|  |  |
| --- | --- |
|  |  |
|  |  |

Project name

Business Requirements Document

|  |  |
| --- | --- |
| **Definition** | A **Business Requirements Document (BRD)** details the business solution for a project including the documentation of customer needs and expectations. |
| **Purpose** | The purpose of this **BRD** is to define the system needs and expectations of the customer, and to ensure that the system components are compatible and comply with the enterprise-wide standards and direction defined by the Agency. **This is a living document; requirements can be refined, added upon, clarified, or even removed.** |
| **Ownership** | The project development team is responsible for preparing the **BRD** document. Prior to proceeding, the document must include approvals from the key Stakeholders. |
| **Applicability** | A **BRD** is a required deliverable on all system development projects. |

|  |  |
| --- | --- |
| **Author:** | SDLC Team |
| **Version:** | 1.0 |
| **Status** | Draft |
| **Date:** | 2013-05-30 |

| **Document Revision History** | | | |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Submitters OF FINAL DOCUMENT

| **Role** | **Name** | **Signature** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |

Reviewers OF FINAL DOCUMENT

| **Role** | **Review Completed** | **Date** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Approvers OF FINAL DOCUMENT

| **Role** | **Name** | **Signature** | **Date** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

*By accepting this* ***Business Requirements Document*** *you are agreeing to the details contained in this document. No changes will be made to this agreement without additional acceptance from each party signed above.*

Table of Contents

[Important Notes for Completing this Document 5](#_Toc338150536)

[1 Introduction 7](#_Toc338150537)

[1.1 Project Background 7](#_Toc338150538)

[1.2 Intent 7](#_Toc338150539)

[1.3 Scope 7](#_Toc338150540)

[1.4 Stakeholders 7](#_Toc338150541)

[1.5 Definitions, Acronyms and Abbreviations 7](#_Toc338150542)

[1.6 Assumptions 8](#_Toc338150543)

[1.7 Dependencies 8](#_Toc338150544)

[1.8 Constraints 8](#_Toc338150545)

[1.9 Critical Success Factors 8](#_Toc338150546)

[2 Business Requirements Overview 9](#_Toc338150547)

[2.1 Description 9](#_Toc338150548)

[2.2 Recommended Wording for Writing Requirements 9](#_Toc338150549)

[3 Business Requirements 11](#_Toc338150550)

[3.1 As-Is State 11](#_Toc338150551)

[3.1.1 Current Assessment of the System Environment 11](#_Toc338150552)

[3.1.2 As-Is Diagrams 11](#_Toc338150553)

[3.1.2.1 System Context Diagram 11](#_Toc338150554)

[3.1.2.2 Business Process Model 11](#_Toc338150555)

[3.2 To-Be State 12](#_Toc338150556)

[3.2.1 To-Be Diagrams 12](#_Toc338150557)

[3.2.1.1 System Context Diagram 12](#_Toc338150558)

[3.2.1.2 Business Process Model 12](#_Toc338150559)

[4 Functional Requirements 13](#_Toc338150564)

[4.1 Functional Area 1-N – [Identify area] 13](#_Toc338150565)

[4.2 Functional Area Summary 13](#_Toc338150566)

[4.3 Business Rules 13](#_Toc338150567)

[4.4 Reporting Requirements 15](#_Toc338150568)

[4.5 Interface/Integration Requirements 15](#_Toc338150569)

[4.6 Input/output File Descriptions 15](#_Toc338150570)

[4.7 Database Mapping 16](#_Toc338150571)

[4.8 Data Dictionary 16](#_Toc338150572)

[4.9 ER Diagrams (Logical Model) 16](#_Toc338150573)

[5 Detailed Functional Requirements 17](#_Toc338150576)

[5.1 Sub-system 17](#_Toc338150577)

[5.2 Component Type 17](#_Toc338150578)

[5.3 Component 1 17](#_Toc338150579)

[6 Non-Functional Requirements 19](#_Toc338150581)

[6.1 Technical Requirements 19](#_Toc338150582)

[6.1.1 Applicable Current Standards 19](#_Toc338150583)

[6.1.2 Accessibility 20](#_Toc338150584)

[6.1.3 Encryption 21](#_Toc338150585)

[6.1.4 Hosting 21](#_Toc338150586)

[6.1.5 Environment 21](#_Toc338150587)

[6.1.6 Business Continuity 21](#_Toc338150597)

[6.1.6.1 Disaster Recovery 22](#_Toc338150598)

[6.1.7 Security Analysis 22](#_Toc338150599)

[6.1.7.1 Application Security 22](#_Toc338150600)

[6.1.7.1.1 Security Requirements 23](#_Toc338150601)

[6.1.7.1.2 Application Access and Decision Making Authorization 23](#_Toc338150602)

[6.1.7.1.3 Security Matrix 23](#_Toc338150603)

[6.1.7.1.4 ISO Security Model (Data Classification) 23](#_Toc338150604)

[6.1.8 Data Integrity 24](#_Toc338150605)

[6.1.9 Design Constraints 24](#_Toc338150606)

[6.1.10 Non-User Interfaces 25](#_Toc338150607)

