Business Requirements Guidelines

August 25, 2001

Version 1.0

Important Information

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. [The company] may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, [the company] products, programming, or services that are not announced in your area. Such references or information must not be construed to mean that [the company] intends to announce such products, programming, or services in your area.

Notice

[The company] assumes no responsibility for any technical inaccuracies or typographical errors that may be contained herein. In no event will [the company] be held responsible for direct, indirect, special, incidental, consequential or any other loss or damage caused by errors, omissions, misprints or misinterpretation of the information found in this publication, even if advised of the possibility of such damages. [The company] expressly disclaims any and all liability to any person, in respect of anything done or omitted, and the consequences of anything done or omitted, by any such person in reliance on the contents of this publication.

Trademarks and Service Marks

The following terms are trademarks or service marks of [the company] in the United States and/or other countries:

Contents

Revisions

Date	Document Version	Software Version	Author	Pages Affected	Comments
8-25-2001	1.0		J.Michko	All	

Introduction

A Business Requirement usually begins with a statement of a specific strategy or goal developed by a leadership team. The team identifies business strategies and goals to meet business needs arising from competitive, regulatory, operational, and other business pressures. The leadership team then assembles a project team to accomplish the strategy or goal. This guide provides an overview of Business Requirements, as well as recommendations and techniques for effectively developing such requirements.

The immediate question confronting the project team is ***How*** they will accomplish the strategy or goal and ***How Much*** will it cost in resources, time, and funding. To answer these questions, the project team must first clearly define ***What*** must be accomplished. This can best be accomplished by following the requirements definition process as described in this guideline.

Business requirements provide answers about what must be accomplished for the project to be considered a success. The following are some of the typical questions that must be answered:

- What business functions are to be performed?
- What information is required?
- What results are expected?
- At what locations?
- For whom?
- How often?

Business requirements provide the criteria by which a delivered system is judged to determine the success of the final system. Moreover, well-defined business requirements become the starting point for setting stakeholder expectations, as well as ongoing project communications, status, deliverables, and milestones. Consequently, well-defined requirements are critical for a project team to be effective and essential for a project to be an ultimate success.

The importance of the project team and stakeholder allocating the time and effort to develop good business requirements cannot be overstated. The costs of correcting a problem after the introduction of a new product can be as much as 100 times greater than the cost of solving the problem during the development of requirements and the design of the product.

Note: The labels applied to requirements and designs in this guide are the standards defined within the the company's methodology. The concepts are universal but one can find many different labels applied to these same concepts across different methodologies.

Business Requirements Overview

Types of Business Requirements

Typically, most projects consist of two types of Business Requirements. Both types represent levels of detail in the requested business functionality: one high-level view and one more detailed view. Both levels are specified at different points during a project to identify ***What*** must be accomplished.

Type of Requirement	Characteristics
High-Level Business Requirements	The following are characteristics of High-Level Business Requirements: • Provide little detail, are conceptual in nature, and serve the strategic management and decision-making process. • Stakeholders normally include senior and executive levels of management. • An example of such a requirement for an airline might be "Provide real-time flight crew scheduling and re-routing."
Detailed Business Requirements	The following are characteristics of Detailed Business Requirements: • Describe how business functions, such as billing and scheduling, are to be performed and the expected results for users. • Provide details and clarity that effectively communicate user needs and expectations. • Are the critical component in the successful development of a solution.
System Requirements	The following are characteristics of System Requirements: • Developed in parallel to Business Requirements. • Answer questions about a solution's technical parameters required to provide the requested business functionality. • The combination of Business and System Requirements provide the information needed for a project team to answer the question of ***How*** to provide a solution and ***How Much*** the solution will cost. **Note:** This document does not provide guidelines for developing System Requirements.

Levels of Design

The combination of Business and System Requirements provides the information needed for a project team to answer the questions ***How*** do we provide a solution and ***How Much*** will the solution cost. This information is defined and documented during the Solution Design process.

Normally, there three levels of design, each representing a degree of detail needed at a particular point in the project to:

- Determine if there is a viable solution to the requirements
- Determine the cost effectiveness of the solution
- Provide the design details needed to proceed to the next stage of development.

