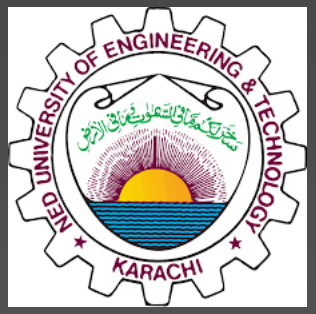


Software Requirements Engineering (SE-208)

Dr. Shehnila Zardari

Introduction

- Requirements form the basis for all software products
- Requirements engineering is the process, which enables us to systematically determine the requirements for a software product

Lecture # 1

Software Requirements

Requirement

- Something required, something wanted or needed
  - Webster's dictionary
- There is a huge difference between *wanted* and *needed* and it should be kept in mind all the time

Software Requirements - 1

- A complete description of *what* the software system will do without describing *how* it will do it is represented by the software requirements

Software Requirements - 2

- Software requirements are complete specification of the desired external behavior of the software system to be built
- They also represent External behavior of the system

Software Requirements - 3

- Software requirements may be:
  - Abstract statements of services and/or constraints
  - Detailed mathematical functions

Software Requirements - 4

- Software requirements may be:
  - Part of the bid of contract
  - The contract itself
  - Part of the technical document, which describes a product

IEEE Definition

- A condition or capability that must be met or possessed by a system…to satisfy a contract, standard, specification, or other formally imposed document
  - IEEE Std 729

Sources of Requirements

- Stakeholders
  - People affected in some way by the system
- Documents
- Existing system
- Domain/business area

Levels of Software Requirements

- Stakeholders describe requirements at different levels of detail
  - *"What versus How"*
  - *"One person's floor is another person's ceiling"*

What Versus How

User needs

Product space

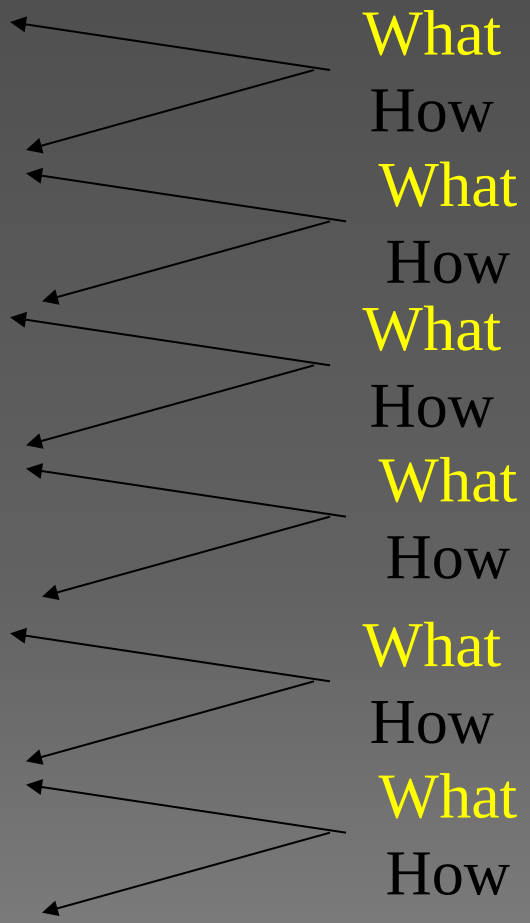
Actual product's behavior

Architecture/data flow

Module specifications

Algorithms

Code



Importance of Software Requirements

- The hardest single part of building a software system is deciding what to build...No other part of the work so cripples the resulting system if done wrong. No other part is difficult to rectify later
  - Fred Brooks

Examples of Requirements - 1

- The system shall maintain records of all payments made to employees on accounts of salaries, bonuses, travel/daily allowances, medical allowances, etc.

Examples of Requirements - 2

- The system shall interface with the central computer to send daily sales and inventory data from every retail store

Examples of Requirements - 3

- The system shall maintain records of all library materials including books, serials, newspapers and magazines, video and audio tapes, reports, collections of transparencies, CD-ROMs, DVDs, etc.

Examples of Requirements - 4

- The system shall allow users to search for an item by title, author, or by International Standard Book Number
- The system's user interface shall be implemented using a web browser

Examples of Requirements - 5

- The system shall support at least twenty transactions per second
- The system facilities which are available to public users shall be demonstrable in ten minutes or less

Kinds of Software Requirements

Kinds of Software Requirements

- Functional requirements
- Non-functional requirements
- Domain requirements
- Inverse requirements
- Design and implementation constraints

Functional Requirements

Functional Requirements - 1

- Statements describing what the system does
- Functionality of the system

Functional Requirements - 2

- Statements of services the system should provide
  - Reaction to particular inputs
  - Behavior in particular situations

Functional Requirements - 3

- Sequencing and parallelism are also captured by functional requirements
- Abnormal behavior is also documented as functional requirements in the form of exception handling

Functional Requirements - 4

- Functional requirements should be complete and consistent
- Customers and developers usually focus all their attention on functional requirements

Functional Requirements Example # 1

- The system shall solve a quadratic equation using the following formula

$$x = (-b \pm \text{sqrt}(b^2 - 4*a*c))/2*a$$

Functional Requirements Example # 2

- The user shall be able to search either the entire database of patients or select a subset from it (admitted patients, or patients with asthma, etc.)

Functional Requirements Example # 3

- The system shall provide appropriate viewers for the user to read documents in the document store

Functional Requirements Example # 4

- Every order shall be allocated a unique identifier (ORDER_ID) which the user shall use to access that order

Functional Requirements Example # 5

- The system shall allow customers to return non-perishable items within fifteen days of the purchase. A customer must present the original sale receipt to return an item

Comments on Examples

- Notice the level of detail in different requirements described above. Some are very detailed compared to others

Comments on Examples

- Notice the ambiguity in the requirement, which uses the term ‘appropriate viewers’
- This requirement does not mention the formats of documents and types of viewers, which can be used

Comments on Examples

- Notice the ambiguity in the requirement for solving the quadratic equation. The requirement does not speak about the possibility when the value of 'a' is zero

$$x = (-b \pm \text{sqrt}(b^2 - 4*a*c))/2*a$$

Comments on Examples

- Incomplete and ambiguous requirements are open to multiple interpretations and assumptions
- This can lead to the development of poor quality, or faulty, software products

Summary

- Requirements form the basis of all software engineering projects
- Functional requirements capture the behavioral aspects/functions of the proposed automated system
- Functional requirements are the backbone of all software products

References

- 'Requirements Engineering: Processes and Techniques' by G. Kotonya and I. Sommerville, John Wiley & Sons, 1998
- Software Requirements: Objects, Functions, and States by A. Davis, PH, 1993
- Software Engineering 6th Edition, by I. Sommerville, 2000
- Software Engineering 5th Edition, by R. Pressman