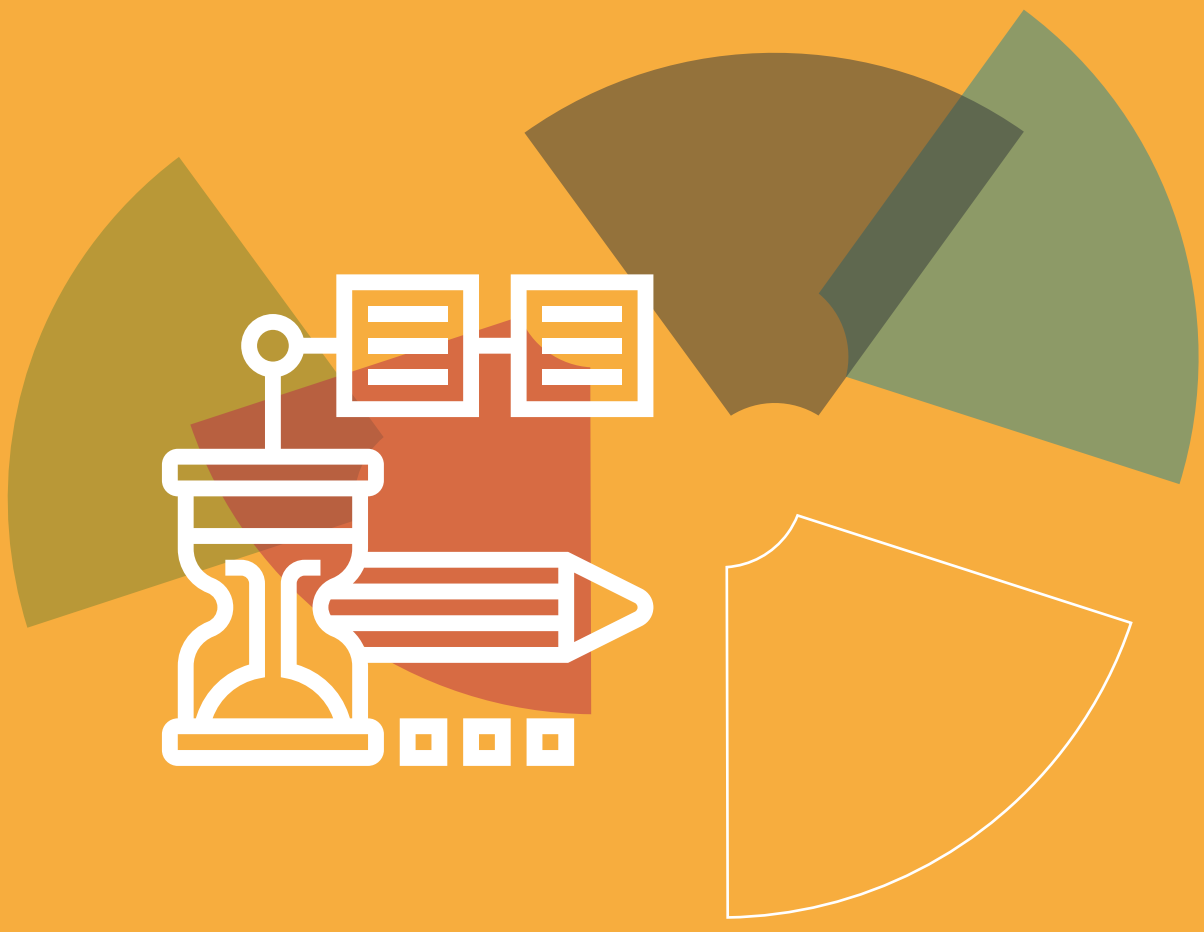
Non-functional Requirements

Making sure that NFRs are in place in order to deliver a fully packaged solution



Non-Functional Requirements

Introduction

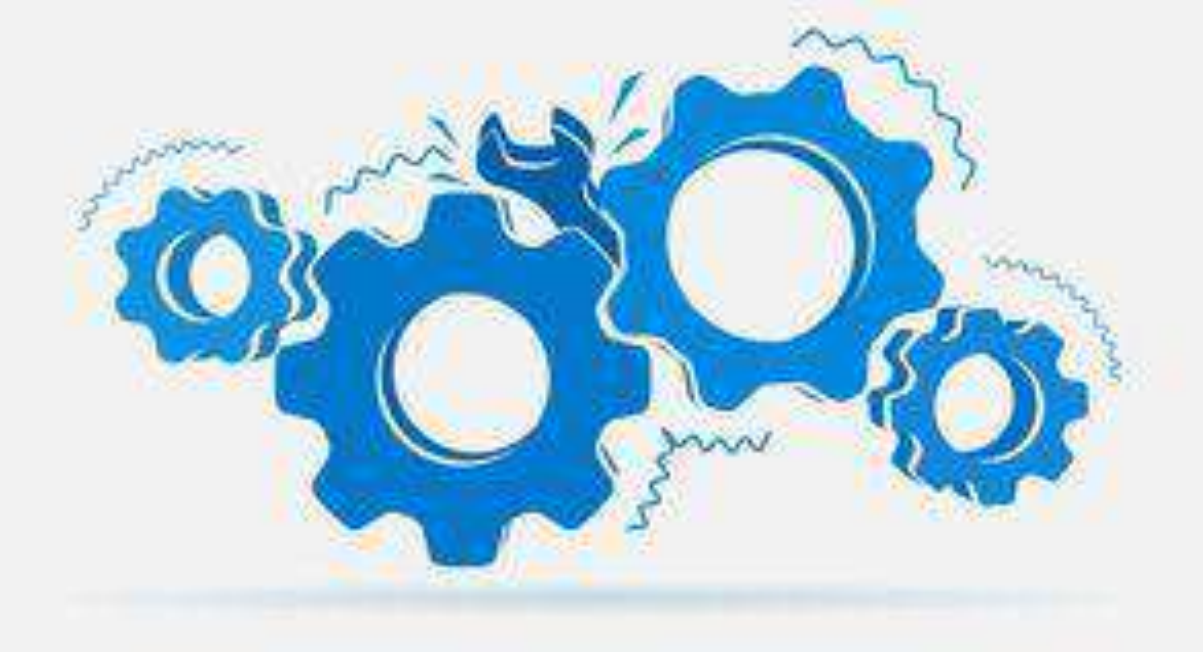
In systems engineering and requirements engineering, a **Non-Functional Requirement** is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Previously this would be mainly the **architecture** of the system. For example: scalability, accessibility, capacity. Which are all still important, however, we can dive a little deeper to categorize these NFR's.

So where do non-functional requirements fall into the world of product management?

We can use NFR's to make sure we deliver a fully packaged solution, we can produce a solution that solves the problem perfectly, but is that always enough to be a product success… or as much of a success as it could be? Using NFR's as stepping stones, we, as product managers, can deliver a world class product that delivers a solution as well as fulfilling on uptake, demand, delivery and support.

Crucially getting NFR's right is more important than ever right now. From an architecture perspective, you need to get this right or you will have heavy costs fixing issues in the future.

But more importantly you should be setting out to the business what are your expectations of the overall customer experience. You have to remember that software is not a solution on its own, everything from the purchasing journey to the training journey and how you are going to support your customers, are also part of the overall solution, and getting this right will retain customer loyalty.



info@tarigo.co.uk

Types of NFR

Whether the requirement is functional or non-functional depends on the detail of the requirement and, obviously, the service itself. It's clear that for a secure banking app, security is part of its primary function and therefore a functional requirement, in order to function and solve the problem, it needs to do 'this'. Once you have determined your non-functional requirements you can group them further into either execution qualities or evolution qualities, depending on what each NFR delivers.

Evolution qualities

These qualities are related to the static make-up of the system/service/product. The functions that fit into this category are therefore used to monitor aspects of system or service and thus inform product teams of the areas that need improvement, focus or enhancement. These NFR's can inform product teams with vital information about user features or help identify problem points in performance.

Testability	**Maintainability**
Reporting	**Modifiability**
Extensibility	**Scalability**

We can, therefore, use NFR's with evolution qualities to assess and collect information such as specific usage or activity in order to help us prioritise updates and customer needs.

info@tarigo.co.uk

Non-Functional Requirements

Execution qualities

Mitigate against creating pain points for the user with regards to the way the software or service is made and the **way** it works, these are observable during run time, and clearly show how the system or solution should ***be***. These are qualities that are closely linked to the capabilities surrounding user experience and observable by all.