The following are the three levels of design:

Level of Design	Characteristics
Conceptual Design	The following characterize the Conceptual Design: • Derived from High-Level Business Requirements. • Provides a high-level view or concept of how the requirements are to be met. • States details in broad terms. • Provides an estimate of (1) whether the requirements can be met, (2) how they might be met, and (3) the effort needed to complete the project.
Logical Design	Based on the Detailed Business Requirements and the Conceptual Design, the Logical Design: • Provides an additional level of detail on what needs to be accomplished to meet the requirements. • Refines the approach and necessary efforts to complete the project.
Physical Design	This is the final level of detail in developing and implementing a solution. • The Physical Design is derived directly from the Logical Design. • At this stage, technical teams require exact specifications of what is to be developed.

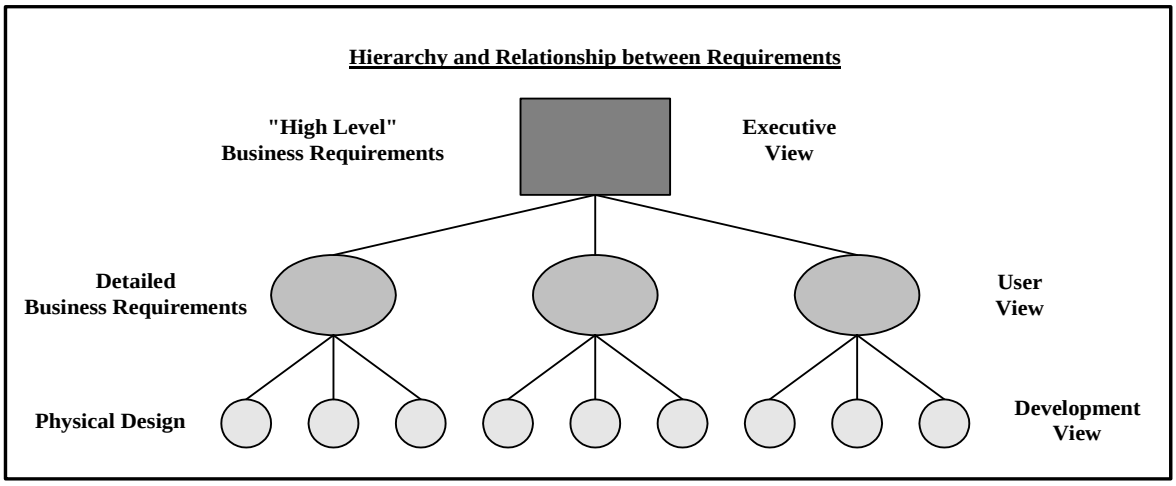


Figure 1. Types of Requirements

High-Level Business Requirements

Overview

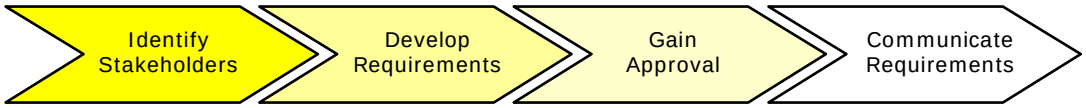
At the start of any project, business executives have a concept of what they want to accomplish. This initial concept is usually a single statement regarding a business strategy or an operational goal. When the concept is reviewed and broken down into the different components that need to meet the strategy or goal, the components become the High-Level Business Requirements for a project.

Typically, High-Level Business Requirements:

- Present little detail.
- Focus on business needs and not how the requirements will be met.
- When viewed together, provide a clear picture of what the project team must accomplish to be successful.
- Require each feature of the solution be in support of a requirement.

Activity Flow

In developing High-Level Business Requirements, it is important to follow a structured task and activity flow. Such a process ensures the success of the project by providing a clearly defined project scope and a method for managing the scope. The following is a graphical view of this process.



