

## **Why is the difference between functional and Non-functional requirements?**

The end goal of a project is to deliver a high quality product exactly as the customer asked for. Functional requirements are the primary way that a customer communicates their requirements to the project team. Functional requirements help to keep project team going in the right direction. Unclear requirements leads to a poorly defined scope that creates a lot of challenges from the beginning of the project. A poorly defined scope leads to extension in the schedule and increase in cost. The customer may not have the time and money to invest, so they just accept a product with low quality.

Typically, the customer has both needs and wants. After seeing the cost estimate, they may ask to reduce the scope. Usually removing some of the non-functional requirements reduces the scope. A lot of non-functional requirements can quickly drive up the cost, while insufficient non-functional requirements may lead to bad user experience.

Understanding the difference between functional and non-functional requirements will help both, the client and the IT supplier as they will be able to understand their requirements clearly. This leads to scope refinement, optimized cost, and finally a happy customer.

If there is any one thing any project must have in order to prevent failure, is a sensible and comprehensive collection of both the functional and non-functional requirements.

**Any project's requirements need to be well thought out, balanced and clearly understood** by all involved, but perhaps of most importance is that they are **not dropped or compromised halfway through** the project.

However, what exactly is the difference between ‘functional’ and ‘non functional’ requirements? It’s not that complex, and once you understand the difference, the definition will be clear. The official definition of ‘a functional requirement’ is that it essentially **specifies something the system should do**.

Typically, functional requirements will specify a behavior or function, for example: “Display the name, total size, available space and format of a flash drive connected to the USB port.” Other examples are “add customer” and “print invoice”.



A functional requirement for a milk carton would be “ability to contain fluid without leaking”

## What are Functional Requirements?

The definition of a functional requirement is:

*“Any Requirement Which Specifies **What The System Should Do.**”*

In other words, a functional requirement will describe a particular behavior or function of the system when certain conditions are met, for example: “Send email when a new customer signs up” or “Open a new account”.

A functional requirement for an everyday object like a cup would be: “ability to contain tea or coffee without leaking”.

## Some of the more typical functional requirements include:

- Business Rules
- Transaction corrections, adjustments and cancellations
- Administrative functions
- Authentication
- Authorization levels
- Audit Tracking
- External Interfaces
- Certification Requirements
- Reporting Requirements
- Historical Data
- Legal or Regulatory Requirements

So what about Non-Functional Requirements? What are those, and how are they different?

Simply put, the difference is that **non-functional requirements describe how the system works**, while **functional requirements describe what the system should do**.

The definition for a non-functional requirement is that it essentially specifies **how the system should behave** and that it is a constraint upon the systems behaviour. One could also think of non-functional requirements as quality attributes for of a system.



A non-functional requirement for a hard hat might be “must not break under pressure of less than 10,000 PSI”

## **What are non-functional requirements?**

The definition of a non-functional requirement is:

*“Any Requirement That Specifies **How** The System Performs A Certain Function.”*

In other words, a non-functional requirement will describe how a system should behave and what limits there are on its functionality.

Non-functional requirements cover all the remaining requirements which are not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviors, for example: “Modified data in a database should be updated for all users accessing it within 2 seconds.”

A non-functional requirement for the cup mentioned previously would be: “contain hot liquid without heating up to more than 45°C”.

Even in the case when the non-functional requirements are not met the basic functionality will not be impacted.

If the functionality of the product is not dependent on non-functional requirements then why are they important? The answer is in usability. Non-functional requirements affect the user experience as they define a system’s behavior, features, and general characteristics.

Non-functional requirements when defined and executed well will help to make the system easy to use and enhance the performance.

Non-functional requirements focus on user **expectations**, as they are product properties.

Let’s take an example of a functional requirement. A system loads a webpage when someone clicks on a button. The related non-functional requirement specifies how fast the webpage must load. A delay in loading will create a negative user experience and poor quality of the system even though the functional requirement is fully met.

## **Some typical non-functional requirements are:**

- Performance – for example Response Time, Throughput, Utilization, Static Volumetric
- Scalability
- Capacity
- Availability
- Reliability
- Recoverability
- Maintainability
- Serviceability
- Security
- Regulatory
- Manageability
- Environmental
- Data Integrity
- Usability
- Interoperability

As said above, non-functional requirements specify the system’s ‘quality characteristics’ or ‘quality attributes’.

