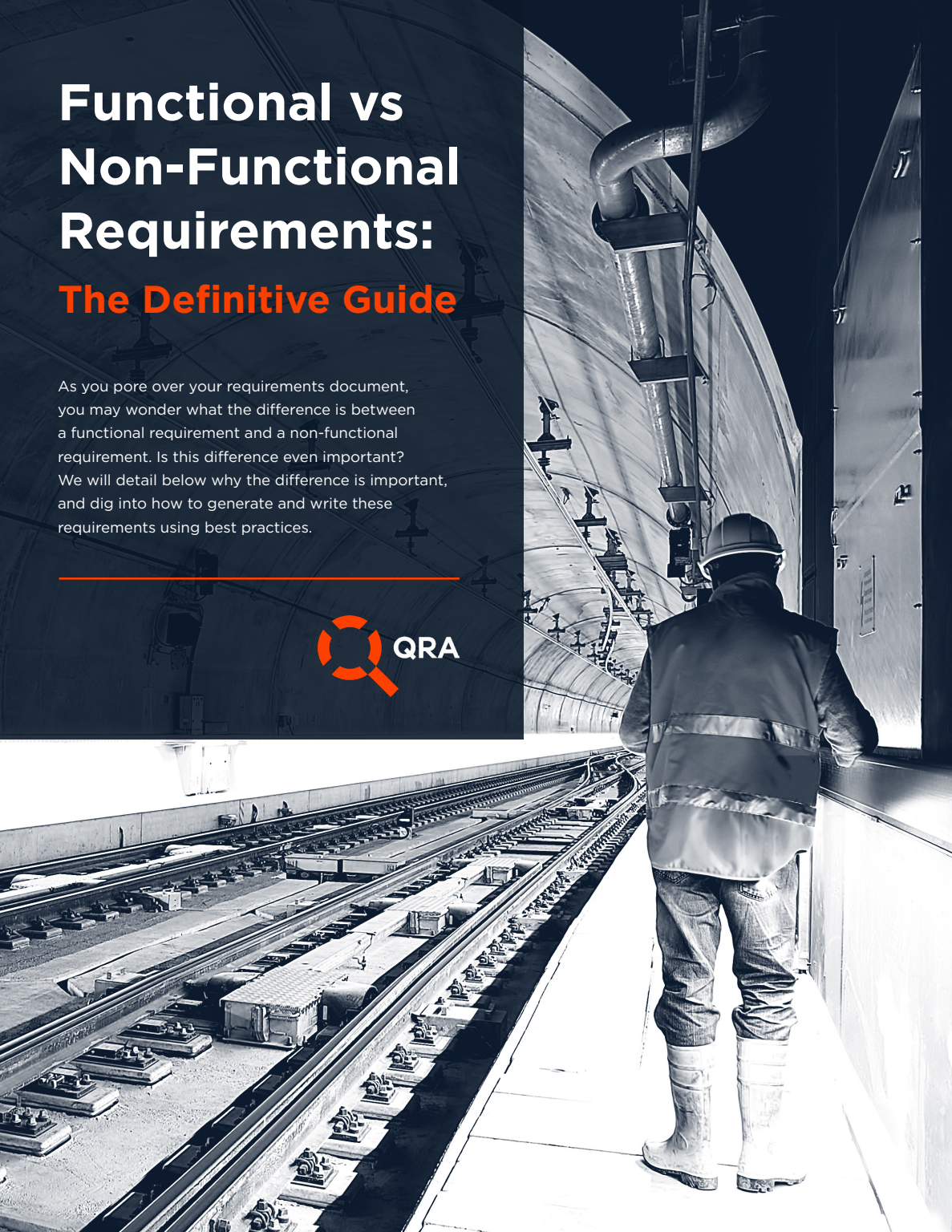
Functional vs Non-Functional Requirements:

The Definitive Guide

As you pore over your requirements document, you may wonder what the difference is between a functional requirement and a non-functional requirement. Is this difference even important? We will detail below why the difference is important, and dig into how to generate and write these requirements using best practices.

QRA



INTRODUCTION

Like many professions, the world of engineering and project management has its own "terms of art" that can be confusing to experts and novices alike. As you pore over your requirements document, you may wonder what the difference is between a *functional requirement* and a *non-functional requirement*. Is this difference even important? We will detail below why the difference is important, and dig into how to generate and write these requirements using best practices.