1. Identify Owners and Decision Makers	1.Gather Input	1. Perform formal requirements review	1. Provide requirements to all users and stakeholders
2. Identify User Groups	2. Identify Requirements by Business Area	2. Refine requirements (as needed) based on review results	2. Begin formal management of Requirements
3. Identify User SME's	3. Ensure Requirements are within project scope	3. Gain consensus among stakeholders	
	4. Resolve areas of uncertainty		

Items to Consider

When developing High-Level Business Requirements, it is important to remember one does not have to provide a lot of detail at this point. The following is a list of items to consider when developing requirements.

- What is the business need being requested?
- Who are the sponsors and decision-makers for the project?
- What benefits will be derived from the solution?
- How will this requirement impact other areas of the business?
- Is the requirement unique or related to another requirement either within the requesting business area or within another area?
- Are there any fundamental constraints, such as time, resources, funding, that should be identified?
- Do the new requirements align with the strategic goals of the business?
- What will be the major consequences when implementing the solution?
- Can the requirement be met with existing technology or capabilities?
- What is the priority of each requirement?

An Example of the Process

Although High-Level Business Requirements are conceptual in nature and do not provide a great amount of detail, they should provide enough information for the project team to gain a general idea of the business need, impact on the business and other systems, and costs to build and implement a new system.

The following table presents an example of a business need, a set of High-Level Business Requirements developed to meet that need, and a sampling of Project Team responsibilities.

Step	Example
Identify the business need	By July 1st, our company must be in compliance with new federal guidelines regarding workplace injuries.
Review federal guidelines and interview stakeholders	This step presents a typical method for identifying the High-Lever Business Requirements for the project.
Develop High-Level Business Requirements	1. By June 1st, our company must develop and implement a process to deal with work-related injuries that is in compliance with federal guidelines. 2. By June 1st, our company's ability to track work-related injuries must be in place. 3. By July 1st, our company must develop and implement a new "Accident Awareness" program that is in compliance with federal guidelines. 4. By July 1st, our company's "new hire orientation program" must be update to include information on these new processes. 5. Review all elements of High-Level Business Requirements to ensure they comply with the new federal guidelines.
Develop an action plan	1. Determine the impacts and costs associated with compliance of federal guidelines. 2. Gain approval of stakeholders and users for selected course of action. 3. Develop Conceptual Design that meets the needs defined by High-Level Business Requirements. 4. Determine the estimated project costs. 5. Present findings to Executive Sponsors for approval.

Detailed Business Requirements

Overview

Before developing Detailed Business Requirements, the project team should have:

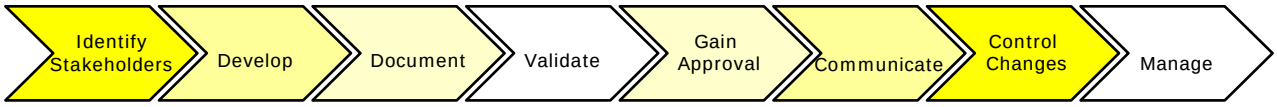
- Documented the High-Level Business Requirements.
- Developed a Conceptual Design for the solution.
- Determined the project's estimated cost.
- Presented all findings to the project sponsors for approval.

After the sponsors have given their approval, the next step involves identifying the Detailed Business Requirements. These requirements:

- Provide the project team with a level of detail about what must be accomplished.
- Enable the project team to develop a detailed Logical Design from which they can further estimate the solution's development and implementation.
- Must be directly related to a High-Level Business Requirement.

Activity Flow

In the previous section of this document, we discussed the importance of following a structured task and activity flow when developing High-Level Business Requirements. The same is true when developing Detailed Business Requirements. The following presents a graphical view of this process.



