Group-3

Team Members: Varunraj Rachakatla (Team Lead), Nikith Kumar Bandaru, Srikanth Boppudi, Mahesh Gudla, Sai Krishna Koduri, Uday Singamala

Investigation: Requirements Engineering

## What is requirements engineering?

The process of establishing, recording, and upholding requirements for a system is known as requirements engineering. To make sure that the finished product satisfies the requirements and expectations of its stakeholders, it is an essential stage in the development of software or systems. The main objectives are Gathering Requirements, Validation, Management, Traceability, Prioritization.

## In an agile/iterative software development setting, will you undergo this process once or multiple times?

Requirements engineering is usually not a one-time event in an agile/iterative software development environment, but rather is done repeatedly over the course of the development lifecycle. In contrast to conventional waterfall models, which collect requirements up front and organize them into phases, agile methodologies welcome flexibility and change. This is how requirements engineering functions in an iterative and agile environment such as Initial Requirements Gathering, Regular Stakeholder Involvement, Backlog Management, Iterative Delivery, Retrospectives.

## What are the different requirements engineering activities? Who takes part in those activities

A number of crucial tasks engage in requirements engineering, all of which help to successfully define, record, and manage requirements. The primary undertakings comprise:

Elicitation:

Goal: Compile data regarding stakeholder expectations and needs.

Participants include end users, customers, project managers, product owners, and business analysts.

Evaluation and Discussion:

Goal: Examine and improve requirements obtained, addressing discrepancies or inadequacies.

Participants include stakeholders, the development team, business analysts, and system analysts.

Details:

Goal: Clearly and concisely describe requirements in documentation.

Participants include technical writers, stakeholders, system architects, and business analysts.

Verification:

Goal: Verify that the requirements are accurate, comprehensive, and consistent.

Participants include business analysts, the quality assurance (QA) team, and stakeholders.

Validation and Examination:

Goal: Verify that the deployed system satisfies the required specifications.

Participants include the development team, the QA team, and testing experts.

Supervisory:

Goals: Manage versions, keep track of requirements changes, and maintain scope control.

Participants include business analysts, project managers, and the change control board.

Trackability:

Goal: Create and preserve a traceable relationship between development artifacts and requirements.

Participants include the development team, project managers, and business analysts.

Setting priorities:

Goal: Rank requirements according to significance and effect.

Project managers, stakeholders, and product owners are involved.

Making prototypes:

Goal: Create a functioning prototype or model to assist stakeholders in clarifying and visualizing requirements.

Developers, designers, and stakeholders are involved parties.

Review and Approval:

Goals: Gather input, discuss requirements with interested parties, and get approval.

Participants include business analysts, project managers, and stakeholders.

Record-keeping:

Goal: Ensure that requirements are thoroughly and currently documented.

Project managers, technical writers, and business analysts are among the participants.

Interaction:

Goal: Promote efficient communication between stakeholders and team members.

Participants: Facilitators of communication, stakeholders, and every member of the project team.

## Define each of the participants in the above activities (example: developer, stakeholder)

Business Analyst:

Function: Examines and comprehends business requirements, converts them into requirements, and collaborates with stakeholders to guarantee that the developed solution and business goals are in line.

Product Owner:

Function: Speaks for customers or other interested parties, ranks feature in the product backlog according to importance, and collaborates closely with the development team to make sure the product satisfies business goals.

Project Manager:

Function: In charge of organizing, conducting, and wrapping up projects. oversees the scheduling, allocation of resources, and compliance of project objectives with corporate goals.

Developer:

Function: Develops, codes, and puts into use software solutions in accordance with the given specifications. Collaborates with other members of the team to deliver the product.

Tester/Quality Assurance Expert:

Function: Verifies that the product satisfies the requirements by testing and validating it. finds and reports errors, works with the development team to resolve problems.

System Architect:

Function: Creates the system's general architecture and structure. makes certain that all of the system's parts are in harmony with one another and meet all of the requirements.

technical writer:

Function: Produces written materials such as system and user manuals and other documentation to assist stakeholders in comprehending and utilizing the product.

Board for Change Control:

Function: An organization in charge of examining, endorsing, or disapproving suggestions for modifying the project's scope, including modifying its requirements. ensures that modifications meet project objectives.

stakeholder:

A person or organization with a stake in the project's success plays this role. Customers, end users, investors, and anybody else having an impact on the project fall under this category.