WHY IS THE DIFFERENCE BETWEEN FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS IMPORTANT?

Ultimately, you want to deliver the product the customer asked for. Functional requirements are the primary way that the customer communicates their needs to the team. They keep everyone on the project team going in the same direction. Without an agreed functional requirements document to clearly define the scope, the end product is likely to miss the mark.

Initially delivering the wrong scope is clearly a problem, but it also creates other issues. To fix the scope, the schedule is extended and the cost increases. The customer may not have the time and money to fix the errors, so they just accept them and consider your product to have quality defects.

Not all scope is equally important, though. Typically, the customer has both needs and wants. After seeing the cost estimate, they may ask to cut scope. Often, scope cutting exercises focus on the non-functional requirements. Excess non-functional requirements can quickly drive up the cost, while insufficient non-functional requirements lead to bad user experiences.

Knowing the difference between functional and non-functional requirements will help both the client and the provider understand their needs in-depth, which will lead to better scope refinement, optimized cost, and ultimately a satisfied customer.

WHAT IS A FUNCTIONAL REQUIREMENT?

Functional requirements define the basic system behaviour. Essentially, they are **what** the system does or must not do, and can be thought of in terms of how the system responds to inputs. Functional requirements usually define if/then behaviours and include calculations, data input, and business processes.

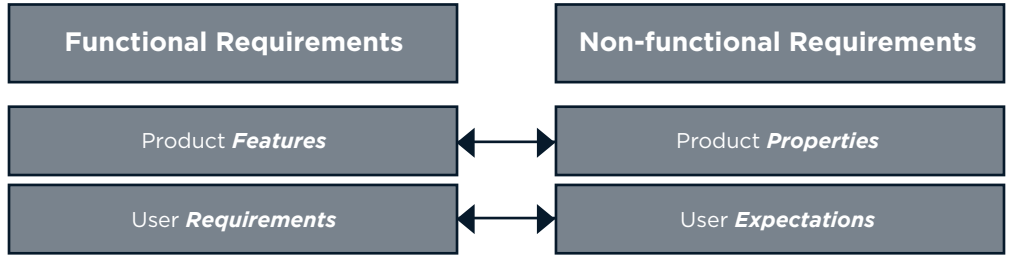
Functional requirements are features that allow the system to function as it was intended. Put another way, if the functional requirements are not met, the system will not work. Functional requirements are product **features** and focus on user **requirements.**

WHAT IS A NON-FUNCTIONAL REQUIREMENT?

While *functional* requirements define what the system does or must not do, *non-functional* requirements specify **how** the system should do it. Non-functional requirements do not affect the basic functionality of the system (hence the name, *non*-functional requirements). Even if the non-functional requirements are not met, the system will still perform its basic purpose.

If a system will still perform without meeting the non-functional requirements, why are they important? The answer is usability. Non-functional requirements define system behaviour, features, and general characteristics that affect the user experience. How well non-functional requirements are defined and executed determines how easy the system is to use, and is used to judge system performance. Non-functional requirements are product **properties** and focus on user **expectations**.

There are more examples below, but for now, think about a functional requirement that the system loads a webpage after someone clicks on a button. There should be a related non-functional requirement specifying how fast the webpage must load. Without it, the user experience and perception of quality are at risk if they are forced to wait too long, even though the functional requirement is fully met.



HOW ARE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS GATHERED?

One of the best ways to gather requirements is to get all of the stakeholders together for a guided brainstorming session. Remember that in many cases the upper-level stakeholders are not the users. Include user representatives on the team, who are one of the best sources for non-functional requirements. They are the ones who will tell you things like – If I could enter the data in this order..., if it could be displayed at least this large so I can see it across the room..., etc.

As you brainstorm functional requirements, consider these groups:

- **Business requirements.** This usually contains 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.** These are the routine things the system will do, such as reporting.
- User requirements. These 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.

After defining the functional requirements, 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?

Although it is ideal to define all functional requirements first, then move on to non-functional requirements, the reality is that people will think of both as they brainstorm. The facilitator of the meeting must be well versed in the difference so they can classify them as the discussion progresses.

During the initiation phase of the project, keep your ears open during even casual discussions with users and stakeholders. You may find they bring up requirements that they did not think of during the initial requirements gathering.