Many different stakeholders have a vested interest in getting the non-functional requirements right particularly in the case of large systems where the buyer of the system is not necessarily also the user of the system.

The importance of non-functional requirements is therefore not to be trifled with. One way of ensuring that as few as possible non-functional requirements are left out is to use non-functional requirement groups. For an explanation on how to use non-functional requirement group, read this blog post which will give you four of the main groups to use.

### **Difference between functional and non-functional requirements:**

<b>Functional Requirements</b>	<b>Non-Functional Requirements</b>
They define a system or its component.	They define the quality attribute of a system
It specifies, "What the system should do?"	It specifies, "How should the system fulfill the functional requirements?"
User specifies functional requirement.	Non-functional requirement is specified by technical peoples e.g. Architect, Technical leaders and software developers.
It is mandatory to meet these requirements.	It is not mandatory to meet these requirements.
It is captured in use case.	It is captured as a quality attribute.
Defined at a component level.	Applied to a whole system.
Helps you to verify the functionality of the software.	Helps you to verify the performance of the software.
Functional Testing like System, Integration, End to End, API testing, etc are done.	Non-Functional Testing like Performance, Stress, Usability, Security testing, etc are done.
Usually easy to define.	Usually more difficult to define.

### **Examples of functional and non-functional requirements:**

Below you can check the list of functional and non-functional requirements examples:

### **Functional Requirements Example:**

1. Authentication of a user when he/she tries to log into the system.
2. System shutdown in the case of a cyber attack.
3. Verification email is sent to user whenever he/she registers for the first time on some software system.

### **Non-functional Requirements Example:**

1. Emails should be sent with a latency of no greater than 12 hours.
2. Each request should be processed within 10 seconds.
3. The site should load in 3 seconds when the number of simultaneous users are > 10000

### **How to gather functional and non-functional requirements?**

Guided brainstorming session is one of the best ways to gather requirements by getting all stakeholders together. You should include user representatives who are the best sources of non-functional requirements.

Basically functional requirements can be divided into 4 groups which are:

**Business requirements.** They contain the ultimate goal, such as an order system, an online catalogue, or a physical product. It can also include things like approval workflows and authorization levels.

**Administrative functions.** They are the routine things the system will do, such as reporting.

**User requirements.** They are what the user of the system can do, such as place an order or browse the online catalogue.

**System requirements.** These are things like software and hardware specifications, system responses, or system actions.

Once the functional requirements are defined then its time to think about the non-functional requirements, such as:

**Usability.** This focuses on the appearance of the user interface and how people interact with it. What colour are the screens? How big are the buttons?

**Reliability / Availability.** What are the uptime requirements? Does it need to function 24/7/365?

**Scalability.** As needs grow, can the system handle it? For physical installations, this includes spare hardware or space to install it in the future.

**Performance.** How fast does it need to operate?

**Supportability.** Is support provided in-house or is remote accessibility for external resources required?

**Security.** What are the security requirements, both for the physical installation and from a cyber perspective?

### **How to write functional and non-functional requirements?**

There are different ways to write functional and non-functional requirements.

The most common way to write functional and non-functional requirements is through a *Requirements Specification Document*. It is a written description of the required functionality.

It states the project objective and includes an overview of the project to provide context, along with any constraints and assumptions. The requirements specification document should include visual representations of the requirements to help non-technical stakeholders understand the scope.

Closely related to a requirements specification document is a *Work Breakdown Structure* or WBS. This breaks down the entire process into its components by “decomposing” the requirements into their elements until they cannot be broken down any further.

Another approach is *User Stories*. They describe the functionality from the perspective of the end-user and states exactly what they want the system to do.

It effectively states “As a <type of user>, I want <goal> so that <reason>”. One benefit of user stories is that they do not require much technical knowledge to write. User stories can also be used as a precursor to a requirements specification document by helping define user needs.

*Use Cases* are similar to user stories in that no technical knowledge is necessary. Use cases simply describe in detail what a user is doing as they execute a task. A use case might be “purchase product”, and describes from the standpoint of the user each step in the process of making the purchase.