

# REQUIREMENT ENGINEERING

Lecture # 16

Lecture by Engr. Sidra



# OBJECTIVES

- When you have read the chapter you will:
  - understand the concepts of user and system requirements and why these requirements should be written in different ways;
  - understand the differences between functional and nonfunctional software requirements;
  - understand how requirements may be organized in a software requirements document;
  - understand the principal requirements engineering activities of elicitation, analysis and validation, and the relationships between these activities;
  - understand why requirements management is necessary and how it supports other requirements engineering activities



# REQUIREMENTS ENGINEERING

- The process of establishing the services that the customer requires from a system and the constraints under which it operates and is developed.
- The requirements themselves are the descriptions of the system services and constraints that are generated during the requirements engineering process.
- The process of collecting the software requirement from the client then understand, evaluate and document it is called as requirement engineering.
- Requirement engineering constructs a bridge for design and construction.



# WHAT IS A REQUIREMENT?

It may range from a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.



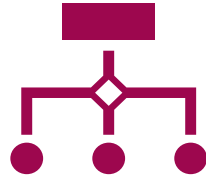
This is inevitable as requirements may serve a dual function:

May be the basis for a bid for a contract - therefore must be open to interpretation;

May be the basis for the contract itself - therefore must be defined in detail



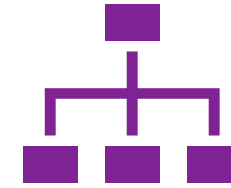
## TYPES OF REQUIREMENTS



### User requirements

Often referred to as user needs, describe what the user does with the system, such as what activities that users must be able to perform.

Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.



### System requirements

These are the building blocks developers use to build the system

A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.





# TYPES OF REQUIREMENTS:

## FUNCTIONAL REQUIREMENTS

- Functional user requirements may be high-level statements of what the system should do.
- Functional system requirements should describe the system services in detail
- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- Describe functionality or system services.
- Depend on the type of software, expected users and the type of system where the software is used.
- Requirements should be both complete and consistent
  - Complete: They should include descriptions of all facilities required
  - Consistent: There should be no conflicts or contradictions in the descriptions of the system facilities



# TYPES OF REQUIREMENTS: NON-FUNCTIONAL REQUIREMENTS

- It specifies criteria that can be used to judge the operation of a system, rather than specific behaviors
- These define system properties e.g. reliability, response time, and storage requirements.
- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Often apply to the system as a whole rather than individual features or services.
- Process requirements may also be specified mandating a particular IDE, programming language or development method.
- They may be more critical than functional requirements. If these are not met, the system may be useless.
- They may be very difficult to state precisely and imprecise requirements may be difficult to verify



# TYPES OF NON FUNCTIONAL REQUIREMENTS

