

|  |
| --- |
| Non-functional requirements |
| guidelines & structure |
|  |
| Abhijit Majumdar | october 02, 2022 |

Table of Contents

[**1.0** **Introduction** 4](#_Toc297895566)

[1.1 Purpose 4](#_Toc297895567)

[1.2 Definition of Non Functional Requirements 4](#_Toc297895568)

[1.3 Objective 4](#_Toc297895569)

[1.4 Intended Audience 4](#_Toc297895570)

[1.5 Acronyms 4](#_Toc297895571)

[1.6 Steps to elicit Non-Functional Requirements: 5](#_Toc297895572)

[1.7 Challenges 7](#_Toc297895573)

[1.8 NFR List 8](#_Toc297895574)

[**2.0** **Non-Functional Requirements** 9](#_Toc297895575)

[2.1 Adaptability 9](#_Toc297895576)

[2.2 Administration 10](#_Toc297895577)

[2.3 Application Controls 11](#_Toc297895578)

[2.4 Audit 12](#_Toc297895579)

[2.5 Availability 13](#_Toc297895580)

[2.6 Communication interfaces 14](#_Toc297895581)

[2.7 Compatibility 14](#_Toc297895582)

[2.8 Configuration Management 15](#_Toc297895583)

[2.9 Content Management 16](#_Toc297895584)

[2.10 Consistency 16](#_Toc297895585)

[2.11 Correctness 17](#_Toc297895586)

[2.12 Data Retention 18](#_Toc297895587)

[2.13 Delivery 19](#_Toc297895588)

[2.14 Disaster Recovery 19](#_Toc297895589)

[2.15 Fault Tolerance 20](#_Toc297895590)

[2.16 Flexibility 20](#_Toc297895591)

[2.17 Infrastructure 21](#_Toc297895592)

[2.18 Interoperability 21](#_Toc297895593)

[2.19 Legislative 23](#_Toc297895594)

[2.20 Localizability 23](#_Toc297895595)

[2.21 Maintainability 23](#_Toc297895596)

[2.22 Performance 25](#_Toc297895597)

[2.23 Portability 26](#_Toc297895598)

[2.24 Privacy 28](#_Toc297895599)

[2.25 Reliability 28](#_Toc297895600)

[2.26 Robustness 29](#_Toc297895601)

[2.27 Safety 30](#_Toc297895602)

[2.28 Security & Access Controls 31](#_Toc297895603)

[2.29 Serviceability 34](#_Toc297895604)

[2.30 Usability 34](#_Toc297895605)

[2.31 Volume and scalability 35](#_Toc297895606)

[2.32 Online Report 37](#_Toc297895607)

[2.33 Session Management 38](#_Toc297895608)

[2.34 Installability 38](#_Toc297895609)

[2.35 Internationalization Features 39](#_Toc297895610)

[2.36 Documentation and Training 39](#_Toc297895611)

[2.37 Extensibility 40](#_Toc297895612)

[2.38 Reusability 41](#_Toc297895613)

[2.39 Standards 42](#_Toc297895614)

[2.40 Messaging 42](#_Toc297895615)

[**3.0** **Special Considerations for projects with Sarbanes-Oxley (SOX) consideration** 44](#_Toc297895616)

[**4.0** **Segregation of Duties** 45](#_Toc297895617)

[**5.0** **NFR Testability and Measurability** 46](#_Toc297895618)

[5.1 Definition 46](#_Toc297895619)

[5.2 Sample Requirements 46](#_Toc297895620)

[5.3 Related NFR/ Categories 46](#_Toc297895621)

[5.4 Context 46](#_Toc297895622)

[5.5 Sample Questions/ Considerations 46](#_Toc297895623)

[**6.0** **References** 47](#_Toc297895624)

[6.1 Information Source/References 47](#_Toc297895625)

### **Introductio****n**

## Purpose

The purpose of this document can broadly be listed as follows

* + - Provide a list and definition of commonly used non-functional requirements (NFR)
    - Questions that could be considered while eliciting the requirements for these NFR.
    - Sample requirements that can be stated as “specifications” that can be leveraged while eliciting requirements
    - Guidelines on how to validate and use the non functional requirements in a project

The list provided in this document can be revised based on specific project needs and additional NFR can be added to the list as needed. The NFR listed in this document, must align with the Requirements Management Plan created for a project.

## Definition of Non Functional Requirements

A software requirement that describes how the software will function rather than what the software will do is called non-functional requirement (NFR). A few examples are software performance requirements and software usability attributes.

## Objective

Non-Functional Requirements are gathered from the business and technical users and these complement the functional requirements for the solution. Often these requirements are overlooked until the later stages of the project. This may lead to missed requirements which can result in a negative experience for the end user. Therefore, it is very important that the requirements team pays special attention in the elicitation of the non-functional requirements.

## Intended Audience

This document is intended for new and existing Business Analysts and Business Architects, who are being on-boarded, in order to provide an understanding of non-functional requirements as well as provide a list of non-functional requirements which can be used by the Business Architects as a point of reference when eliciting requirements during the requirements and architecture stage of a project.

## Acronyms

| **Acronyms** | **Description** |
| --- | --- |
| BU | Business Unit |
| DB | Database |
| Config | Configuration |
| URL | Uniform Resource Locator |
| GUI | Graphical User Interface |
| PDA | Personal digital assistant |
| PPT | Power point |
| SLA | Service Level Agreement |
| CPU | Central processing unit |
| QA | Quality Assurance |
| MIS | Management Information System |
| SEC | Securities and Exchange Commissions |
| BA | Business Architect |
| OS | Operating Systems |
| NFR | Non Functional Requirement |
| SLA | Service Level Agreement |
| ERISA | Employee Retirement Income Security Act |

## Steps to elicit Non-Functional Requirements:

The following diagram gives an overview of the approach the requirements team can use to effectively elicit the non-functional requirements.

**Step 1: Identify NFR**

The requirements team should reference the Project Charter and the Integrated Plan in order to get an understanding of the high level scope of the project for which they will be eliciting requirements. Through these documents, the requirements team will also get an understanding of the area (both, in terms of domain and technology) in which the project will function. This understanding will enable the requirements team to create an initial list of potential non-functional requirements (from the complete NFR list which is available in the job aid spreadsheet, which is also available in the NFR Guideline document and the Requirements Management Plan) based on their understanding of the scope of the project. The job aid spreadsheet contains a list of NFR as well as their standard values that can be re-used, by the requirements team, across various projects. The classification of the requirements should be referenced from the NFR Guideline in this stage.

**Step 2: Analyze Benchmarks and Elicit Non Functional Requirements**

The requirements team should reference the Job Aid spreadsheet in order to validate the applicable NFR from the spreadsheet during the requirement elicitation sessions. The requirements team should also ensure that the NFR are applicable for the project and their values are accurate and still apply to the organization (through discussions with the technical teams).

As part of this step, the requirements team must validate the classification of requirements done as part of STEP 1

**Step 3: Update the Requirements Management Plan with any new Non Functional Requirements**

If new requirements/classifications of requirements are validated as part of Step 1 and 2, additions must be updated in any job aid spreadsheet or NFR Guideline document and the Requirements Management Plan. It is important to ensure that there is traceability between NFR and the details listed in the NFR Guide and the Requirements Management Plan.

## Challenges

Some of the common reasons why Non Functional requirements are difficult to identify and test:

* They can be interpreted differently by different users. For example, for an expert, certain documentation and training material may be easy to understand and learn, however, for an entry level person this may prove to be a difficult task.
* The relevance or importance of a NFR might vary depending on the system or products and services under consideration. For example, in the financial services industry, accuracy or correctness of information and software security would be amongst the most important quality attributes.
* Goals of two different NFR can be conflicting. For example, the extra code and processing requirements to increase usability usually increases processing run-time, thus hampering performance.

## NFR List

This section lists NFR along with their classifications. It should be noted that some of the NFR have overlap with others. The validation of the applicable NFR needs to be done against this list.

