GANPAT UNIVERSITY

U. V. PATEL COLLEGE OF ENGINEERING

2CEIT502 SOFTWARE ENGINEERING

UNIT 4

REQUIREMENT ANALYSIS AND SPECIFICATION OR REQUIREMENT ENGINEERING

Prepared by: Prof. Ravi Raval (Asst. Prof in C.E Dept. , UVPCE)

Unit 4: Requirements Analysis and Specification

Unit 4: Requirements Analysis and Specification

Requirement Engineering:

- The broad spectrum of tasks and techniques that lead to an understanding of requirements is called *requirements engineering*.
- Requirements engineering builds a bridge to design and construction. But where does the bridge originate?
  - One could argue that it begins at the feet of the project stakeholders (e.g., managers, customers, end users), where business need is defined, user scenarios are described, functions and features are delineated, and project constraints are identified.
  - Others might suggest that it begins with a broader system definition, where software is but one component of the larger system domain.

Unit 4: Requirements Analysis and Specification

Requirement Engineering:

- Requirements engineering provides the appropriate mechanism for understanding
  - what the customer wants
  - analyzing need
  - assessing feasibility
  - negotiating a reasonable solution,
  - specifying the solution unambiguously,
  - validating the specification,
  - and managing the requirements as they are transformed into an operational system.

- It encompasses seven distinct tasks:

1) inception,
2) elicitation,
3) elaboration,
4) negotiation,
5) specification,
6) validation, and
7) management

Unit 4: Requirements Analysis and Specification

<u>Requirement Engineering:</u>

(1) Inception(આરંભ, શરૂઆત): How does a software project get started?

- Is there a single event that becomes the catalyst for a new computer-based system or product, OR
- Does the need evolve over time?
- In some cases, a casual conversation is all that is needed to precipitate a major software engineering effort.
- But in general, most projects begin when a business need is identified or a potential new market or service is discovered.
- Business mangers
- Marketing people
- Product manager

<u>Stakeholders</u>

- Do feasibility analysis
- Define business case
- Identify depth and breadth of the market

Unit 4: Requirements Analysis and Specification

Requirement Engineering:

(2) Elicitation(સ્પષ્ટતા): Ask customer users and others what the objectives for the product are:

- What is to be done?
- How solution fits to business?

- Problems you may encounter during elicitation:
  - **Problem of Scope:** ill-defined boundaries or unnecessary technical details specified.
  - **Problems of Understanding:** When customers/users are not sure what is needed, have a poor understanding of the capabilities and limitations of their computing environment, don't have a full understanding of the problem domain, have trouble communicating needs to the system engineer, omit information that is believed to be "obvious," specify requirements that conflict with the needs of other customers / users, or specify requirements that are ambiguous or untestable.
  - **Problems of volatility:** requirements changes over time.

Unit 4: Requirements Analysis and Specification

<u>Requirement Engineering:</u>

(3) **Elaboration:** expand and refine the information received during inception and elicitation.

- Focuses on software function, behavior and information
- Develop scenarios – how the user will interact with the system.
- Develop class diagrams

(4) **Negotiation:** Customers/users ask for more than can be achieved, given limited business resources

- It's also relatively common for different customers or users to propose conflicting requirements, arguing that their version is "essential for our special needs."
- You need to negotiate the requirements in order to:
  - Rank requirements
  - Discuss conflicts in priority
  - Assess their cost and risk.

Unit 4: Requirements Analysis and Specification

Requirement Engineering:

(5) Specification: A specification can be a written document, a set of graphical models, a formal mathematical model, a collection of usage scenarios, a prototype, or any combination of these.

