**TEMPLATE**

Nonfunctional Requirements Definition

<Insert Project Name>

Release is a number for the document being released for review. A release can go through many changes between the versions. That’s why the SaveDate field below is important – to distinguish between Versions of a Release.

Version: Major.Minor

This template developed by Requirements Quest® is based on IEEE (Institute of Electrical and Electronics Engineers) 830 Standard SRS (Software Requirements Specification). For additional information, visit <http://www.ieee.org>. To request an electronic copy of this template, send an email to inquire@requirementsquest.com or visit [www.RequirementsQuest.com](http://www.RequirementsQuest.com).



Table of Contents

[1. Introduction 3](#_Toc471827508)

[1.1. Purpose of this document 3](#_Toc471827509)

[1.2. Reference Materials 3](#_Toc471827510)

[1.3. Specific Terms and Acronyms 3](#_Toc471827511)

[1.4. Assumptions 3](#_Toc471827512)

[1.5. Constraints 3](#_Toc471827513)

[1.6. Dependencies 3](#_Toc471827514)

[2. Nonfunctional Requirements 4](#_Toc471827515)

[2.1. Nonfunctional Requirement Statements 4](#_Toc471827516)

[2.2. Common Information 5](#_Toc471827517)

[3. Appendices 6](#_Toc471827518)

[3.1. Revision History 6](#_Toc471827519)

[3.2. Validation History 6](#_Toc471827520)

[3.3. Requirements Issues 6](#_Toc471827521)

[3.4. Attachments 6](#_Toc471827522)

# Introduction

[Insert Relationship Map- provides a broad view of the entities impacted by the project scope.]


## Purpose of this document

This document contains all nonfunctional requirements for this project.

This document must establish an agreement between the customer and suppliers on what the system must do. The contents of this document should endeavor to reduce development effort by reducing redesign due to ambiguous statements. It should contain enough detail to provide a basis for estimating cost and schedules. Each version should provide a baseline for verification and validation. It should facilitate “transfer” of knowledge, commitment, intent and finally transfer of the viable product to the stakeholders. The final version of this document should accurately and completely describe the final product and serve as a basis for future enhancements.

Both the writer and the reader should strive for these goals in the elicitation, analysis, presentation and validation of this document. Through baselining, versions of this document should clearly identify all changes.

## Reference Materials

There are many other documents that together describe the complete set of requirements for this project.

Other documents such as standards, regulations, business process flows, or descriptions of existing functionality to be modified should be named here. Often there are copies of common information distributed throughout the company. The definitive source of the information and/or the person who manages the information should be noted so that discrepancies may be quickly resolved with clear authority.

| **Reference Document Name** | **Brief Description** | **Location of Definitive Source** |
| --- | --- | --- |
|  |  |  |

## Specific Terms and Acronyms

Terms here are specific to understanding the content of this document.

Do not copy terms from the Glossary to this section. Describe the meaning and usage of any terms that are used in any of the requirements documents. Especially include a term if its meaning could be interpreted differently by members of the project team. If the company has a common definition of terms document or list of acronyms, it may be attached separately but the document must be referenced here.

| **Term or Acronym** | **Description** |
| --- | --- |
|  |  |

## Assumptions

Identify anything that adds clarification to or provides background information about the nonfunctional requirement statement, or other related item.

Assumption statements are not specific requirements themselves. They help clarify requirements that may be misinterpreted due to different definitions of terms or different opinions about how some business operation flows. During validation of requirements, assumptions may help discover omitted requirements, highlight project risks, or encourage discussion of various interpretations. Examples include: technology such as a specific operating system or database; and expectations of the development team.

| **ID** | **Assumption Statement** | **Related To** |
| --- | --- | --- |
| A1 |  |  |

## Constraints

Identify anything that puts limits on implementing the nonfunctional requirements.