This document provides the definitions of these NFR, related NFR, Categories, and sample questions/considerations which would help elicit requirements.

| **NFR Name** | **NFR Group** | **Stage Applicable** |
| --- | --- | --- |
| [Adaptability](#_Adaptability) | Revision | Architecture |
| [Administration](#_Administration) | Revision | Architecture |
| [Application Controls](#_Application_Controls) | Operation | Architecture |
| [Audit](#_Audit) | Operation | Architecture |
| [Availability](#_Availability) | Operation | Architecture |
| [Communication interfaces](#_Communication_interfaces) | Operation | Architecture |
| [Compatibility](#_Compatibility) | Transition | Architecture |
| [Configuration](#_Configuration_Management) | Revision | Architecture |
| [Content Management](#_Content_Management) | Operation | Architecture |
| [Consistency](#_Consistency) | Operation | Architecture |
| [Correctness](#_Correctness) | Operation | Architecture |
| [Data Retention](#_Data_Retention) | Revision | Architecture |
| [Delivery](#_Delivery) | Operation | Architecture |
| [Disaster Recovery](#_Disaster_Recovery) | Revision | Architecture |
| [Fault Tolerance](#_Fault_Tolerance) | Operation | Design |
| [Flexibility](#_Flexibility) | Operation | Design |
| [Infrastructure](#_Infrastructure) | *Overlap with scalability, disaster recovery, data backup, security, performance and serviceability* | |
| [Interoperability](#_Interoperability_) | Transition | Architecture |
| [Legislative](#_Legislative) | Operation | Architecture |
| [Localizability](#_Localizability) | Operation | Design |
| [Maintainability](#_Maintainability) | Revision | Architecture |
| [Performance](#_Performance) | Operation | Architecture |
| [Portability](#_Portability) | Transition | Architecture |
| [Privacy](#_Privacy) | Operation | Architecture |
| [Reliability](#_Reliability) | Operation | Architecture |
| [Robustness](#_Robustness) | Operation | Architecture |
| [Safety](#_Safety) | Operation | Architecture |
| [Security & Access Controls](#_Security_&_Access) | Operation | Architecture |
| [Serviceability](#_Serviceability) | Revision | Architecture |
| [Usability](#_Usability) | Operation | Architecture |
| [Volume & Scalability](#_Volume_and_scalability) | Revision | Architecture |
| [Online report](#_Online_Report) | Operation | Architecture |
| [Session Management](#_Session_Management) | Operation | Architecture |
| [Installability](#_Installability) | Operation | Design |
| [Internationalization](#_Internationalization_Features) | Operation | Architecture |
| [Documentation and Training](#_Documentation_and_Training) | Operation | Development and Testing |
| [Extensibility](#_Extensibility) | Revision | Design |
| [Reusability](#_Reusability) | Transition | Design |
| [Standards](#_Standards) | Operation | Design |
| [Messaging](#_Messaging) | Operation | Design |

### **Non-Functional Requirements**

## Adaptability

* + 1. **Definition**

Adaptability is a measure of a system to adapt to different environments, configurations, and user expectations. An adaptable system is designed in such a manner so that any new requirement changes in the near future would have minimal impact on the system’s current design and/or code. This is also referred to as flexibility.

* + 1. **Sample Requirements**
* The system shall be able to incorporate changes in error messages without having to change the source code.
* The system shall be able to incorporate a new business rules without having to change the source code.
  + 1. **Related NFR/ Categories**

[Configuration](#_Toc292135536), [Extensibility](#_Toc292190476), [Flexibility](#_Toc292127054), [Reusability](#_Reusability), [Scalability](#_Toc292216232)

* + 1. **Context**

This would be applicable for systems which are required to handle multiple objects at the same time, each of which typically has its own user interface with data that are distinctly separate from the others. For example, a multi-lingual system will have different user interfaces for each language supported. This might also be applicable in systems which need to switch from using a specific set of components to using another set. For example, security software shall be configured to switch from workday mode to weekend mode.

Flexibility is generally more applicable when building Commercial-Off-The-Shelf (COTS) software or a software product that is intended to be sold to multiple organizations.

* + 1. **Sample Questions/ Considerations**

1. Are there any special requirements regarding adaptation of the software (including upgrading)?
2. Should deployment tools be provided with the development framework, or should it integrate smoothly with third party deployment tools?
3. Should third party component or self-written components be easily “pluggable” with the application?
4. Is there a limit on how long a new component may talk to a “Plug in” from start to finish, including testing?

## Administration

* + 1. **Definition**

System administration is related to installing and maintaining computer systems, and responding to service outages and other problems.

* + 1. **Sample Requirements**
* The system shall have an administrative console to monitor and administer systems/applications/services remotely.
* The system shall provide the capability for site administration, including managing security roles and privileges as well as configurations (if any).
  + 1. **Related NFR/ Categories**

[Availability](#_Availability), [Maintainability](#_Toc292135553), [Reliability](#_Reliability), [Robustness](#_Toc292135559), [Serviceability](#_Toc292135567)

* + 1. **Context**

This is relevant when there is a requirement to monitor and administer systems/applications/services.

* + 1. **Sample Questions/ Considerations**

1. How does the IT owner and/or business owner plan to tackle the administration of the system?
2. What are the functionalities / privileges that needs to be handled by the system administrator?
3. Is there a requirement to provide an administrative console to monitor and administer systems/applications/services remotely?

## Application Controls

* + 1. **Definition**

Application controls are programmed and/or manual procedures designed to prevent or detect and correct errors or anomalies in the processing of business transactions.

* + 1. **Sample Requirements**
* All processing changes to date shall be logged in a table
* Reports containing information of confidential nature shall be printed under controlled conditions.
  + 1. **Related NFR/ Categories**

*See section 3.0 related to* [*SOX*](#_Special_Considerations_for)

* + 1. **Context**

This standard applies to new system development work effort, that is, all new code must align with this standard.

* + 1. **Sample Questions/ Considerations**

**Input**

* Ensure that all the transactions are properly authorized
* Ensure that all transactions are received, accounted for and only received once
* Ensure transactions are input according to established processing schedules and in proper time period
* Ensure data entry errors are prevented through on line programming features
* Ensure applications contain sufficient reporting capabilities to track user activity

**Processing**

* Ensure completeness and accuracy of processing
* Ensure all changes to date are recorded

**Output/Reporting**

* Ensure sensitive reports containing non public personal information are printed under controlled conditions
* Ensure user controlled reporting software and queries are secured
* Ensure outputs are organized and clear
* Ensure effective record retention procedures are in place
* Ensure computer generated transactions are properly reported to user management

**Application Security**

* Ensure accounts are adequately secured
* Ensure application contains sufficient reporting capabilities to enable user activity tracking
* Access to program files and data is restricted to authorized personnel

**User documentation and training**

* Ensure user manuals are prepared
* Ensure users are properly trained in how to use the system

## Audit

* + 1. **Definition**

Audit is concerned with the ability to identify the person, group or system that performed, or is responsible for actions affecting information, as well as the specific actions performed and when it is to be performed. A log or audit trail provides the “who, what, when and where” of any access to or change in information.

* + 1. **Sample Requirements**

1. The system shall maintain an audit trail for a given record type or transaction.
2. The system shall maintain the audit log in a table.
3. The system shall maintain an audit trail of activities undertaken by every user.
   * 1. **Related NFR/ Categories**

[Online report](#_Online_Report)

* + 1. **Context**

This is applicable depending on whether the transactions generated as a part of system functionalities are needed to be tracked in future for regulatory or administrative needs.

* + 1. **Sample Questions/ Considerations**
* Is audit capability required for the system? What level of auditing is needed?
* Do audit trail logs have to be maintained?
* What are the record-types for which the audit trails have to be maintained?
* How it is going to be maintained? Is it through the default facility provided by any standard database/ product or is it written as a part of the application?

## Availability

* + 1. **Definition**

Availability is the measure of the extent to which users can depend on the system to be able to function during usual operating hours. It indicates when a system is operational and how reliable it is during operational periods.

* + 1. **Sample Requirements**
* The online claim filing system shall be available for not less that 99% of the time between 6 AM and 11 PM, Eastern Standard Time
* The online claim processing system shall not be available between 11 PM and 6 AM, Eastern Standard Time to facilitate the nightly processing of claims.
  + 1. **Related NFR/ Categories**

[Administration](#_Administration), [Maintainability](#_Toc292135553), [Reliability](#_Reliability), [Robustness](#_Toc292135559), [Serviceability](#_Toc292135567), [Disaster Recovery](#_Toc292135544)

* + 1. **Context**

This is applicable depending on whether there is a high impact of system downtime on business and whether the users will tolerate reduced levels of service. For example, a mortgage amortization schedule shall be available to the customer within 10 seconds for 99% of the times that it is requested at any time, but an Online Payment System shall be available for use between the hours of 7:00 a.m. and 10:00 p.m. EST

* + 1. **Sample Questions/ Considerations**
* What is the expected availability of the application? Is there a need for all the modules/features to be available in the expected availability levels?
* What is the downtime impact on the business? Is a high degree of availability critical or just nice to have?
* What are the peak times; what are the performance SLA?
* What is the maximum permissible maintenance service time?
* What is the expected frequency and duration of the maintenance service?
* Many companies run a maintenance cycle at end-of-day. Does it have to run every day? Could the maintenance be subdivided and spread out across a day?
* What housekeeping tasks must be performed during normal operation without user awareness? What degree of inferior performance is tolerable?
* Should the batch tasks be performed only during the preferred batch window?
* What is the existing maintenance support structure?
* What is the expected frequency in which the application upgrades and bug fixes be performed?
* Is there data available for frequency and duration of unexpected system failures?

## Communication interfaces

* + 1. **Definition**

Communication Interfaces is the mechanism through which the application will communicate with external systems/components.

