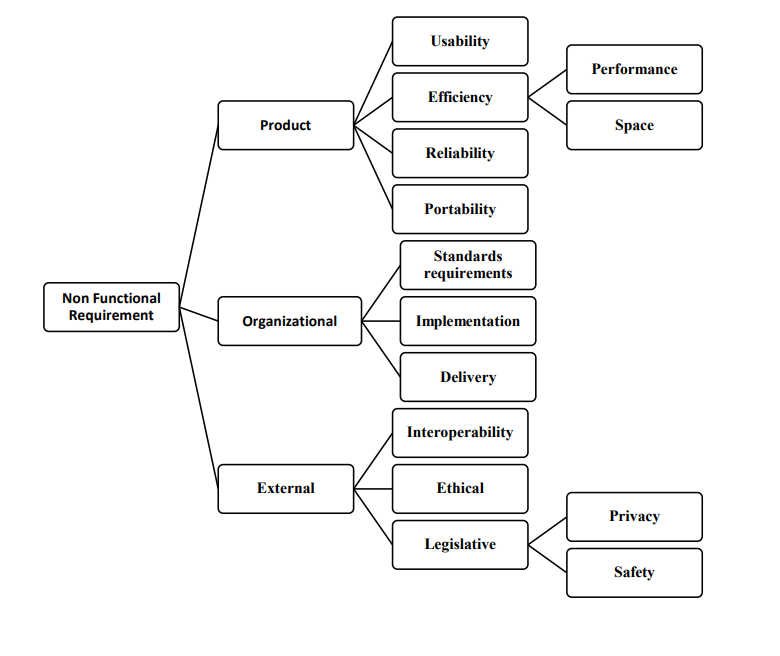
Name : Muhammad Shoaib Akhter Qadri

Enrollnment # : 02-131212-009

Registration # : 79290

Subject : Software Requirement Engineering

**Question 1:** For each type of Non Functional Requirements give two example each and identify Goal, Objective and Quantitative verifiable metric where possible to create proper non-functional statements.

****

**NON-FUNCTIONAL REQUIREMENTS**

* **Usability:**

Usability is a non-functional requirement, because in its essence **it doesn't specify parts of the system functionality, only how that functionality is to be perceived by the user**, for instance how easy it must be to learn and how efficient it must be for carrying out user tasks.

**Example No:1:**

* **Efficiency of use:**

Goals are easy to accomplish quickly and with few or no user errors.

**Example No:2:**

* **Intuitiveness:**

The interface is easy to learn and navigate; buttons, headings, and help/error messages are simple to understand.

**Goal:**

Goals include **effective, efficient, engaging, error tolerant, easy to learn** .

**Objective:**

The goal is **to identify any usability problems, collect qualitative and quantitative data and determine the participant's satisfaction with the product**.

**Metrics**:

Usability metrics are **a system of measurement of the effectiveness, efficiency, and satisfaction of users working with a product**.

* **EFFICIENCY:**

 Efficiency is defined as the ratio of the amount of energy used to the amount of work done to create an output. But when it comes to software development, “efficiency” is **the amount of software developed or requirement divided by the number of resources used like time, effort.**

**Example of Performance:**

**A telephony switch may be required to process 100,000 calls per hour**. Performance is the degree to which a software system or component meets its objectives for timeliness.

**Space Example:**

Space complexity **includes both Auxiliary space and space used by input**. For example, if we want to compare standard sorting algorithms on the basis of space, then Auxiliary Space would be a better criterion than Space Complexity.

**Goal:**

Efficient product.

**Objective:**

**It helps to test the effectiveness of the software**. **Metrics (space):** Main memory.

**Metrics:**

Efficiency metric is **a performance measurement tool that helps organizations understand the different contributing factors towards net organizational output**.

* **PORTABILITY:**

Software portability is **the possibility to use the same software in different environments**. It applies to the software that is available for two or more different platforms or can be recompiled for them.

**Example No 1:**

**Port an application to Linux** means to modify the program so that it can be run in a Linux environment. Portability refers to the ability of an application to move across environments, not just across platforms.

**Example No 2:**

When referring to computer hardware, portability describes an external device that can be moved from one place to another without disrupting its operation.

**Goal:**

To make the computer program run in the new environment.

**Objective:**

To determine whether a system can be ported to each of the environmental characteristics.

**Metrics:**

Iterative and incremental development cycle

* **STANDARD REQUIREMENTS:**

Any standard, protocol, or similar document that outlines the rules and processes for the creation of software products.

**Example No 1:**

Apple standard for building their projects.

**Example No 2:**

SDLC or V-Model or agile is used for building projects.

**Goal:**

Satisfying their customers' quality requirements

**Objective:**

Consistent with the quality policy and relevant to product and service conformity

* **IMPLEMENTATION:**

Software implementation refers to **the process of adopting and integrating a software application into a business workflow**

**Example No 1:**

Developing and then executing a new marketing plan to help increase sales of the company's products to consumers.

**Example No 2:**

Software development tools contain implementations of a programming language.

**Goal:**

Helps to find and deploy the right software for your business's needs.

**Objective:**

Make it complete according to the user’s point of view.

* **DELIEVERY:**

Delivery is the entire process of getting a software product to customers, from conceptualization, through development, and ending in the actual purchase and installation of the product's license.

**Example No 1:**

**Waterfall, Agile, and DevOps** is the example of Delivery.

**Example No 2:**

The entire system should be up and running in the user’s production environment by May 20,2021.

**Goal:**

Trusted Product.

**Objective:**

Providing visibility, improving communication and eliminating disconnected teams and manual processes as much as possible.

* **INTEROPERABILITY:**

The term “software interoperability” refers to **the capability of different solutions to communicate with one another freely and easily.**

**Example No 1:**

Systems that are interoperable exchange information in real-time, without the need for specialized IT support or behind-the-scenes coding.

**Example No 2:**

Interoperability testing is done between smartphones and tablets to check data transfer via Bluetooth.

**Goal:**

To make it so that different systems are able to “talk” and “understand” the information they pass to one another.

**Objective:**

Helps clinicians deliver safe, effective, patient-centered care

* **ETHICAL:**

Ethics **focuses on providing high-quality software**.

**Example No 1:**

Software engineers shall act consistently with the public interest.

**Example No 2:**

CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

**Goal:**

Product efficiency.

**Objective:**

Analyzing, specifying, designing, developing, testing, and maintaining software