Analysis 2

Validating, Prioritizing, and Managing Requirements

Requirements Validation

Definition

- Requirements validation is a systematic process for ensuring that the **specified requirements** accurately **reflect stakeholder needs** and are **correct**, **complete**, and **consistent**.

Objectives

- Ensure requirements **reflect stakeholder intent**.
- Verify that **no essential requirements are missing**.
- Identify and **resolve conflicts** between requirements.
- Ensure that requirements are **feasible**.
- Ensure all stakeholders **understand and agree** on requirements.

Deliverables

- Requirements Validation Report

Validation Criteria

- Business Analysts apply a set of validation criteria to assess quality, clarity, and completeness, evaluating whether the requirements are fit for purpose.

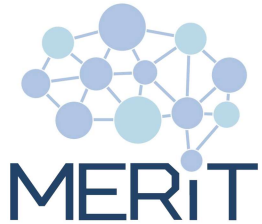
Validation Criteria

- **Correctness:** The requirement accurately reflects stakeholder intent.
- **Completeness:** All necessary requirements are present.
- **Consistency:** The requirement does not conflict with any other requirement.
- **Unambiguity:** The requirement can be understood in only one way by all readers.

Validation Criteria

- **Feasibility:** The requirement is realistic given the available resources, technology, schedule, and budget.
- **Testability:** The requirement can be objectively tested or measured.
- **Traceability:** The requirement can be traced to its origin.
- **Modifiability:** The requirement allows for changes without affecting other requirements.

Validation Techniques



- Stakeholder business analysts, developers, testers, product owners, and business stakeholders **review requirements**.
- **Checklists**, which consist of a list of validation criteria, are used during reviews.
- **Prototypes** (such as mockups, wireframes) are used to validate requirements.

Validation Techniques

- **Models** (such as use case diagrams, process flows, ER diagrams) are used to ensure consistency and completeness.

Validation Process

- Prepare validation **criteria**.
- Choose validation **techniques**.
- **Conduct** validation and **document findings**.
- **Update requirements** if needed based on feedback.
- Get **approval** from stakeholders.

Requirements Validation Report

- ☐ 1. Introduction
  - ✔ Scope
  - ✔ Purpose
- ☐ 2. Participants
  - ✔ Names, Roles, Responsibilities
- ☐ 3. Criteria
- ☐ 4. Techniques

Requirements Validation Report

- 5. Findings
- 6. Actions Taken
- 7. Open Issues
- 8. Approval

Requirements Prioritization

Definition

- Requirements prioritization is a systematic process for evaluating, **ranking**, and **ordering requirements** based on their relative importance, value, urgency, risk, or other project constraints.

Objectives

- Ensure that the highest-value requirements are implemented early to **deliver impact sooner**.
- **Optimize** the use of **limited resources**.
- Enable stakeholders to **make informed choices** when facing constraints.
- Guide **incremental and iterative** development.
- Mitigate **risks**.

Deliverables

- Requirements Prioritization Matrix
- Prioritized Requirements List
- Product Backlog
- MoSCoW Categorization

Prioritization Factors

- Business Value: revenue, efficiency, competition
- User Importance
- Urgency / Time Sensitivity
- Risk and Complexity: take high risk early
- Cost / Effort to Implement
- Dependencies
- Stakeholder Demand

Prioritization Techniques

- MoSCoW Method
  - ✔ Must Have: essential for delivery
  - ✔ Should Have: important but not critical
  - ✔ Could Have: desirable but not necessary
  - ✔ Won’t Have: explicitly out of scope

Prioritization Techniques

- Kano Model
  - ✔ Assigns scores to each requirement based on factors like value, cost, risk, and urgency.
  - ✔ Factors are weighted (e.g., Value = 40%, Risk = 30%, Cost = 30%)
  - ✔ Final scores guide the ranking

Prioritization Techniques

- Weighted Scoring Model
  - ✔ Basic Needs: absence causes dissatisfaction
  - ✔ Performance Needs: drive satisfaction linearly
  - ✔ Delighters: unexpected features that create high satisfaction
  - ✔ Indifferent or Reverse: low or negative impact features

Prioritization Techniques

- 100-Point Method
  - ✔ Stakeholders are given 100 points to distribute across all requirements.
  - ✔ Allows each stakeholder to express the relative importance of each item.

Conflicts

- Conflicts naturally occur when different stakeholders prioritize differently due to their roles, motivations, or pressures.
- The sales team pushes for demo features, while developers want technical enablers first.
- Compliance demands security; UX focuses on usability.
- Business users want speed; IT wants long-term maintainability.

Conflict Resolution

- A neutral facilitator ensures each voice is heard and drives consensus.
- Stakeholders' votes are weighted by their influence or the importance of their roles.
- Objective criteria help move the conversation from opinions to facts.
- In unresolved cases, escalate the issue to the product owner, steering group, or sponsor.

Requirements Management

Definition

- Requirements management is the **continuous process** of **documenting**, **analyzing**, **tracking**, and **controlling** changes to requirements throughout the SDLC.

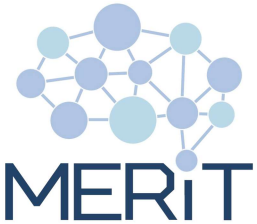
Objectives

- Establish a **systematic approach** for managing changes in requirements.
- Make sure requirements are kept as **current living** documents.
- Establish and maintain **traceability** links between phases.
- Ensure **stakeholders and teams are aligned** on requirement status, priorities, and updates.

Deliverables

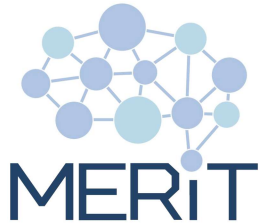
- Requirements Traceability Matrix (RTM)
- Change Requests (CRs)
- Impact Analysis Reports
- Audit Trail / Version History

Change Management



- A **change request** (CR) gets submitted by a stakeholder (developer, tester, or business analyst).
- The **business analyst** conducts an **impact analysis**.
- The **product owner** conducts a **review** and approves or rejects the CR.
- Decisions are **documented**, **versioning** is implemented, and **traceability** is ensured.

Impact Analysis



- Impact analysis is a systematic process for evaluating the consequences of a change in a requirement.
- It allows decision-makers understand the ripple effects before approving a change.
- Effective impact analysis helps prevent hidden costs, technical debt, and unexpected failures.

Versioning and Baseline

- Versioning is the practice of assigning and tracking unique identifiers to different versions of requirements documents to **record their evolution**.
- A baseline is a **formally approved version** of the requirements document used as a stable foundation for design, implementation, testing, compliance, and audits.

Traceability

- Requirements traceability is the ability to link each requirement to related elements across the SDLC.
- Requirements Traceability Matrix maps requirements to: requirement ID (RQ-101), design module (UI-Login), code (login.js), test case (TC-001), risk ID (R-09).