**What is Requirements Engineering?**

A systematic and strict approach to the definition, creation and verification of requirements for a software system is known as requirements engineering. In order to guarantee the effective creation of a software product, the requirements engineering process entails a number of tasks that help in understanding, recording and managing the demands of stakeholders.

**Steps in Requirements Engineering Process**

The requirements engineering process is an iterative process that involves several steps, including:

**Requirements Elicitation**

This is the process of gathering information about the needs and expectations of stakeholders for the software system. This step involves interviews, surveys, focus groups, and other techniques to gather information from stakeholders.

**Requirements Analysis**

This step involves analyzing the information gathered in the requirements elicitation step to identify the high-level goals and objectives of the software system. It also involves identifying any constraints or limitations that may affect the development of the software system.

**Requirements Specification**

This step involves documenting the requirements identified in the analysis step in a clear, consistent, and unambiguous manner. This step also involves prioritizing and grouping the requirements into manageable chunks.

**Requirements Validation**

This step involves checking that the requirements are complete, consistent, and accurate. It also involves checking that the requirements are testable and that they meet the needs and expectations of stakeholders.

**Requirements Management**

This step involves managing the requirements throughout the software development life cycle, including tracking and controlling changes, and ensuring that the requirements are still valid and relevant.

**Requirement Engineering**

The Requirements Engineering process is a critical step in the software development life cycle as it helps to ensure that the software system being developed meets the needs and expectations of stakeholders, and that it is developed on time, within budget, and to the required quality.

Requirement Engineering is the process of defining, documenting and maintaining the requirements. It is a process of gathering and defining service provided by the system. it is the disciplined application of proven principle , methods ,tools and notations to describe a proposed system’s intended behaviour and its associated constraints.

**Tools Involved in Requirement Engineering**

* observation report
* Questionnaire ( survey , poll )
* Use cases
* User stories
* Requirement workshop
* Mind mapping
* Role playing
* Prototyping

**Requirements Engineering Process Consists of the Following Main Activities**

* Requirements elicitation
* Requirements specification
* Requirements verification and validation
* Requirements management

**1. Requirements Elicitation**

 It is related to the various ways used to gain knowledge about the project domain and requirements. The various sources of domain knowledge include customers, business manuals, the existing software of same type, standards and other stakeholders of the project. The techniques used for requirements elicitation include interviews, brainstorming, task analysis, Delphi technique, prototyping, etc. Some of these are discussed [here.](https://www.geeksforgeeks.org/software-engineering-requirements-elicitation/) Elicitation does not produce formal models of the requirements understood. Instead, it widens the domain knowledge of the analyst and thus helps in providing input to the next stage.

Requirements elicitation is the process of gathering information about the needs and expectations of stakeholders for a software system. This is the first step in the requirements engineering process and it is critical to the success of the software development project. The goal of this step is to understand the problem that the software system is intended to solve, and the needs and expectations of the stakeholders who will use the system.

There are several techniques that can be used to elicit requirements, including:

* **Interviews**: These are one-on-one conversations with stakeholders to gather information about their needs and expectations.
* **Surveys**: These are questionnaires that are distributed to stakeholders to gather information about their needs and expectations.
* **Focus Groups**: These are small groups of stakeholders who are brought together to discuss their needs and expectations for the software system.
* **Observation**: This technique involves observing the stakeholders in their work environment to gather information about their needs and expectations.
* **Prototyping**: This technique involves creating a working model of the software system, which can be used to gather feedback from stakeholders and to validate requirements.

It’s important to document, organize and prioritize the requirements obtained from all these techniques to ensure that they are complete, consistent and accurate.

**2. Requirements Specification**

This activity is used to produce formal software requirement models. All the requirements including the functional as well as the non-functional requirements and the constraints are specified by these models in totality. During specification, more knowledge about the problem may be required which can again trigger the elicitation process. The models used at this stage include ER diagrams, data flow diagrams(DFDs), function decomposition diagrams(FDDs), data dictionaries, etc.

Requirements specification is the process of documenting the requirements identified in the analysis step in a clear, consistent, and unambiguous manner. This step also involves prioritizing and grouping the requirements into manageable chunks.

The goal of this step is to create a clear and comprehensive document that describes the requirements for the software system. This document should be understandable by both the development team and the stakeholders.

**There are several types of requirements that are commonly specified in this step, including**

1. **Functional Requirements:** These describe what the software system should do. They specify the functionality that the system must provide, such as input validation, data storage, and user interface.
2. **Non-Functional Requirements**: These describe how well the software system should do it. They specify the quality attributes of the system, such as performance, reliability, usability, and security.
3. **Constraints:** These describe any limitations or restrictions that must be considered when developing the software system.
4. **Acceptance Criteria**: These describe the conditions that must be met for the software system to be considered complete and ready for release.

In order to make the requirements specification clear, the requirements should be written in a natural language and use simple terms, avoiding technical jargon, and using a consistent format throughout the document. It is also important to use diagrams, models, and other visual aids to help communicate the requirements effectively.

