Table of Contents

How do we work with Non-Functional Requirements (NFRs)?

Non-functional Requirements (NFRs) are effectively constraints on what we are working to deliver. Even if the solution we deliver meets all the acceptance criteria, if solution doesn't also address the non-functional requirements there will be questions around whether we should deliver the item into production.

An example of a non functional requirement is your security standards. If you solution does not meet your security standards for delivery to your production system, then you can expect that you will not be able to deliver your solution to production.

If you don not address NFRs you will reduce you ability to deliver value to your customers. For this reason, as you write features (for example) you should capture and understand your nonfunctional requirements and make sure they are being incorporated in the discussion.

You might want to consider creating an "enabler", contributing to your architectural runway, to help you work through NFRs that have not been address as part of your normal activities.

In general and like all quality and compliance approaches, you are trying to "shift" this work "left" in the work process so that normal delivery of value just takes care of the requirement, rather than trying to build in the capability after you have designed and built the capability. To do this well you will need to reach out to the subject matter expert on the NFR (e.g. security, operations, etc.) to ensure you are really addressing the issue.

Non-Functional Requirements come in many different types, including:

1. Speed of transaction throughput (how many transactions per second are expected and can the code support it?)
2. Security (is the code being built secure, have any relevant security teams reviewed or assessed the code?)
3. Capacity (does the code being developed have resilience to high volumes of traffic built in?)
4. Stability (is the code stable to peak demands and other potential disruptions?)
5. Supportability (is the code supportable? For example is there alerting with quality documentation in place?)
6. Performance (is the code performant? Can it support expected volumes in normal traffic as well as peak demands?)

Want to Know More?

- https://www.scaledagileframework.com/nonfunctional-requirements/

FAQ, SAFe, NFRs, Features