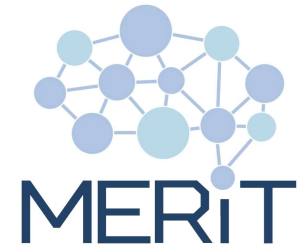


Analysis 1

Eliciting and Specifying
Requirements



Introduction to Analysis



Definition

- The analysis is the process of understanding the system's **intended functionality**.
- It focuses on the **problem space**, not the solution space.
- It describes “**what** the system should do”, without being concerned with “**how** the system should work”.

Definition 2

- The business stakeholders have a **mental representation**, M , of the system to be developed.
- Assume M is the **best possible outcome** that can be achieved at the end of the SDLC.
- The analysis is the process of creating a **formal representation**, F , of the system.
- The goal of the analysis is to **minimize** $\| F - M \|$.



Objectives

- Define **system boundaries**.
- Set **clear expectations** and **scope**.
- Establish a **formal foundation** for the **next phases**.
- Establish a **mutual understanding** among stakeholders.
- Reduces the risk of **costly rework** and **project failure**.



Key Activities

- **Requirements Elicitation:** Discovering, extracting, and clarifying business stakeholder needs using structured techniques such as interviews, workshops, observation, and document reviews.
- **Requirements Specification:** Formally documenting functional and non-functional requirements in a structured format.



Key Activities

- **Requirements Validation:** Checking that documented requirements are accurate, complete, consistent, and aligned with stakeholder intent.
- **Requirements Prioritization:** Assessing and ranking requirements by importance, feasibility, and business value.



Key Activities

- **Requirements Management:** Creating links between requirements and their related design, implementation, and testing elements. Tracking changes, managing versions, and resolving conflicts throughout the system's lifecycle.



Key Deliverables

- **System Requirements Specification (SRS):** Documents all functional and non-functional requirements, serving as a reference for design and testing.
- **Requirements Models:** Visual representations, such as use case diagrams, user stories, activity diagrams, entity-relationship diagrams (ERD), data-flow diagrams (DFD), state diagrams, and class diagrams.



Key Deliverables

- **Requirements Validation Report:** Documents the feedback loop with business stakeholders and the resolution of inconsistencies or gaps.
- **Requirements Prioritization Matrix:** Identifies the priority of requirements based on business value.
- **Requirements Traceability Matrix (RTM):** Maps requirements to design, implementation, and testing for full lifecycle tracking.



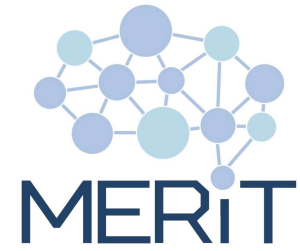
Key Deliverables

- **Requirements Change Log:** Contains a tracked record of changes to requirements over time, including rationale, versioning, and impact notes.



Key Role: Business Analyst

- A Business Analyst plays a key role in **engaging with stakeholders** and is primarily responsible for **carrying out key activities** and **producing key deliverables** during the analysis phase.
- They serve as a **bridge between stakeholders**, speaking both non-technical and technical languages, and providing a balance between business needs and technical constraints.



Requirements Elicitation



Definition

- Requirements elicitation is a systematic process for **discovering, clarifying, and defining** the stakeholders' **needs, expectations, and constraints**.
- It acts as the first step in transforming **vague ideas** into **formal requirements**.



Objectives

- Understand **stakeholder needs**.
- Capture **functional and non-functional requirements**.
- Define the **scope and boundaries** of the **system**.
- Uncover **constraints, assumptions, and dependencies**.



Deliverables

- Interview Notes
- Workshop Output Documents
- Survey/Questionnaire Results
- Elicitation Log
- Preliminary Requirements List
- Prototypes or Mockups
- Informal Use Cases or Scenarios

Elicitation Sources

- Stakeholders
- Work Environment
- Existing Documents
- Existing Systems



Elicitation Techniques

- Conduct **interviews** with stakeholders to gather their input.
- Implement **surveys** to reach large groups or when direct interaction with stakeholders is limited.
- Facilitate **workshops** with multiple stakeholders to collaboratively explore and refine requirements.
- **Observe** stakeholders in their work environment to reveal hidden requirements.



Elicitation Techniques

- Review **existing documents** to understand relevant rules, flows, and constraints.
- Review **existing systems** to identify current functions.
- Visualize ideas using **prototypes** (such as mockups, wireframes, or interactive demos) to uncover hidden expectations.
- Describing **user interactions** with the system to uncover hidden scenarios.



Challenges

- Stakeholders may **not know** their needs or **struggle to express** them.
- Requirements may **change** over time.
- Requirements may be **in conflict** across stakeholders.
- Stakeholders often have **implicit expectations**.
- Language differences** can hinder understanding.
- Key stakeholders may be **unavailable or disengaged**.



