# Non­­­ - ­­­FUNCTIONAL

**“How the system should behave while performing the operations.”**

* These are the constraints on the services which the system is offering, e.g., Time taken to perform an operation.
* Also known as Quality Attributes.
* For Example:
  + Recoverability, Response Time, Safety, Security, Usability, Reliability etc.
* Explained Example
  + Database should load the data in **2 seconds**.

## Types of Non – Functional Requirements

1. Performance Requirements
2. Interface Requirements
3. Operational Requirements
4. Resource Requirements
5. Verification Requirements
6. Acceptance Requirements
7. Documentation Requirements
8. Security Requirements
9. Portability Requirements
10. Quality Requirements
11. Reliability Requirements
12. Maintainability Requirements
13. Safety Requirements

## Classification of Non – Functional Requirements

1. Process Requirements
2. Product Requirements
3. External Requirements

### Process Requirements

Requirements on development standards and methods which must be followed.

CASE tools which should be used.

The management reports which must be provided.

Example of Process Requirement:

* The system must be developed using the XYZ suite of CASE tools.
* Management reports setting out the effort expended on each identified system component must be produced every two weeks.

### Product Requirements

The desired characteristics that a system must possess.

Some product requirements can be formulated precisely:

1. Performance.
2. Capacity.

Some are difficult to quantify:

1. Usability.

**Example:**

1. The system service X must have an availability of 99%
   1. That mean out of every 100 request of the service 99 must be satisfied.
2. System Y must process a minimum of 8 transactions per second.
   1. Performance Requirement.

### External Requirements

May be placed on both product and the process.

External Requirements are based on:

* + Application domain information
  + Organisational considerations
  + The need for the system to work with other systems
  + Health and safety or data protection regulations

## Nfrs for critical system

1. Reliability
   1. Constrain on run – time behavior of the system.
   2. **Availability:** The system available for service requested by the user.
   3. **Failure Rate:** How often does the system fail to provide the service required by the user.
2. Performance
   1. Constrain the speed of operation of a system
   2. Response Requirements, Timing Requirements.
3. Security
   1. These are included to ensure that Unauthorised access to the system and its data is not allowed.
   2. For example: The access permission for system data may only be changed by system data administrator.

All system data must be backed up every 24 hours and the backup copies stored in a secure location which is not in the same building as the system

1. Usability
   1. Well-structured user manuals, informative error handling messages, help facilities, and consistent interfaces enhance usability.
2. Safety

# SRS

* A set of precisely stated properties or constraints which a software system must satisfy.
* A software requirements document establishes boundaries on the solution space of the problem of developing a useful software system.
* A software requirements document allows a design to be validated - if the constraints and properties specified in the document are satisfied by the software design, then that design is an acceptable solution to the problem.

## Six requirements For srs

* + it should specify only external system behavior,
  + it should specify constraints on the implementation,
  + it should be easy to change,
  + it should serve as a reference tool for system maintainers,
  + it should record forethought about the life cycle of the system, and
  + it should characterize acceptable responses to undesired events.

## System Model

* This conceptual model is a very high-level view of the system in which the major user services are identified and their relationships documented.
* It is necessary to establish an explicit, precisely defined system model at an early stage and to use this model to understand the system.

## Functional Description

* The functional system requirements are those system services which are expected by the user of the system.
* In principle, the functional requirements should be both complete and consistent.
* Completeness means that all user-required services are specified.
* Consistency means that no one requirement definition should contradict any other.

### Expressing Functional requirements

There are 3 ways of expressing functional requirements

1. In natural language.
2. In a structured language.
3. In a formal specification language with rigorously defined syntax and semantics.

## 10 Tips for getting useful information from users

1. Include the real end users.
2. Don’t ask users to do your job.
3. Overcome resistance to change.
4. Use data to settle differences of opinion.
5. Leave room for users to change their minds.
6. Keep an open mind.
7. Live in their camp for a while.
8. Get some communications help.
9. Don't rely on memory or general impressions.
10. Don't rush to write things off as too difficult.

## Characteristics of srs

1. Unambiguous.
2. Complete.
3. Consistent.
4. Verifiable.
5. Traceable.
6. Modifiable.
7. Maintainable.

## 7 sins of Specification

1. Noise

Noise elements are those which introduce no new elements and can obscure the text. That means referencing the same concept in different ways 🡪 This is a bad thing in specification.

1. Silence

The existence of a feature of the problem that is not covered by any element of the text.

1. Over-specification

Overspecification in requirements can be annoyingly close to silence. The reader is told too much about the *solution* while he or she is desperately trying to grasp the *problem* and figure out – by themselves – features not covered by the text.