[6.1.10.1 Hardware Interfaces 25](#_Toc338150608)

[6.1.10.2 Software Interfaces 25](#_Toc338150609)

[6.1.10.3 Communications Interfaces 26](#_Toc338150610)

[6.1.11 Legal, Copyright and Other Notices 26](#_Toc338150611)

[6.1.12 Purchased Components and Licensing Requirements 26](#_Toc338150612)

[6.2 Operational Requirements 27](#_Toc338150613)

[6.2.1 System Performance 27](#_Toc338150614)

[6.2.1.1 Performance Requirements 28](#_Toc338150615)

[6.2.1.2 Availability 28](#_Toc338150616)

[6.2.1.3 System Capacity and Scalability Requirements 28](#_Toc338150617)

[6.2.1.4 Usability Requirements 28](#_Toc338150618)

[6.2.2 Flexibility Requirements 29](#_Toc338150619)

[6.2.3 Robustness Requirements 29](#_Toc338150620)

[6.2.4 Audit and Controls 29](#_Toc338150621)

[6.2.5 Software Quality Assurance (SQA) 29](#_Toc338150622)

[6.2.6 System Administration 30](#_Toc338150623)

[6.2.7 Backup and Recovery 30](#_Toc338150624)

[6.2.8 Archiving and Retention Requirements 30](#_Toc338150625)

[6.3 Transitional Requirements 31](#_Toc338150626)

[6.3.1 Data Conversion 31](#_Toc338150627)

[6.3.2 Training Requirements 31](#_Toc338150628)

[6.3.2.1 Roles of Customers/Consumers of the System 31](#_Toc338150629)

[6.3.3 Support Requirements 31](#_Toc338150630)

[6.3.4 Documentation 32](#_Toc338150631)

[6.3.5 Help 32](#_Toc338150632)

[6.3.6 Deployment 33](#_Toc338150633)

[6.3.7 Release Validation 33](#_Toc338150634)

Important Notes for Completing this Document

1. Each section of the **Business Requirements Document** must be completed in full. If a particular section is not applicable to this project, then you must write "Not Applicable" and provide a reason.
2. This page and the Business Requirements Overview can be deleted once the document has been completed. No other sections are to be deleted from this document.
3. Do not change the format or fonts used in the template.
4. All tables should have landscape orientation.
5. Text contained within [ ] provides information on how to complete that section and can be deleted once the section has been completed.
6. Delete examples under each section once the section has been completed.
7. If additional categories are required, feel free to add an “Other” category to the end of the document.
8. Reviews and Approvals. All reviews should be completed prior to obtaining approval signatures. Project Manager Review should be completed prior to Program Area Review.
9. This document should be stored in the Project repository.

# Introduction

[A **Business Requirements Document (BRD)** details the business solution for a project including the documentation of customer needs and expectations. The introduction of the **Business Requirements Document** should provide an overview of the entire document. It should include the purpose, scope, definitions, acronyms, abbreviations, references and overview of this **Business Requirements Documen**t. **This is a living document; requirements can be refined, added upon, clarified, or even removed.**]

## *Project Background*

[Provide an introduction to the Project. This includes describing the business context of the project and an overview of the Business Area’s operation. Include a high-level overview of each of the functional business areas.]

## *Intent*

[Describe the intent of the solution. What is to be provided? If an RFP is to follow, identify what the Business Area is buying from the vendor – professional services, hardware, software, Software as a Service (SaaS), support, peripheral devices etc.]

## *Scope*

[This section of the document should describe scope of document.]

## *Stakeholders*

[List the stakeholders for the project and the contributors to the requirements in the document and their roles. A stakeholder is an individual or group that has a vested interest, directly or indirectly, in the project or a decision or proposed action that affects the project.]

## *Definitions, Acronyms and Abbreviations*

[A reference should be made to the glossary which will contain the definitions for this project].

|  |  |
| --- | --- |
| **Term** | **Definition** |
|  |  |

## *Assumptions*

**Assumptions** are factors that are believed to be true, but have not been confirmed. Assumptions may affect all aspects of the project and pose a certain degree of risk if they do not prove to be true.

| **#** | **Assumptions** |
| --- | --- |
|  |  |

## Dependencies

[Identify any factors that are linked together where each has some effect on the other. These may include things like availability of project resources, users or stakeholders, equipment, business processes and regulatory approvals.]

| **#** | **Dependencies** |
| --- | --- |
|  |  |

## Constraints

**Constraints** are defined as those regulatory, technological or business realities that legitimately constrain solution development.

| **#** | **Constraints** |
| --- | --- |
|  |  |

## Critical Success Factors

[It is a **critical factor** or an activity required for ensuring the success of a project or enhancement. Therefore, include issues vital to an organization's current operating activities and to its future success. They represent those managerial or enterprise area, that must be given special and continual attention to bring about high performance.]

| **Critical Success Factors** |
| --- |
|  |

# Business Requirements Overview

## Description

***Business Requireme****nts describe* ***WHAT*** *the system, process or product/service must do in order to fulfill the business need(s) and are categorized into different priorities based on MoSCoW analysis.*

