RESEARCH AND THOUGHTS ON REQUIREMENTS FORMAT

There are a few ways of writing requirements and they are explained along with an example of a simple banking application:

1. **Natural Language Specification:**

* **Description:** Natural language specification is a common and straightforward way of expressing requirements using plain language.
* **Example:**

Requirement: The banking application should allow registered customers to check their account balance by logging into their accounts. After logging in, they should be able to view their account balance in real-time.

1. **Structured Specification:**

* **Description:** Structured specification adds organization and formatting to requirements for improved clarity and understanding.
* **Example:**

Requirement ID: BA-001

Title: Account Balance Inquiry

Description: The banking application shall provide a feature for registered customers to check their account balance.

Priority: High

Acceptance Criteria:

1. Customers can access the account balance inquiry feature by logging into their accounts.
2. After logging in, customers can view their account balance in real-time.
3. **Form-Based Specification:**

* **Description:** Form-based specification uses standardized forms or templates to capture requirements in a consistent form
* **Example:**

Requirement ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Title: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Description: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Priority: [High][Medium][Low]

Acceptance Criteria:

1. \_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_

1. **Tabular-Based Specification:**

* **Description:** Tabular-based specification uses tables or grids to present requirements in a structured manner.
* **Example:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Title** | **Description** | **Priority** |
| BA-001 | Account Balance Inquiry | The banking application shall allow customers to check their account balance. | High |
|  |  | Customers must log in to access this feature. |  |
|  |  | Upon logging in, the system should display the real-time account balance. |  |

The different formats for writing requirements and differences between them are:

1. **Organization:**

* Structured Format: Requirements are organized hierarchically into sections and subsections, often following a well-defined outline structure.
* Tabular Format: Requirements are organized linearly in rows and columns within a table or grid, typically without a hierarchical structure.

1. **Attributes:**

* Structured Format: Each requirement includes multiple attributes, such as a unique ID, title, description, acceptance criteria, dependencies, priority, and references. These attributes provide comprehensive details for each requirement.
* Tabular Format: Requirements are represented using a limited set of attributes, often including ID, title, description, priority, status, and references. Tabular formats may have fewer attributes compared to structured formats.

1. **Detail Level:**

* Structured Format: Structured requirements tend to provide a high level of detail, making them suitable for complex projects and comprehensive documentation.
* Tabular Format: Tabular formats are often more concise and may not provide the same level of detail as structured formats. They are better for high-level summaries.

1. **Complexity:**

* Structured Format: Well-suited for complex projects with many stakeholders, intricate dependencies, and extensive traceability needs.
* Tabular Format: Simpler and more straightforward, making them suitable for smaller projects or Agile environments where simplicity is valued.

If I were to choose between structured and tabular formats for writing requirements, I would choose the structured format. Because:

1. **Comprehensive Documentation:** Structured formats permit the addition of many different qualities and specifics for each requirement. This level of specificity is beneficial for complicated projects where thorough documentation is necessary for understanding, traceability, and meeting legal or regulatory requirements.
2. **Clear Hierarchy:** The structured format provides a clear hierarchical organization of requirements, making it easier to understand the relationships between different requirements, sub-requirements, and sections. This is beneficial for projects with intricate dependencies.
3. **Traceability:** The structured format enables strong traceability, allowing you to monitor the effects of changes and comprehend how requirements are intertwined. The management of modifications in large-scale projects depends on this.
4. **Suitability for Big Projects:** Structured formats work well for developing clear and comprehensive requirements documentation that is simple to share and review for big, complicated projects with lots of stakeholders.
5. **Reference Standards:** Structured formats frequently follow recognized norms, such as IEEE Std 830-1998, which offers instructions for structuring software requirements specifications. For consistency and best practices, this alignment may be advantageous.

Reference link:

1. <https://ieeexplore.ieee.org/document/720574>
2. <https://www.agilealliance.org/glossary/user-story-template/>