IN order to decrease the project's cost, increase the development of team productivity, and develop a successful and cost-effective project, must determine the class of requirements

Precise functional and non-functional requirements are essential to reduce development costs because, when the requirements are clear, the team can develop the project much faster. The difference between functional and non-functional requirements is as follows:

Functional requirements are easy to define because the business idea drives them. They include all the features of your future project and the ways users engage with it.

Experience drives Non-functional requirements. In order, to identify them, you need to analyze the product's performance and make it convenient and useful. Such requirements may appear when the product is being used on a regular basis.

Functional requirements

Requirements that describe system behavior under specific conditions and include the product features and functions which developers must add to the solution. Such requirements should be precise both for the development team and stakeholders.

Non-functional requirements: are quality attributes that describe ways your product should behave. The list of basic non-functional requirements includes:

a- Usability

Usability is the degree of ease with which the user will interact with your products to achieve required goals effectively and efficiently.

b- Legal or Regulatory Requirements

Legal or regulatory requirements describe product adherence to laws. If your product violates these regulations, it may result in legal punishment, including federal fines.

For example, look at legal requirements for our recent project, a mobile taxi platform:

"To operate in London, the platform should be licensed by the local transport authority - Transport for London."

c- Reliability

Such a metric shows the possibility of your solution to fail. To achieve high reliability, your team should eliminate all bugs that may influence the code safety and issues with system components.

d- Performance

Performance describes how your solution behaves when users interact with it in various scenarios. Poor performance may lead to a negative user experience and jeopardize system safety.

Example:

"The application shows cars nearby for three seconds."

as a summary:

Functional requirements: describe what the system does.

Non-functional requirements: describe how the system works.

The Comparison Between Functional Vs. Non-Functional Requirements

Parameters	Functional Requirement	Non-Functional Requirement
Requirement	It is mandatory	It is non-mandatory
Capturing type	It is captured in the use case.	It is captured as a quality attribute.
End-result	Product feature	Product properties
Capturing	Easy to capture	Hard to capture
Objective	Helps you verify the functionality of the software.	Helps you to verify the performance of the software.
Area of focus	Focuses on user requirement	Concentrates on the user's expectation and experience.
Documentation	Describe what the product does	Describes how the product works
Product Info	Product Features	Product Properties

Examples:

1- All new customers need to be added to the system. **(FR)**

2- Users must be able to browse for events by name and date. **(FR)**

3- (Daily) reports of sales must be generated. **(NFR)**

4- All data is to be backed up (nightly). **(NFR)**

5- The system shall be available in Arabic and English. **(FR)**

6- The system shall allow the user to check the availability and type of buses on line to all users. **(FR)**

(7/24 NFR)

7- The system shall support minimum 1000 reservation per hour. **(NFR)**

8- The system shall allow to customer to cancel their reservation at any moment. **(FR)**