*Must - Requirements identified as must have to be satisfied in the final solution without which the operation of the proposed system is not possible. They represent features that the Business Area cannot function without.*

*Should - Requirements identified as a high priority that need to be satisfied if possible. This is considered a critical requirement but has work around, if needed. Though not mission critical, these provide significant benefit to the organization.*

*Could – Requirements identified are considered desirable and would represent helpful or convenient features that would be beneficial to the Business Area.*

*Won’t – These are the requirements that the stakeholders have agreed that not be implemented in the current release but may be considered for future releases or upgrades.*

## Recommended Wording for Writing Requirements

*Characteristics of Quality Requirements; the requirements should be written such that they are:*

1. *Concise*

* *requirements should be stated clearly and to the point*
* *easily read and understood by non-technical people*
* *able to be clearly understood by someone moderately familiar with the business*

1. *Testable*

* *requirements should be measurable*
* *possible to observe and evaluate whether the system met the requirement*

1. *Unambiguous*

* *requirements should be stated in such a way that there is only one way to interpret it*

1. *Unique*

* *each requirement should be uniquely identified/numbered*

1. *Complete*

* *all requirements may not be known in detail at the beginning of a project but they should be kept up to date as the project proceeds*
* *written using complete sentences that reflect a full thought*
* *contains all the information needed to define the system function that it is intended to address*

1. *Consistent*

* *requirements should not conflict with other requirements or with the business process*
* *same terminology is used throughout the document*
* *does not create redundancy among requirements*

1. *Traceable*

* *each requirement should be traceable to the higher level business need*
* *the source of the requirement can be easily identified*

1. *Feasible*

* *the requirements should be realistic within the constraints of the project*
* *can be achieved within the budget*
* *can be achieved within the schedule*

1. *Design independent*

* *defines what functionality will be provided by the system*
* *does not specify how that functionality can or should be implemented*
* *allows the system developer to determine which technology is best suited to achieve the desired functionality*

*It is also recommended to use active voice instead of passive voice when writing requirements;*

*Passive voice example: “The passive voice is to be avoided.”*

*Active voice example: “Avoid using the passive voice.”*

*Passive voice example: “A report can be run that includes the completed surveys.”*

*Active voice example: “The application will provide a report that lists the counties which have submitted completed surveys.”*

*Textual statements of requirements can be supplemented with one or more diagrams, tables, flowcharts, task flow diagrams.*

# Business Requirements

## As-Is State

### Current Assessment of the System Environment

[Describe the current business process and its system environment – does the Business Area have an existing application? What it is? What are some of the key challenges with the system? If no automated system, is everything done manually? Describe it. Is there any scope for business process improvement or re-engineering? Has an enterprise analysis been performed – are there any other systems or business units that can benefit from a common solution? What are the improvement opportunities? Does the solution automate or simplify work people perform; does it reduce the complexity of the interfaces or eliminate redundancy? Can the system be improved to fulfill enterprise needs?]

### As-Is Diagrams

An **As-Is Diagram** is a description of the current behavior of a process, including sub processes and activities. [Please include the following diagrams:]

#### System Context Diagram

**System Context Diagram** is a simple diagram showing how this system currently interacts with other existing systems.

***Link:*** *[please provide the link to the location of where this document resides]*

#### Business Process Model

This is a model of the current state of the process ("as is" model). One or more of the following flow diagrams are included depending on the complexity of the model

* **Business Process Flow Diagram** details the work flow of a specific business operation. This is accomplished by showing specific functional areas within a business. It is the steps in the business process from start to finish ensuring objectives and goals of the department **are** currently being met. Business flow also shows all major components of the business process.
* **Work Flow Diagram** is a simple form representing how the system currently flow of tasks or actions from one person or group to another.
* **Data Flow Diagram** shows how the system currently moves the data through a system between processes, entities, and data stores.

***Link:*** *[please provide the link(s) to the location of where this document resides]*

## To-Be State

### To-Be Diagrams

[The ‘To-Be’ graphical representations of the overall system should be included:]

#### System Context Diagram

A **System Context Diagram** is a simple diagram showing how this system **will** interact with other existing systems.

**Link:** [please provide the link to the location of where this document resides]

#### Business Process Model

This is a model of the future state of the process ("to be" model). [One or more of the following flow diagrams are included depending on the complexity of the model.]

* **Business Process Flow Diagram** details work flow (sequence of tasks) of a specific business operation. This is accomplished by showing specific functional areas within a business. It is the steps in the business process from start to finish ensuring objectives and goals of the department **will** be met. Business flow also shows all major components of the business process.
* **Work Flow Diagram** is a simple form representing how the system **will** flow tasks or actions from one person or group to another.
* **Data Flow Diagram** shows how the system **will** move data within a system between processes, entities, and data stores.