* + 1. **Sample Requirements**
* The system shall be able to retrieve the <credit scores of a user> by interfacing with <Experian, TransUnion and Equifax.
* The system shall interface with <Single-Entry Multiple Company Interface (SEMCI) programs to provide insurance quotes to agents>( Data and external system names can be updated).
  + 1. **Related NFR/ Categories**

[Compatibility](#_Compatibility), [Interoperability](#_Interoperability), [Portability](#_Portability)

* + 1. **Context**

This is applicable when the system interfaces with one or more external system particularly when there are issues related to data exchange between the two systems.

* + 1. **Sample Questions/ Considerations**
* Are there any external systems with which this system must interface?
* Are there any constraints on the nature of the interface between this system and any external system, such as the format of data passed between these systems, and any particular protocol used?

## Compatibility

* + 1. **Definition**

Compatibility is the capability of orderly, efficient integration and operation with other elements in a system with no modification or conversion required. Software incompatibility occurs many times for new software released for a newer version of an operating system which is incompatible with the older version of the operating system.

This may occur because the new operating system may miss some of the features and functionality on which the old software depended.

* + 1. **Sample Requirements**
* The portal shall compatible with Version 7 and above of the Internet Explorer web browser.
* The application shall be compatible with Windows XP, Windows Vista and Windows 7.
  + 1. **Related NFR/ Categories**

[Communication interfaces](#_Toc292135532), [Interoperability](#_Interoperability), [Portability](#_Portability)

* + 1. **Context**

There might be an explicit requirement for new software released for a newer version of an operating system to be compatible with the older version of the operating system because the older OS may miss some of the features and functionality on which the software depends.

* + 1. **Sample Questions/ Considerations**
* Are there any requirements regarding this system and its compatibility with previous versions of this system or legacy systems providing the same capability?
* What are the platforms/ environments that the system should be compatible with i.e. Apple Mac, Windows CE, Windows 95/98/NT/2000, etc.?
* What are the browsers that must support the applications with versions?
* Is the application to be used by mobile/handheld device users? If so, what type of data are they looking for to be viewed through a PDA or other Mobile device?

## Configuration Management

* + 1. **Definition**

Configuration refers to the way a system is set up and defines the values of parameters with which the system can be configured.

* + 1. **Sample Requirements**
* The portal shall enable business users to configure workflows for creation of new content and updates to existing content
* The portal shall enable users to select a preferred web site theme from a set of available options.
  + 1. **Related NFR/ Categories**

[Adaptability](#_Adaptability), [Extensibility](#_Toc292190476), [Flexibility](#_Toc292127054),[Reusability](#_Reusability), [Scalability](#_Toc292216232)

* + 1. **Context**

This is applicable in systems where there are requirements for configurable parameters (e.g. language preference).

* + 1. **Sample Questions/ Considerations**
* Are there any limitations / guidelines on configuration vs. customization (for the IT product)?
* Will the product be configured after it has been deployed? In what way will the system be configured?
* What part of the system needs to be configurable?
* Is there a need for any configuration parameters that are customizable to suit the environment of production such as a URL prefix?
* Do we need runtime configuration capabilities?

## Content Management

* + 1. **Definition**

Content Management is related to the set of procedures and technologies which are used to support collection, managing, and publishing of information.

* + 1. **Sample Requirements**
* A third-party built content management system shall be used to manage content on the portal.
* The portal shall enable business users to configure workflows for creation of new content and updates to existing content.
  + 1. **Related NFR/ Categories**

N/A

* + 1. **Context**

This is applicable for systems which are workflow based and where there is a need to handle large volumes of electronic documents. For example, IT problem resolution system in any enterprise

* + 1. **Sample Questions/ Considerations**
* Does the application need any type of Content management features?
  + If yes, are they looking at any particular content management tool?
* What kind of search and indexing feature are required for content management?
* Is there any kind of workflow required? For example, approval facilities at various levels.

## Consistency

* + 1. **Definition**

Consistency refers to the uniformity in functionality/look and feel between two applications belonging to the same family. For example, Word, Excel and PPT being part of the Microsoft Office suite have certain uniformity in functionality/look and feel.

* + 1. **Sample Requirements**
* The portal shall have a consistent look and feel (fonts, colors, navigation) across web pages
  + 1. **Related NFR/ Categories**

[Standard](#_Toc292216254)

* + 1. **Context**

This is relevant for applications which are used by multiple BUs within an organization. Therefore, an advisor portal, meant for the life insurance Business Unit, can be expected to have some uniformity with an advisor portal built for the retirements BU.

* + 1. **Sample Questions/ Considerations**
* Are there any special requirements regarding the consistency of the user interface, both within the system and with other systems?

## Correctness

* + 1. **Definition**

It is a measure of how well the data is maintained by the software system in terms of accuracy, consistency, authenticity and without corruption.

* + 1. **Sample Requirements**
* The system shall ensure the correctness of data. For example, data stored should be correct till X places after decimal.
* The system shall ensure that the data, which is stored in multiple databases, are in sync.
* If a user is in the middle of entering some data when a timed change trigger is reached, the change trigger shall be put into effect after the user finishes entering the data.
* Commit and rollback functionalities shall be used in Stored Procedures.
  + 1. **Related NFR/ Categories**

[Privacy](#_Privacy), [Security & Access Controls](#_Toc292135565), [Session Management](#_Toc292216236)

* + 1. **Context**

This is applicable when there is a high need for precision of data and accuracy of information portrayed is critical for business. For example, for financial system, all monetary amounts must be accurate to two decimal places whereas a system displaying weather conditions to the public might vary by 2 degrees in displaying temperature.

* + 1. **Sample Questions/ Considerations**
* What is the degree of numeric precision needed in calculations, storage, and presentation?
* How should the coordination of data stored in multiple data stores be managed?
* How should the data relationships be maintained?
* What data manipulation processes are performed at each location?
* What steps are needed to ensure data are represented in the manner intended by the source/authority?
* What should happen if a user is in the middle of entering some data when a timed change trigger is reached?
* From where should users restart entering information later, from the point they had reached previously?
* How should data commit and rollback functions be implemented?
* What should be the data encryption methods?
* What are the business needs regarding data governance?

## Data Retention

* + 1. **Definition**

Data Retention or Backup refers to making copies of data so that these additional copies may be used to restore the original after a data loss.

* + 1. **Sample Requirements**
* The different entities, which will be covered in the back-up, shall be defined. For example, the different entities covered in backup are E1, E2.
* The system shall have a defined frequency of backup. For example, once every T1 days.
* The system shall maintain the backup data in the database for a defined period of time. For example, the backup data must to be maintained in the database for T2 days.
* The system shall archive the data after a defined period of time. For example, the frequency of archiving would be once every T3 days.
* The system shall have reporting requirement on the archived data.
* The backup processes shall be monitored by a third party monitoring center (say, M1).
* The system shall have a defined period of time within which the data, that has been lost, shall be restored. For example, the data lost should be restored in HH hours.
  + 1. **Related NFR/ Categories**

[Disaster Recovery](#_Toc292135544)

* + 1. **Context**

This is relevant for applications which need to process/store data which is of critical importance to the organization.

* + 1. **Sample Questions/ Considerations**
* What is the organizational policy for data retention?
* What are the different entities covered in backup? What is the frequency of backup?
* How long the data has to be maintained in the database? Is it okay to go for hard delete of complete/old data? Or is it required to make a soft delete and then archive accordingly? What is the frequency of archiving?
* What is the process by which owners of data can get information about how their data was backed up?
* Is there any reporting requirement on the archived data?
* Is the backup processes going to be monitored by a third party monitoring center? This center alerts users to any errors that occur during automated backups.
* What is the min/max latency of the data? Is it different by consumer?

## Delivery

* + 1. **Definition**

It is the mechanism by which a product or service can be delivered or fulfilled. It is also referred to as fulfillment.

* + 1. **Sample Requirements**
* The system shall have defined formats and mediums of delivery. For example, the project should be distributed in XX file format in pen driver/DVD.
  + 1. **Related NFR/ Categories**

N/A

* + 1. **Context**

This is dependent on the nature of the product or service. Examples include:

* The product should be distributed as a zip file
* The product shall be of a size such that it can fit in one CD ROM
  + 1. **Sample Questions/ Considerations**
* Is there any requirement related to delivery mechanism?

## Disaster Recovery

* + 1. **Definition**

Disaster recovery is the policy and procedure related to preparing for recovery or continuation of technology infrastructure critical to an organization after a natural or human-induced disaster. Disaster recovery nonfunctional requirements play an important role for both onsite solutions as well as hosted solutions. Two key concepts for disaster recovery are “Recovery Time Objective (RTO)” and “Recovery Point Objective (RPO)”. Recovery time objective is “How long can we live without it?” an example of recovery time objective; if a disaster occurred at 4:00 pm in the afternoon how soon do you have to have the activity or information available. The result of this determination of how often do we back up the information, system availability after a disaster, and will disaster recovery interfere with current availability.

Recovery point objective is “How far back do you have to recover to or what is the acceptable loss?” an example of recovery point objective; if the last backup was midnight the evening before and the disaster occurred at 4:00 pm is it acceptable to lose the activity that occurred from the last back up time to when the system was restored? The result of this helps determine the acceptable loss and or how often or when back up of information needs to occur.

