Functional vs Non-Functional Requirements: The Definitive Guide

Knowing exactly what features and functionalities a customer wants in the app is quite challenging for a software development team. To avoid any misunderstandings, a customer and a software development team need to define project requirements: both functional and non-functional requirements for the future application. In this article, we explain the difference between the two types of requirements and share the best practices on how they are gathered.

What are Functional Requirements?

In software development, functional requirements determine the functions an entire application or just one of its components should perform. A function consists of three steps: data input – system behavior – data output. It can calculate, manipulate data, carry out business processes, establish user interaction, or do any other tasks.

In other words, a functional requirement is WHAT an application must or must not do after some data input.

Functional requirements are important as they show software developers how the system is intended to behave. If the system doesn't meet functional requirements it means that it doesn't work properly.

What are Non-Functional Requirements?

Non-functional requirements determine the performance standards and quality attributes of software, e.g. system usability, effectiveness, security, scalability, etc.

While functional requirements determine what the system does, non-functional requirements describe HOW the system does it. For example, a web application must handle more than 15 million users without any decline in its performance, or a website must not load more than 3 seconds.

If an app doesn't meet non-functional requirements, it continues to perform its basic functions, however, it won't be able to provide a great user experience.

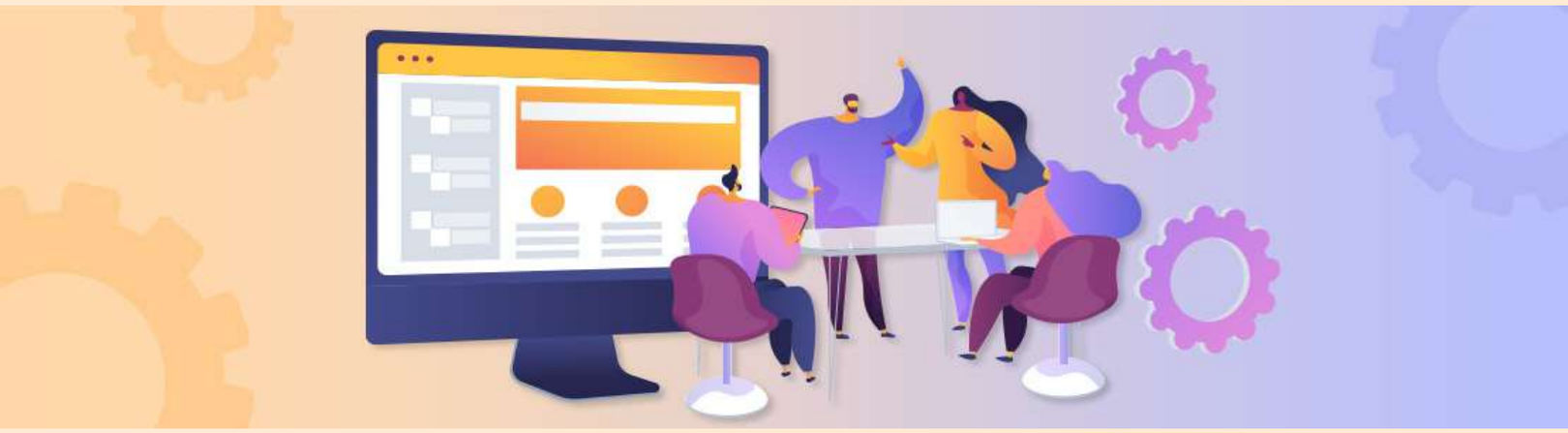
Non-functional requirements are important as they help software developers to define the system abilities and constraints that are essential for developing high-quality software. Therefore, non-functional requirements are as important as functional requirements for successful product adoption.

Why is the Difference Between Functional and Non-Functional Requirements Important?

Well-defined functional and non-functional requirements help software developers to build the product that exactly corresponds to the client's needs. However, is it really necessary to know the difference between functional and non-functional requirements?

The main reason to know the difference between functional and non-functional requirements is that they define the scope of work for a project. Software developers need to keep up with this scope to develop an application within its timeframe and budget.

If the scope of work constantly changes, the development team has to extend the deadlines and the development costs increase. This may lead to adverse consequences for a project.



The importance of distinguishing between the two types of requirements is paramount when creating an MVP. A development team and a customer should discuss which features and functionalities to implement in the app first. A customer may have his own vision of the project and its requirements. If a customer decides that they want to remove or modify some feature, it's essential to realize what type of requirement it is. Most of the time, software developers can simply change the non-functional requirements while functional requirements will demand more work and profound changes.

When a customer and a software development provider know the difference between functional and non-functional requirements, it helps them to more precisely define the scope of work, more accurately range the requirements by importance, optimize the project costs, and better meet the customer's needs.

How are Functional and Non-Functional Requirements Gathered?

Ideally, before turning to a software development company, customers should already have all the functional and non-functional requirements at hand. Therefore, they need to prepare them in advance on their own or ask a 3d party provider.

Let's have a look at what each type of requirements include.

The functional requirements can be divided into three groups:

- business requirements – describe the project goals and expectations, the benefits the project can bring, possible project constraints, and its scope;
- user requirements – involve the user needs and what activities a user will be able to perform in the system;
- system requirements – include system actions, specifications of software and hardware, and so on.

