**Requirements Elicitation and Analysis**

**Definition**

Requirements Elicitation is a crucial process in software engineering that focuses on identifying and understanding the needs and constraints of stakeholders. This involves engaging with users, customers, and suppliers to gather comprehensive system requirements. The process aims to uncover both explicit and implicit needs to ensure that the final product aligns with user expectations, forming a solid foundation for the development stages of the online store.

**Stages include:**

* **Requirements discovery,**
* **Requirements classification and organization,**
* **Requirements prioritization and negotiation,**
* **Requirements specification**

****

 **Requirements Discovery**  
 This stage involves discovering and identifying the fundamental requirements of the project through direct interaction with stakeholders, such as clients or potential users.

* **Project Application**: We start by gathering insights from end-users to understand their needs and expectations from the system. For example, in an "online hardware store" project, users might need a robust product search feature, clear price listings, and detailed specifications.

 **Requirements Classification and Organization**  
Once requirements are collected, they are organized and classified according to categories, such as functional (e.g., core system features) and non-functional (e.g., performance, security), or by priority (essential, optional).

* **Project Application**: Requirements are grouped based on priority for the online store. For instance, a "secure payment system" would be essential, while a "smart product recommendation" feature could be classified as an optional enhancement.

 **Requirements Prioritization and Negotiation**  
During this stage, requirements are prioritized, and negotiations take place with stakeholders to determine what should be included in the initial release versus what can be deferred.

* **Project Application**: Some features may not be feasible in the first phase due to budget or time constraints. For example, we may decide to begin with a basic "product display system" and defer an advanced "product filtering system" for a later phase.

 **Requirements Specification**  
Finally, requirements are documented in detail to ensure that all members of the development team have a clear understanding of what needs to be built.

* **Project Application**: In this phase, detailed specifications are created for each part of the project, such as user interface requirements for displaying product lists, search functions, and cart functionalities in the online store.

**Problems of requirements analysis**

### 1. Unclear Requirements

* **Problem:** Requirements may be vague or not well-defined, leading to misunderstandings between the technical team and stakeholders.
* **Impact:** This results in a system that may not fully meet the customer’s expectations, increasing the likelihood of rework and scope changes later on.

**2. Frequent Changes in Requirements**

* **Problem:** Customers may request changes to requirements during development due to shifts in business needs or a better understanding of their needs.
* **Impact:** Frequent changes can affect the timeline and cost, leading to what is known as “scope creep.”

**3. Difficulty Defining Non-Functional Requirements**

* **Problem:** Non-functional requirements, like performance, security, and usability, are often overlooked or poorly defined.
* **Impact:** This can lead to an inefficient or insecure system, impacting user satisfaction and overall system effectiveness.

**4. Ineffective Communication with Stakeholders**

* **Problem:** Analysts may struggle to communicate effectively with non-technical stakeholders, leading to gaps in understanding.
* **Impact:** This may result in a system that does not align with the client’s or end-user’s expectations.

**5. Lack of Domain Knowledge**

* **Problem:** If analysts are not familiar with the client’s industry, they may find it challenging to accurately identify the requirements.
* **Impact:** This can lead to a system that is poorly aligned with the client's needs and expectations.

**6. Not Engaging All Stakeholders**

* **Problem:** Ignoring input or needs from certain stakeholders can result in missing critical requirements.
* **Impact:** This leads to an incomplete system that may not meet the needs of all user groups.

**7. Resistance to Change from Users**

* **Problem:** Users may resist new systems, preferring the familiarity of the old system.
* **Impact:** User resistance can hinder successful system adoption and usage after deployment.

**8. Difficulty Validating Requirements**

* **Problem:** Some requirements may be difficult to verify or are only partially verifiable.
* **Impact:** This complicates testing to confirm that the system meets requirements during the testing phase.

**9. Conflicting Requirements**

* **Problem:** Requirements may conflict with each other due to differing needs of stakeholders.
* **Impact:** This creates misalignment within the system and often requires adjustments to reconcile competing requirements.

**10. Using Inappropriate Techniques for Gathering Requirements**

* **Problem:** Using unsuitable methods to gather requirements, like interviews when fieldwork might be more appropriate.
* **Impact:** This can lead to incomplete or inaccurate information, affecting the quality of requirements analysis