Business Requirements

Document

Document Control / Version History				
Version	**Date**	**Author**	**Description**	**Approved**
0.1	11/14/2019	RMJ	Initial draft	MT
0.2	11/27/2019	RMJ	2nd draft	MT
0.3	12/4/2019	RMJ	3rd draft	MT
1.0	12/10/2019	RMJ	Incorporated IS & PMO edits & forwarded revisions	KS & MT
2.0	12/19/2019	RMJ	Incorporated IS & PMO edits, forwarded revisions & sent to President	KS & MT
3.0	1/7/2020	RMJ	Greenlighted by all/ proofed/test printed/removed all occurrences of '*This page is intentionally blank*' for online consumption (but kept in hardcopy version), as requested by IS Mgr	NP, KS & MT
4.0	1/13/2020	RMJ	Final proof & forwarded final draft to IS Mgr, PMO & President	NP, KS & MT
5.0	1/18/2020	RMJ	Signoff/approval of final draft by IS Mgr, PMO & President/uploaded to IS & PMO SP sites	NP, KS & MT

Notice

Contents

1 Introduction

ABC's *Business Requirements Standards Document* is a comprehensive, mandatory, and integral part of Information Services (IS) projects. Implemented through the Project Management Office (PMO), this document is designed to support the documentation of all software and internal projects developed for and by the company.

It is essential for aligning stakeholders and guiding the development of effective solutions that meet our business needs. It not only serves as a blueprint for ABC's projects but also as a dynamic document that evolves through collaboration and continuous review.

1.1 Purpose

The Business Requirements Standards Document defines the format and content of documentation for software acquisition, development, and sustaining engineering. Its purpose is to provide a framework and model for recording essential information needed throughout the development life cycle and maintenance of software systems and/or internal processes. Additionally, it promotes consistency of documentation across the business and provides status on all corporate projects.

1.2 Scope

This Business Requirements Standard Document ensures:

Scope	
1.	Documentation goals of each project are adequately satisfied.
2.	Clear descriptions of the software management, engineering, and assurance processes and products are provided.
3.	Consistency in format across project documentation is achieved.
4.	Traceability of each phase of the development life cycle is maintained.

Furthermore, this document focuses on the overall process of requirements documentation and answers the following questions:

- What are requirements?
- Why are requirements important?
- Who is involved in determining and documenting requirements?

As set forth by IS, the Project Management, Office, and the Office of the President, compliance with ABC's Business Requirements Documentation Standards, policies, and procedures is mandatory.

2 Requirements

2.1 Overview

A *Requirement* is a statement defining a capability, characteristic or constraint for a product, service or event. Requirements define capabilities needed by a business user to solve a problem or achieve an objective. These capabilities must be met or possessed by a product to satisfy a contract, standard, specification or other formally imposed documentation.

Generally, a requirement is something wanted or needed. In the context of ABC's IS projects, requirements are the documented objectives a project should achieve. Requirements are developed based on the business case for the project or problem to be solved.

The *Project Initiation Agreement (PIA)* describes what areas are within or outside the scope of the project. It also lists project deliverables. Requirements are defined shortly after the PIA is approved. **Note**: No development work may start until requirements are analyzed and accepted by the business users.

Requirements should fit within the agreed-upon scope of the project. Well-formed and specific requirements also become the basis for developing test plans as well as justification for the formal acceptance of test results. Eventually, as business users accept deliverables and approve changes to be implemented as part of IS projects, they can confirm how well each requirement has been met.

The development process should allow for changes to stated requirements including new requirements that were not obvious during the analysis stage. Oftentimes business activities and efforts must be completed for the project to be successful. And such efforts should be included in the *User Requirements Document* and the *Project Plan*, and if appropriate, in the *Project Risk Assessment*.

2.2 Functional Requirements

Functional Requirements are imposed by business users. They also include requirements from external regulatory agencies. These requirements pertain to what systems must do from a user's perspective or the systems' capabilities provided to users and other parties.

Functional requirements are also referred to as *User Requirements*.