***Link:*** *[please provide the link(s) to the location of where this document resides]*

# Functional Requirements

*[Process requirements identify what the product should do in order to fulfill the* ***business need (e****.g., “The system should notify the program area when a new permit is submitted”). Functional requirements are a complete description of how the system will function from the user’s perspective. They describe the behavior and capabilities of an application, including the information or data that the application will manage. Functional requirements typically have one or more* *business rules associated with them.]*

## Functional Area 1-N – [Identify area]

*Description: [Provide a brief description of the functional area to be included in the solution. (E.g. Scheduling, Purchase orders, Suppliers, Vessel Maintenance, et cetera) The description should summarize the detailed requirements identified in the table below. Every requirement listed below should logically map back to the* ***Functional Area Summary*** *description. Functional Area descriptions represent the mandatory process requirements of the system within the resulting Request for Proposal (RFP). Use the word must within the summary paragraph only. Use the word should in the Requirements column below. ]*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
| 1 |  |  |  |
| 2 |  |  |  |

## Functional Area Summary

*[Copy and paste the Functional Area Description from each Functional Area into the table below. This table will be copied into the resulting RFP as the mandatory Functional Requirements.]*

| **Functional Areas** |
| --- |
| 1 |
| 2 |

## Business Rules

*The goal of defining the business rules for the <project name> is to clarify the data and access validation efforts to be incorporated into the developed system. This section will provide the development team with data requirements and validation parameters to be applied when executing the required functions of the system. Knowing and documenting the <project name> rules will enable development efforts to be implemented effectively and will also provide the necessary awareness of these rules to those involved with the system definition efforts.*

*Further, once the rules have been separated from the requirements, it is easier to identify which rules are volatile. This will enable the development team to make smart decisions for future maintenance considerations about when to abstract the implementation, and when to embed it. User roles may also be further “restricted” based on additional parameters and these too will need to be provided in developing specifications for the system and for the system security model. Reasons for Business Rules:*

* *Can translate into specific code that validates data entered or enforces desired behavior from the user*
* *Describe what may or may not be done in a specific scenario*
* *Provides the criteria, conditions and exceptions for a scenario*
* *Can exist independent of requirements, they hold true whether they are incorporated in an automated system or a manual process*
* *Enforce policy*
* *Determine when information may change or when values are valid*
* *Define how decisions are made in a process*
* *Are more volatile since many are based on ever changing regulations and strategies*

| **Business Rule and Validation Requirements** | **Rule Type**  **(Data Validation, Business Constraint, Computation)** |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

*[Depending on the complexity of the system, a separate Business Rules Document. May be linked here:]*

***Link:*** *[please provide the link to the location of where this document resides.]*

## Regulatory Requirements

***Regulatory requirements*** *address legislative or legal requirements of an agency. They can be internal or external regulations.  These are non-negotiable. [Simply specifying that the system should adhere to a law or legislation is not an acceptable requirement. Elaborate on how the business system will meet the regulation. e.g., “The system must supply to the FHWA a complete Linear Referencing System (LRS) which includes all public roads using GIS by June of 2014 per FHWA MAP-21.”]*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

## Data Reporting Requirements

***Data Reporting Requirements*** *identify what reports and files the application and/or system must be able to manage. [E.g. Frequency of report, required run dates/times, recipients of reports, type of format, data source, distribution methods, and storage as it applies to reporting. A detailed report specification will be completed during the design phase for each report.] Identify impacts on Enterprise Data, Business Intelligence and Data Governance involvement [Inclusion of data in Enterprise views, avoidance of duplicating data, usage of Enterprise views in reports].*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

## Interface/Integration Requirements

***Interface Requirements*** *identify what information is needed to ensure that the system will communicate properly with external components. [E.g. Application integration (e.g. SFS, BDIS, Local Programs, et cetera), user interfaces (e.g. message display, navigation links, et cetera), communication interfaces (e.g. e-Mail, Web Browser, Electronic forms, et cetera), interfaces to other systems to complete daily/weekly data exchanges, et cetera.]*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

## Database Mapping

***Database Mapping*** *is the process by which a link between two distinct data models is created. Data mapping is used as a first step for many complex tasks associated with data integration which include data transformation or data mediation between a data source and its destination; identification of relationships in data which is vital in analysis of data lineage or consolidation of many databases into one while identifying redundancy.*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

## Data Dictionary

*A* ***Data Dictionary*** *is a collection of descriptions of the* [*data*](http://searchdatamanagement.techtarget.com/definition/data) *objects or items in a data model. This should include: name,* [*description*](http://www.businessdictionary.com/definition/description.html)*, and data type of each data element.*

***Link:*** *[please provide the link to the location of where this document resides.]*

## ER Diagrams (Logical Model)

*An* ***ER Diagram*** *is a graphical representation of entities and their relationships to each other within a database. This is a very high level, requirements oriented set of diagrams that will be elaborated upon by analyst personnel during the design phase. This section may be limited to simply identifying database systems and their interactions with other applications. Alternatively, it may include actual database schema diagrams and descriptions. The amount of detail will vary by project complexity.*

***Link:*** *[please provide the link to the location of where this document resides.]*

# Detailed Functional Requirements

[The **Detailed Functional Requirements** section lists functional specifications for each aspect of the system. **This will grow further, in more detail, in the Design Phase.** The structure of this section is dependent on system organization. For example, if the system is organized to follow the business unit structure, with each sub-system supporting a specific Customer or Consumer group, then each **Sub-system Description** should list main characteristics and functions of that group; on the other hand, if the system is organized by type of interface (data entry, reporting, etc.), then the Sub-system Description should outline common characteristics of those system components. **Due to the possible lengthiness of this section, this can be included in this document or added as an appendix/link.]**

## Sub-system

[This section describes **sub-system** components.]