Standard Template of SRS :

Table of Contents (Index)

Unit 4: Requirements Analysis and Specification

<u>**Standard Template of SRS :**</u>

4. Operating environment
5. Design and implementation constraints
6. User documents
7. Assumptions and dependencies

3. System Features
    1. System features 1
    2. System features 2 (and so on)
4. External Interface requirements
    1. User Interfaces
    2. Hardware interfaces
    3. Software Interfaces
    4. Communication interfaces
5. Other non-functional requirements
    1. Performance requirement
    2. Safety requirements
    3. Security requirements
    4. Software quality attributes
6. Other requirements

Appendix A: Glossary of terms

Appendix B: Analysis Models

Appendix C: List of Issues

Unit 4: Requirements Analysis and Specification

Requirement Engineering:

(6) Validation:

- All requirements stated unambiguously?
- All inconsistencies, omissions and error detected / corrected?
- Work product confirms the standard?

- Not check and verify that:
  - Errors in content / interpretation
  - Areas where clarification still requires
  - Missing info and / or inconsistencies
  - Conflicting requirements / Unrealistic requirements

(7) Requirements management: is a set of activities that helps the project team to identify, control, and track requirements and changes to requirements anytime as the project proceeds.

Unit 4: Requirements Analysis and Specification

Requirement Elicitation:

- Also called "Requirements gathering"
- Who will do this task? → System Analyst
- **System Analyst:** The engineers who gathers and analyse customer requirements and then write the document are known as system analyst.
- Tough task to gather requirements from large no. of people and document them and then to understand clearly.
- Get help if available currently working model is exist.
  - Obtain input/output data formats and operation process details.
- Visit customer site and carry-out questionnary survey, task analysis, scenario analysis, and form analysis.

But still the question remains…How to gather requirements?

Unit 4: Requirements Analysis and Specification

Requirement Elicitation:

How to gather requirements?

1. Study Existing documents:
   - Study before visit to know the context
   - Basic purpose and the stakeholders
2. Interviews: organize interviews with all types of users:
   1. Understand requirements and draft a document.
   2. Circulate it among all types of user to check and verify their requirements.
   3. Refine the document for comments and feedback received.
3. Task Analysis: The users view a software as a box, that provides a set of services, A service is “task”. Ex: “Issue a book” in library management service.
4. Scenario analysis: A task can have multiple scenarios.

   Ex. Scenarios of “Issue a book”
   - Book is successfully issued. Book issue slip is printed.
   - The book is reserved. Cannot be issued.
   - Member has already issued MAX books earlier.
5. Form Analysis: Identify input and output data form.

Unit 4: Requirements Analysis and Specification

<u>Requirement Analysis:</u>

The main purpose of the requirements analysis activity is to analyse the gathered requirements to remove all ambiguities, incompleteness, and inconsistencies from the gathered customer requirements and to obtain a clear understanding of the software to be developed.

Why to analyse requirement? Or outcomes of requirement analysis:

1) What is the problem?
2) Why is it important to solve the problem?
3) What exactly are the data input to the system and what exactly are the data output by the system?
4) What are the possible procedures that need to be followed to solve the problem?
5) What are the likely complexities that might arise while solving the problem?
6) If there are external software or hardware with which the developed software has to interface, then what should be the data interchange formats with the external systems?

Unit 4: Requirements Analysis and Specification

Requirement Analysis:

It was observed that during requirement analysis three types of problems can be raised and to deal with

1) **Anomaly/Ambiguity:** several interpretations of that requirement are possible

- Ex1: Process control application

Stakeholder1: When the temperature becomes high, the heater should be switched off.

But "high" means exactly how much? Proper quantification is required for such words like "good", "high", "low", "bad"

- Ex2: Learning Management System (LMS)

Stakeholder1: during the final grade computation, if any student scores a sufficiently low grade in a semester, then his parents would need to be informed.

Here, It lacks any well defined criteria for "sufficiently low grade"

Unit 4: Requirements Analysis and Specification

Requirement Analysis:

2) **Inconsistency:** Where one requirement contradicts with other.

- Ex1: Process control application

Stakeholder1: The furnace should be switched-off when the temperature of the furnace rises above 500° C.

Stakeholder2: When the temperature of the furnace rises above 500° C, the water shower should be switched-on and the furnace should remain on.

- Ex2: Learning Management System (LMS)

Stakeholder1: a student securing fail grades in three or more subjects must repeat the courses over an entire semester, and he cannot credit any other courses while repeating the courses.