1. Identify Decision Makers	1. Gather Input	1. Categorize requirements	1. Validate requirements are well defined	1. Perform formal requirements review	1. Provide requirements to all users and stakeholders	1. Identify change control board	1. Begin formal management of requirements
2. Identify User Groups	2. Identify Requirements by Business Area	2. Prioritize requirements	2. Validate requirements against project scope	2. Refine requirements (as needed) based on review results		2. Implement change control process	2. Monitor requirements
3. Identify User SME's	3. Ensure Requirements are within project scope	3. Document using template or tool	3. Update Design & Prototype as needed	3. Gain consensus among stakeholders			
	4. Resolve areas of uncertainty						

Benefits of Well-Defined and Managed Detailed Requirements

The steps taken to develop well-defined Detailed Business Requirements result in many benefits. The following are just a few of the major benefits:

- Builds a shared understanding of expected results between the project team and users.
- Involves users in defining what is needed.
- Supports the likelihood that what is built matches what is needed and was requested.
- Reduces the chances of system rejection.
- Ensures needed functionality, not unused bells and whistles, are delivered.
- Manages possible trade-offs caused by time, budget, and other constraints.
- Ensures projects meet budget and resource estimates.

Characteristics of Well-Defined Requirements

The following characterize well-defined Business Requirements:

Characteristic	Definition
Correct	Each requirement accurately describes the functionality to be delivered. The reference for correctness is the source of the requirement, such as a customer, or it can be traced to a High-Level Business Requirement.
Feasible	It must be possible to implement each requirement within the known capabilities and limitations of the system and its environment. To avoid unacceptable requirements, a developer should work with analysts and users throughout the development process.
Necessary	Each requirement should be a valid customer need, or something required to conform to an external requirement, an external interface, or an enterprise standard.
Unambiguous	The reader of a requirements statement should be able to draw only one interpretation from the statement. Moreover, multiple readers of a requirement should arrive at the same interpretation. **Note:** Natural language is highly prone to ambiguity. Avoid subjective words like user-friendly, easy, simple, rapid, efficient, several, state-of-the-art, improved, maximize, and minimize.
Verifiable	Determine if you can devise tests or use other verification approaches, such as inspection or demonstration, to determine whether each requirement is properly implemented in the product. The requirement is not valid if it can not be verified through testing.

Examples of Detailed Business Requirements—The Good and the Ugly

Example 1	
Poorly Defined	The system should be user friendly. **Note:** Obviously, no one wants a system that is not user friendly. The real issue is there will be a time constraint in training users.
Well Defined	The system's design must be such that front-line users will be proficient at using the application after a one-day training session.
Example 2	
Poorly Defined	The system should have a quick response time. **Note:** This statement does not provide enough details.
Well Defined	The system should provide sub-second response times for screen refresh resulting in a user's ability to resolve customer inquiries within 60 seconds or less during peak business hours of 8 AM-5 PM Monday-Friday.
Example 3	
Poorly Defined	The system should be Web-based and developed using MS Active X. **Note:** Concentrate on "What" and not "How." The IT team will develop the technical requirements and a system design, and will answer how the Business Requirement will be met.
Well Defined	1. The application should be easily accessible from a standard Internet browser. 2. The application should allow for easy transfer of information (cut and paste) to the standard office suite of products.
Example 4	
Poorly Defined	Validate charge numbers online against the master corporate charge number list, if possible. **Note:** The term "if possible" does not deliver any information. Provide the specifics and assign a low priority to the requirement.
Well Defined	The system shall validate the charge number entered against the online master corporate charge number list. If the charge number is not found on the list, an error message shall be displayed and the order shall not be accepted.

Pitfalls to Avoid in Defining Detailed Business Requirements[1]

The following is a list of some of the many pitfalls project teams routinely experience if they do not follow a detailed approach to requirements definition and management. A more detailed explanation presenting the symptoms and solutions for each pitfall appears in Appendix A.