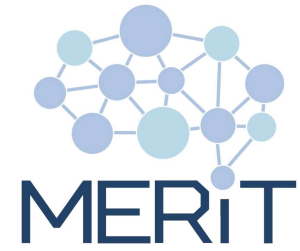
Best Practices

- ❑ **Use multiple** elicitation techniques.
- ❑ **Engage** a broad range of stakeholders.
- ❑ **Prepare thoroughly** and understand the domain and context in advance.
- ❑ **Clarify vague** inputs.
- ❑ **Avoid assumptions.**
- ❑ **Validate early.**



Importance

- It establishes a **shared understanding** among the stakeholders.
- Every design, implementation, and testing decision **links** to the requirements.
- Mistakes made during elicitation are the **most expensive to fix**.



Requirements Specification



Definition

- Requirements specification is a systematic process of converting requirements into a clear and complete functional and non-functional description of the system.



Objectives

- Formalize the **system's functional and non-functional behaviors.**
- Establish a **contract between business and technical stakeholders.**
- Provide a **foundation** for design and testing.



Deliverables

- System Requirements Specification (SRS)
- Requirements Models



System Requirements Specification

- SRS formally **defines functional and non-functional requirements** in a format that is **understandable to non-technical stakeholders** and **actionable for technical stakeholders**.
- The structure may vary slightly by organization or standard, but a widely accepted format is based on the IEEE 830 or ISO/IEC/IEEE 29148 standards.



System Requirements Specification

- 1. Introduction
 - ✓ Purpose
 - ✓ Definitions, Acronyms, Abbreviations
 - ✓ Intended Audience and Use
 - ✓ System Overview and Context

System Requirements Specification

- 2. Overall Description
 - ✓ System Perspective
 - ✓ Users
 - ✓ Operating Environment
 - ✓ Constraints
 - ✓ Assumptions and Dependencies

System Requirements Specification

- 3. External Interfaces
 - ✓ User
 - ✓ Hardware
 - ✓ Software
 - ✓ Communication



System Requirements Specification

- 4. Functional Requirements
- 5. Non-Functional Requirements
 - ✓ Performance
 - ✓ Safety
 - ✓ Security
 - ✓ Quality
 - ✓ Business Rules

System Requirements Specification

- 6. Other Requirements
 - ✓ Legal
 - ✓ Regulatory
- 7. Appendices
 - ✓ Glossary
 - ✓ References



Functional Requirements

- Functional requirements define **what the system should do**: behaviors, features, and services it should provide under specific conditions.
- They are directly traceable to user needs or business rules.
- They can be modeled using requirement models.
- They should be clear, testable, and unambiguous.



Functional Requirements

- The system shall allow users to log in using their username and password.
- The system shall generate a monthly sales report.
- The system shall validate payment information before submission.
- When the temperature exceeds 100°C, the system shall trigger an alert.



Non-Functional Requirements

- Non-functional requirements define **how well the system performs its functions**: quality attributes, operational constraints, and system-level properties.
- They apply across multiple functional areas.
- They can include performance, safety, security, usability, reliability, maintainability, scalability, etc..
- They may require specific measurement or monitoring mechanisms.



Non-Functional Requirements

- The system shall support 500 concurrent users.
- Response time for any user action shall not exceed 2 seconds.
- The system shall be available 99.9% of the time.
- The system shall comply with GDPR regulations.

Modelling Techniques

- Use Case Diagrams (UML)
 - ✓ Show interaction among the different actors of the system.
 - ✓ Best for identifying actors and high-level functional requirements.



Modelling Techniques

- Activity Diagrams
 - ✓ Represent workflows or step-by-step behavior.
 - ✓ Best for:
 - ❖ Understanding workflows and user behavior.
 - ❖ Modeling business processes and system interactions.

Modelling Techniques

- Data Flow Diagrams (DFD)
 - ✓ Illustrate how data moves through the system.
 - ✓ Best for:
 - ❖ Visualizing how data moves through the system.
 - ❖ Early analysis or documenting legacy systems.



Modelling Techniques

- Entity-Relationship Diagrams (ERD)
 - ✓ Model data entities, their attributes, and relationships.
 - ✓ Best for defining the data structure and constraints in the system.



Modelling Techniques

- State Diagrams
 - ✓ Show the states an object can be in and how it transitions between them.
 - ✓ Best for modeling behavior of stateful systems such as order lifecycle: New → Processing → Shipped → Cancelled.

Modelling Techniques

- Sequence Diagrams
 - ✓ Detail the order of messages or interactions between system components or between users and the system.
 - ✓ Best for explaining the order and flow of system interactions, such as the login sequence.