Stakeholder2: expressed the following requirement—there is no provision for any student to repeat a semester; the student should clear the subject by taking it as an extra subject in any later semester.

Unit 4: Requirements Analysis and Specification

Requirement Analysis:

3) **Incompleteness:** Some requirements are overlooked. Or lake of feature felt much later when using the software.

- Ex1: Chemical plant automation

suppose one of the requirements is that if the internal temperature of the reactor exceeds 200° C then an alarm bell must be sounded. However, on an examination of all requirements, it was found that there is no provision for resetting the alarm bell after the temperature has been brought down in any of the requirements.

- Ex2: Learning Management System (LMS)

If the Grade Point Average (GPA) of a student is less than 6 then intimate his performance to his parents via postal mail and email as well.
However, the feature (textbox to enter in a registration form for ex.) where postal address or email address should be entered is missing.

Unit 4: Requirements Analysis and Specification

<u>Requirement Documentation (SRS):</u>

- **Users of the SRS document:**

1. Users, Customers and Marketing personnel
2. Software Developers
3. Test Engineers
4. Project Managers
5. User document writer
6. Maintenance Engineers

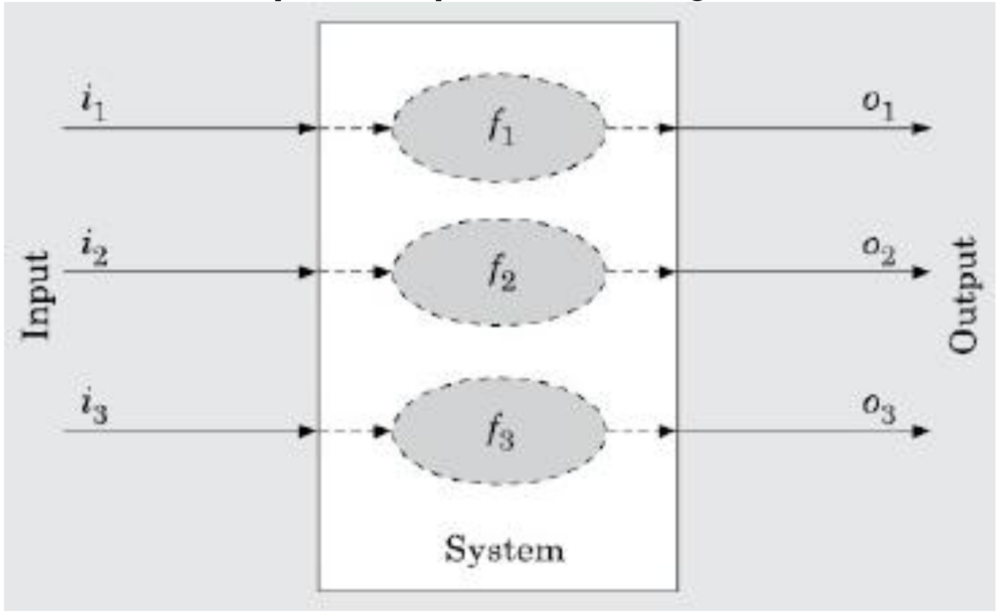
- **Characteristics of a good SRS document**

1. Concise
2. Implementation-independent (black-box view) (see next slide)
3. Traceable
4. Modifiable
5. Identification of undesired events
6. Verifiable

Unit 4: Requirements Analysis and Specification

Requirement Documentation (SRS):

Black box view of system performing a set of functions



Unit 4: Requirements Analysis and Specification

Requirement Documentation (SRS):

- **Why to spend time and resources to develop an SRS?**
  - Forms an agreement between the customer and the developers
  - Reduces future reworks
  - Provides a basis for estimating cost and schedules
  - Provides a baseline for validation and verification
  - Facilitates future extensions
- **Attributes of bad SRS document:** try to avoid this problem while writing SRS.
  - Over-specification: to explain “what to” part don’t get started “how to” explanation.
  - Forward reference: aspect that are discuss much later in SRS document, reduces readability
  - Wishful thinking: description of aspects which would be difficult to implement.
  - Noise: irrelevant details. E.g clerk report for work between 8 to 5 in register_cust function.