Development team:

Role: Comprises developers, testers, and other specialists responsible for implementing and evaluating the solution. Collaborates closely to deliver increments of the product.

## For each of the roles in the above two questions, identify who fills those roles (for example, your team is the developer(s), etc.). Remember that one person/group can play multiple roles!

Roles and Assignments:

1. Business Analyst:
   * Assigned to: Nikith Kumar Bandaru
2. Product Owner:
   * Assigned to: Srikanth Boppudi
3. Project Manager:
   * Assigned to: Mahesh Gudla
4. Developer:
   * Assigned to: Sai Krishna Koduri, Varunraj Rachakatla
5. Tester/Quality Assurance Expert:
   * Assigned to: Uday Singamala
6. System Architect:
   * Assigned to: Varunraj Rachakatla
7. Technical Writer:
   * Assigned to: Uday Singamala
8. Board for Change Control:
   * Assigned to: Varunraj Rachakatla, Nikith Kumar Bandaru, Srikanth Boppudi, Mahesh Gudla, Uday Singamala, Varunraj Rachakatla
9. Stakeholder:
   * Assigned to: [Nikith Kumar Bandaru, Srikanth Boppudi, Mahesh Gudla, Sai Krishna Koduri, Uday Singamala, Varunraj Rachakatla, Client, Northwest Missouri State University]
10. Development Team:
    * Comprises: Developers (Varunraj Rachakatla, Nikith Kumar Bandaru, Srikanth Boppudi, Testers (Mahesh Gudla, Uday Singamala, Varunraj Rachakatla) and other specialists.

## Define functional requirement.

A functional requirement is a precise and comprehensive statement of the features, capabilities, or behavior that a product, software, or system needs to have in order to satisfy its users or other stakeholders. Functional requirements are essential for directing the design, development, and testing phases of a system because they specify the functionality or operations that the system is expected to perform. These specifications outline the interactions between the system and its users as well as what the system is expected to do in specific scenarios.

Characteristics of functional requirements include Specificity, Measurability, User- Centric, Objective

## Define non-functional requirement.

A non-functional requirement is a kind of requirement that outlines characteristics or attributes that define how the system should operate rather than criteria that are related to the functionality or features of the system. Performance, dependability, security, usability, and other elements that enhance the system's overall efficacy and quality are all covered by non-functional requirements. These specifications are necessary to guarantee that the system satisfies user needs and adheres to certain standards.

Key characteristics of non-functional requirements include Quality Attributes, Constraints, Measurability, Global Characteristics

## What project artifacts (documents, etc.) should result from requirements engineering?

Throughout the software development life cycle, requirements engineering activities produce a variety of project artifacts that are used as communication and documentation tools. The particular artifacts might change depending on the project's size and complexity as well as the methodology (waterfall, agile, etc.) that was employed. Typical project artifacts from requirements engineering include the following:

Requirements Document:

Goal: Detailed document outlining every requirement, both functional and non-functional.

Contents: System requirements, business rules, use cases, user stories, and any other pertinent information.

User Stories:

Goal: The goal is to provide concise, end-user-focused explanations of system functionality with an emphasis on agile development.

Use Cases:

Goal: Providing in-depth explanations of how users or systems, or actors, interact with the system to achieve particular objectives.

Actors, scenes, pre- and post-conditions, and the sequence of events are among the contents.

Specification of System Requirements (SRS):

Goal: A thorough manual outlining the features, limitations, and interfaces of the system.

Contents: Data model, system architecture, functional and non-functional requirements, and additional technical information.

Traceability Matrix:

The goal is to create and preserve a chain of custody for requirements in relation to other project components.

Contents: Hyperlinks to test cases, design documents, requirements, and other project elements.

Model:

Goal: A graphic depiction of the functionality or user interface of the system.

Contents: Interactive prototypes, mock-ups, or wireframes that aid in helping stakeholders see the finished product.

Test cases:

Goal: Describes how each requirement will be put to the test to make sure it has been applied correctly.

Contents: Test data, test scenarios, and anticipated outcomes.

Requests for Changes:

Documents suggested requirements modifications and their effects on the project.

Contents: Change description, justification, impact assessment, and status of approval.

Backlog (Agile):

Goal: A dynamic list of user stories or features that are prioritized for implementation in future iterations.

