



Non-Functional Requirements Template

Prepared By: Shehzad Rafique

Business Analyst

(Strategic Planning & Architecture Wing)

Electronic Government Directorate

Non-Functional Requirements.....	3
Categorization of Non-Functional Requirements	3
· Qualities of the System.....	3
· <i>Run-time Qualities: Informally we can think of functional requirements capturing what the system must do, and the run-time qualities as describing how well these functional requirements are satisfied, Run-time qualities include:.....</i>	3
· <i>Development-time Qualities.....</i>	5
· Constraints of the System.....	5
Sources of Non-Functional Requirements	6
· System Constraints	6
· User Objectives, Values and Concerns	6
· Analysis of Features.....	6
· Development Organization Objectives, Values and Concerns	7
Template for Capturing and Documenting Non-Functional Requirements	7

Non-Functional Requirements

The qualities we desire of a problem solution other than those concerning its functionality, e.g. its robustness, its efficiency, its security, its extensibility, its maintainability, its portability, etc.

Categorization of Non-Functional Requirements

To make it easier to capture non-functional requirements, we broadly organize them into two categories:

- **Qualities of the System:** *Qualities* are properties or characteristics of the system that its stakeholders care about and hence will affect their degree of satisfaction with the system. Qualities can further be divided into two categories based on executing system and work products.
 - **Run-time Qualities:** Informally we can think of functional requirements capturing what the system must do, and the run-time qualities as describing how well these functional requirements are satisfied, Run-time qualities include:

Usability	Describes the ease with which the system can be learned or used. A typical usability requirement might state: <ul style="list-style-type: none">• The system should allow users to install and operate it with little or no training.
Supportability	Refers to the software's ability to be easily modified or

	<p>maintained to accommodate typical usage or change scenarios.</p> <p>A typical example of supportability requirement is:</p> <ul style="list-style-type: none"> • The system shall allow users to create new workflows without the need for additional programming.
Availability/ Reliability	<p>Specifications for reliability typically refer to availability, mean time between failures, mean time to repair, accuracy, and maximum acceptable bugs. For example:</p> <ul style="list-style-type: none"> • The system shall meet the terms of a Service Level Agreement.
Quality of Service Requirements	<p>It typically refers to response time, transaction throughput, and capacity. For example:</p> <ul style="list-style-type: none"> • All Web pages must download within three seconds during an average load, and five seconds during a peak load
Security	<p>Security refers to the ability to prevent and/or forbid access to the system by unauthorized parties. Some examples of security requirements are:</p> <ul style="list-style-type: none"> • User authentication shall be via the corporate Single Sign-on system.
Scalability	<p>The ease with which a system or component can be modified to handle a large increase in users, workload or</p>

	transactions .
--	----------------

- **Development-time Qualities:** In the case of the work products, the qualities are driven by the development organization’s goals. We will refer to these as *development-time qualities*. Examples of development-time quality requirements are:

Localizability	Ability to make adaptations due to regional differences
Modifiability or Extensibility	Ability to add (unspecified) future functionality
Evolvability	Support for new capabilities or ability to exploit new technologies
Composability	Ability to compose systems from components
Reusability	Ability to (re)use in future system

- **Constraints of the System:** *constraints* are characteristics of the environment (i.e., the larger system into which the system under development will fit) or the development organization that constrain the development in some way. Examples include the target operating system or hardware platform in the case of the user environment, or the skill-set of available developers in the case of the development organization.

Sources of Non-Functional Requirements

Run-time non-functional requirements arise from the operating environment, the user(s), and competitive products:

- ***System Constraints.*** Here one is looking for elements of the environment into which the system must fit, that may serve as constraints on the system. This may be true of the installed infrastructure (e.g., hardware and OS platforms) or legacy applications, or may be in the form of organizational factors or the process that the system will support.
- ***User Objectives, Values and Concerns.*** In establishing the run-time qualities for a system, it is important to identify all the categories of user (including other systems) that will interact with the system, and understand what quality attributes they care about.. The requirements team should nonetheless be alert to requirements that users take for granted or are not able to articulate directly. The qualities may need to be translated by developers from user-level objectives, values and concerns into specific technical quality requirements. For example, a user's requirement not to be impeded by slow system performance in conducting a task may be translated into requirements on transaction throughput.
- ***Analysis of Features.*** Run-time qualities are often associated with product features. *Features* are generally thought of as the characteristics of the product that establish its competitiveness, frequently distinguishing the product functions (base-line and unique product differentiators) with at least one quality attribute. For example, many web-based

catalog services have on-line payment options. To allay market concern, the electronic payment feature includes transaction security as an essential attribute.

- ***Development Organization Objectives, Values and Concerns.***

Stakeholders in the development organization include strategic management (e.g., general manager and R&D/IT manager), program and project managers, architects, developers, quality assurance (testers), marketing and manufacturing engineers, etc.

Their objectives, values and concerns may relate to the business performance, schedules, productivity and effectiveness, etc. For example, strategic management establishes the product portfolio plan, including planned releases (which modules in what timeframe). The architects and technical managers may translate those portfolio objectives into development-time quality requirements such as extensibility, evolvability and reuse, knowing that the portfolio cannot be accomplished without these characteristics..

Template for Capturing and Documenting Non-Functional Requirements

<u>Category</u>	<u>Property</u>	<u>Example</u>
------------------------	------------------------	-----------------------

Run-time Qualities		
	Usability	The online help should be incorporated into the system
	Supportability	
	Availability/Reliability	Downtime <= 1 Minute
	Scalability	
	Security	Digital Certificates should be used in transactions for authentication purpose
	Quality of Service Requirements	System should be able to process 100 transactions per minute
Development-time Qualities		
	Localizability	System should support Multicurrency
	Extensibility/Modifiability	Development of 3-tier or N-tier application
	Evolvability	
	Composability	Application components must be developed in platform-independent and portable language for example Java
	Reusability	Application components must be developed in platform-independent and portable language for example Java
Constraints		

	Interface Constraints	System should be able to communicate with the neighboring systems through web services
	Protocol Constraints	Applications should be CORBA and IIOP Protocol compliant
	Platform Constraints	Application server should run on Unix
	Architectural Constraints	J2EE architecture
	Design and development Constraints	Development of 3-tier or N-tier application
	Networking Constraints	Use CAT 6 for LAN cabling