This section is often misunderstood. Pay particular attention to this description. A constraint is a statement that expresses measurable bounds on the implementation of a system. It may limit the technology or environment in which the system may be built. It may list the certain options that may be considered for implementation. It may describe limitations or conditions that users must work under. In all cases, a “constraint” is differentiated from business, user and functional requirements in that it does not describe the functionality but rather limits the design or implementation options. Examples of constraints include: hardware constraints because the software may reside on a server with other applications and compete for resources such as memory, ports, and processor speed; reliability constraints such as how often the system must be backed up or even if hot back such as RAID is required; and “criticality” refers to how long the system would be allowed to be out of operation due to some failure. These are just a few of the constraints that may make the project a success.

| **ID** | **Constraint Statement** | **Related To** |
| --- | --- | --- |
| C1 |  |  |

## Dependencies

Detail any external event, condition, or system that must be in place for a requirement to be implemented.

| **ID** | **Dependency Statement** | **Related To** |
| --- | --- | --- |
| D1 |  |  |

# Nonfunctional Requirements

## Nonfunctional Requirement Statements

Nonfunctional requirements focus on the qualities that must be applied to design and implement a system. These are specific standards and attributes in support of the other requirements. For detailed information about nonfunctional requirements, including over 2,000 suggested elicitation questions, reference
***The Quest for Software Requirements***, by Roxanne E. Miller, [www.RequirementsQuest.com](http://www.RequirementsQuest.com).

Column Header Key:
BR = Business Rules Identifier, CI = Common Information Identifier, S = Status, P = Priority

| **ID** | **Nonfunctional Requirement Statements** | **BR** | **CI** | **S** | **P** |
| --- | --- | --- | --- | --- | --- |
| **OPERATION Requirements**: How well does the system perform for daily use?Describe the user concern for using the functionality. The user perceives the system as a tool to automate tasks. |
| **Access Security** How well is the system guarded against unauthorized access?The extent to which the system is safeguarded against deliberate and intrusive faults from internal and external sources. |
| N-ACS1 |  |  |  |  |  |
| **Accessibility** How easy is the system to use by people with varying capabilities?The extent to which the software system can be used by people with the widest range of capabilities to achieve a specified goal in a specified context of use. |
| N-ACC1 |  |  |  |  |  |
| **Availability** How dependable is the system during normal operating times?The degree to which users can depend on the system to be up (able to function) during “normal operating times.” |
| N-AVL1 |  |  |  |  |  |
| **Confidentiality** How well does the system make sensitive data available to authorized users?The degree to which the software system protects sensitive data and allows only authorized access to the data. |
| N-CNF1 |  |  |  |  |  |
| **Efficiency** How fast can it process? How many can be processed? How well does the system respond?The extent to which the software system handles capacity, throughput, and response time. |
| N-EFC1 |  |  |  |  |  |
| **Integrity** How accurate and authentic are the data?The degree to which the data maintained by the software system are accurate, authentic, and without corruption. |
| N-INT1 |  |  |  |  |  |
| **Reliability** How immune is the system to failure?The extent to which the software system consistently performs the specified functions without failure. |
| N-REL1 |  |  |  |  |  |
| **Safety** How well does the system prevent harm to people and the environment?The degree to which a software system prevents harm to people or damage to the environment in the intended context of use. |
| N-SAF1 |  |  |  |  |  |
| **Survivability** How resilient is the system from failure?The extent to which the software system continues to function and recovers in the presence of a system failure. |
| N-SRV1 |  |  |  |  |  |
| **Usability** How easy is it to learn and operate the system?The ease with which the user is able to learn, operate, prepare inputs, and interpret outputs through interaction with a system. |
| N-USE1 |  |  |  |  |  |
| **REVISION Requirements**: How easy is it to correct errors and add functions?Describe the user concern for changing source code or data that drive the system. The user perceives the system as programmed language statements. |
| **Flexibility** How easy is it to modify to work in different environments?The ease with which the software can be modified to adapt to different environments, configurations, and user expectations. |
| N-FLX1 |  |  |  |  |  |
| **Maintainability** How easy is it to upkeep and repair the system?The ease with which faults in a software system can be found and fixed. |
| N-MNT1 |  |  |  |  |  |
| **Modifiability** How easy is it to change the software system, and at what cost?The degree to which changes to a software system can be developed and deployed efficiently and cost effectively. |
| N-MDF1 |  |  |  |  |  |
| **Scalability** How easy is it to expand or upgrade the system’s capabilities?The degree in which the system is able to expand its processing capabilities upward and outward to support business growth. |
| N-SCL1 |  |  |  |  |  |
| **Verifiability** How easy is it to show the system performs its functions?The extent to which tests, analysis, and demonstrations are needed to prove that the system will function as intended. |
| N-VER1 |  |  |  |  |  |
| **TRANSITION Requirements**: How easy is it to adapt to changes in the technical environment?Describe the user concern for managing the upkeep of the software. The user perceives the system to have characteristics similar to hardware. |
| **Installability** How easy is it to install, uninstall, and reinstall the software system?The ease with which a software system can be installed, uninstalled, or reinstalled into a target environment. |
| N-INS1 |  |  |  |  |  |
| **Interoperability** How easy is it to interface with another system?The extent to which the software system is able to couple or facilitate the interface with other systems. |
| N-IOP1 |  |  |  |  |  |
| **Portability** How easy is it to transport?The ease with which a software system can be transferred from its current hardware or software environment to another. |
| N-POR1 |  |  |  |  |  |
| **Reusability** How easy is it to convert for use in another system?The extent to which a portion of the software system can be converted for use in another. |
| N-REU1 |  |  |  |  |  |