The non-functional requirements fall under various categories, including:

- usability – defines how easy a user can interact with the app's interface, e.g. the screen color, buttons size, etc.;
- availability – ensures that the app will work stably for a certain time period, e.g. rare downtimes throughout the year 24/7;
- reliability – defines that the app will work in a certain environment or for a specific period of time without any failures;
- recoverability – ensures that the app can recover all the data after the system failure or restore the system to certain parameters;
- scalability – determines that the app will continue functioning properly after its size or volume changes;
- performance – assesses how fast the app is;
- supportability – defines if the app is easy to support and maintain throughout its life cycle and what kind of support it requires e.g. an in-house team or remote support;
- security – defines how secure the app should be, for example, FinTech and banking apps should meet the international and regional security standards;
- capacity – assesses the amount of data or services the app can handle.

Examples and Best Practices

There is a wide range of other formats that can help make up the project requirements. Let's have a look at the most effective ones.

User Stories

It is a common practice to formulate the requirements in a form of user stories. User stories are the requirements conveyed by a user. They usually take the form of several simple sentences which have the same pattern:

As a (user), I want to (goal) so that (reason).

A user story example: As a project manager, I want to understand the software development team's progress so that I can report on the outcomes to the CEO and project stakeholders.

Software developers usually compile requirements using user stories when they want to communicate the ideas on product features and functionality to non-technical members.

Use Cases

Use cases have a broader scope than user stories. They include types of users and all the possible actions a user can do in an app. Unlike user stories that describe the end purpose of a feature, use cases involve the flow of steps which lead to the purpose.

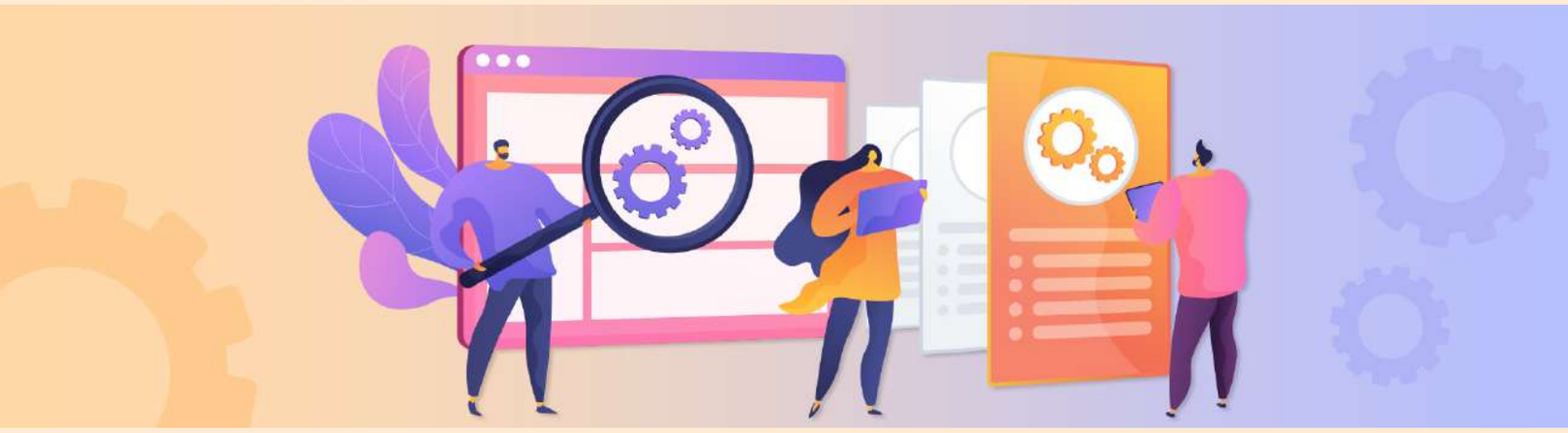
For example, if you want to create an application for logistics and supply chain management, the number of roles software developers should think about are: sellers, buyers, suppliers, managers, dispatchers, and many others.

The actions for these roles can look like these:

- both seller and buyer can view the product route in the app;
- all the dispatchers can track products in warehouses;
- all the users can view the delivery time on items in their accounts, etc.

Software Requirements Specification Document

Software Requirements Specification (SRS) is the document that a software development team relies on when creating an app. It includes all the customers' needs and wants translated into a clear for the development team language – a detailed description of all the product functions and features.



The main sections that are usually included in an SRS document are:

- introduction with the product purpose, glossary of terms, references to specific literature and materials related to the app development;
- description contains a detailed description of product features, coding standards, data exchange policies, and others;
- the system features section which explains how the app's features should function;
- external interface requirements define hardware, software, or databases the app has to interact with;
- non-functional requirements – all the performance standards, an app's quality attributes, and security requirements.

The creation of an SRS, user cases, and user stories are essential for effective application development.

However, there are other documents that are equally important for a successful project launch and development.

Conclusion

With custom software services companies can build any type of app for their effective business development. However, for an app to really meet their business needs, it should have detailed functional and non-functional requirements.

To form functional and non-functional requirements, you can search for help from your software development company, third-party businesses, or do it on your own. Well-elaborated requirements ensure that your software development partner will fully understand how to develop a digital solution that would entirely meet your expectations and satisfy your business needs.

Feel free to contact us and ask any questions: info@scand.com