

Analyzing Requirements (Cont'd)



- In addition, you:
 - Need to analyze the identified requirements to understand their intent fully.
 - Must ensure that there is no ambiguity and lack of clarity.
 - Should analyze the requirements for completeness and contradictions.

Requirements can be apparent or hidden. They may be:

- Explicitly stated
 - Left unsaid
-
- Remember, what the customer says may not always be the exact requirement. You have to deduce the requirement from what is said.

Defining the Problem Domain

- Problem domain refers to the area or set of circumstances, in which the problem exists and for which you need to provide a solution.
- The domain is:
 - Limited by the goals and constraints expressed by the problem owner.
 - An area that you need to analyze and explore to obtain a solution that will operate in it.

Sample Problem Problem.



Recording Requirements



- As you analyze the requirements, you start documenting these in the form of requirement models.
- Three crucial modes of recording requirements are:
 - Vision document
 - Software requirements specification
 - Use case descriptions and diagrams

Requirements are the foundation of the entire development process and are referred to throughout. Therefore, they should be:

- Complete
- Consistent
- Accurate

Validating Requirements

- Once the requirements are recorded, you need to validate them to ensure that they are well-defined.
- The factors that qualify requirements to be well-defined are shown in the following figure.



8

Correct: Correctness refers to keeping the specification up to date when you find things that are not correct.

Unambiguous: An SRS is unambiguous if every requirement stated in it has only one interpretation.

Complete: This should be all that is needed by the software designers to design the software.

Consistent: The SRS should be consistent within itself and consistent to its reference documents.

Verifiable: Do not place requirements such as 'It should provide the user a fast response' or, 'the system should never crash'. Instead, provide a quantitative requirement such as 'Every key stroke should provide a user response within 100 milliseconds'.

Ranked for importance: Sometimes, a new system has requirements that are marketing wish lists and may not be achievable. It is useful to provide this information in the SRS, but rank it in order of importance for a feasibility check.

Modifiable: Having the same requirement in more than one place may not be wrong, but it tends to make the document not maintainable.

Traceable: In most organizations, it is sometimes useful to connect the requirements in the SRS to a higher-level document. Requirements traceability is concerned with documenting the life of a requirement and to provide bidirectional

traceability between various associated requirements. It enables users to find the origin of each requirement and track every change that was made to that requirement. For this purpose, it may be necessary to document every change made to a requirement.

Validating Requirements (Cont'd)



- To validate requirements, you must:
 - Go through the requirements specifications iteratively
 - Identify violation of the preceding criteria for well-defined requirements
- Testers will also write test cases from the requirements and use them to validate the requirements.

Requirements Elicitation Techniques



- Requirements elicitation is the first step in the requirements capturing process.
- During this step, an analyst can use a combination of the techniques to bring out “what” a software system must do according to the customer.
- Some of the commonly used elicitation techniques are:

**Background
Investigation**

Workshops

Interviewing

Questionnaires

Observation

Elicit Conclusion



- As analysts start eliciting requirements by using one or more of the techniques, they:
 - Gather information
 - Analyze it to specify software requirements for the system
- The software requirements are then recorded in various forms by the development team.

Ways of Recording Requirements



- As you elicit and analyze requirements, you need to start recording them to ensure that they can be used for OOAD and managed effectively across the OO development process.
- A number of formats are used for recording requirements, two most common and useful ones are:
 - Vision statement
 - Software requirements specification document

Vision Statement

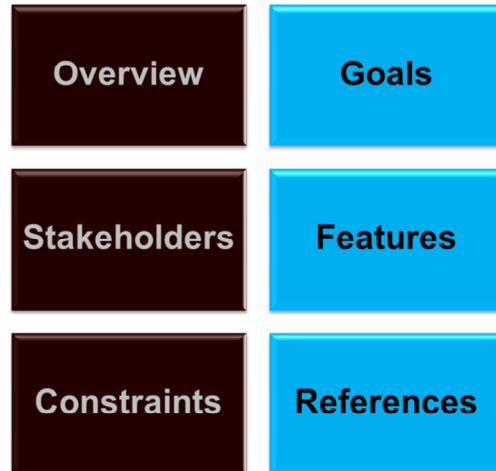


- A vision statement is a document that:
 - Provides an overview of the expectations of the stakeholders from the system
 - Is normally written during the initial elicitation sessions with the customer
 - Lists the top-level features of the system
 - Acts as a scope definition document for the:
 - Development team
 - Customer

As more detailed documents and models are developed for the system through the analysis and design phases, a vision document helps the developers remain focused on the basic and critical needs of the system.

Vision Statement (Cont'd)

- The main contents of a vision statement are:



14

The main contents of a vision statement are:

Overview: This is a summarized description of context (such as client's environment), problem, and proposed solution.

Goals: This can include one or more goals defining what the stakeholders are trying to achieve through the software. The core component of a vision statement is this set of goals of the software system. The goals describe what the developers are trying to develop, for whom, and why. Sometimes the goals also include some marketing spiel about how the product will improve the lives of its users, especially over another competitor's product. However, the focus of the vision statement within the development team is to use the goals to be reminded of the sole purpose of the development process. Developers often refer to it when they begin to lose focus during analysis and design or begin to deviate from the scope.

Stakeholders: This is a list of people with some interest in the software, such as different types of end-users. These stakeholders help you identify actors when you start creating use cases to record requirements in detail.

Features: This is a list of the top-level functionality of the software system. You can refer to this top-level list to start creating your use cases.

Constraints: This is a list of top-level non-functional requirements of the software system, such as availability and performance. It may also contain any known assumptions or dependencies.

References: As more detailed software requirements are written, you can add references to them at the end of the vision document for better management of requirements.

Exercise



- **Write the Vision Statement for ...**

15

Banking?

Amazon?

Software Requirements Specification Document



- The software requirements specification (SRS) document is a detailed description of:
 - Requirements
 - Constraints
 - Dependencies
 - Assumptions related to the software



- An SRS document:
 - Contains functional requirements described in the form of use cases on the basis of which you further analyze and model your system
 - Contains non-functional and domain requirements that need to be considered during design and/or implementation
 - Is the basis for all subsequent models and documentation
 - Is used as the base document for preparing testing and validation plans

Modeling Requirements



- Recording requirements properly is important to ensure that all the members of the development team are aware of and can understand the problem that they need to solve. For this purpose, usually:
 - Functional requirements are modeled using use cases.
 - Non-functional requirements are typically described by using natural language for ease of comprehension and understanding.

Use case

- *Buy something*
 - The Requestor initiates a request and sends it to her or his Approver, who completes the request for submission and sends it to the Buyer. The Buyer finds the best vendor, initiates PO with Vendor.
 - At any time prior to receiving goods, Requestor can change or cancel the request. Canceling it removes it from any active processing.