Once the requirements are specified, they must be reviewed and validated by the stakeholders and development team to ensure that they are complete, consistent, and accurate.

**3. Requirements Verification and Validation**

**Verification:** It refers to the set of tasks that ensures that the software correctly implements a specific function.

**Validation:** It refers to a different set of tasks that ensures that the software that has been built is traceable to customer requirements. If requirements are not validated, errors in the requirement definitions would propagate to the successive stages resulting in a lot of modification and rework. The main steps for this process include:

1. The requirements should be consistent with all the other requirements i.e no two requirements should conflict with each other.
2. The requirements should be complete in every sense.
3. The requirements should be practically achievable.

Reviews, buddy checks, making test cases, etc. are some of the methods used for this.

Requirements verification and validation (V&V) is the process of checking that the requirements for a software system are complete, consistent, and accurate, and that they meet the needs and expectations of the stakeholders. The goal of V&V is to ensure that the software system being developed meets the requirements and that it is developed on time, within budget, and to the required quality.

1. Verification is the process of checking that the requirements are complete, consistent, and accurate. It involves reviewing the requirements to ensure that they are clear, testable, and free of errors and inconsistencies. This can include reviewing the requirements document, models, and diagrams, and holding meetings and walkthroughs with stakeholders.
2. Validation is the process of checking that the requirements meet the needs and expectations of the stakeholders. It involves testing the requirements to ensure that they are valid and that the software system being developed will meet the needs of the stakeholders. This can include testing the software system through simulation, testing with prototypes, and testing with the final version of the software.
3. V&V is an iterative process that occurs throughout the software development life cycle. It is important to involve stakeholders and the development team in the V&V process to ensure that the requirements are thoroughly reviewed and tested.

It’s important to note that V&V is not a one-time process, but it should be integrated and continue throughout the software development process and even in the maintenance stage.

**4. Requirements Management**

Requirement management is the process of analyzing, documenting, tracking, prioritizing and agreeing on the requirement and controlling the communication to relevant stakeholders. This stage takes care of the changing nature of requirements. It should be ensured that the SRS is as modifiable as possible so as to incorporate changes in requirements specified by the end users at later stages too. Being able to modify the software as per requirements in a systematic and controlled manner is an extremely important part of the requirements engineering process.

Requirements management is the process of managing the requirements throughout the software development life cycle, including tracking and controlling changes, and ensuring that the requirements are still valid and relevant. The goal of requirements management is to ensure that the software system being developed meets the needs and expectations of the stakeholders and that it is developed on time, within budget, and to the required quality.

There are several key activities that are involved in requirements management, including:

1. **Tracking and controlling changes:** This involves monitoring and controlling changes to the requirements throughout the development process, including identifying the source of the change, assessing the impact of the change, and approving or rejecting the change.
2. **Version control**: This involves keeping track of different versions of the requirements document and other related artifacts.
3. **Traceability**: This involves linking the requirements to other elements of the development process, such as design, testing, and validation.
4. **Communication:** This involves ensuring that the requirements are communicated effectively to all stakeholders and that any changes or issues are addressed in a timely manner.
5. **Monitoring and reporting**: This involves monitoring the progress of the development process and reporting on the status of the requirements.

Requirements management is a critical step in the software development life cycle as it helps to ensure that the software system being developed meets the needs and expectations of stakeholders, and that it is developed on time, within budget, and to the required quality. It also helps to prevent scope creep and to ensure that the requirements are aligned with the project goals.

**Advantages and Disadvantages**

The advantages and disadvantages of the requirements engineering process in software engineering include:

**Advantages**

* Helps ensure that the software being developed meets the needs and expectations of the stakeholders
* Can help identify potential issues or problems early in the development process, allowing for adjustments to be made before significant
* Helps ensure that the software is developed in a cost-effective and efficient manner
* Can improve communication and collaboration between the development team and stakeholders
* Helps to ensure that the software system meets the needs of all stakeholders.
* Provides a clear and unambiguous description of the requirements, which helps to reduce misunderstandings and errors.
* Helps to identify potential conflicts and contradictions in the requirements, which can be resolved before the software development process begins.
* Helps to ensure that the software system is delivered on time, within budget, and to the required quality standards.
* Provides a solid foundation for the development process, which helps to reduce the risk of failure.

**Disadvantages**

* Can be time-consuming and costly, particularly if the requirements gathering process is not well-managed
* Can be difficult to ensure that all stakeholders’ needs and expectations are taken into account
* Can be challenging to ensure that the requirements are clear, consistent, and complete
* Changes in requirements can lead to delays and increased costs in the development process.
* As a best practice, Requirements engineering should be flexible, adaptable, and should be aligned with the overall project goals.
* It can be time-consuming and expensive, especially if the requirements are complex.
* It can be difficult to elicit requirements from stakeholders who have different needs and priorities.
* Requirements may change over time, which can result in delays and additional costs.
* There may be conflicts between stakeholders, which can be difficult to resolve.
* It may be challenging to ensure that all stakeholders understand and agree on the requirements.