Contents: A prioritized to-do list that is frequently maintained using Jira or Trello.

Minutes of meetings and correspondence:

Documents agreements, discussions, and decisions reached during meetings pertaining to requirements.

Minutes of meetings, emails, and other correspondence are included in the contents.

## Define requirements elicitation.

The process of methodically obtaining, identifying, and recording requirements from stakeholders in order to comprehend their needs, expectations, and limitations with regard to a project, software, or system is known as requirements elicitation. This stage of the requirement’s engineering process is crucial because it establishes the parameters for defining the goals and behavior of the system. Ensuring that all parties involved in the development process have a mutual understanding is the aim of requirements elicitation.

Key aspects of requirements elicitation include:

Stakeholder Involvement, Communication, Techniques and Methods, Understanding Context, Documenting Requirements, Iterative Process, Conflict Resolution, Prioritization

## What is the sequence of steps one should take during requirements elicitation? Which step is potentially the hardest?

Depending on the size, scope, and development methodology of the project, there can be variations in the order in which the requirements are elicited. Nonetheless, the following is a general flow of actions that can be taken when gathering requirements:

Determine the Stakeholders:

List and identify all pertinent parties with an interest in or involvement in the system, such as customers, business analysts, end users, subject matter experts, and other pertinent parties.

Specify the goals and parameters:

Establish the project's scope and clearly state the goals of the requirements elicitation process. Recognize the objectives and the constraints that the system must work within.

Create an elucidation plan:

Create an elicitation plan that details the objectives, parties involved, methods to be applied, and a timetable for gathering requirements.

Perform Background Study:

assemble basic data regarding the sector, field of business, systems in place, and any applicable laws or guidelines. This background information aids in context comprehension.

Select Elicitation Methods:

Choose the best elicitation methods for the job based on its specifics. Interviews, questionnaires, workshops, ideation sessions, use cases, and prototypes are examples of common methodologies.

conducting elicitation sessions:

To obtain requirements, actively engage stakeholders using the methods you have chosen. This entails posing queries, urging stakeholders to share their requirements, goals, and limitations, and facilitating dialogues.

Document Requirements:

Take note of and record the requirements that are elicited in an organized manner. Use cases, user stories, system requirements, and any other pertinent artifacts may be included in this documentation.

Verify and Authenticate the Needs:

Verify the requirements with the relevant parties to make sure they are accurate, comprehensive, and clear. Clear up any miscommunications or inconsistencies that surface during the validation procedure.

Prioritize Requirements:

Prioritize requirements with stakeholders according to their significance and influence. This aids in keeping the development process's focus on key features and functionalities.

Repeat and Make Improvements:

Understand that gathering requirements is a process that is iterative. Reexamine earlier steps to update and refine the requirements when added information becomes available or as circumstances change.

Review with interested parties:

Hold review meetings with interested parties to go over the specified requirements and get input. By doing this, it is ensured that everyone is aware of the requirements.

Document Elicitation Outcomes:

After the requirements elicitation process is complete, record the final requirements as well as any modifications you made and the reasoning behind them.

Get the Go-Ahead:

Get official consent from important parties indicating that they concur with the requirements that have been elicited. With this approval, the development phase is officially underway.

Moving Forward with Requirements Analysis:

Transfer the requirements that were elicited to the requirements analysis phase so that they can be refined, examined more closely, and turned into comprehensive development specifications.

The step that is often considered potentially the hardest is:

Conducting Elicitation Sessions:

To gather requirements, this step entails interacting with stakeholders using a variety of approaches. Diverse viewpoints, communication obstacles, identifying implicit requirements, handling change resistance, and weighing stakeholder input can all present challenges. In this step, having adept facilitation skills and navigating intricate interpersonal dynamics are essential.

## What are key things you should try to identify when working with the client? What kinds of questions should you ask?

During requirements elicitation, it is important to collaborate with a client to identify essential information that will help to clarify the project's objectives, scope, and constraints. To extract pertinent information, you must ask the right questions. The following are important items to look for and questions to ask:

Goals and Objectives for Business:

Determine the main aims and objectives of the company.

Questions such as: "What are the primary business objectives for this project?”, "What goals are you aiming to achieve with the new system?"

Project Scope:

Make the project's parameters and restrictions clear.

Questions such as: "What functionalities or features do you envision for the system?”, "Are there any specific areas or processes to be included or excluded?"