Examples of functional requirements include:

1. '*The system must exclude adjustments to invoices dated more than six months ago*.'

2. '*Category Management reports should be sorted by buyer number within category.'*

2.3 Non-Functional Requirements

Non-functional Requirements support the functional requirements in areas such as security, performance, availability, reliability, and resistance to failure, capacity, and more. Constraints are a major type of non-functional requirement.

Business users may or may not state non-functional requirements directly, however these requirements must be satisfied for functional requirements to be fully satisfied.

Examples of non-functional requirements include:

1. '*The system should be accessible during normal work hours, 6:30 AM through 7:00 PM, Mondays through Fridays.'*

2. '*The new Category Management database will be established and maintained as DB2 tables.'*

3. '*The conversion should not occur during EXPO shipment periods, from early July through late October.'*

Requirements are described and agreed upon between the business users and the development team. They are the detailed specifications that the development team must meet for business users to accept deliverables.

Without well-formed, documented, and agreed-upon requirements, the following issues can occur:

- Developers may be wasting effort on items that are not required.
- Developers may be focusing more on low-priority, low-risk tasks than truly critical deliverables.
- Business users have no clear indicator to establish that deliverables are acceptable.
- Business users and developers would have no guidance in developing test plans and test scripts.

Note: Requirements that are misunderstood or misstated during the early stages of the development process may lead to expensive rework and fixes. Whereas requirements that are documented and accessible facilitate traceability. Each requirement can be traced backward or forward to determine:

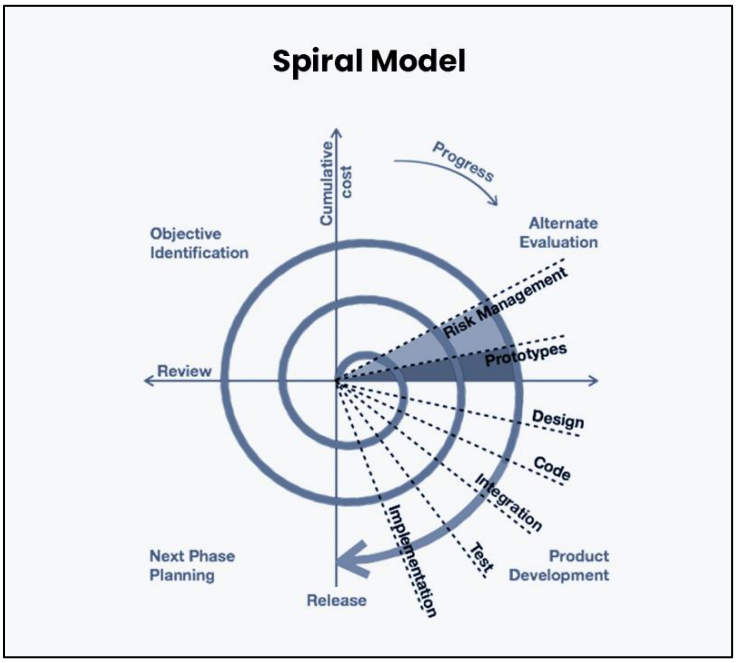
- Who or where the requirement came from
- Whether the requirement is within scope
- Which test cases help ensure the requirement is met with satisfaction
- Which of the delivered system functions meet stated requirements
- Whether a proposed function or change is **not** part of agreed-upon requirements

2.4 Requirements Models

2.4.1 The Spiral Model

The Spiral Model, illustrated below, describes an iterative effort at determining and verifying requirements. Proposed requirements are continually analyzed and verified.

Even after stated requirements are formally accepted and a baseline is achieved, new requirements may be introduced and known requirements re-evaluated. However, if any new requirements are introduced after receiving formal signatures to approve the stated requirements, the Project Lead will need to submit and receive approval on a formal change request form, as described in the Project Management Office's procedures. *(Refer to the Change Management Procedure document that appears in the "Managing Change" section of the Project Management Office's department web page on E-Net.)*



The Spiral Model, also known as the *spiral lifecycle model*, is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the *prototyping* model and the *waterfall* model, and is favored for large, expensive, and complicated projects.