## Component Type

[Component type description]

## Component 1

* Component Description
* Component Mockup (where appropriate)
* Component Business Flow
* Business Requirement(s) Traceability Matrix
* Logical Data Model
* Process Models
* Use Case/User Stories/ Functional Decomposition
  + **Functional Decomposition**: The main purpose of functional decomposition is to break up a large or complex business process or function into smaller and more manageable chunks in order to make the function more easily understood. The individual elements of the process and their hierarchical relationship to each other are commonly displayed in a diagram called a functional decomposition diagram. When Functional Decomposition is used as the means of specifying the functional requirements it must include the purpose, inputs, operations and outputs.

***Link:*** *[please provide the link to the location of where this document resides.]*

* + **Use Case**: A Use Case describes what an actor wants a system to do that provides some value to the actor. A Use Case specification includes the textual properties of the use case, a diagram depicting the interaction between all of the actors of the system with the use case and the graphical user interface specifications for the functionality defined.

***Link:*** *[please provide the link to the location of Use Case Specification.]*

* + **User Stories** (If you are using the Agile Methodology): User Stories are an extremely high-level definition of a requirement, containing enough information so that the developers can produce a realistic estimate of the effort to implement it.

**Epics** are large user stories too big to implement in a single iteration.

The following template is provided for a User Story.

Story points: each user story should be accompanied by story points. Each user story is expressed in story points, which is an estimate of the relative complexity of a story

***Link:*** *[please provide the link to the location of where this document resides.]*

* Class Diagrams show the static structure of the model, in particular, the things that exist such as classes, their internal structure, and their relationships to other classes. Class diagrams do not show temporal information.

A class diagram is presented as a collection of (static) declarative model elements, such as classes, packages, and their relationships, connected as a graph to each other and to their contents. Class diagrams may be organized into (and owned by) packages, showing only what is relevant within a particular package.

**Link**: [please provide the link to the location of where this document resides.]

# Non-Functional Requirements

## Technical Requirements

Technical requirements identify technical aspects and constraints that must be considered when defining the new system. Considerations may include accessibility needs of Consumers, whether or not the storage and handling of data must follow specified encryption regulations, or whether the system will operate on internal agency hardware or will be hosted at either an internal or external data center. **The requirements below would be anything outside of the current standards.**

### Applicable Current Standards

The Development Standards identifies or adopts standard techniques and practices related to the development of application software. Project Development Standards are adopted from ITS Department standards. References to Standards Documents and Policy Guides are included in Systems Design documentation.

| **Standard** | | **Description** | **Reference** |
| --- | --- | --- | --- |
| Coding | | Coding standards are specific to the development tool and define, naming conventions and the structure and formatting of programs in the new system. Coding standards include SQL standards which define techniques for data access. | Reference ITS Coding standards.   * JDeveloper * Oracle Warehousing * PL/SQL * Java |
| Web Presentation (GUI) | | GUI standards establish the look and feel of the user interface and how components are assembled. This includes applications styles (e.g. dockable menus), application frameworks, (e.g., Microsoft MFC) and standardized GUI components (e.g., message boxes, menus etc.) | Reference ITS Web Development Standards. |
| Help System | | Help System standards define the tools and overall design standards for the application help system. This includes standards for page and field level help, how help content is defined and maintained and how help interfaces with other definitional aspects of the application (i.e. data dictionary, system and user documentation). | Reference ITS Help System Standards. |
| Metadata | | Metadata standards define the methods and naming conventions used to develop the data model. | Reference ITS Metadata Standards. |
| Software Configuration Management (SCM) | | SCM standards define the tool(s) and methods for managing and tracking changes to application software as it evolves through the development lifecycle. | Reference ITS Software Configuration Standards. |
| Software Quality Assurance (SQA) | SQA standards define the processes and metrics used to ensure and measure software quality. | Reference ITS Software Quality Assurance Standards. |

### Accessibility

**Accessibility** is the degree to which a system is available to as many people as possible. Specifically, ensuring people with disabilities have access to and use of information and data that is comparable to the access and use by [those] who are not individuals with disabilities, unless an undue burden would be imposed on the agency. ([Fed. Sect. 508 definition](http://www.section508.gov/index.cfm?fuseAction=stds))

| **Standard** | **Description** | **Reference** |
| --- | --- | --- |
| Accessibility for people with disabilities | New York State's Information Technology Policy on Accessibility of Web-Based Information and Applications is based on Federal Section 508 Standards, Subpart B, section 1194.22 and Subpart C, section 1194.3  It establishes minimum accessibility requirements for web-based Information and Applications developed, procured, maintained or used by state entities. | **STATEWIDE:**  NYS IT Policy [NYS-P08-005 Accessibility of Web-Based Information and Applications](http://www.oft.state.ny.us/policy/99-3.htm)  Contact the State Agency Accessibility Coordinator with questions and for guidance about the policy, the interpretation, and QA/QC activities for a particular agency.  [NYS-P08-005 State Agency Accessibility Coordinators](http://www.its.ny.gov/policy/2010NYSAccessibilityCoordinators2.pdf)  Guide to implementing the Section 508 Standards for Electronic and Information Technology is available at [US Access Board](http://www.access-board.gov/sec508/guide/)  Web Accessibility Initiative (WAI) on World Wide Web Consortium offers strategies, guidelines, and resources for designing, developing and implementing accessible Web applications. <http://www.w3.org/WAI/> |