User Roles and Stakeholders:

Determine which user roles and important stakeholders are involved.

Questions such as: "Who are the primary stakeholders for this project?”, "Can you describe the various user roles and their responsibilities?"

Present Problems and Difficulties:

Recognize the difficulties, problems, or inefficiencies that exist now.

Questions such as: "What issues are you currently facing with the existing system/process?”, "Are there any pain points that you would like the new system to address?"

User Requirements & Expectations:

Examine end users' requirements and expectations.

Questions such as "What are the most critical needs of the end-users?”, "What features would bring the most value to them?"

Success Standards:

Establish the criteria for measuring success.

Questions such as: "How will you measure the success of the project?”, "What key performance indicators (KPIs) are important to you?"

Restrictions and Restraints:

Determine any restrictions or limitations, including financial constraints, deadlines, and legal requirements.

Concerns: "Are there any budget constraints for this project?”, "Are there specific regulatory requirements that need to be considered?"

Requirements for Technology and Integration:

Examine any integration needs or preferences you may have with technology.

Questions such as: "Do you have any preferences regarding the technology stack?”, "Are there existing systems that the new system needs to integrate with?"

Requirements for Security and Compliance:

Recognize the implications for compliance and security.

Questions such as: "What security measures are crucial for the system?”, "Are there any compliance standards that need to be adhered to?"

Prospective Scalability and Development:

Find out if plans exist for system evolution or scalability in the future.

Questions such as: "Do you anticipate the need for the system to scale in the future?”, "Are there upcoming changes in the business that the system should accommodate?"

Process of Making Decisions:

Recognize how decisions are made within the company.

Issues such as "How are decisions related to the project typically made?”, "Who will be involved in the decision-making process?"

Needs for Assistance and Training:

Determine the end users' training and support needs.

Issues such as "What level of training will end-users need?”, "Are there specific support expectations for maintaining the system?"

Feedback and Preferences for Communication:

Find out the client's preferred method of receiving updates and feedback.

Questions such as "What is your preferred mode of communication?”, "How often would you like project updates?"

Additionally, include any information you think may be helpful as you gather requirements while meeting with the client.

Project Overview:

Briefly describe the project's goals, scope, and objectives.

Key Features and Priorities:

Discuss and prioritize key features and functionalities.

Constraints and Assumptions:

Clarify any constraints or assumptions.

Success Criteria:

Define how success will be measured.

Regulatory and Compliance Requirements:

Identify any regulatory or compliance requirements.

Integration Points:

Discuss existing systems or third-party services.

Data Management and Security:

Gather information about data handling and security measures.

Feedback and Approval Process:

Establish a clear process for client feedback and approvals.

Training and Support:

Discuss training or support requirements for end-users.

Timeline and Milestones:

Outline the project timeline and key milestones.

Communication Plan:

Define a communication plan, including regular update meetings.

Statement of Work:

This Statement of Work (SOW) outlines the scope, deliverables, and key terms of the project. It serves as a comprehensive guide for all project stakeholders, detailing the expectations and responsibilities of each party involved.

References:

Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies:

Fernández, D. M., Wagner, S., Lochmann, K., Baumann, A., & de Carne, H. (2016). *Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies*. <https://doi.org/10.1016/j.infsof.2011.09.001>

Empirical Investigation of Critical Requirements Engineering Practices for Global Software Development:

Habib Ullah Khan, Mahmood Niazi, Mohamed El-Attar, Naveed Ikram, Siffat Ullah Khan, & Asif Qumer Gill. (2021). *Empirical Investigation of Critical Requirements Engineering Practices for Global Software Development. IEEE Access, 9, 93593–93613.* <https://doi.org/10.1109/ACCESS.2021.3092679>

Agile Software Requirements Engineering Challenges-Solutions—A Conceptual Framework from Systematic Literature Review:

Hoy, Z., & Xu, M. (2023). *Agile Software Requirements Engineering Challenges-Solutions—A Conceptual Framework from Systematic Literature Review. Information (2078-2489), 14(6), 322.* <https://doi.org/10.3390/info14060322>

A systematic literature review of requirements engineering education:

Daun, M., Grubb, A. M., Stenkova, V., & Tenbergen, B. (2023). *A systematic literature review of requirements engineering education. Requirements Engineering, 28(2), 145–175.* <https://doi.org/10.1007/s00766-022-00381-9>