These Include:

Accessibility	**Efficiency**
Usability	**Operability**
Maintainability	**Certification**
Security	**Stability**

Execution NFR's take into account aspects of the software service that are maybe taken for granted or simply expect to be in place in order to operate, use and access. If we are going to sign up to a subscription-based site then a payment mechanism that securely captures payment information has to be in place, it's expected of a platform but not required in order to deliver the actual solution.

The same can be said about a user-friendly navigation system, it would create a pain point for the user that doesn't need to exist if the navigation of the software was difficult to use or non-existent. Think of these qualities as smoothing out the experience of using your product.

info@tarigo.co.uk

Non-Functional Requirements

In a SaaS World

In the world of Software as a Service, NFR's are critical in delivering an exceptional all-round package – you can tick off all of the functional requirements and deliver a service that **DOES** what is needed but if you don't deliver any of your NFR's then how the system is supposed to **BE** could end up very different than intended. The result of this, customer resentment – a service that is unusable or difficult to understand, inefficiency and a lack of future proofing could result in total product failure. Even if the product is capable of delivering the right solution.

As an example of this – let's say we are about to release a new software service. An adware detection software that scans every new file that is read by the hard drive of your device. New downloads are automatically scanned for adware and so on. This is fairly straightforward from a functional requirement perspective, i.e. what the software has to **DO**. It needs to stay up-to-date in order to detect the latest adware and upon detecting any adware, be able to alert the user and contain the adware before it infects the system.
Great, the software **DOES** exactly what is required of it.

What about the NFR's that this software would need in order to deliver the full package? In order to work and function well for the user.
It wouldn't be a very good platform if the software acted as a bottleneck on files being downloaded, it would be an irritation to the user. The software would still **DO** what was required, but it wouldn't **BE** done well.

info@tarigo.co.uk

Important Considerations

When thinking of your NFR's, think about the experience you want your customers to have and draw out all these journeys, so that every area of the business is aware of their responsibility. Some questions to ask are:

- Are we going to offer free trials? If so, how are we going to support them? What do we need from development, sales, pre-sales, marketing?
- What is the work flow for customers to purchase? Do we have Account Managers selling? Are we expecting customers to purchase from a portal? Who across the business needs to be involved?
- Are we going to do all training online? Will this be self-service? Do we have partners who need to train the software? Are we going to sell the environments? Who across the business needs to be involved?
- How are we going to support our customers, once they are trained and on-board?

info@tarigo.co.uk

Non-Functional Requirements

Performance Testing

You can test the NFR's and identify any problems that need addressing with performance testing – of which there are three types you can use to highlight and prepare the service/product for market.

Load testing – under the conditions of usage, does the performance of the service live up to what is expected?

Stress testing – similar to load testing but to the extreme, this type of testing is designed to determine the maximum load that the service can handle and identify a possible breakpoint(s).

Volume testing – how does the system handle large volumes of data? Are there any glitches caused by high volumes of data traffic?

Testing is easily neglected but important in determining market readiness. Make sure, when planning your product roadmap, that you allow enough time for testing of the software. It's easy to forget that testing needs to be undertaken but it is necessary for assessing adequacy of the software and performance as well as evaluating the products readiness.

info@tarigo.co.uk

Conclusion

There are many more questions to answer and NFR's to think about, all of them will have a potential impact on the overall solution (incl. architecture) and customer satisfaction. Once you have been through this exercise, most NFR's will be the same for subsequent solutions. But always review them.

In summary, we are looking to deliver the best possible version of the solution that we can, this means a fully rounded out solution that can adapt and grow as it progress through its lifecycle. The key to this is understanding the environment that your software or service will exist in. How will each user journey look and what NFR's will make the journey smoother and easier? Looking at this from the perspective of the user allows you to identify NFR's that would otherwise be missed.

With existing services this can be done by conducting observational research, which is useful in highlighting large numbers of irritations to the user that when added together can cause frustration but on their own wouldn't even be considered.

info@tarigo.co.uk

Non-Functional Requirements

Get involved

We appreciate your feedback and thoughts. Join or start a discussion on market mapping. Share your examples, ask for feedback, let us know how it made a difference to your business.

More support

Want some more advice? Contact us on Expert@tarigo.co.uk and we'll be happy to help.