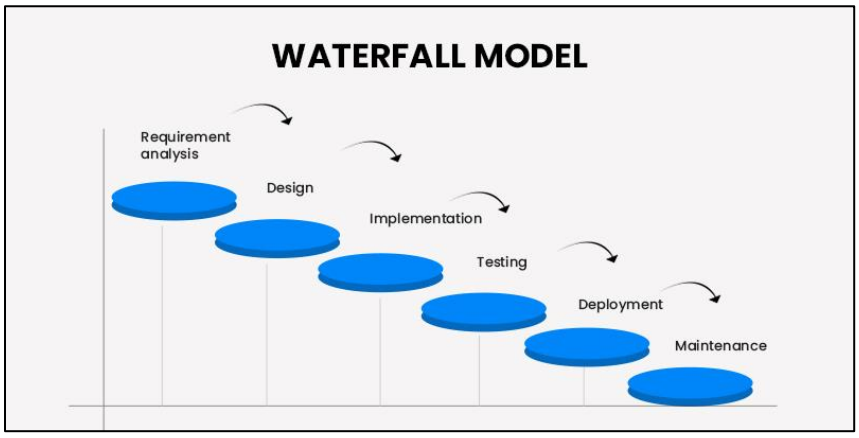
Generally, the steps in the Spiral Model reflect:

Spiral Model	
1.	The new system requirements are defined in detail and usually involve interviewing a few users representing all external or internal users and other aspects of the existing system.
2.	A preliminary design is created for the new system.
3.	The first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system and represents an approximation of the characteristics of the final product.
4.	A second prototype evolves by a fourfold procedure: a. evaluating the first prototype in terms of its strengths, weaknesses, and risks b. defining the requirements of the second prototype c. planning and designing the second prototype d. constructing and testing the second prototype
5.	At the customer's option, the entire project can be aborted if the risk is deemed too great. Risk factors may involve development cost overruns, operating-cost miscalculations, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
6.	The existing prototype is evaluated in the same manner as the previous prototype. And if necessary, another prototype is developed according to the fourfold procedure outlined above.
7.	The preceding steps are iterated until the customer is satisfied that the refined prototype represents the desired final product.
8.	The final system is constructed based on the refined prototype.
9.	The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

2.4.2 The Waterfall Model

The Waterfall Model is a popular version of the systems development life cycle model for software engineering. Often considered the classic approach to the systems development life cycle, this model describes a linear and sequential development method.

Waterfall development has distinct goals for each phase of development. Imagine a waterfall on the cliff of a steep mountain. Once the water has flowed over the edge of the cliff and has begun its journey down the side of the mountain, it cannot turn back. It is the same with waterfall development as once a phase of development is completed, the development proceeds to the next phase and there is no turning back.



Advantages

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping or iterative steps.

Disadvantages

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well thought out in the concept stage. Alternatives to the waterfall model include joint

application development (JAD), rapid application development (RAD), synch and stabilize, build and fix, and the Spiral model.

2.5 Determining and Documenting Requirements

Stakeholders are defined as people or parties who are directly or indirectly affected by the project. They include anyone with a vested, substantive interest in the project, as well as individuals and organizations that are actively involved in the project, or whose interests may be affected because of project execution or project completion.

Stakeholders for the company's IS projects shall include business users, project team members, and any impacted IS functional or technical groups.

Project Leaders are responsible for determining and documenting requirements. Other members of the development team such as Business Systems Consultants, acknowledged Business Experts, as well as Outside Consultants may also be involved in the process of determining and analyzing requirements.

Business Users primarily the Project Sponsor and business representatives to the Project Team, will be involved in the collaborative effort to describe, analyze and document requirements.

Other stakeholders, apart from those who are directly involved with the project, will also be engaged in the requirements determination process. These may include external parties such as our trading partners (i.e., customers and vendors) or government regulatory agencies.

The *Development Team* represented by the Project Leader proposes the analyzed requirements in a formal document. The business users and stakeholders, represented by the Project Sponsor, will approve and accept, or otherwise send back the proposals for rework.

