**Definition:**

* A **non-functional requirement** describes operational qualities rather than behavioral qualities.
* In other words, it talks about “how the system will do” rather than focusing on “what the system will do “
* In simple terms: A **non-functional requirement** is something that can be measured
* Non-functional testing verifies how well the system meets these requirements by examining its performance, usability, reliability, etc.  For example, a non-functional test of an e-shop would check how many people could simultaneously place an order.

**Benefits of Non-Functional testing**

* Increase security of the product
* Improving the performance of the system
* Improving the user experience and ease of operating the software

**Problems of Ignoring Non-Functional testing:**

* Running slowly or becoming unresponsive while several users use it simultaneously (Poor performance)
* Becoming unavailable when it is required most, resulting in a direct impact on business (poor Availability or poor Reliability)
* Providing poor usability that frustrates users (poor usability)
* Breaking the security of confidential data (poor security)

**Functional vs Non-Functional Requirement**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Functional Testing** | **Non-Functional testing** |
| **Functionality** | Verifies what the product does | Verifies how the product works |
| **Execution** | Is performed before non-functional testing | Is performed after functional testing |
| **Focus area** | Is based on customers’ requirements | Focuses on customers’ expectations |
| **Manual testing** | Can be executed either manually or using automation tools | Is preferably performed using automation tools |
| Examples | A login page must show text boxes to enter a username and password. | The dashboard should load in 2 seconds. |
|  | The search function in the application should not be case sensitive. | Software should be installable on all versions of Windows and Mac. |
|  | A verification email is sent to a user whenever he/she registers for the first time. | The GUI should provide help and tooltips to make it easy to use. |

**Types of Non-Functional testing**

* **Compatibility:** 
  + What are the minimum hardware requirements? What operating systems (including versions) must be supported?
* **Performance:**
  + How quick is the response to a user’s actions? How much time does a user wait for a specific operation to happen?
* **Capacity**
  + What is the maximum number of users allowed to use the application simultaneously—for example, on Black Friday? Is the user experience compromised at this time?
* **Security**
  + Is every endpoint adequately secured? Does every endpoint require an authentication? Is confidential data, such as the last names of end users, shared in response?
* **Reliability and Availability**
  + How often do critical failures happen to the system? Do users need round-the-clock access to the system?
* **Scalability**
  + Is your system able to handle the growing number of users by adding extra resources (e.g., servers, storage, memory) to it?
* **Maintainability and Manageability**
  + How long does it take to fix components? How easily can an administrator service the system? How easy is it to improve performance and adapt to a new operating system?
* **Usability:**
  + What determines the user’s satisfaction with the product? How easy is it to use the application? Is your site easy to navigate? Does the user understand the primary purpose of the application?
* **Accessibility**
  + Could people with visual impairments or color blindness use the application? Can users with impaired mobility navigate the system with a keyboard? Will people with cognitive disabilities, or those who are deaf or hard-of-hearing, have full access to the content? Does the application meet all legal requirements for accessibility?

There are more than 150 types of Nonfunctional testing including Localization, Portability

While the types of Non functional testing differ based on the requirement. There are 2 major categories

* Security testing
* Performance testing

Performance Testing

Performance testing is a software testing practice implemented to determine how a system performs in terms of speed, response time, stability, reliability, scalability, and resource usage of a software application under the expected workload.

Types of performance testing:

* Load testing
  + verifies an application’s ability to perform under anticipated user loads.
* Stress testing
  + checks an application under extreme workloads to see how it handles high traffic or data processing
* Endurance testing
  + Checks whether software can handle the expected load over a long period of time.
* Scalability testing
  + determines how effectively an application scales up to support an increase in user load.

Automating Non functional testing:

Using manual testing for Non functional testing is time consuming and might not achieve desired results. In fact, tests like performance testing is impossible without Automation tests

Examples of automation tools are

Jmeter

Loadrunner

Forecast