HOW ARE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS WRITTEN?

Both functional and non-functional requirements can be written in several different ways. Here are a few of the more common ones, although anything that effectively communicates the information to everyone involved is perfectly acceptable.

The most common way to write functional and non-functional requirements is a *requirements specification document*. This is simply a written description of the functionality that is required. It states the project objective and includes an overview of the project to provide context, along with any constraints and assumptions. Although much of a requirements specification document is narrative, it is a good idea to 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 individual elements until they cannot be broken down any further.

Another approach is *user stories*. These are a description of the functionality from the perspective of the end-user, and describes 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.

WHAT ARE SOME EXAMPLES OF FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS?

Although there are gray areas in classifying a requirement as either functional or non-functional, here are some examples:

Functional

- When the user enters the information, the system shall send an approval request.
- The server shall log all changes to existing data.
- When 24 hours have passed since the last database backup, the server shall automatically back up the database.
- The local terminal shall automatically print a list of orders every 4 hours.
- When the local time is 0800, the server shall email a report of open issues to all managers.
- The system shall only allow managers to view customer banking data.

Non-functional

- Database security shall meet HIPAA requirements.
- The layout shall allow users to reach their profile data from any page within 3 clicks.
- If a user has not changed their password for 56 days, then the system shall require a password change upon login.
- The system must accommodate a minimum of 3 million concurrent users.
- All web pages shall load within 4 seconds.
- The server room shall accommodate a future doubling of installed hardware.
- The server room shall be accessible by authorized employees 24 hours per day.
- The background colour for all screens shall be #fff4b6.
- System programming shall not use deprecated code.

A functional requirement frequently has a related non-functional requirement. Remember that the difference is essentially the *what* and the *how*, which are both necessary. Here are some examples:

Functional Requirements	Related Non-Functional Requirement
When a site visitor creates an account, the server shall send a welcome email.	When sending welcome emails, the server must send them within 10 minutes of registration.
When order status changes to fulfillment, the local printer shall print a packing slip.	When packing slips are printed, they must be on both sides of 5" x 8" sheets of white paper.
The system must allow the user to fill out and submit a service form.	When the form is requested from the server, it must load with 1 second. When the submit button is pressed, it must complete upload within 2 seconds.

WHAT ARE BEST PRACTICES FOR FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS?

Regardless of the form they take, the key to effective functional and non-functional requirements is that they are clear and as easy to understand as possible. Each audience is different, but in all cases the closer the requirements are to being in natural language, the better. And when it comes to language, the use of active voice is preferred over passive voice. Active voice usually results in clearer and shorter requirements by ensuring there is an "actor" in each requirement statement.

When writing requirements, make sure they are complete and accurate and avoid vagueness. At the same time, avoid including extraneous information that may confuse people. Use "must" instead of "should" as you write the requirements document. Be consistent in the use of terminology and units, and be consistent in the format and language used.

As you seek to make the requirements clear, translate stakeholder and user input into discrete requirements. If they say the system must be "fast", what does that mean? When they say the web site must be able to handle "a lot" of users, what does that mean? Turn these types of requirements into a number and quantify them.

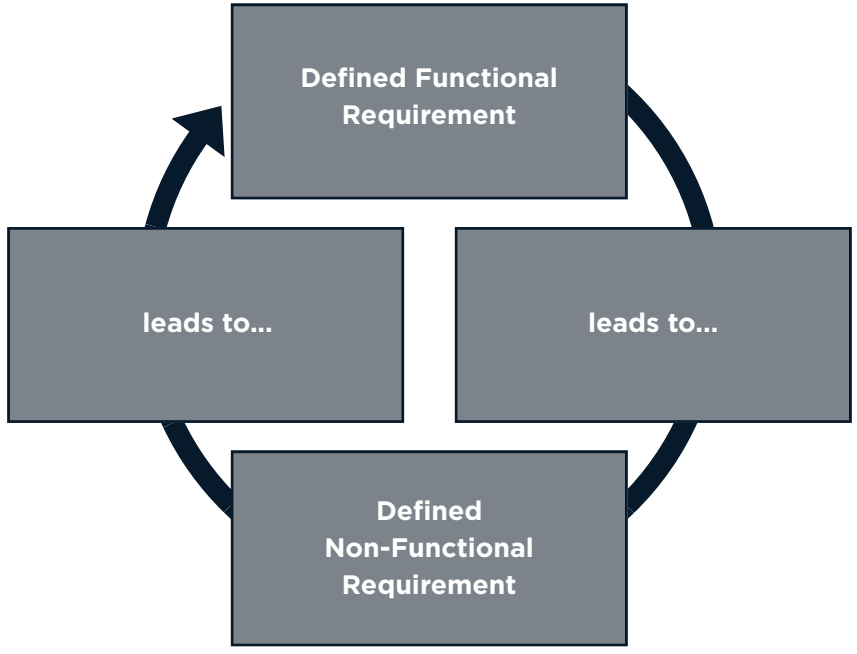
A good requirement is testable. How will you know when the requirement has been delivered successfully? Write requirements in a granular way so that no two requirements are ever combined. This will make it easier for the people who are turning your requirements into reality, and aid in later testing.