***Accessibility*** *requirements applicable to this project:*

(Add rows needed for requirements that exceed or otherwise deviate from the above standards.)

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  | APPLICABLE CURRENT STANDARDS-Accessibility |  |  |
|  | Exemption from APPLICABLE CURRENT STANDARDS-Accessibility (Full/Partial) |  |  |
|  | Assistive Technology required for one or more employees |  |  |
|  |  |  |  |

### Encryption

Whether or not the storage and handling of data must follow specific Encryption regulations.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Hosting

Whether it is required that the system will operate on internal agency hardware or will be Hosted at either an internal or external data center.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Environment

**Physical Environment** in which the system must function. Location (indoors, outdoors, residency, main office, etc.), number of locations, temperature and climate constraints, dimension constraints, stability, mobility, safety and durability are some of the factors to consider.

**Link**: [please provide the link to the location of where this document resides.]

### Business Continuity

**Business Continuity** describes the overall sensitivity of the system processes and data in terms of required availability and acceptable downtimes. Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner).

Example: Following are the anticipated impacts of system downtime.

* It is expected that system downtime…
* In the event of sustained outages…
* The risk is highest for requests for…
* Servers are located in different physical locations…
* An impact is present of external users...

#### Disaster Recovery

***Disaster Recovery*** *addresses the ability to recover from power failures, lost data, system failure, acts of nature, or sabotage. Discussion points include criticality of the system, acceptable response time to recover (RTO), point of restoration (RPO), redundancy and failover plans. Disaster Recovery is a subset of business continuity. The Recovery Time Objective (RTO) is the duration of time and a service level within which a business process must be restored after a disruption in order to avoid unacceptable consequences associated with a break in* *business continuity. The Recovery Point Objective (RPO) is the point in time to which you must recover data as defined by your organization. This is generally a definition of what an organization determines is an "acceptable loss" due to a disruption in availability.*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Security Analysis

**Security Analysis** begins with an identification of the security decision makers, the systems administrator, the “delegated administrators” and the general system users. These authorization levels are defined in detail in the Security Standards and Guidelines. A major consideration during Security Analysis is identification of data restrictions and requirements based on ownership, intellectual property, privacy, confidentiality and accuracy. For detailed items designed to highlight security issues, tasks and deliverables see the ITS SDLC document – Security Checklist.

| **Standard** | **Description** | **Reference** |
| --- | --- | --- |
| Security Standards | Security Standards identify user types and access controls that are standard across applications. | Reference ITS- Security Standards and Guidelines. |

#### Application Security

The intent of this section of the document is to detail the requirements for authorization to use the <project name> application. This is used to build application security at the functionality level. Define the following:

##### Security Requirements

**Security Requirements** identify the security, confidentiality, integrity and privacy issues affecting access to the product, use of the product, and protection of the data the product uses or creates. [E.g. Roles, access areas, user privileges, number of people in each role, et cetera.]

*Categorize requirements as to level (system, software, function, element, etc.) and whether they are requirements or other kinds of constraints.*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

##### Application Access and Decision Making Authorization

Define the role and describe the access that it has. If the role is defined generally (e.g. “maintain account”) this may include information about permissions to create, update, read or delete (accounts). Describe any restrictions that may exist relative to the different roles. Restrictions may include limiting access to data (e.g. classified as confidential), or data that is not relevant to the end user (e.g. regional data for regions other than end user).

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Description** | **Permissions** | **Restrictions** |
| <role1> |  |  |  |
| <role2> |  |  |  |
| <role3> |  |  |  |
| Etc. |  |  |  |

##### Security Matrix

Access of groups and individuals, groups and systems to specific systems functionality.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Events v. Roles** | <role 1> | <role 1> | <role 1> | <role 1> | <role n>… |
| <event 1> | x | x |  |  | x |
| <event 2> | x | x | x |  |  |
| <event 3> |  |  |  |  | x |

##### ISO Security Model (Data Classification)

Communicate with the Information Security Office (ISO) through the life cycle of the project. The ISO has a model for data security classification as follows:

C – Confidentiality

I – Integrity

A - Availability

Include the ISO on the team in the concept phase, prior to the initiation of a project. When the User begins to think about a project, considerations regarding CIA need to be made. The Requirements Phase is the best stage to Identify and classify risks. Consider the following:

* What the data is – are there confidentiality issues?
* Who will need access and how will they get it?
* How “right” does the information have to be?
* What will be the impact if the information is unavailable?