## Common Information

In the Nonfunctional Requirement Statements above, specific information that is referenced multiple times may be described once here. This “named information” may then be referenced by its name with quotes around it in the rest of the document.

| **ID** | **Named Information** | **Related Req. ID** | **Definition or Business Usage / Business Elements** | **Definitive Source** |
| --- | --- | --- | --- | --- |
| CI1 |  |  |  |  |
| CI2 |  |  |  |  |
| CI3 |  |  |  |  |

# Appendices

## Revision History

Each time a change is made to the document (even if several are released together) make a short note describing the change briefly. Do not use non-descript statements like: “Modified issues” or “Changed some assumptions”. Rather, make statements like: “added issues 21 through 27, resolved issues 8, 17, 18.” To determine what version a change was released, review the “save date” of the file on the title page. Verify that all participants are using the same version of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| **Change Date** | **Changed by** | **Description of Change** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Validation History

The Requirements Management Process suggests the following 5 stages: Elicitation, Analysis, Representation, Validation, and Change Control. Upon successful review and approval, the version of the document Approved should be “Baselined”. That is, a copy of the version should be archived so that future changes can be compared

**Participant Index**

|  |  |  |
| --- | --- | --- |
| **ID** | **Stakeholder Name** | **Specific Role or Area of Expertise** |
| Supplier 1 |  |  |
| Supplier 2 |  |  |
| Supplier 3 |  |  |
| Supplier 4 |  |  |
| Receiver 1 |  |  |
| Receiver 2 |  |  |
| Receiver 3 |  |  |
| Receiver 4 |  |  |

**Outcomes: A = Accept, C = Accept with Conditions, R = Reject**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Review Date** | **Overall Outcome** | **Supplier Outcome(s)** | **Receiver Outcome(s)** | **Identified Issues** |
| S1 | S2 | S3 | S4 | R1 | R2 | R3 | R4 |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

## Requirements Issues

| **ID** | **Description** | **Raised by** | **Assigned to** | **Status** |
| --- | --- | --- | --- | --- |
| IS1 |  |  |  |  |
| IS2 |  |  |  |  |
| IS3 |  |  |  |  |
| IS4 |  |  |  |  |

## Attachments

Below are supplemental documents that help to illustrate and define the nonfunctional requirements.

Either name here and manually attach separate documents or link to those documents from this document. Use “Insert > File…” to include other documents. Attach Context Diagrams, Graphical Models, Use Cases or other documents as needed.

### Attachment