1. Ambiguity

An element that makes it possible to interpret a feature of the problem in at least two different ways.

1. Contradiction

Two or more elements define a feature of the system in an incompatible way.

1. Forward Reference

Implicit forward references (uses of a concept that come before the proper definition of the concept without particular warning to the reader) are the problem. This is why a glossary is so important.

1. Wishful Thinking

An element that defines a feature of the problem in such a way that a candidate solution cannot realistically be validated with respect to this feature.

## Rules of order for specifying software Requirements

1. Use an industry standard for:

Completeness check, and establishment of consistency between documents

1. Use models for:

Functional Relationships, Data flow, Data structure and to express complete requirements.

1. Limit the structures of paragraphs to individual sentences:

To increase the traceability and modifiability of each requirement and to increase the ability to check for completeness.

1. Use simple sentences:

This is to increase the verifiability (testability) of each requirement.

1. Limits the use of verbs and objects to a specific set:

This improves consistency and reduces ambiguity.

1. Use words common to the end user:

In order to increase user understanding of the requirements.

1. Limit the verbs and objects to actions and items that are visible external to the product:

# Elicitation

1. Application Domain Understanding
2. Problem Understanding
3. Business Understanding
4. Understandi006Eg the needs and constraints of system stakeholders

## Elicitation Stages

1. Objective setting

An outline description of the system to be solved, why the system is necessary and the constraints on the system.

1. Background Knowledge acquisition

Background knowledge of the organization where the system is to be installed.

1. Knowledge organization

The information/knowledge collected must be organized in some possible fashion.

1. Stakeholder Requirements collection

Stakeholder should be engaged to discover their requirements.

## Analysis Check

1. Necessity Checking
   1. No useless requirement should be included.
2. Consistency and Completeness checking.
3. Feasibility Checking
   1. The requirements must be with in the budget.

## Requirements Negotiation

1. **Requirements discussion**

Requirements which are highlighted as problem are discussed with the Stakeholders.

1. **Requirements Prioritization**

Disputed requirements are prioritized to help decision making.

1. **Requirements Agreement**

Finalization of requirements.

## Elicitation Techiniques

1. Interviews
2. Scenarios
3. Requirements reuse
   1. Taking the requirements which have been developed for one system and using them for another.
   2. Saves time and effort.
4. Prototyping
   1. Throw – Away
   2. Evolutionary

## Requirement Analysis

The goal is to discover problems, incompleteness, and inconsistency in the elicited requirements.

### (Check slide no 29) and Chapter Summary

## Requirements Negotiation

1. When there are many stakeholders, disagreements about requirements are inevitable.
2. Requirements negotiation is the process of discussing the conflicts with all the stakeholders and reaching a compromise that all stakeholders agree too.

# Requirements Management

#### Some problems due to changing requirements

1. Unmatched and outdated requirements specification causing confusion and unnecessary rework.
2. Time spent for coding, writing test cases and documentation for the requirements that no longer exist.

#### Requirement Management Activities

Requirements management includes all activities intended to maintain the integrity and accuracy of expected requirements.

1. Manage changes to agreed requirements.
2. Manage changes to baseline.
3. Keep project plans synchronized with requirements
4. Manage relationships between requirements

#### Requirement change Factors

* **Requirements errors, conflicts, and inconsistencies**
  + May be detected at any phase (when requirements are analyzed, specified, validated, or implemented).
* **Evolving customer/user knowledge of the system**
  + When the requirements are developed, customers/users simultaneously develop a better understanding of what they really need.
* **Technical, schedule, or cost problems**
  + Difficult to plan and know everything in advance.
  + We may have to revisit the list of requirements and adapt it to the current situation.
* **Changing Customer Needs.**

## Requirements volatility

Requirements change while they are being elicited, analyzed, specified, and validated and after the system has gone into service.

* Stable requirements are concerned with the essence of a system and its application domain
  + Derived from the client’s principal business activities or the domain model
  + They change more slowly than volatile requirements
  + E.g., a hospital will always have doctors, nurses, patients…
* Volatile requirements are specific to the instantiation of the system in a particular environment for a particular customer at a particular time
  + E.g., in a hospital, we can think of requirements related to the policies of the government health system

#### Types of Volatile Requirements

1. **Mutable Requirements**

These are requirements which change because of changes to the environment in which the system is operating.

1. **Emergent Requirements**

Requirements that emerge as the customer's understanding of the system develops during the system development. The design process may reveal new emergent requirements.