Unit 4: Requirements Analysis and Specification

<u>Requirement Documentation (SRS):</u>

- **Important categories of customer requirements**

An SRS document should clearly document the following aspects of a software:

1) Functional requirements
2) Non-functional requirements
   - — Design and implementation constraints
   - — External interfaces required
   - — Other non-functional requirements
3) Goals of implementation.

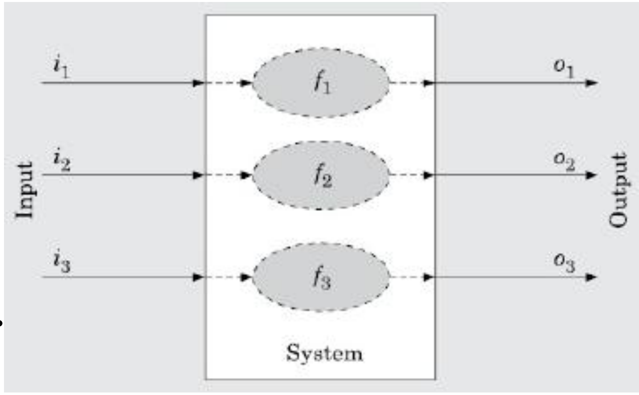
Unit 4: Requirements Analysis and Specification

Requirement Documentation (SRS):

- **Important categories of customer requirements**

1) **Functional requirements:** The functional requirements capture the functionalities required by the users from the system.

- A software is set of offering functions $\{f_i\}$ to the user.
- $f : I \rightarrow O$. meaning that a function transforms an element (i_i) in the input domain (I) to a value (o_i) in the output (O).
- Each function f*i* of the system can be considered as reading certain data i*i*, and then transforming a set of input data (i_i) to the corresponding set of output data (o_i).



Unit 4: Requirements Analysis and Specification

Requirement Documentation (SRS):

- **Important categories of customer requirements**

2) **Non-functional requirements:**

- The non-functional requirements are non-negotiable obligations that must be supported by the software.
- The non-functional requirements capture those requirements of the customer that cannot be expressed as functions (i.e., accepting input data and producing output data).
- Non-functional requirements usually address aspects concerning external interfaces, user interfaces, maintainability, portability, usability, maximum number of concurrent users, timing, and throughput (transactions per second, etc.)

Unit 4: Requirements Analysis and Specification

Requirement Documentation (SRS):

- **Important categories of customer requirements**

2) **Non-Functional requirements:**

a) **Design and implementation constraints:** some items or issues which imposes limits on developers.

Some Examples of constraints:

- corporate or regulatory policies that needs to be honoured;
- hardware limitations;
- interfaces with other applications;
- specific technologies, tools, and databases to be used;
- specific communications protocols to be used;
- security considerations;
- design conventions or programming standards to be followed, etc.

b) **External interfaces required:**

Example of interfaces are:

- hardware, software and communication interfaces,
- user interfaces,
- report formats, etc.

The description may include:

- sample screen images,
- any GUI standards or style guides that are to be followed
- screen layout constraints
- standard buttons and functions (e.g., help) that will appear on every screen
- keyboard shortcuts
- error message display standards,

c) **Other non-functional requirements:** performance, reliability, accuracy, security issues

Unit 4: Requirements Analysis and Specification

Requirement Documentation (SRS):

- **Important categories of customer requirements**

3) **Goal of Implementation:**

- This section offers some general suggestions regarding software to be developed. The develops are free to follow or unfollow those suggestions.
- A goal, in contrast to the functional and non-functional requirements, is not checked by the customer for conformance at the time of acceptance testing.
- This section might contains information like:
  - issues such as easier revisions to the system functionalities that may be required in the future,
  - easier support for new devices to be supported in the future,
  - Reusability issues, etc.