Software Engineering – CSC 342

Chapter 3

Software Requirements

King Saud University
College of Computer and Information Sciences
Department of Computer Science

Dr. S. HAMMAMI

Objectives

- To introduce the concepts of user and system requirements
- To describe functional and non-functional requirements
- To explain how software requirements may be organised in a requirements document

Outcomes

When you have read the chapter, you will

- Understand the concepts of user requirements
- Understand the concepts of system requirements
- Understand why these requirements should be written in different ways
- Understand the differences between functional and non-functional software requirements
- Understand how requirements may be organized in a software requirements document.

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.

What is a requirement?

Requirements Analysis and Definition

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

- The requirements analysis and definition establish the system's services, constraints and goals by consultation with users. They are then **defined** in a manner that is understandable by both users and development staff.

Requirements define the function of the system FROM THE CLIENT'S VIEWPOINT.

Types of requirement

- **User requirements**
  - Statements in natural language plus diagrams of what services the system is expected to provide and its operational constraints. Written for customers.
- **System requirements**
  - 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 requirement

User Requirement:

Let us assume that we have a word-processing system that does not have a spell checker. In order to be able to sell the product, it is determined that it must have a spell checker. Hence the business requirement could be stated as:

- ***user will be able to correct spelling errors in a document efficiently.***

- **Hence, the Spell checker will be included as a feature in the product.**

Types of requirement

System Requirement:

- **After documenting the user's perspective in the form of user requirements, the system's perspective: what is the functionality provided by the system and how will it help the user to accomplish these tasks. Viewed from this angle, the functional requirement for the same user requirement could be written as follows:**
- ***The spell checker will find and highlight misspelled words. Right clicking a misspelled word, will then display a dialog box with suggested replacements. The user will be allowed to select from the list of suggested replacements. Upon selection it will replace the misspelled word with the selected word. It will also allow the user to make global replacements.***

Definitions and specifications

User requirement definition

Library System (LIBSYS) shall keep track of all data required by copyright licensing agencies.

System requirements specification

- On making a request for a document from LIBSYS, the requestor shall be presented with a form that records details of the user and the request made.

- LIBSYS request forms shall be stored on the system for five years from the date of the request.

- All LIBSYS request forms must be indexed by user, by the name of the material requested and by the supplier of the request.

- LIBSYS shall maintain a log of all requests that have been made to the system.

- For material where authors' lending rights apply, loan details shall be sent monthly to copyright agencies that have registered with LIBSYS

Functional and non-functional requirements

Software system requirements are often classified as functional requirements, non-functional requirements:

- Functional requirements
  - Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

- Non-functional requirements (Quality Requirements)
  - Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.

Functional requirements

- Describe functionality or system services.
- Depend on the type of software, expected users of the software and the general approach taken by the organisation when writing requirements.
- Functional user requirements may be high-level statements of what the system should do but functional system requirements should describe the system services in detail.

Example: The LIBSYS system

Functional requirements may be expressed in a number of way.

Example: LIBSYS used by students and faculty to order books and documents from other libraries.

- A library system that provides a single interface to a number of databases of articles in different libraries.
- Users can search for, download and print these articles for personal study.
- The user shall be able to search either all of the initial set of databases or select a subset from it.
- The system shall provide appropriate viewers for the user to read documents in the document store.

Non-functional requirements

- These define system properties and constraints e.g. reliability, response time and storage requirements.
- They may define constraints on the system such as the capabilities of I/O devices and the data representations used in system interfaces.
- Process requirements may also be specified mandating a particular CASE system, programming language or development method.
- Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless.

Non-functional classifications

- Product requirements
  - Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability, etc.
- Organisational requirements
  - Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.
- External requirements
  - Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.

User requirements

- Should describe functional and non-functional requirements in such a way that they are understandable by system users who don't have detailed technical knowledge.
- User requirements are defined using natural language, tables and diagrams as these can be understood by all users.

Problems with natural language

Various problems can arise when requirements are written in natural language sentences in a text document:

- Lack of clarity
  - Precision is difficult without making the document difficult to read.
- Requirements confusion
  - Functional and non-functional requirements tend to be mixed-up.
- Requirements amalgamation
  - Several different requirements may be expressed together as a single requirement.

Guidelines for writing requirements

To minimise misunderstandings when writing user requirements, It is recommended that you follow some simple guidelines:

- Invent a standard format and use it for all requirements.
- Use language in a consistent way. You should always distinguish between mandatory and desirable requirements.
- Use text highlighting to identify key parts of the requirement.
- Avoid the use of computer jargon.

System requirements

- More detailed specifications of system functions, services and constraints than user requirements.
- They are intended to be a basis for designing the system.
- They add detail and explain how the user requirements should be provided by the system.
- The system requirements should simply describe the external behaviour of the system and its operational constraints.
- They should not be concerned with how the system should be designed or implemented.

Requirements and design

- In principle, requirements should state **what** the system should do and the design should describe **how** it does this.
- In practice, requirements and design are inseparable
  - A system architecture may be designed to structure the requirements;
  - In most cases, systems must interoperate with other existing systems. These constrain the design, and these constraints impose requirements on the new system;

Problems with Natural Language (NL) specification

- Ambiguity
  - The readers and writers of the requirement must interpret the same words in the same way. NL is naturally ambiguous so this is very difficult.
- Over-flexibility
  - The same thing may be said in a number of different ways in the specification.
- Lack of modularisation
  - NL structures are inadequate to structure system requirements.

Because of these problems, requirements specification written in natural language are prone to misunderstandings.

Alternatives to NL specification

Notation	Description
Structured natural language	This approach depends on defining standard forms or templates to express the requirements specification.
Design description languages	This approach uses a language like a programming language but with more abstract features to specify the requirements by defining an operational model of the system. This approach is not now widely used although it can be useful for interface specifications.
Graphical notations	A graphical language, supplemented by text annotations is used to define the functional requirements for the system. An early example of such a graphical language was SADT. Now, use-case descriptions and sequence diagrams are commonly used.
Mathematical specifications	These are notations based on mathematical concepts such as finite-state machines or sets. These unambiguous specifications reduce the arguments between customer and contractor about system functionality. However, most customers don't understand formal specifications and are reluctant to accept it as a system contract.

Structured language specifications

- The freedom of the requirements writer is limited by a predefined template for requirements.
- All requirements are written in a standard way.
- The terminology used in the description may be limited.
- The advantage is that the most of the expressiveness of natural language is maintained but a degree of uniformity is imposed on the specification.
- Structured language notations limit the terminology that can be used and use templates to specify system requirements.

Form-based specifications

- Definition of the function or entity.
- Description of inputs and where they come from.
- Description of outputs and where they go to.
- Indication of other entities required.
- Pre and post conditions (if appropriate).
- The side effects (if any) of the function.

Graphical models

- Graphical models are most useful when you need to show how state changes or where you need to describe a sequence of actions.
- Different graphical models are explained in next chapters.

The requirements document

- The requirements document is the official statement of what is required of the system developers (SRS).
- Should include both a definition of user requirements and a specification of the system requirements.
- It is NOT a design document. As far as possible, it should set of WHAT the system should do rather than HOW it should do it

Key points

- Requirements set out what the system should do and define constraints on its operation and implementation.
- Functional requirements set out services the system should provide.
- Non-functional requirements constrain the system being developed or the development process.
- User requirements are high-level statements of what the system should do. User requirements should be written using natural language, tables and diagrams.

Key points

- System requirements are intended to communicate the functions that the system should provide.
- A software requirements document is an agreed statement of the system requirements.
- The IEEE standard is a useful starting point for defining more detailed specific requirements standards.