Practical Example of QAW: Task and project management system

Project Context: Continuing with the same task and project management system, we will now apply Quality Attribute Workshop (QAW) to identify and prioritize quality attributes in collaboration with various stakeholders.

1. Identification of Quality Attributes: In meetings with the work team, we discussed possible quality attributes that could be targeted for study and application within the project. Some of the attributes identified are:

- **Usability**
- **Performance**
- **Availability**
- **Customization**
- **Maintainability**

2. Definition of Quality Scenarios: With the active participation of stakeholders, we define detailed scenarios for each attribute:

Usability Scenario: "Ease of creation and/or assignment of tasks".
Performance Scenario: "Efficient handling of task and project management".
Availability Scenario: "The system will be accessible 24/7".
Personalization Scenario: "The user will be able to set his preferences on task labels".
Maintainability Scenario: "Ease of updating and that the system has a complete documentation that facilitates the resolution of problems".

3. Prioritization of Quality Attributes: Using voting techniques and collaborative discussions, stakeholders prioritize quality attributes. We found that usability is critical to end users, followed by availability, performance, maintainability and customization.

4. Metrics Definition: For each quality attribute, we define specific metrics that will help us evaluate system performance on those aspects. For example:

- To evaluate usability, user testing could be performed to collect data on learning time, user errors, and user satisfaction.
- To evaluate performance, data collection on response time and efficiency could be performed.
- To assess availability, tests on downtime and recovery time could be performed.
- To assess customization, one could inquire about the number of customization options, the level of customization depth, and the ease of customization.
- To assess maintainability, data on code complexity, code modularity and documentation would need to be collected.

5. Detailed Scenario Development: We work on developing detailed scenarios to better understand how quality attributes manifest themselves in specific situations.

- **Usability Scenario:** "A new employee needs to learn how to use the system to manage his tasks. The system should be easy to learn and use, with an intuitive interface that is easy to understand. The user should be able to add, edit and complete tasks without making mistakes".
- **Performance Scenario:** "The manager needs to generate a detailed project report for a meeting, so the system should be efficient enough to handle a larger number of users or tasks without affecting performance."
- **Availability Scenario:** "A user is trying to access the system during a period of high traffic, so the system should be available to be accessed by this user and most other users during this period."
- **Customization Scenario:** "A student needs to tailor assignments and projects to his specific needs, so the system should easily allow him to customize it to his liking and such changes can be made without difficulty."
- **Maintainability Scenario:** "The system code should be clean and easy to understand so that the developer can make the necessary changes efficiently. The system should also have adequate documentation so that the developer can find the information he needs to make changes."

Results:

Based on the second analysis we performed, we obtained a solid basis for architectural decision making and definition of quality requirements. We performed evaluative metrics that have helped us to measure quality attributes and thus, prioritize them. Along with this evaluation, we developed more detailed quality scenarios where the impact of these attributes can be better understood.

The prioritization was done in such a way that we defined usability as critical for end users, because the system must be easy to use and intuitive, causing that enjoyable experience to all users. Following this, we consider that availability goes hand in hand, since the system must allow access at any time a user wants it. Performance, maintainability and customization end with this prioritization, because the focus is more on user comfort, thus defining the importance of these quality attributes against the system.

It could be concluded that the identification and prioritization of quality attributes of the task and/or project management system helped to reduce system development costs and to focus on the important attributes of the system, thus creating a more effective development team, leading to greater collaboration and communication, improving the quality of the software.