Pitfall	An Example of a Symptom
Confusion over "Requirements"	An executive's perception of requirements involves a high-level business strategy or goal, but a developer or engineer's requirements involves a detailed design or engineering specification.
Inadequate customer Involvement	Users often think: • Developers already know what they need. • The technical stuff does not apply to them. • They are too busy to gather and refine requirements.
Ambiguous and vague Requirements	• A requirement statement has several different meanings and there is uncertainty as to which one is correct. • Developers have to ask the analyst or customer many questions, and sometimes have to guess what is really intended.
Requirements that have not been prioritized	• A high percentage of all requirements have been classified as high priority. • Users are reluctant to prioritize because they fear the developers will restrict the project to the highest priority items.
Building functionality no one uses	• Users request specific features but never use them. • Developers add functionality that the users "are just going to love."
Analysis Paralysis	All requirements must be modeled "six ways from Sunday," the entire system must be prototyped, and development will be held up until all changes cease.
Scope Creep	The product scope was never clearly defined and new requirements are being added during development.
Inadequate change process	There is no specific process for dealing with requirements changes and new functionality becomes evident only during system or beta testing.
Insufficient change impact analysis	There has been no careful analysis of the implications of changes. The change may be too complex, take longer than promised, be technically impossible, or conflict with other requirements.
Inadequate version control	Approved changes are not incorporated periodically into the Requirements document. Project participants are unclear as to what is in the Requirements baseline.
Inadequate estimates for gathering requirements	The project has missed major milestones and there have been cost overruns.

[1] This material adapted, with permission from Karl E. Wiegers, Ph.D., 2001 "10 Requirements Traps to Avoid".

Reviewing and Organizing Requirements as a Group

After gathering all Detailed Business Requirements, it is important to review and organize the requirements as a group to ensure they are complementary and provide the necessary details to accomplish the project's High-level Business Requirements. The following are important additional characteristics to be considered when organizing requirements into a group.

Characteristic	Definition
Complete	• There should be no missing requirements or information. • Organize the requirements in a hierarchical fashion to help reviewers understand the structure of the functionality described and identify missing items. • A set of requirements must identify all impacted functionality represented by the High-Level Business Requirements.
Consistent	• Consistent requirements do not conflict with other requirements. • Disagreements within a group of requirements must be resolved before development can proceed. • It may be difficult to determine which requirement is correct until a review of each is completed. • Be careful when modifying requirements, as inconsistencies may go undetected if you review only the specific change and not all related listings.
Prioritized	• Assign an implementation priority to each requirement to indicate its importance in a particular product release. • If all requirements are regarded as equally important, the project manager is less able to react to new requirements added during development, or to budget cuts, schedule overruns, or the departure of a team member. • Priority is a function of the value provided, the relative cost of implementation, and the relative technical risk associated with implementation.
Modifiable	• It is important to revise a Requirements document when necessary and maintain a history of changes for each requirement. • Each requirement must be uniquely labeled and expressed separately from other requirements to provide its own identity. • To ease the process of modifying requirements, organize them so that related requirements are grouped together. Create a Table of Contents, Index, and Cross Reference List.
Traceable	• Ensure that each requirement can be traced to its source. • Link each requirement to the design elements, source codes, and test cases constructed to implement and verify the requirement. • Traceable requirements are uniquely labeled and are written in a structured, fine-grained way, as opposed to large narrative paragraphs or bulleted lists.

A Summary of the Process

1. There are three types of requirements.
   - High-Level Business Requirements - Described in broad terms, these requirements support a business strategy or goal
   - Detailed Business Requirements - Specifics of the business functionality that will be required to accomplish the high-level business requirements
   - System Requirements - Technical details regarding such items as performance, locations, and security. These requirements have not been covered in this document.
2. Remember the questions that business requirements must answer.
   - What functions are to be performed?
   - What information is required?
   - What results are expected?
   - At what locations?
   - For whom?
3. Start a requirements definition by developing the High-Level Business Requirements.
4. Avoid including design issues and specifications in business requirements.
5. Include sponsors, impacted users groups, and IT teams in definition process.
6. Each Detailed Business Requirement should be stated as a unique objective with the following attributes:
   - Correct – Accurate description of a feature or process
   - Feasible – The requirement can be achieved
   - Necessary – It is truly needed
   - Unambiguous – Only one interpretation
   - Verifiable – Stated in concrete terms, testable, and measurable
7. After individual requirements are gathered, they should be reviewed as a group to ensure they are:
   - Complete – Each requirement describes one result to be achieved
   - Consistent – No conflicts between requirements
   - Prioritized – Must have versus like to have
   - Modifiable – Necessary changes can be easily made
   - Traceable – Each requirement can be traced back to its origin
8. The use of a standard documentation template or tool will help to facilitate configuration control, tractability, and testing.
9. A formal change control process must be used to identify, control, track, and report changes.