* + 1. **Sample Requirements**
* The organization policy on business continuity plan or disaster recovery shall be according to a specified document (say, D1).
* The system shall have specific controls to prevent an event from occurring. For example, specific control C1 is needed for preventing an event E1 from occurring.
* The system shall have specific controls for detecting a defined event as and when that event occurs. For example, specific control C2 is needed for detecting an event E1 as and when it occurs.
* The system shall be restored to X backup within X time frame after an event or a disaster has occurred.
* The system shall remain available during an event or disaster.
* Upon an event or disaster the acceptable loss of activity data or information is X.
* The system shall have specific controls for restoring the system after the occurrence of a defined event. For example specific control C3 is needed for restoring the system after an event E1 occurs.
* The system shall have specific controls for restoring the system after the occurrence that would not interfere with current system availability.
  + 1. **Related NFR/ Categories**

[Data Retention](#_Toc292135541)

* + 1. **Context**

This is dependent on the business critical nature of the system being considered.

* + 1. **Sample Questions/ Considerations**
* What is the organization policy on business continuity plan or disaster recovery?
* What specific controls are needed for preventing an event from occurring?
* What specific controls are needed for detecting unwanted events?
* What is the expected time for application or data restoration?
* What specific controls are needed for correcting or restoring the system after disaster or an event?
* How soon does the system or information need to be available after an event or disaster occurs?
* How far back do you have to recover or what is the acceptable loss?

## Fault Tolerance

* + 1. **Definition**

Fault tolerance refers to the ability of a system to perform, possibly at a reduced level, rather than failing completely, when some part of the system fails. That is, the system as a whole is not stopped due to problems either in the hardware or the software

* + 1. **Sample Requirements**
* The portal shall be display a text-only version of the web site if a denial of service hacking attack is detected on its servers.
* The portal shall allow a user to place an online order for the purchase of a phone even if it is unable to retrieve the credit score information of the user from any of the credit rating agencies.
  + 1. **Related NFR/ Categories**

[Reliability](#_Reliability)

* + 1. **Sample Questions/ Considerations**
* How critical is the application?
* How likely is a particular component of the application to fail?
* How expensive is it to make the application fault tolerant?
* What is the business risk and/or impact of each of the component’s failure?

## Flexibility

* + 1. **Definition**

This is related to flexibility in delivering the product.

* + 1. **Sample Requirements**
* The various mediums of delivery shall be defined. For example, the product can be distributed through email, pen drive, CD etc.
  + 1. **Related NFR/ Categories**

[Adaptability](#_Adaptability), [Configuration](#_Toc292135536), [Extensibility](#_Toc292190476), [Reusability](#_Reusability), [Scalability](#_Toc292216232)

* + 1. **Context**

This is used in delivery context when there are constraints in movement of the product from one location to another.

* + 1. **Sample Questions/ Considerations**
* The product shall be distributed as zip file
* The product shall be of a particular size such that it can fit into a CD ROM
* Does the system need to be able to accept and/or export files/data in any format?

## Infrastructure

* + 1. **Definition**

This NFR is similar to some other NFR stated in this document. It overlaps with scalability, disaster recovery, data backup, security, performance and serviceability. Refer to 2.16.3 related NFR categories.

* + 1. **Sample Requirements**
* The System needs a minimum RAM configuration of 4GB to support 100 concurrent users.
* The Server space needed for standard response time is 1Terabyte.
  + 1. **Related NFR/ Categories**

[Scalability](#_Toc292216232), [Disaster recovery](#_Toc292135544), [Data backup](#_Toc292135541), [Performance](#_Performance), [Serviceability](#_Toc292135567)

## Interoperability

* + 1. **Definition**

Interoperability is a property referring to the ability of diverse systems to couple or facilitate the interface with other systems to work together through exchanging data via a common set of exchange formats, to read and write the same [file formats](http://en.wikipedia.org/wiki/File_format), and to use the same [protocols](http://en.wikipedia.org/wiki/Protocol_(computing)). For example a .NET and a Java based system can interact through using web services.

* + 1. **Sample Requirements**
* The system shall have defined internal/external systems that will interface with the system under development / installation.
* Homegrown modules, which must be included into the solution, shall be identified and defined. For example, homegrown modules/applications that need to be included into the solution are M1 and M2.
* The system shall have a table (say, T1) which would record error or acknowledgment of receipt from other systems.
* The system shall have defined data formats that will be supported. For example, the different data formats that need to be supported by the application are F1 and F2.
* Interoperable system testing shall be conducted between the system and another system (say, E2) to ensure that, in the production scenario, they actually will inter-operate as expected
* A common technology, such as Internet Protocol (IP), shall be defined to speed up and reduce complexity of interoperability.
* The system shall have a defined time period after which it will be upgraded. For example, the interface is upgraded every X months.
* The system shall have configuration parameters for data exchange. For example, the configuration parameters P1, P2 are required for data exchange.
  + 1. **Related NFR/ Categories**

[Communication interfaces](#_Toc292135532), [Compatibility](#_Compatibility), [Portability](#_Portability)

* + 1. **Context**

This is applicable when a system needs to collaborate with partner applications and external operations and there is a need to add or remove interfaces without disrupting the core system.

* + 1. **Sample Questions/ Considerations**
* What internal/external departments will interface with the system under development or installation?
* Are there any homegrown modules/applications that need to be included into the solution?
* Are there any external applications that need to interface with the system to be developed?
* What are the standards and abilities of the external supplier?
* Are there different data formats that need to be supported by the application?
* Are there any standards available that were followed for similar systems to make it interoperable?
* How are conflicts with the standards reconciled?
* Is there any Interoperable system testing needed to ensure that, in the production scenario, they actually will inter-operate as expected?
* Is there a need for a common technology, such as Internet Protocol (IP) to speed up and reduce complexity of interoperability?
* Is there a need to record what was sent / received on the interface?
* What kind of data exchange is envisioned (for example, query, update etc)?
* What kind of data is important to the business partners? How is the data exchanged?
* How frequent is the interface upgraded? How is the interface upgraded?

## Legislative

* + 1. **Definition**

This NFR is related to documenting all the legislative requirements.

* + 1. **Sample Requirements**
* The portal shall comply with the Employee Retirement Income Security Act (ERISA)
  + 1. **Related NFR/ Categories**

Sarbanes-Oxley ([SOX](#_Special_Considerations_for)) impacts

* + 1. **Context**

This is needed particularly for systems which deal with licensing, contracts, financial statements and other disclosure which are made mandatory by the regulatory agencies like the SEC.

* + 1. **Sample Questions/ Considerations**
* Are there any specific legislative requirements?
* Does it meet ERISA standards?

## Localizability

* + 1. **Definition**

Localizability refers to the ease with which an application’s output can be customized in different geographic locations.

* + 1. **Sample Requirements**
* The system shall produce search results that are relevant to the default location set by the user.
* The weather widget shall display weather information pertaining to the current location of the user.
* Based on the Profile information available on the system provide recommendations for the local information e.g “Chicago” data vs “Springfield” based on the user location.
  + 1. **Related NFR/ Categories**

[Internationalization](#_Internationalization_Features)

* + 1. **Sample Questions/ Considerations**
* Does the system need to be accessible by more than one location?

## Maintainability

* + 1. **Definition**

The ease with which faults in a software system can be found and the system modified in order to correct defects or make future maintenance easier, is called maintainability.

* + 1. **Sample Requirements**
* The system shall have documented SLAs for maintenance activities. For example, SLA1, defined for maintenance activities, is to be referred for defining the SLA for the maintenance of the system.
* The time period, when support features will be available, shall be defined. For example, support features are needed every weekday for maintenance regardless of whether the resource is utilized or not.
* A user shall be sent a notification that a defect or fault has been encountered.
* A user shall be responsible for analyzing the problem.
* There shall be a list of defined skill-sets to carry out maintenance activities on the system. For example, skill-set needed for maintaining the system are SK1, SK2 and SK3.
* There shall be defined tools which are used in assisting system releases, upgrades and distribution for a defined application For example, tools used in assisting system releases, upgrades and distribution for the existing application are T1 and T2.
* The frequency and volume of change in data and/or rules shall be defined. For example, the expected frequency and volume of change in rules, data is N1 every T1 month.
  + 1. **Related NFR/ Categories**

[Administration](#_Administration), [Availability](#_Availability), [Reliability](#_Reliability), [Robustness](#_Toc292135559), [Serviceability](#_Toc292135567)

* + 1. **Context**

Maintainability is applicable depending on the complexity of system design/code and the expected frequency or quantum of change in functionality in future. For example, a pensions system can be expected to have frequent changes in functionality due to regulatory changes, so maintainability is a very crucial NFR.