2.5.1 Requirements Determination Process

The mandated process derives from standards and recommends best practices by PMI (Project Management Institute) and IEEE (Institute of Electrical and Electronics Engineers). Those guidelines are adapted for use within ABC.

The process follows the Spiral Model whereby the elicited, analyzed and documented requirements are continually validated, and changes are properly managed.

There are five major steps within this process including:

1. Requirements Elicitation
2. Requirements Analysis
3. Requirements Specification
4. Requirements Validation
5. Requirements Management

2.5.2 Gathering Requirements

Gathering requirements for a project demands documentation for approval and used for reference throughout the entire project. Listed and documented are the Requirements Process steps and the Requirements Worksheets in logical sequence in the spiral requirements development process.

As an example. the Stakeholder Identification Worksheet must be completed prior to the Scope Definition Worksheet because without knowing who the stakeholders are, we could not possibly know the correct scope of the project. An in-depth order of use is shown in the *Requirements Standards – Templates Flowchart*.

Depending on the size of the project, some of these worksheets are mandatory and some are not. Those that are **not** mandatory, their use may be ‘strongly recommended.’ For rules governing worksheet use, reference the Requirements Process Checklist. Note: No attempt has been made in this document to define what is or is not necessary.

Keep in mind that during the requirements process, these forms are a 'work in progress' until the final signoff of the User Requirements Document. As more information becomes known about the requirements, these worksheets will be updated to reflect the new information.

2.5.3 Structuring a Requirements Document

After the *Business Case Document (REQ-05)* has been approved, the project team starts to develop the more detailed user requirements. During this process, the team will develop three key sets of worksheets that will be maintained throughout the project's life cycle. The worksheets include:

2.5.3.1 Requirements Analysis/Status Worksheet (REQ-08)

This worksheet is the primary repository for the documentation and maintenance of all user requirements. User requirements can be added to or modified on this worksheet at any time during the Requirements Process or anytime during the project's life cycle. It is intended to be THE working document for recording the user requirements and maintaining the status of those requirements. If a requirement is no longer necessary, it will continue to reside on this worksheet for documentation purposes, but its status would be identified as "eliminated."

2.5.3.2 Requirements Test Record Worksheet (REQ-11)

This worksheet is used to document test cases that will be conducted during User Acceptance Testing to verify whether the user requirements have been met. As requirements are modified, added, or eliminated, the Requirements Test Record Worksheet (REQ-11) needs to be updated in accordance with those requirements changes.

2.5.3.3 Requirements Traceability Matrix (REQ-12)

Used in conjunction with the Requirements Analysis/Status Worksheet (REQ-08) and the Requirements Test Record Worksheet (REQ-11), this worksheet is also maintained throughout the project's life cycle. It acts as the roadmap between the business requirements, user requirements, and test cases. As requirements are modified, added, or eliminated, this matrix is used to determine what other requirements are impacted, as well as which test cases are impacted.

Stakeholder Identification Worksheet (REQ-01)				
Project Name		**Project No.**		
Stakeholder Department or Organization	**Name / Title**	**Project Role (PIA resource)**	**Degree of Involvement (High, Med, Low)**	**Point of Contact**

Existing System Documentation Worksheet (REQ-02)			
Project Name		**Project No.**	
ID	**Potential Sources**	**Location of Documentation From/To**	**Relevance to Current Project**

Scope Definition Worksheet (REQ-03)	
Project Name	**Project No.**
Statement of Problem/Opportunity/Need: *(What are we doing?)*	
Justification and Rationale: *(Why are we doing this?)*	
Scope: *(Boundaries for the project – what's in, what's out)*	
Business Constraints: *(When, where, other things to know)*	
Technical Constraints: *(hardware, software, network, etc.)*	

USER CLASSIFICATION MATRIX (REQ-06)			
Project Name: _____ Project No._____			
ORGANIZATION / USER	**USER TYPE A**	**USER TYPE B**	**USER TYPE C**