# Investigation: Requirements Engineering - Modify

**Group – 04**

**Wiki Page Link:** [**https://github.com/bhargavvarmasagi/GDP-Team04/wiki/Investigation:-Requirements-Engineering-%E2%80%90-Modify**](https://github.com/bhargavvarmasagi/GDP-Team04/wiki/Investigation:-Requirements-Engineering-%E2%80%90-Modify)

**1. What is Requirements Engineering?**

Requirements Engineering is the process of figuring out what a system or software needs to do. It involves talking to the people who will use or benefit from the software (called stakeholders), writing down their needs, and making sure that everyone understands what is required before the actual development starts. This helps reduce misunderstandings and ensures the final product does what it’s supposed to.[1]

**2. In an Agile/Iterative Software Development Setting, Will You Undergo This Process Once or Multiple Times?**

In Agile or iterative development, you won’t just gather requirements once. Since Agile is all about being flexible and making changes as needed, you will go through this process many times. With each sprint or iteration, new requirements might come up or old ones might change based on feedback from users or stakeholders.[1]

**3. What are the Different Requirements Engineering Activities? Who Takes Part in Those Activities?**

Here are the main activities in Requirements Engineering:

Elicitation: This means gathering all the requirements by talking to people who will use the software or are affected by it.

Who is involved: Stakeholders, Business Analysts, Product Owners.

Specification: Writing down the requirements in a way that everyone can understand.

Who is involved: Business Analysts, Developers.

Validation: Making sure that the requirements are correct and that the system can actually be built based on them.

Who is involved: Developers, QA Engineers, Stakeholders.

Management: Handling any changes to the requirements as the project moves along.

Who is involved: Project Managers, Product Owners, Business Analysts.[2]

**4. Define Each of the Participants in the Above Activities**

Developer: The person who writes the code and builds the system.

Stakeholder: Anyone who has an interest in the system, like the client, end-users, or company managers.

Business Analyst: The person who talks to the stakeholders and writes down what they want in a way that developers can use.

Product Owner: The person responsible for making sure the team builds what the client needs.

Project Manager: The person who makes sure the project stays on track, on time, and within budget.

QA Engineer: The person who tests the system to make sure it works as expected and meets the requirements.[3]

**5. Roles in the Above Two Questions and Who Fills Them In your group, these roles might be filled by the following people:**

Project Manager: Venkata Srinivasa Bhargav Varma Sagi

Front end developer: Praneeth Shanigaram

Back end developer: Varsha Pusupati

Tester: Akanksha Mamidaga

Business Analyst: Padma Sri Garikimukku

Client: Mark Chai

6. Define Functional Requirement

A functional requirement describes the actions, behaviors, and tasks that a system should carry out in order to achieve its goals. It outlines how a user and the system interact, with a particular attention on inputs, processes, and outputs. These specifications are essential for directing the process of development and making sure that the system achieves its goals. [4]

**7. Define Non-Functional Requirement.**

Functionality of a system, with significance on features like speed, security, and usability, is described by a non-functional requirement. It states not what the system should do, but rather how well it should perform particular tasks, such as loading quickly or ensuring the security of user data. These specifications guarantee the system's ease of use and dependability. [4]

**8. What Project Artifacts Should Result from Requirements Engineering?**

Here are some of the documents or artifacts you might create:

Requirements Document: A document that lists all the requirements for the system.

User Stories: Short descriptions of features from the user’s perspective.

Diagrams: Visual representations, like flowcharts or use case diagrams, showing how the system works.[5]

**9. Define Requirements Elicitation**

Requirements elicitation is just a fancy way of saying “gathering requirements.” It involves talking to the client, users, and stakeholders to figure out what the software needs to do.[5]

**10. What Are the Sequence of Steps During Requirements Elicitation? Which Step is Potentially the Hardest?**

Here are the steps for gathering requirements:

Identify stakeholders: Find out who will use the system and who will give input.

Collect information: Ask questions, interview stakeholders, or observe how they currently do things.

Document the requirements: Write down everything you gathered so that everyone can understand it.

Validate the requirements: Make sure the requirements are correct and possible to achieve.

The hardest step is often Collecting information because stakeholders might not always know exactly what they want, or they might have different opinions.[5]

**11. Key Things to Identify When Working with the Client. What Kinds of Questions Should You Ask?**

When meeting with the client, it’s important to figure out:

The goals: What do they want to achieve with this project?

The key features: What are the most important things the system needs to do?

User expectations: How do they expect the system to behave?

Any limitations: Are there any constraints, like time, budget, or existing systems that need to be considered?

Questions to ask might include:

What’s the main problem you want to solve?

Who will use this system, and what do they need?

Are there any specific features you can’t live without?

What are the software changes will you do in the future?

**References :**

1. Lucia, Andrea De, and Abdallah Qusef. (2010). “Requirements Engineering in Agile Software Development.” *Joursisnal of Emerging Technologies in Web Intelligence 2 (3).* <https://doi.org/10.4304/jetwi.2.3.212-220>.
2. Pandey, S., Pokharel, S. and Hassan, R. (2018) Towards “Cyber-Physical Requirement Engineering Elicitation Tool Support.” *2018 World Automation Congress (WAC), Stevenson, WA,* 1-5.<https://doi.org/10.23919/WAC.2018.8430399>
3. Van Lamsweerde, A. (2009). “Requirements Engineering: From System Goals to UML Models to Software Specifications.” Wiley.<https://www.academia.edu/33019629/Requirements_Engineering_From_System_Goals_to_UML_Models_to_Software_Specifications_Fundamentals_of_RE_Chapter_4_Requirements_Specification_and_Documentation>
4. Chung, L., Nixon, B.A., Yu, E. and Mylopoulos, J. (2012) “Nonfunctional Requirements inSoftwareEngineering.”.<https://www.academia.edu/62009607/Non_functional_requirements_in_software_engineering>
5. **Zowghi, D., & Coulin, C. (2005). “Requirements Elicitation: A Survey of Techniques.”** *In* Advances in Requirements Engineering, 1-20. <https://doi.org/10.1007/978-1-4020-3642-4_1>