* + 1. **Sample Questions/ Considerations**
* Are there any SLA or performance metrics defined for maintenance activities?
* What are the support features needed for maintenance regardless of whether the resource is utilized on not?
* Is there any need for in-built features in the product itself or need for support tools to help satisfy maintenance performance requirements with less effort?
* Who receives a notification that a defect or fault has been encountered? What information is necessary to report a problem?
* What is the skill-set needed for maintaining the system?
* Are there any special requirements regarding system maintenance?
* What are the tools used in assisting system releases, upgrades and distribution for the existing application?
* Is there any expectation in the extent of documentation for the system to facilitate maintenance?
* What is the expected frequency and volume of change in rules, data etc.?
* Which modules of the application had frequent maintenance activities in the existing application?
* Are common changes configurable?

## Performance

* + 1. **Definition**

Performance is characterized by the amount of useful work accomplished by a system compared to the time and resources used. This requirement expresses the expectations with regard to response time, throughput and capacity.

* + 1. **Sample Requirements**
* The system shall have an average/expected response time for data submission/retrieval. For example, the average expected response time for data submission/retrieval transaction is XX.
* The system shall have an average/maximum number of simultaneous transactions for each type of functionality. For example, the average/maximum number of simultaneous transactions for each type of functionalities is YY.
* For a defined user group, the system shall have a defined response time. For example, for user group U1, the response time would be Ux1.
* The system shall have a response time for a defined number of concurrent users. For example, when number of concurrent users is CU1, the expected response time would be T1.
* The system shall have an average expected throughput. For example, the average expected throughput would be Tr1.
* Throughput shall be monitored by a defined system (say, S1).
* The time period, during which a pre-defined maximum number of users can access the system, shall be defined and the number of user shall be restricted to the pre-defined number. For example, during time period T1-T2 everyday, the number of users accessing the system would be limited to N1.
* The system shall have a defined and documented set of best practices to optimize system processing of data storage/validation/retrieval. For example, to optimize system processing of data storage/validation/retrieval the following best practices are to be followed: ABC.
* Load balancing shall be done by utilizing a defined application server (say, XY).
* The number of active users over a defined time period (say, T1-T2) shall be monitored in order to identify peak processing periods.
* When storage capacity is exceeded, the user shall be informed by mail.
  + 1. **Related NFR / Categories**

N/A

* + 1. **Context**

Performance is applicable when response expected for a typical transaction is very critical i.e. there is a high need for minimizing user frustration while waiting for a response.

* + 1. **Sample Questions/ Considerations**

**Response time**

* What is the expected response time for a typical transaction?
* What is the expected range in response time for a data submission screen (UI)?
* What is the expected range in response time for a data retrieval screen?
* Are there plenty of images in each page? What is the size of each page (including images)?
* What is the number of simultaneous transactions for each type of functionalities (e.g. database transaction, reporting etc.)?
* What is the peak and average number of transactions?
* Would users of different user-groups of the application expect the same response time?
* Are there any specific groups of users, like customer support executives where delay in response time cannot be allowed?
* What is the expected increase in load of number of users on a given day?
* What internal monitoring can be done by the system to measure response time?
* Is there a need to notify the user of possible wait times, give progress information during the wait and camouflage the wait?

**Throughput**

* Is there a need to monitor system throughput? (Throughput is the rate at which the system can perform input and output processing)
* Are there peak processing periods when it might be necessary to limit or stagger the number of users accessing the system?
* What are the possible actions needed to improve throughput? For example background processing might run on separate systems.
* What are the methods to organize user inputs /outputs to optimize system processing of data storage/validation/retrieval?

**Capacity**

* When is it necessary to limit the number of users logging in due to presence of simultaneous users?
* Is there a need for load balancing?
* What information might be helpful in monitoring user activity? For example tracking the number of active users over a time period can identify peak processing periods.
* Who should be notified when storage capacity is exceeded?
* What procedures should be taken to prevent the system from getting over cluttered with inactive data? For example, this will drive the data archiving need.

## Portability

* + 1. **Definition**

Portability is the [software](http://en.wikipedia.org/wiki/Software) codebase feature to be able to reuse the existing code instead of creating new code when moving software from one environment (which includes hardware, platform and/or operating system) to another.

* + 1. **Sample Requirements**
* The platforms with which the code base is compatible shall be defined. For example, the code base developed should be compatible in platform P1, P2 and OS W1, W2 and W3.
* The system shall be independent of the products of web-server, app-server and defined database versions (say, V1, V2 etc)
* The system shall (or shall not) be independent of component stereotypes such as rules engine, logging, security infrastructure etc.
* The program shall have the same GUI across defined platforms (say, P1, P2 etc)
  + 1. **Related NFR/ Categories**

[Communication interfaces](#_Toc292135532), [Compatibility](#_Compatibility), [Interoperability](#_Interoperability)

* + 1. **Context**

This is applicable depending on whether the application would run on machines having same environment or otherwise, whether the system is intended for a single site, different sites or departments, different organizations, different industries, or different regions and countries. For example, there might be a need for an application to be developed which would run on both Microsoft Vista and Macintosh operating system platforms.

* + 1. **Sample Questions/ Considerations**
* What platforms must the system support? Which operating system versions must be supported?
* Is there a need for separate installation kits for installation in different platforms?
* Is Web access required? If so, then which Web browsers and versions must be supported?
* Is there a need for the system to be independent of the products of web-server, app-server and database?
* Is there a need for the system to be independent of component stereotypes like rules engine, logging, security infrastructure etc.?
* Will the system be ported to new platforms in the future (over and above those explicitly stated for the system)? What could be the expected changes to development environments?
* Is there any roadmap for replacing specific modules or features of the application post implementation?
* Will data files be transferable and run as is on all platforms? Will the code run exactly the same on all platforms?
* Will the program have the same GUI across platforms?
* What portability requirements must be communicated to customers/users?
* Who is accountable for the portability requirements?
* What Implementation strategies should be used to develop portable software?
* Is it configurable; can we make a change to business rules without code changes?

## Privacy

* + 1. **Definition**

Data privacy refers to the evolving relationship between technology and the legal right or public expectation of privacy in the collection and sharing of data about one’s self. Privacy concerns exist wherever unique data relating to a person are collected and stored, in digital form or otherwise.

* + 1. **Sample Requirements**
* Data shall be stored in defined database(s) (say, D1, D2 etc) with administrative access given to a defined user group (say, UG1) only.
* Data write access shall be given to a defined user group only (say, UG2).
* A data masking procedure shall be followed in order to ensure that the entire data cannot be read.
* When the data is updated, an audit trail record shall be inserted in a defined table (say, T1)
  + 1. **Related NFR/ Categories**

[Correctness](#_Correctness), [Security & Access Controls](#_Toc292135565), [Session Management](#_Toc292216236)

* + 1. **Context**

This is applicable for all applications which need to handle sensitive data (e.g. SSN, payroll information etc).

* + 1. **Sample Questions/ Considerations**
* Are there specific concerns around how data is collected, stored, and associated?
* Who should be given access to information? Should there be any data masking?
* Whether an individual has any ownership rights to data about them, and/or the right to view, verify, and challenge that information?
* Can a participant be uniquely identified by the data ( i.e. Date of Birth and Last Name)

## Reliability

* + 1. **Definition**

Reliability is the ability of a [system](http://en.wikipedia.org/wiki/System) or component to perform its required functions consistently without failure under stated conditions for a specified period of time. Requirements are specified using reliability parameters.

* + 1. **Sample Requirements**
* The system shall have a defined mean time between failures. For example, the Mean Time between failures would be MTF1.
* For a defined business unit / location (say BU1), a defined variance of Mean Time (say, X %) between failures shall be allowed.
* The system shall have a defined time period to recover from a failure. For example, the system would recover from failure in time T1.
* A defined modeling technique (say, MT1) shall be available for testing reliability.
* Notifications to a defined user/event Log entry in a defined table (say, T1) shall be necessary when the system fails.
* Training and documentation shall be provided to a defined user group (say, UGx) to prevent system failure.
* System failure reports shall be sent every defined time interval (say, Tx) to a defined user (say, Manager M1).
  + 1. **Related NFR/ Categories**

[Administration](#_Administration), [Availability](#_Availability), [Maintainability](#_Toc292135553), [Robustness](#_Toc292135559), [Serviceability](#_Toc292135567), [Disaster Recovery](#_Toc292135544)

* + 1. **Context**

Reliability is applicable based on the how much mission critical the system/system modules are and what is cost of failure of the system. Sometimes costs associated with system failure often far exceed development costs.

* + 1. **Sample Questions/ Considerations**
* What features of the system are mission-critical?
* Which features of the system are in the not-so-critical category?
* Will reliability vary by business unit or location?
* From historic data, which of the modules or features didn’t scale up to the expected reliability?
* Are there any requirements regarding system “up time”? Note: This may be specified in terms of Mean Time between failures.
* Are there any special requirements regarding recovery from a system failure?
* Are there any modeling techniques that are available for testing reliability?
* What preventive measures are to be taken to avoid failure?
* Is there a need for defects classifications? If yes, what are they?
* What notifications are necessary when the system fails?
* What training and documentation must be provided to prevent system failure?
* What system failure reports are needed? Is there a need to maintain an event log?