Make sure that requirements fully cover every scenario, which means including requirements detailing what the system shall *not* do, but be careful not to over-specify. Focus on the features that the users truly need. More is not always better and often leads to increased cost, diluted impact, and a bloated product that confuses users and is difficult to use. You should be able to trace back every requirement to one of the project objectives. It can be useful to benchmark against other companies in the same industry to understand what they are doing. Sometimes this means you aim for "just as good" as the competition, and sometimes it gives you a point from which to be better.

It is important when talking with users and stakeholders to try to understand the bigger picture of where their requirements are coming from and how they relate to the project objective. What are they really trying to do? Sometimes they will present a solution when they really should present a problem so the best solution can be brainstormed by a larger team. Also, be sure to understand their authority to make functional requirement requests. Usually, a project manager has the final say and should be consulted before adding additional requirements. As you gather requirements, document assumptions in a requirements traceability matrix so you can later go back to the person who requested the feature with any questions you may have. Specification development is frequently an iterative process. As the requirements are developed, be sure they are feasible and non-conflicting.

Defining requirements can be confusing. Although it is not always possible, try to make them understandable by non-technical stakeholders, and use visuals as much as possible to reinforce the information.

As a final takeaway, never consider non-functional requirements to be unimportant, despite the name. A website that takes 30 seconds to load might meet its functional requirements but is still not usable.



ABOUT QRA

QRA Corp's mission is to accelerate the design process across industries who are tackling the most complex systems by empowering them to build tomorrow's safe, secure, and incredibly powerful products. QRA's technology, patented toolsets and capabilities have been used to avoid stressful reworks, enable confident engineering, and find previously undetected catastrophic flaws.

QRA's requirements analysis tool, QVscribe, harnesses Natural Language Processing to automatically apply the best requirements analysis tactics by leading industry experts. Automated requirements analysis empowers engineering teams to build faster by identifying errors where they matter most – in the requirements.

To learn more about QVscribe and find additional helpful resources for improving your requirements and your RE processes, visit qracorp.com/qvscribe. To discover how QVscribe can help your organization improve and accelerate its requirements definition and analysis processes, click here to schedule an online demonstration.

REFERENCES

"EARS: The Easy Approach to Requirements Syntax." International Academy, Research, and Industry Association, www.iaria.org/conferences2013/filesICCGI13/ICCGI_2013_Tutorial_Terzakis.pdf.

"Functional Requirement." Wikipedia, 24 Aug. 2019, en.wikipedia.org/wiki/Functional_requirement.

"Functional and Nonfunctional Requirements: Specification and Types." AltexSoft, www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/.

"Functional Vs Non-Functional Requirements – Essential Elements of SDLC." Evoke Technologies, 26 June 2018, www.evoketechnologies.com/blog/functional-non-functional-requirements-sdlc/.

"Functional vs. Non-Functional: What's the Difference?" Systemation, 28 Sept. 2015, www.systemation.com/functional-and-non-functional-requirements-a-primer/.

Labunskiy, Evgeniy. "What Comes First: Functional or Non-Functional Requirements?" Medium, Scrum Ukraine, 24 Feb. 2017, medium.com/agiletransformation/what-comes-first-functional-or-non-functional-requirements-b3ee96424742.

"Non-Functional Requirement." Wikipedia, 23 Sept. 2019, en.wikipedia.org/wiki/Non-functional_requirement.

version 1

qracorp.com