# Software Requirement Specification

# What is Requirements Analysis?

Requirements Analysis is the **process of understanding the customer needs and expectations from a proposed system** or application and is a well-defined stage in the Software Development Life Cycle model.

# Why is Requirements Analysis necessary?

insufficien

- Inadequate attention to Software Requirements Analysis at the beginning of a project is the most common cause for critically vulnerable projects that often do not deliver even on the basic tasks for which they were designed.
- Requirements Analysis Processes as a prerequisite to successful projects that align with the client's business goals and meet the project's requirement specifications.

# Steps in the Requirements Analysis Process

**I. Fix system boundaries**

This initial step helps in identifying how the new application integrates with the business processes, how it fits into the larger picture and what its scope and limitations will be.

**II. Identify the customer**

In more recent times there has been a focus on identifying who the ‘users’ or ‘customers’ of an application are.

gathering

# III. Requirements elicitation

- Information is gathered from the multiple stakeholders identified. The Requirements Analyst draws out from each of these groups what their requirements from the application are and what they expect the application to accomplish.

# PROBLEMS FACED IN REQUIREMENTS ELICITATION

unclear

- Ambiguous understanding of processes
- Inconsistency within a single process by multiple users
- Insufficient input from stakeholders
- Conflicting stakeholder interests
- Changes in requirements after project has begun

# Tools used in Requirements Elicitation

Traditional methods of Requirements Elicitation included

- stakeholder interviews and focus group studies.
- Other methods like flowcharting of business processes
- and the use of existing documentation like user manuals, organizational charts, process models and systems or process specifications,
- on-site analysis, interviews with end-users,
- Market research and competitor analysis were also used extensively in Requirements Elicitation.

## IV. Requirements Analysis Process

- Once all stakeholder requirements have been gathered, a structured analysis of these can be done after modeling the requirements.
- Some of the Software Requirements Analysis techniques used are requirements animation, automated reasoning, knowledge-based critiquing

# V. Requirements Specification

- Requirements, once elicited, modeled and analyzed should be documented in clear, unambiguous terms. A written requirements document is critical so that its circulation is possible among all stakeholders including the client, user-groups, the development and testing teams.

# VI. Requirements Management

organizing, checking, and updating the list of what the system must do

- Requirements management practices guarantee that all system requirements are stated unambiguously, that omissions and errors are corrected and that evolving specifications can be incorporated later in the project lifecycle.

# Types of Requirements

Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management:

## Customer Requirements

- Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints and measures of effectiveness and suitability.

## Functional Requirements

- Functional requirements explain what has to be done by identifying the necessary task, action or activity that must be accomplished.

## Performance Requirements

- The extent to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness or readiness.

## Design Requirements

- The “build to,” “code to,” and “buy to” requirements for products and “how to execute” requirements for processes expressed in technical data packages and technical manuals.

## Derived Requirements

- Requirements that are implied or transformed from higher-level requirement. For example, a requirement for long range or high speed may result in a design requirement for low weight.

# What is a Software Requirements Specification?

- An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

# A well-designed, well-written SRS accomplishes four major goals:

- **It provides feedback to the customer.**
- **It decomposes the problem into component parts.**
- **It serves as an input to the design specification.**
- **It serves as a product validation check.**

# WHAT ARE THE BENEFITS OF A GREAT SRS?

The IEEE 830 standard defines the benefits of a good SRS:

- **Establish the basis for agreement between the customers and the suppliers on what the software product is to do.**
- **Reduce the development effort.**
- **Provide a basis for estimating costs and schedules.**
- **Provide a baseline for validation and verification.**
- **Serve as a basis for enhancement.**

# What Kind of Information Should an SRS Include?

- Interfaces
- Functional Capabilities
- Performance Levels
- Data Structures/Elements
- Safety
- Reliability
- Security/Privacy
- Quality
- Constraints and Limitations