## Robustness

* + 1. **Definition**

Robustness is the ability of a computer system to continue functioning after recovery from a system failure during execution or the ability of an algorithm to continue to operate despite abnormalities in input, calculations etc.

* + 1. **Sample Requirements**
* The process (say, P1 and P2) should terminate gracefully when some faults occur.
* The system shall have a fault detection processes which would run every defined time interval. For example, fault detection processes would run every time interval T1.
* The indentified Commercial product (say, P1) shall be used to perform robustness testing.
* A defined user/user group (say, U1) shall be alerted when default values replaces invalid data
  + 1. **Related NFR/ Categories**

[Administration](#_Administration), [Availability](#_Availability), [Maintainability](#_Toc292135553), [Reliability](#_Reliability), [Serviceability](#_Toc292135567)

* + 1. **Context**

This is applicable for systems which must continue to deliver essential business-critical services to legitimate users while the system is under attack, or after part of the system has been damaged as a consequence of an attack or a system failure. As such, robustness requirements are closely related to reliability, availability, and maintainability as these nonfunctional requirements are compromised by system failures.

* + 1. **Sample Questions/ Considerations**
* Is there any specific need for the system to be very robust? Is yes, to what degree?
* What are the user expectations for system robustness?
* What measures should be introduced to safeguard or recover software and hardware from natural disasters/unauthorized access?
* How should exception handling be done?
* What is the likely source of corrupt data? What measures can be taken to reduce or clean corrupt data?
* When default values are to be replaced for invalid data? Which users need to be alerted when the same happens?
* Which processes should be terminated when faults occur?
* Is there any need of any commercial products to perform robustness testing?
* How frequently should fault detection processes run?

## Safety

* + 1. **Definition**

This is related to physical safety in context of the potential damage to people, property and the environment.

* + 1. **Sample Requirements**
* The system shall have the ability to implement and process the financial information securely.
* The system shall have processes to ensure that personal Information is not compromised.
  + 1. **Related NFR/ Categories**

[Security & Access Controls](#_Toc292135565)

* + 1. **Context**

Example: If electromagnetic environment is present in one section of a workshop, one might be asked to wear rubber sole shoes. This NFR may not apply to MM Business Context very often, given the nature of business.

## Security & Access Controls

* + 1. **Definition**

The term system security means the collective processes and mechanisms by which sensitive information are protected from unauthorized activities or untrustworthy individuals.

* + 1. **Sample Requirements**
* For a given location (say, L1) the security access shall be “Read only”.
* When system is accessed at a given time (say, T1), a defined security feature (say, F1) shall be applicable.
* The system shall have a defined password format. For example, the password format should be XXXYYYYY
* The system shall have a defined time period after which passwords should be changed. For example, passwords should be changed every time interval T1.
* For a special event (say, E1), a one time user id/password shall be provided.
* For granting/revoking access, a defined user group (say, UG1) shall need to provide approval.
* For a defined user (say, U1), the records R1 shall be available in Read/Update/Delete mode.
* The system shall have a user hierarchy for the approval process. For example, the user hierarchy for approval of transaction X1 is as follows: U1>U2>U3.
* The system shall have a defined duration for a user session. For example, the duration of a user session would be time Tx.
* The system shall allow users to re-authenticate themselves by asking a secret question.
* On closing Internet browsers, users shall be automatically logged off.
* Users shall need to click OK on an inactivity related message with timer otherwise they would be logged off after a defined time period (say, 5 min).
* If users make a defined number of invalid attempts (say, N1 invalid attempts) to enter their password, they would be blocked.
* The data types (say D1), that would require encryption, shall be defined.
  + 1. **Related NFR/ Categories**

[Correctness](#_Correctness), [Privacy](#_Privacy), [Safety](#_Safety), [Session Management](#_Toc292216236)

* + 1. **Context**

This NFR is applicable depending on to what extent the system needs to be safeguarded against deliberate and intrusive faults from internal and external sources. As an example users might be asked to change the initially assigned login authentication information (password) immediately after the first successful login to increase the level of security for a banking system.

* + 1. **Sample Questions/ Considerations**

**Location:**

* Does the system needs any kind of location based secured access?

**Time of Access:**

* Does the application needs some security feature depending on access time? This is especially true for critical finance related data.

**User Registration**

* Is self registration of users allowed?
* What should be the password format? For example, whether it should contain special characters or it should be Alpha-numeric only. Setting password policies requires trade-off considerations, such as how many characters can be easily remembered by a typical user versus how easily passwords might be cracked
* How frequently should passwords be changed?
* Is De-registration of users allowed?
* What should be the level of user data protection? It should be in accordance with federal and state privacy laws.
* Is there a need to accommodate One-time or one-day users particularly during special events?
* Does the application need to control access rights?
* What is the process for granting or removing access?

**User authorization**

* Is there a need for delegating authority? For example, a Project Manager might need an administrative assistant who needs access to his/her calendar and email.
* Is there a need for some special event which might require extra authorization?
* Can the user modify, add or delete a record or can the user view it in “read only mode”?
* What are the user hierarchy and security structure/layers?
* Is there any Approval cycle etc. required as part of security?
* If there is an approval cycle, then what are the various channels for approval? For example multiple approvers layer, and if the main approver is absent then it flows to the other?
* Is it required in the above case to generate e-mails to let the approver know some things are pending for approval?
* Is there any time constraint after which the things automatically pass to the next level?
* Are all the modules visible to all the users?
* Can the users generate all reports?
* Who has the right to generate users and change the roles?
* Can any user manipulate data directly in the back-end?
* How User access rights are used to maintain segregation of duties (SOD)?
* User roles and responsibilities need to be clearly defined
* Can a user have multiple roles within an application?

**Authentication:**

* Is there a need for limiting the duration of a user session? If yes what should be the duration
* How should the system allow users to re-authenticate themselves (e.g. by answering a secret question)?
* How should users be de-authenticated or logged off? This prevents others from performing functions as if done by the users previously signed in.
* How to interpret implicit log off intentions from users? For example, on closing Internet browsers users should be automatically logged off.
* When users modify profile information, how should necessary notification and authentication be done?
* How to handle user absence or inactivity? It might be necessary to have users click OK on an inactivity related message with timer otherwise they would be logged off
* How to block or unblock users? For instance, if users make five invalid attempts to enter their password, they would be blocked.
* What kind of session information is to be stored? This usually includes session start and stop date/time, workstation used, authentication method and cause of session termination.
* Does role based security relate with only the application or it has to be tagged with the OS level of groups and the role?
* Is it to be integrated with other applications running in the same server (i.e. common login for all applications)?
* Are the different roles hierarchical in nature? Can a user have multiple roles for the same application?
* Does the role depend on the location?
* Is Encryption needed for the password, while on the Intranet?

**Encryption:**

* Is data security required while authenticating?
* Are there any secured pages?
* Does the application need data encryption in the data stores also?

## Serviceability

* + 1. **Definition**

It refers to the ability of technical support personnel to install or monitor computer products, identify exceptions or faults, and provide hardware or software maintenance in pursuit of restoring the product into service.

* + 1. **Sample Requirements**
* A defined service tool shall be needed that provides capabilities and data so as to service (analyze, monitor, debug, repair, etc) the application.
* Help desk notification of exceptional events shall take place by electronic mail.
* Network monitoring and event logging shall be performed by a defined user group/system (say, Team T1/System S1).
* The product shall be designed to allow recovery from exceptional events without intervention by technical support staff.
  + 1. **Related NFR/ Categories**

[Administration](#_Administration), [Availability](#_Availability), [Maintainability](#_Toc292135553), [Reliability](#_Reliability), [Robustness](#_Toc292135559)

* + 1. **Context**

This is relevant for applications which require regular monitoring because it might be highly complex in nature.

* + 1. **Sample Questions/ Considerations**
* Is there a need for a service tool that provides capabilities and data so as to service (analyze, monitor, debug, repair, etc) the application?
* Is there a need for help desk notification of exceptional events (e.g., by electronic mail) or by sending text to a pager?
* Is there a need for network monitoring and event logging / tracing?
* Is there a need for graceful degradation, where the product is designed to allow recovery from exceptional events without intervention by technical support staff?

## Usability

* + 1. **Definition**

Usability is quality of a visitor’s experience when using an application/website. This includes how a user is able to learn, operate, prepare inputs, and interpret outputs through interaction with an application/website.

* + 1. **Sample Requirements**
* Users should be easily able to print information.
* The web site should not display unsolicited windows or graphics
* The web site should allow users to perform tasks in the same sequence and manner across similar conditions.
* There shall be tooltips (such as, hints, tips and shortcuts) that appear as the user hovers mouse over an application control.
* For warning messages and error messages, sufficient details shall be provided.
  + 1. **Related NFR/ Categories**

[Internationalization](#_Internationalization_Features), [Messaging](#_Messaging)

* + 1. **Context**

This is applicable depending on how many varied users would be using the system. If a system is used for numerous user groups, particularly by people who are not IT savvy, usability becomes a critical factor.