### Data Integrity

Specify precision (resolution) and accuracy (by some known standard) that is required in the systems output. Example: Precision will be determined for individual computations.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Design Constraints

This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.

| **#** | **Constraints** |
| --- | --- |
|  |  |

#### Infrastructure and Application Architecture Requirements

It is important that all project initiatives and their requirements comply with existing technology standards. Use this section to position this project within that framework. Remember that specific technologies are not normally a part of requirements. If presented as such, they often become constraints. For example, a user might state “the solution must use a Microsoft SQL Server® database”. Since this might be a violation of the approved information architecture and technology infrastructure of our agency, you must verify compatibility, documenting any variance here. Do not pre-judge approval or disapproval of technology constraints that are presented as requirements. Simply raise the red flag, providing information to support a decision. However, if the decision is made during business analysis activities, document both the requirement and the decision, indicating parties involved and when the decision was made.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Non-User Interfaces

This section defines the non-user interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, etc, so that the software can be developed and verified against the interface requirements. Specifies an external item with which a system must interact, or constrains on formats, timings, or other factors used by such an interaction.

|  |  |
| --- | --- |
| **Identifier** | Name of External Interface (filename or report identifier). |
| **Source** | Source of External Interface (function, system or organization) |
| **Destination** | Destination of External Interface (function, system or organization) |
| **Type** | Media of Interface (EDI, XML, tape, EFT) |
| **Reference** | Reference to supporting documentation including file descriptions and required data definitions. |

#### Hardware Interfaces

This section defines any **hardware interfaces** that are to be supported by the software, including logical structure, physical addresses, expected behavior, etc.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

#### Software Interfaces

This section describes **software interfaces** to other components of the software system. These may be purchased components, components reused from another application, or components being developed for subsystems outside of the scope of this project, but with which this software application must interact.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

#### Communications Interfaces

Describe any **communications interfaces** to other systems or devices such as local area networks, remote serial devices, etc.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

### Legal, Copyright and Other Notices

This section describes any necessary legal disclaimers, warranties, copyright notices, and patent notice, trademark, or logo compliance issues for the software.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

### Purchased Components and Licensing Requirements

This section defines any licensing enforcement requirements or other usage restriction requirements which are to be exhibited by the software.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

## Operational Requirements

Specify any administrative constraints or expectations that must be supported by the system. These requirements may include system performance expectations, technical infrastructure constraints, security mechanisms that must be followed, the need to regularly archive data, and any mandated audit and control processes.

### System Performance

System Performance refers to response times, throughput (transactions per second), and system resource (CPU, memory, network and disk I/O) utilization levels of the application and the systems that support it. In general, response time is an end-user concern, throughput is a business concern, and resource utilization is a technical concern.

This section documents the specific performance needs for a particular system by identifying transaction volumes, user workloads, business usage scenarios, and performance requirements and objectives. These may be found in a variety of sources such as project documents including: ITIR, RFP, Consultant or Vendor Contracts, SLA, Project Definition and Scope. Technical requirements and objectives may be found in current software development policies and standards and resource utilization requirements from technical services groups.

The information for this section will evolve and be elaborated through the initiation, requirements, and design phases. And, then will be used as input for test planning and development in the construction phase.

Definitions:

* Performance **requirements** are those criteria that are absolutely non-negotiable due to things like contractual obligations, service level agreements (SLAs), or fixed business/technical needs. Failure to meet a requirement may lead to a decision to delay a release until the criterion is met.
* Performance **objectives** are measures that equate to the most desirable performance characteristics and typically focus on metrics that can be directly related to user satisfaction or product quality.
* Performance **thresholds** are the upper limits of acceptable measures and typically are used to evaluate what combination of configuration settings will result in the most desirable performance characteristics.

**Performance Requirements**

| **#** | **Requirement** | **Source** | **Additional Information** |
| --- | --- | --- | --- |
| e.g. | Enterprise Dashboards must not take more than 6 seconds to load. | Contract CMS-1234 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Performance Objectives and Thresholds**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Objective with Threshold** | **Source** | **Additional Information** |
| e.g. | Normal response time should not be greater than 3 seconds to return each page or requested item. Threshold=6 seconds during extreme load conditions. | Contract CMS-1234 'response should be equal to or better than current system/version' | Response metrics were developed with a MO Project Leader performing the functions in v6.1 and QMU analyst using a stopwatch to record response times. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Instructions:

**#** is reference to a requirements management system number or, a sequential number assigned here for identification and reference.

**Requirements** and **Objectives** are stated in SMART terms = Specific, Measurable, Attainable, Realistic, Time-bound (if applicable).

**Source** identifies the source of the data e.g. Contract [id], SLA [ver, date], Scope [ver, date], Standard [id], Policy [id]

***Threshold*** *defines the upper limit of acceptability.*

#### Availability

**Availability Requirements** identify a measurement of the planned uptime during which the system is actually available for use and fully operational. These requirements should be documented in terms of application and required availability. [E.g. Available 24x7, 365 days a year, et cetera.]

