Introduction	Describe the system functions and explain how it will work with other systems. It should describe objectives of the organization commissioning the software.
Glossary	Define the technical terms used in the document. No assumptions should be made about the experience of the reader.
System models	Should set out one or more system models showing the relationships between the system components and the system and its environments. These might include object and data-flow models.

Table 5.1: The structure of a software requirement document.



Al-Mustaqbal University
College of Science
Intelligent Medical System Department

Functional Requirements definition	Should be described the services provided for the user, may use natural language, diagrams or other notations that are understandable by customers.
Non-functional requirements definition	Describe the constraints that imposed on the software and the restrictions on the freedom of the designer. This might include details of specific data representation, response time and so on.
System evolution	Should describe the fundamental assumptions on which the System is based and anticipated changes due to HW evolution, Changing user needs, and so on.
Requirements specification	Should describe the functional requirements in more detail. If necessary , further detail may also be added to the non-functional requirements, for example interfaces to other systems may be defined.

➤ **There are three major problems with *Requirements Definitions Document***

(RDD) written in natural language (NL):

- 1. Lake of clarity:** It is very difficult to use language in a precise and unambiguous way without making the document difficult to read.
- 2. Requirements confusion:** Functional requirements, non-functional requirements, system goals and design information may not be clearly distinguished.
- 3. Requirements amalgamation:** Several different requirements may be expressed together as a single requirement.

➤ **There are three major problems with *Requirements Specification Document***

(RSD) written in natural language (NL):

- 1. Ambiguity:** The readers and writers of the requirement must interpret the same words in the same way. Natural language (NL) is inherently ambiguous, so this is very difficult.
- 2. Over-flexibility:** The same thing may be said in a number of different ways in the specification.
- 3. Lack of modularization:** NL structures are inadequate to structure system requirements.



Al-Mustaqbal University
College of Science
Intelligent Medical System Department