1. **Consequential Requirements**

These are requirements which are based on assumptions about how the system will be used.

Once the system is in place, some of these assumptions will be wrong.

1. **Compatibility Requirements**

These are requirements which depend on other equipment, technology, or processes.

## Expectations of Requirement Management

1. Identification of each individual requirement.
2. Traceability from Highest level to Implementation.
3. Impact assessments of proposed changes
4. Controlled and Consistent access to current project information.
5. Change Control.

## Identification of Requirements

It is essential for requirements to have a unique identification.

#### Requirement Identification Techniques

1. **Dynamic Renumbering** 
   1. As you reorganise your document and add new requirements, the system keeps track of the cross references and automatically renumbers your requirements depending on its chapter, section, and position within the section
2. **Database record Identification**
   1. When a requirement is identified, it is entered in a requirements database and a database record identifier is assigned which is then used for all subsequent references to the requirement
3. **Symbolic Identification** 
   1. Requirements can be identified by assigning them a symbolic name, which is associated with requirement itself.

#### Requirements Status

Help manage requirement lifecycle.

Example of a set of statuses:

1. **Proposed**:
2. **Approved**
3. **Rejected**
4. **Implemented**
5. **Verified**
6. **Deleted**

#### Version Control

This is another important aspect of requirements management.

* Every version of requirement needs to be uniquely identified and every version must be available to all team members.
* A requirement document must be updated which should include: Changes, Dates, Whom, and Why.
* Some version control tools should be used to store and manage revision history.

# Traceability

The ability to follow the life of a requirement in both, forward and backward directions.

* From its origin to its deployment and subsequent use.

A software requirement specification is traceable if origin of each of its requirements is clear and it facilitates the referencing of each requirement in future development.

# Requirements Management Planning

## Requirements Management Planning Success Factors

1. Organizational Commitment
2. Recognizing the Value of Requirements Management Planning
3. Stakeholder Engagement and Collaboration
4. Integration with Project Management Activities

## Requirements Management Planning Activities

1. Stakeholder Analysis and Engagement

Analysis of people, groups, or organizations that may affect, be affected, or have an interest in decision activity or outcome of the project.

1. Requirements Management Planning Initiation
2. Develop the Requirements Management Plan
3. Launch the Requirements Management Plan

### Stakeholders Analysis and Engagement

1. Generate or refine a stakeholder register.
2. Group stakeholders.
3. Manage Stakeholders register.

# Verification and Validation

## Validation Objectives

Checks a document for:

1. Completeness and consistency
2. Conformance to standards
3. Requirements conflicts
4. Technical errors
5. Ambiguous requirements

## Analysis and Validation

#### Analysis

Have we got the right requirements?

#### Validation

Have we got the requirements right?

## Validation Inputs and Output

#### Inputs

1. **Requirements Document**

Complete version of elicited requirements document, not an unfinished draft.

1. **Organizational Knowledge**

Knowledge of the organization which is often used to judge the realism of the requirements.

1. **Organizational Standards**

Local standards for the organization of the requirements document.

#### Outputs

1. **List Problems**

List of discovered problems in the document.

1. **Agreed Actions**

List of agreed responses to the discovered problems in the document.

## Requirements review

A group of people read and analyze the requirements, look for problems, discuss their opinions and agree on action to address those problems.

### Review Activities

#### Plan review

The review team is selected and a time and place for the review meeting is chosen.

#### Distribute documents

The requirements document is distributed to the review team members

#### Prepare for review

Individual reviewers read the requirements to find conflicts, omissions, inconsistencies, deviations from standards and other problems.

#### Hold review meeting

Individual comments and problems are discussed and a set of actions to address the problems is agreed.

#### Follow-up actions

The chair of the review checks that the agreed actions have been carried out.

#### Revise document

The requirements document is revised to reflect the agreed actions. At this stage, it may be accepted, or it may be re-reviewed

## pROBLEM aCTIONS

1. Requirements Clarification
2. Missing Information
3. Requirements conflict
4. Unrealistic Requirements

## Review Checklist

1. Understandability
2. Redundancy
3. Completeness
4. Ambiguity
5. Consistency
6. Traceability

## Checklist Questions

1. Is each requirement uniquely identified?
2. Are specialized terms defined in the glossary?
3. Does a requirement stand on its own or do you have to examine other requirements to understand what it means?
4. Do individual requirements use the terms consistently?
5. Is the same service requested in different requirements? Are there any contradictions in these requests?
6. If a requirement refers to some other facilities, are these described elsewhere in the document?
7. Are related requirements grouped together? If not, do they refer to each other?