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

#### System Capacity and Scalability Requirements

**System Capacity and Scalability Requirements** identify the expected load on the system, how many users are likely to access the system concurrently, system usage and the system attributes. [E.g. Average number of concurrent users; maximum number of concurrent users, average number transactions processed per day, maximum number of transactions processed per day, et cetera.]

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

#### Usability Requirements

**Usability Requirements** identify what users’ abilities and expectations of usage experiences the product must conform to. Meant to address user friendliness and how the user interfaces (screens) are designed with consideration for the different types or users and their skill sets. [E.g. On-line field help, pop-up calendars for date fields, feedback to users on tasks, progress bars, hourglasses, minimize data entry through use of pick lists, save documents in progress, et cetera.]

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Flexibility Requirements

**Flexibility (augmentability and expandability) Requirements** measure the ease of adding new capabilities to the product. [E.g. Ability and ease of adding new functionality, modifying existing functionality, adding additional content, et cetera]

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Robustness Requirements

***Robustness Requirements*** *identify the degree to which the product continues to function properly when confronted with invalid inputs, defects in connected software and hardware components or unexpected operating conditions. [E.g. Ability for the solution to recover all changes made in the file up to one minute prior to the failure, et cetera.]*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Audit and Controls

Addresses the need to trace and log use of the system as to updates in database, operations performed, web page visitors, etc.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Software Quality Assurance (SQA)

[Assurance that the system meets specified requirements and Customer needs and expectations.

*This section addresses product quality, Scalability, Completeness, Absence of bugs, Extensibility, Maintainability, and conformance to requirements or program specification related to Reliability, fault tolerance and robustness, how the system will respond to data exceptions, system failures and hardware failures.]*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### System Administration

Addresses how authorization is assigned and process by which problems are reported and resolved.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Backup and Recovery

Backup and Recovery requirements may include many daily chores such as system backups, change control, and help desk. Backup and Recovery is not something implemented at the time of a disaster; Backup and Recovery refers to those activities performed daily to maintain service, consistency, and recoverability.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Archiving and Retention Requirements

Define the process to retain data after it has served its usefulness. Should data be purged from the system? How long data should be saved? Will there be a need to access historical data? How fast do you need to get it?

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

## Transitional Requirements

Define the realm of conditions that must be satisfied prior to physically implementing the system in a production environment, or to relegating support responsibilities to the Performing Organization. Data conversion requirements and development and delivery of Consumer training programs and materials fall into this category.

### Data Conversion

**Data Conversion** requirements identify the specific requirements pertaining to data conversion and cleansing

[E.g. Migration of employee data from current application to new application, data cleansing efforts, et cetera.]

| **#** | Requirements | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Training Requirements

**Training Requirements** identify all user, help desk and support training required.

[E.g. required materials, level of training, number of participants, duration of training, et cetera.]

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

#### Roles of Customers/Consumers of the System

Identify training needs for specific user roles, documentation needs, system access, hardware needs, deployment logistics, etc.

### Support Requirements

**Support Requirements** provide information relative to the impact the application is expected to have on operational support resources and should include:

* Help Desk Support
* Network Support
* Application Support
* Database Support
* Administrative Support
* Security Support
* Training Support

|  |  |
| --- | --- |
| **Support Type**  Type of support required; Help Desk Support, Network Support, Application Support, Database Support, Security, Training or Administrative Support. Each system may require multiple Support Types. | |
| **Service**  **Level Definition** | Defines expectations related to system performance and availability and the related support needs in a way that can be measured. Support needs include requirements for help desk problem resolution turnaround, support coverage/schedule, defect resolution turnaround, systems availability, security support and other service needs as defined by the customer.  It is not expected that a “Service Level Agreement” will be developed for each application. However, an assessment of the operational support needs of the business application should be developed to determine if those needs fall within the parameters of ITS’ existing support levels. |
| **Role** | The roles expected to be performed by support individuals. Role definitions are available in the Roles and Responsibility appendix. For example: Application Support may require a Business Analyst, Developer and Tester. Database Support may include Data Architect and DBA. |
| **Estimated HRS/WK** | Estimated hours per week of support for roles defined. |

### Documentation

The development of appropriate documentation, such as manuals for operations and maintenance, are required for successful transition. (Examples: User manuals, technical, training, etc.)

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Help

Describe the Help Requirements. Example: Each feature of the <system> will have built-in online help for the user. Online Help shall include step by step instructions on using the System. Online Help shall also include definitions for terms and acronyms.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Deployment

***Deployment*** *addresses all the activities that make the system available for use. Examples: Big Bang Approach where implementation happens in a single instance. All users move to the new system on a given date. Phased Rollout where changeover occurs in phases over an extended period of time. Users move onto new system in a series of steps. Parallel Rollout where both the legacy and new system run at the same time. Users learn the new system while working on the old.*

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |

### Release Validation

**Release Validation** addresses availability of customers for testing, what type of deployment is acceptable (all or piecemeal), customer expectations of system, etc.

| **#** | **Requirements** | **Priority**  M – Must  S – Should  C – Could  W – Won’t | **Additional Information** |
| --- | --- | --- | --- |
|  |  |  |  |