

Non Functional Requirement (NFR)

Balasubramanian Swaminathan

PMP®, ACP®, CSM®, CSP®, SPC4.0®, AHF
Director – Global Programs, Digital Operations
[Enterprise Agile Coach and Leader]
GE Healthcare Digital

Copyright © 2018 by Balasubramanian Swaminathan. Permission granted to INCOSE to publish and use.



NFR – What are they?

- **Wikipedia definition:**
 - ❖ A **non-functional requirement** is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions.
 - ❖ The plan for implementing functional requirements are detailed in the system design.
 - ❖ The plan for implementing non-functional requirements are detailed in the system architecture.

NFR – What are they?

- ❖ NFR are often referred to as “-ilities.” Of course, not all NFR end in “-ility.” We also have security, performance, robustness and so on.
- ❖ It’s all about quality – also known as:
 - Technical requirements
 - Quality attributes
 - Quality of service requirements
- ❖ Specify “how well” the “what” must behave
- ❖ Impose constraints that typically cut across functional requirements

NFR – What are they?

- ❖ A constraint is a condition to make the requirements in line with quality expectations
- ❖ A constraint sets a limit to comply with
- ❖ It helps determine whether you have satisfied the non-functional requirements
- ❖ A constraint is addressed side by side with its linked functional scope
- ❖ A constraint should be satisfied in a finite period of time

Different Levels of NFR

- ❖ NFR's can be at different levels:
 - ❖ Portfolio Level – Themes
 - ❖ Value Stream Level* – Epics
 - ❖ Program Level – Features
 - ❖ Team Level – User Stories



NFR at Portfolio Level

- ❖ Examples of Portfolio level NFRs:
 - ❖ Single Sign-on
 - ❖ Restrictions on Open Source Usage
 - ❖ Common Security Requirements
 - ❖ Regulatory Standards



NFR at Program and Team Level

- ❖ Common place to find NFR's are at Program Level (System Quality)
- ❖ Team Level NFR's are important from implementation perspective
- ❖ If majority of NFR's are inherited at team level – It helps to foster built-in quality methods and avoid postponing NFR testing or delegating it to System Team.
- ❖ If not done correctly at team level (at right time), teams will be producing code that does NOT satisfy NFR and it would be very expensive to add NFR's later in the process



NFR at Team Level

- ❖ Team NFR's are norms of behavior that are agreed within team and between teams.
- ❖ Teams handle NFR as part of their DoR (Definition of Ready) and DoD (Definition of Done)
- ❖ Examples of DoD
 - ❖ Use of TDD (Test Driven Development) and the way tests are maintained
 - ❖ Coding Standard
 - ❖ Documentation Standard



Types of NFRs:

❖ Two types of constraint:

❖ **Internal quality**

- **Rule** is a constraint that sets a limit to comply during software construction
- Constraints to be obeyed during the implementation by the builders

❖ **External quality**

- **Restriction** is a constraint that sets a limit to comply at run time during software execution

Internal Quality: Constraint is a rule

❖ First category:

- maintainability (continuous integration; branching and merging),
- simplicity (ease to understand or explain; code layout convention),
- portability (ease to reuse for multiple platforms),
- Extensibility (ease to takes into consideration future growth)
- modifiability and
- testability (ease to confirm conformance by observing a reproducible behavior: red-green-refactor)

which are barely visible by stakeholders but simplify how to build the software

Story is not done until each rule is confirmed

External Quality: Constraint is a restriction

❖ Second category:

- Performance (response time or throughput testing),
- Reliability (testing over a period of time – memory leaks?) ,
- Robustness (simulate broken component),
- Correctness (acceptance testing),
- security (intrusion testing) ,
- Scalability (load testing),
- Usability (testing with real users)

which carry out the software functions at run time, and as such are not only visible to stakeholders but also highly desirable



Internal Quality vs External Quality

- ❖ Restriction is confirmed by test
- ❖ Restriction has recovery action
- ❖ Restriction has a measurable quality objective
- ❖ Restriction has to be SMART
 - Specific (It should target a piece of functionality that is small, consistent and simple)
 - Measurable (It imposes a limit that is measurable, otherwise how would you know when you have addressed it)
 - Attainable (It is recognized as achievable by the team)
 - Relevant (It is directly related, connected and pertinent to the NFR)
 - Traceable (It is linked with a requirement and a target that justifies why it exists)



Acceptance Criteria (AC)

- ❖ BDD (Behavior Driven Development) AC:
 - **Given a** <pre-condition>
 - **When an** <Action occurs>
 - **Then a** <post-condition>



BDD Acceptance Criteria - Examples

- ❖ BDD driven AC's example-1 (Successful account creation):
 - **Given** an unauthenticated user
 - **When** that user provides correct account sign-up data
 - **Then** an account is created
 - And an unique webpage is created
 - And an email confirmation is sent

- ❖ BDD driven AC's example-2 (Username already taken):
 - **Given** an unauthenticated user
 - **When** that user provides a user name that is associated to an existing account
 - **Then** an error message is presented to the user

NFR Implementation Approaches

❖ Two approaches exist:

- ❖ **All-at-once.** Some NFRs appear as new, immediate, architectural features, and theoretically negotiable or not, you just have to do them now.
 - ❖ For example, a new regulatory rule for derivative trading, if not immediately accommodated, could take the company completely out of the market, or cause a regulatory violation.
- ❖ **Incremental story-by-story path.** Other times the teams have options. For example the need for “substantially improved performance” can be dealt with over time, one story at a time.



NFR Testing

Discussion at conference



Collaboration of System Team and Agile Team – NFR Testing

Discussion at conference



Examples of NFR as a User Story

- ❖ As a customer, I want to be able to run your product on all versions of Windows from Windows XP on.
- ❖ As the CTO, I want the system to use our existing orders database rather than create a new one, so that we don't have one more database to maintain.
- ❖ As a user, I want the site to be available 99.999 percent of the time I try to access it, so that I don't get frustrated and find another site to use.
- ❖ As someone who speaks a Latin-based language, I might want to run your software someday.
- ❖ As a user, I want the driving directions to be the best 90 percent of the time, and reasonable 99 percent of the time.



Reference Materials

- ❖ Essential XP – Ron Jeffries
- ❖ Agile Requirements Breakdown Structure – David Bulkin
- ❖ Addressing Non-Functional Requirements with Agile Practices – Mario Cardinal
- ❖ Mountain Goat Software – Mike Cohn
- ❖ Scaled Agile Framework (SAFe 4.0)





Thank you for attending!
Share your experiences at #HWGSEC