* + 1. **Sample Questions/ Considerations**
* What would be the different user groups of the system? What would be the number of users in each of these groups? What would be the skill-level of these different user groups?
* What would be the total number of users of the system? What would be the number of active and concurrent users of the system?
* Can users easily accomplish intended tasks at their desired speed?
* How much training do users need?
* What documentation or other supporting materials are available to help the user? Can users find solutions in these materials?
* From where can the specific wordings for the error messages and their descriptions be obtained?
* What and how many errors do users make when they interact with the product? Can the user recover from errors?
* Is the user ever expected to do something that is not obvious?
* What degree of user help is necessary?
* Are there hints and tips and shortcuts that appear as the user is using the software?
* Should there be instructions in the manual that actually belong as contextual tips shown in the program?
* How interactive is the process starting from user data input to user receiving output in the system?
* What is the level of detail needed for warning messages and error messages?

## Volume and scalability

* + 1. **Definition**

Scalability is the ability of a system to expand its processing capabilities in order to handle growing amounts of load. To scale horizontally (or scale out) means to add more nodes to a system, such as adding a new computer to a distributed software application. An example might be scaling out from one Web server system to three. To scale vertically (or scale up) means to add resources to a single node in a system, typically involving the addition of CPUs or memory to a single computer.

* + 1. **Sample Requirements**
* The expected growth of the application in terms of number of users per year shall be defined (say, X %).
* Any expectation of a spurt in the number of concurrent users during a given time of the day (say, HH) shall be identified and defined.
* The growth in volume of data shall be no more than Y% per year.
* The magnitude of increase in transactions shall be no more than Z% per year.
* The auditing system growth and expansion shall be monitored by an identified department (say, Department D1).
* Any scaling required due to an increase in the number of users shall be identified.
* Scaling reports shall be sent to a defined department (say, Dept#1).
* An identified department (say, Dept#2) shall be responsible for auditing systems growth and expansion.
  + 1. **Related NFR/ Categories**

[Adaptability](#_Adaptability), [Configuration](#_Toc292135536), [Extensibility](#_Toc292190476), [Flexibility](#_Toc292127054),[Reusability](#_Reusability)

* + 1. **Context**

Scalability is applicable when a system is expected to cope with increasing processing load, expanding business locations or the need to down grade (Scale down).

* + 1. **Sample Questions/ Considerations**
* What volume of users will the system support?
* What is the expected growth of the application in number of users over time period of one year, two years, three years and five years?
* Would there be a spurt in the number of concurrent users during any specific time of the day?
* What volume of data would the users support?
* What is the expected growth in volume of data?
* What is the expected performance for the screens, in case of system with user interface?
* What magnitude of increase in transactions/ products is expected?
* Are there any constraints in investment on infrastructure scaling out [horizontal scaling] or scaling up [vertical scaling]?
* Is there any corporate or departmental standards or preferences for QA, Load testing and Performance testing?
* What historical scalability data are to be utilized by the system? What data would help in identifying growth trends?
* Who is responsible for auditing system growth and expansion?
* Who should receive scalability reports?
* Who monitors growth trends and patterns? What indicators of growth should be monitored?
* What is the affect of scaling, both vertical and horizontal?
* What locations could face problems due to sudden increase in business volume?
* What component or architectural feature will be the greatest challenge to scale? What third-party components are in use/planned for use that will have scalability issues?

## Online Report

* + 1. **Definition**

Online reporting requirements deal with the capability of the application to support the reporting business requirements. These capabilities may include specifications regarding the expected rate of report generation, fonts and formats supported, the types of pages, and information regarding usage of the site.

* + 1. **Sample Requirements**
* System will be able to produce reports in the following formats: Excel, PDF, word.
* The system shall be able to generate and print at least 100 loss run reports in an hour
* The system shall be able to report on the time a specified user signs in to the application
  + 1. **Related NFR/ Categories**

[Audit](#_Audit)

* + 1. **Context**

This is relevant for applications which are very data intensive and part of MIS thus having numerous reporting related requirements.

* + 1. **Sample Questions/ Considerations**
* What is the volume of online reports?
* Is there any graphics involved in the report? Or is it simply text based?
* Are these reports editable?
* What is the file format for these reports? For example, is it PDF, HTML, DOC, XLS?
* Do we need any control to say that the report has already been generated?
* Do we need any log of when the report has been generated or who has generated it?
* Is the report data that is viewed user-dependent?
* Is there any drill down reports? If so up to what level of granularity the drill down is required?

## Session Management

* + 1. **Definition**

It is the process of keeping track of a user’s activity across sessions of interaction with the computer system.

* + 1. **Sample Requirements**
* The System will have a session time out if the user is inactive over 5 minutes from the instance has been logged in.
* The session time out for pop-up windows shall be 10 minutes.
* In the event there are no pop-up windows open the session time out will be 5 minutes.
  + 1. **Related NFR/ Categories**

[Correctness](#_Correctness), [Privacy](#_Privacy), [Security & Access Controls](#_Toc292135565)

* + 1. **Context**

This is relevant for all web based applications when there is a need to track user activities to serve a customer better.

* + 1. **Sample Questions/ Considerations**
* What defines the end of a session? Does a session end when the browser closes?
* Should the user be notified once the session ends?
* Can one store all the data in the client machine (i.e. in cookies, temporary tables, etc.)?

## Installability

* + 1. **Definition**

Installability requirements refer to the requirements concerning the installation process of the application, the people who should carry out the installation, the configuration of the target platforms, and the type of software to be installed.

* + 1. **Sample Requirements**
* The installation manual should state the minimum processor requirements for the application.
* The application shall be installed by technicians who have been certified as Microsoft Certified Systems Administrator
* Before the installation of the application, any previous version of the application shall be uninstalled from the computer
  + 1. **Sample Questions**
* Does the application meet the software standards prescribed by MM?
* Is there a specific vendor that is licensed/ qualified for the installation.
  + 1. **Related NFR/ Categories**

[Infrastructure](#_Infrastructure)

## Internationalization Features

* + 1. **Definition**

The means of adapting [computer software](http://en.wikipedia.org/wiki/Computer_software) to different languages, regional differences and technical requirements of a target market is called Internationalization.

* + 1. **Sample Questions**
* The system will support multiple time Zones.
* The system will be compatible with Operating system software versions released for the specific countries and region from where the users are logged in.
  + 1. **Related NFR/ Categories**

[Messaging](#_Messaging), [Usability](#_Toc292135570)

* + 1. **Context**

This is relevant when an application is accessed by multi language users.

* + 1. **Sample Questions/ Considerations**
* Does the application need support for multiple languages?
* What kind of support is required? For example, does it encompass all the content or is restricted to back-end reports only?
* What are the languages it should support?
* Should the GUI involved also support multiple languages?

## Documentation and Training

* + 1. **Definition**

Documentation is a general term for a multiplicity of documents in a chosen mix of media and with a certain collection. Purpose of documentation is the use to support a tool or a process.

* + 1. **Sample Requirements**
* The repository (say, R1) shall have the ability to store specified documentation. For example the following types of documentation are required: user guides, white papers, on-line help, quick-reference guides etc.
* The method of distribution of the documentation shall be defined. For example, the documentation shall be distributed via websites, software products, and other on-line applications.
  + 1. **Related NFR/ Categories**

[User documentation](#_Toc292135588)

* + 1. **Context**

This is relevant when there a need for some kind of guidance for the user as the system functionalities are too complex to understand.

* + 1. **Sample Questions/ Considerations**
* What kind of documentation is needed? Example: user guides, white papers, on-line help, quick-reference guides etc.
* Is there any preference for hard-copy (paper) or soft-copy (electronic) documentation?
* How will the documentation be distributed? Example: via websites, software products, and other on-line applications.

## Extensibility

* + 1. **Definition**

Extensibility is a measure of the ability to extend particular system functionality to another system and minimizing the level of effort required to implement the extension. Extensions can be through the addition of new functionality or through modification of existing functionality.

* + 1. **Sample Requirements**
* The specific application functionality (say, F1) shall be designed in a manner such that it can be extended to any identified system(s).
  + 1. **Related NFR/ Categories**

[Adaptability](#_Adaptability), [Configuration](#_Toc292135536), [Flexibility](#_Toc292127054),[Reusability](#_Reusability), [Scalability](#_Toc292216232)

* + 1. **Context**

Extensibility is applicable when there is a need to extend the existing software by “plugging in” (also called “plug-and-play”) extra components.

As an example there might be a future need to add a new delivery option for customer mailing method by developing and “plugging in” the functionality necessary to support that delivery option. The new delivery option shall not require changes to the core software of the system to allow its introduction.

* + 1. **Sample Questions/ Considerations**
* Does the system/application need to be extensible?
* Is there any specific application functionality that needs to be built with extensibility in mind?

