

**FACE-to-PHASE:**

**Getting to Know the Phases of Systems Development**

**UNIT 1**

**LESSON 1**

**Requirements**

**Learning Outcomes**

At the end of the lesson, you should be able to:

* Identify correctly and discuss the phases of the Requirements Life Cycle.
* Illustrate the steps of an existing system using a USE CASE Model.
* Create Activity Diagram of a system.

What are requirements?

* Requirements are statements that identify the essential needs of a system in order for it to have value and utility.

Characteristics of Good ReqUIREMENts

1. Describes What, Not How.

2. Atomic. i.e., it should have a single purpose

3. Unique.

4. Documented and Accessible.

5. Identifies Its Owner.

6. Approved. After a requirement has been revised, reviewed, and rewritten, it must be approved by its owner.

7. Traceable. A good requirement is traceable; it should be possible to trace each requirement back to its source.

8. Necessary.

9. Complete.

10. Unambiguous

11. Quantitative and testable

12. Identifies applicable states

14. States Assumptions. All assumptions should be stated.

15. Use of Shall, Should, and Will. A mandatory requirement should be expressed using the word shall (e.g., "The system shall conform to all state laws

16. **Avoids Certain Words**. The words optimize, maximize, and minimize should not be used in stating requirements, because we could never prove that we had achieved them.

**Requirements Life cycle**

**The User**

**Raw Req’ts**

**Organised Req’ts**

**Analysed Req’ts**

**Complete user Req’ts**

**SPECS**

C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0195384.wmf

* **Elicitation Phase**

The starting point of the requirements engineering process is an elicitation process that involves a number of people to ensure consideration of a broad scope of potential ideas and candidate problems

* **Organization Phase**

In this step there is no transformation of the requirements, but simple classification and categorization. For example, requirements may be grouped into functional vs. nonfunctional requirements.

* **Analysis Phase**

This represents a transformation.

* **Prototype Phase**

In this way poorly understood requirements may be tested and perhaps strengthened, corrected, or refined. This activity is often done as a proof of concept and serves to induce feedback from both the stakeholders and engineers.

* **Requirements documentation and specification**

This represents the requirements as the finished product of the stakeholder requirements team. The requirements are compiled into a requirements list or into some equivalent document format. These collected requirements are then transformed into a specification.

* + 1. **Requirements Elicitation, Documentation, and Maintenance**

**Requirements elicitation**

* Requirements determination addresses the gathering and documentingof the true and real requirements for the Information System being developed.
* Requirements is the wants and /or needs of the user within a problem domain.

**Requirements determination questions**

* Who does it?
* What is done?
* Where is it done?
* When is it done
* How is it done
* Why is it done?

**Systems Requirements**

* Characteristics or features that must be included to satisfy business requirements
  + Outputs
  + Inputs
  + Processes
  + Timing
  + Controls
  + Volumes. sizes, and frequencies
* Data/Information collected can be about; people, organization, work and work environment.

**Fact – Finding Methods**

* Sampling (of existing documentation, forms, and databases).
* Research and site visits. (Participation)
* Observation of the work environment.
* Questionnaires.
* Interviews.
* Prototyping.
* JAD/Joint requirements planning (JRP).

**Types of Requirements**

* **User Requirements:** these are statements in Natural language plus diagrams of services the system provides, together with its operational constraints. These can be categorised into 2; functional requirements and non-functional requirements
  + **Functional requirements** 
    - Describe ***what*** the system should do
  + **Non-functional requirements** 
    - Consists of **C*onstraints*** that must be adhered to during development (design and implementation)
    - Remember ‘**Constraints**.’
* **System requirements**
  + What we agree to provide
  + Describes system services
  + Contract between Client and contractor

**Functional requirements**

* + What *inputs* the system should accept
  + What *outputs* the system should produce
  + What data the system should *store* that other systems might use
  + What *computations* the system should perform
  + The *timing and synchronization* of the above

**Non-functional requirements**

* Non-functional requirements are global constraints on a computer system
  + e.g. development costs, operational costs, performance, reliability,
* The challenge of Non-functional requirements:
  + Hard to model
  + Usually stated informally, and so are:
    - often contradictory,
    - difficult to enforce during development
    - difficult to evaluate for the customer prior to delivery
* Define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations.
* Process requirements may also be specified mandating a particular programming language or development method
* Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless.

**Examples of NFR**

* Interface requirements
  + How will the new system interface with its environment?
  + User interfaces and “user-friendliness”
  + Interfaces with other systems
* Performance requirements
  + Time - response time
  + Output - transactions per second
* Security
  + permissible information flows
  + Or who can do what
  + Survivability – e.g. system will need to survive fire natural catastrophes, etc
* Operating requirements
  + Physical constraints (size, weight),
  + Personnel availability & skill level
  + Accessibility for maintenance
  + Environmental conditions
* Lifecycle requirements
  + Maintainability, Enhanciability, Portability, expected market or product lifespan
* limits on development
  + E.g. development time limitations, resource availability and methodological standards.
* Economic requirements
  + E.g. restrictions on immediate and/or long-term costs**.**

**Requirements Documentation**

* There are basically two types of documents realized from the requirements elicitation phase. These include;
  + User Requirements Specification Document
  + System requirements specification Document

**User Requirements Specification –URS/URD**

* The URS document outlines precisely what the User (or customer) is expecting from this system.
* User Requirement Specification may incorporate the functional requirements of the system or may be in a separate document labelled the Functional Requirements Specification - the FRS.

The URD has the following information:

* 1. Functional Requirements
  2. Non-Functional Requirements

**System Requirements Specification Document**

A detailed description of the system services.

* What do we agree to provide?
* A structured document setting out detailed descriptions of the system services.
* Written as a contract between client and contractor.

**TOOLS THAT AID IN DEVELOPING & UNDERSTANDING SYSTEM REQuiremenTS**

* Affinity diagrams
* Force-field analysis
* Ishikawa fishbone (cause-and-effect) diagrams
* Pareto diagrams
* Pugh charts
* Quality function deployment (QFD)