## Reusability

* + 1. **Definition**

This refers to the ability of the system to reuse/extend portion of the software system for use in another system. This includes reusing existing reference architecture framework or some in built code library. For example, development of functionality to support the Electronic Funds Transfer (EFT) payment option can be modularized for reuse by other departments of the organization.

* + 1. **Sample Requirements**
* The specific module/components (say, M1) of the system shall be reused in other applications.
* Defined additional features (say, F1, F2 and F3) shall be included in a given module (say, M1) to increase reusability.
* The module (say, M1) shall reuse existing reference architecture framework (say, RF1).
* A module (say, M1) shall use an in-built code library (say, L1) to increase reusability.
* A module (say, M1) shall use the variable, property and function-naming convention as per a defined Standard.
  + 1. **Related NFR/ Categories**

[Adaptability](#_Adaptability), [Configuration](#_Toc292135536), [Extensibility](#_Toc292190476), [Flexibility](#_Toc292127054),[Scalability](#_Toc292216232)

* + 1. **Context**

Generally, as the size and complexity of the software increases, the feasibility for reuse usually decreases. On the other hand, Small modules and components are more easily designed, tested, and maintained. Another example would be when a module deals with many variables for calculating interest amounts in multiple currencies, then the chances for consistent interpretation are lower, so reusability is less.

* + 1. **Sample Questions/ Considerations**
* What specific elements or components of the system would be reused in other applications?
* Is there any existing reference architecture framework which the system should reuse?
* Is the system supposed to use any in built code library?
* What is the extent of current presentation standards?
* What other projects in the organization covers substantially the same domains or work areas?
* What kind of documents from other projects could be useful?
* What standards have been defined and implemented in previous projects?
* Is it configurable; can changes be handled without code changes?

## Standards

* + 1. **Definition**

This NFR is related to coding or user interface related standards.

* + 1. **Sample Requirements**
* Coding in a particular programming language shall be done according to the organization-wide standards defined for that language.
* User interface / style shall conform to colors/look and feel described in enterprise branding guidelines document.
  + 1. **Related NFR/ Categories**

[Consistency](#_Toc292135538), [Legislative](#_Toc292135551)

Add Legislative as a related category

* + 1. **Context**

This is necessary if there is a requirement for some type of uniformity in GUI enforced by the marketing group or some sort of coding standard enforced for easy maintainability.

* + 1. **Sample Questions/ Considerations**
* Are there any standards to which the system must conform? This may include coding standards or a user interface style guide.
* Are there any enterprise-wide guidelines that need to be incorporated while implementing the standards? Example: branding guidelines that need to be kept in mind while developing a portal; or application control standards for financial transactions.

## Messaging

* + 1. **Definition**

This NFR is related to handling messages.

* + 1. **Sample Requirements**
* The system shall display error messages in <message boxes> / <by redirecting to an error page
* The system shall have the ability to customize error messages in special cases.
* The system shall have the ability to log details for administrators to monitor the system.
* <NS1> shall be the notification system used to render messages
  + 1. **Related NFR/ Categories**

[Audit](#_Audit), [Internationalization](#_Internationalization_Features), [Usability](#_Toc292135570)

* + 1. **Context**

This is relevant when some important information or warning needs to be showed to users (based on some user action or page/application loading) which would help in reducing transaction errors.

* + 1. **Sample Questions/ Considerations**
* Does the applications need that the errors be displayed as message boxes or should it be re-directed to an error page or is it required to refresh the same page with an error message at the top?
* What should be the text of the errors trapped is it to be customized always or the system messages have to be displayed in special cases?
* Does some kind of logging have to be there for the administrators to monitor the system?
* What is the type of notification system that the client is looking for?
* How to display the information? Is it required as a pop-up as the mouse navigates over it? Should it appear in the bottom in the application task bar? Is it required as a hyperlink or any clickable item which displays some popup or redirects to some screen?

### **Special Considerations for projects with Sarbanes-Oxley (SOX) consideration**

SOX is a United States federal law enacted on July 30, 2002, which set new or enhanced standards for all U.S. public company boards, management and public accounting firms. This law does not apply to privately held companies. The act contains 11 sections, ranging from additional corporate board responsibilities to criminal penalties, and requires the SEC to implement rulings on requirements to comply with the new law.

SOX may be applicable to systems that generate transactions impacting the financial and accounting system of the enterprise. Following are some of the IT Application Controls that should be considered for a project that might have SOX impact.

* Data edits
* Segregation of Duties in the context of separation of business functions (e.g., transaction initiation vs. authorization)
* Balancing of processing totals
* Transaction logging
* Error reporting

The objective of the above is to ensure that all the data is “authorized, accurate, complete and correct; all data is processed as intended; all output is accurate and complete.” (Source: IT Standard: Application Control document)

Some of the common NFR that need to be considered in the context of the above listed IT Application controls include:

* [Security & Access Control](#_Toc292135565)
* [Privacy](#_Privacy)
* [Reliability](#_Reliability)
* [Robustness](#_Toc292135559)
* [Standards](#_Toc292216254)
* [Audit](#_Audit)
* [Correctness](#_Correctness)
* [Data Retention](#_Toc292135541)
* [Disaster Recovery](#_Toc292135544)
* [Documentation and Training](#_Toc292135582)
* [Messaging](#_Messaging)
* [Online report](#_Online_Report)
* [Legislative](#_Toc292135551)

### **Segregation of Duties**

Segregation of Duties (SoD) separates the roles and responsibilities to ensure that an individual cannot process a transaction (say, from initiation through to reporting) without the involvement of others and thereby SoD reduces the risk of fraud or error to an acceptable level. For example, a single individual should not be able to execute a transaction (process a check, accept cash etc) and audit the same transaction as well. This is because giving a single individual the ability to perform both of the above operations increases the risk of fraud or error. Thus, a SoD project aims to ensure that conflicting activities that cannot be assigned to the same individual are appropriately segregated.

The development and maintenance of an effective segregation of duties is not solely an IT problem - some activities are performed outside of the ERP system, while others can be enforced within the system. For example, a payment to a supplier may be authorized and released in the system, but the reconciliation of the bank account from which the payment is made may be performed and authorized manually.

Steps to ensure Effective Segregation of Duties:

1. Review the processes / controls / activities defined within the scope of the project. The review will not necessarily need to incorporate all the controls that exist but those that that require an authorization/segregation of duties.
2. Identify the roles that are relevant to each process / control / activity. Once the relevant roles are identified, a matrix will need to be updated to reflect how the controls / activities are assigned across the roles that have been identified.
3. Categorize the type of controls / activities into manual (M) or system (S).
4. Evaluate the Segregation of Duties. The determination of an appropriate segregation of duty is dependent upon the specific roles and responsibilities of individuals and the need to separate those roles and responsibilities appropriately to reduce the risk of error,

With reference to IT, Segregation of Duties involves testing that must be performed, using an established system tool, to review system access settings to ensure that they are in accordance with Segregation of Duties requirements. The reports, that the system generates, must contain the following pieces of information:

1. All user profiles that cause segregation of duties issues
2. All individual users who's profile causes segregation of duties issues
3. A report by individual conflict of all users who cause the conflict
4. All users who can perform critical activities (explained below)
5. All profiles included in the exclusion table
6. The transactions assigned to system job roles and also the access to transactions of User ids of individual users and reports
7. Relevant authorizations/roles which have conflicting transactions
8. Users with access to conflicting transactions due to access to multiple

The system testing, for segregation of duties, is performed in three main instances:

1. Every time a new user is added to the system, to determine if the profile that has been requested for the user is not in conflict of the SoD requirements.
2. Every time a user's profile is amended to determine if the change would cause a conflict with the SoD requirements.
3. On a periodic basis to review SoD in the relevant (ERP) system.

### **NFR Testability and Measurability**

## Definition

Testability is the degree to which tests, analysis, and demonstrations are needed to prove that the software system will function as per expectation. The definition of a non-functional requirement should include an appropriate measure of success for each one so that it can be adequately tested. Some non-functional requirements may seem very subjective (e.g. “intuitive interface”) but careful thought can usually provide an appropriate success measurement. Or, said another way, there must be a way to prove that a non-functional requirement has been fulfilled. Each requirement should be testable – that is, it must be possible to design a test that can be used to determine if a solution has met the requirement or some other means of determining whether to accept a solution that meets the requirement.

## Sample Requirements

* Ninety percent of operators shall be able to use all the functionality of the system after no more than six hours of training.
* The system shall provide 90% of responses in no more than 2 seconds

## Related NFR/ Categories

N/A

## Context

This is applicable when the concerned system interfaces with multiple external systems, there are security issues in respect to testing and there is a constraint in the availability of testing resource.

For example, the maximum number of test cases to cover testing of any particular source code module shall be 20.

## Sample Questions/ Considerations

* The Business Architect should review each NFR and ensure that each is measurable and testable.

### **References**

## Information Source/References

* Quest for Software Requirements by Roxanne E Millers
* Cognizant Templates for capturing NFR
* Wikipedia
* *IT Standard: Application Control* document