Appendix A: Symptoms and Solutions for Pitfalls in Defining Requirements[2]

The following presents some of the many pitfalls that project teams routinely experience by not following a detailed approach to requirements definition and management.

Confusion over "Requirements"	
Symptoms	• An executive's perception of requirements may involve a high-level business strategy or goal. • A developer or engineer's requirements might look like a detailed design or engineering specification. • Customer-provided requirements occasionally read more like solutions. Project stakeholders often do not classify their requirements as high-level or detailed. Project participants, therefore, will have different expectations as to the amount of detail in the requirements. • Although users provide requirements, developers may not be sure what they are supposed to build. • If discussions during the development of requirements focus exclusively on functionality, participants might not understand the various kinds of information that fall under the broad fabric of requirements. Consequently, important stakeholder expectations might go unstated and unfulfilled.
Solutions	1. Recognize that there are several types of requirements. 2. Educate project participants on key requirement's concepts, terminology, and practices. 3. Clearly define the type of requirements being pursued by the team.
Inadequate Customer Involvement	
Symptoms	• Users often think developers already know what they need, or they believe all that technical stuff, such as business requirements, do not apply to them. • Users frequently indicate they are too busy to spend the time it takes to gather and refine the requirements. • Users draw on unprepared users or software developers to supply all of the input to requirements. • Developers make requirements decisions without adequate information and perspective.
Solution	Identify the various types of users. Each user will differ in their frequency of using the product, the features they use, and their access privilege level.

[2] This material adapted, with permission from Karl E. Wiegers, Ph.D., 2001 "10 Requirements Traps to Avoid".

Ambiguous and Vague Requirements	
Symptoms	Ambiguity exists when: • A requirement statement has several different meanings and there is uncertainty as to which one is correct. • Multiple readers interpret a requirement in different ways. Each reader concludes that his or her interpretation is correct, and the ambiguity remains undetected until later—when it is more expensive to resolve. A symptom of vague requirements is when: • Developers have to ask the analyst or customers many questions. • Developers have to guess at what is really intended. The extent of this guessing game might not be recognized until the project is far along and implementation has diverged from what is really required. At this point, expensive rework may be needed to bring things back into alignment.
Solutions	1. Avoid using subjective and ambiguous words when requirements are being written. Terms like minimize, maximize, optimize, rapid, user-friendly, easy, simple, often, normal, usual, large, intuitive, robust, state-of-the-art, improved, efficient, and flexible are particularly dangerous. 2. Avoid the use of terms such as "and/or" and "etc." It is acceptable to include the term "TBD" (to be determined) to indicate current uncertainties, but make sure you resolve them prior to design and construction.
Requirements That Have Not Been Prioritized	
Symptoms	• Declaring all requirements to be equally critical deprives the project manager of a way to respond to new requirements and other considerations such as changes in staff, schedule, and quality goals. • Approximately 90 percent of your requirements are classified as high priority. • Stakeholders might interpret "high priority" differently, leading to mismatched expectations about what functionality will be included in the next release. • Development teams balk at prioritizing requirements because they believe the business and user area have a better understanding of business priorities. • Users are reluctant to prioritize because they fear the developers will automatically restrict the project to the highest priority items and the other items will never be implemented. • Uninformed people are left to make the prioritization decisions, unaware of the implications of those decisions.
Solutions	1. Rank all requirements according to the return benefit each provides to the business. 2. Factor in the dependencies between requirements. In some cases influences outside the underlying business, such as state, federal, and industry regulations, may affect a specific priority.

Building Functionality No One Uses	
Symptoms	• System features that user groups said they needed but are never used or used infrequently. • Glitzy user interface that must be present for the software to be useful. • “Gold plating” from the developers, which adds unnecessary functionality or features that "the users are just going to love." • Proposed functionality that is not clearly related to known user tasks or achieving business goals.
Solutions	1. Trace each Detailed Business Requirement back to its origin, such as a specific High-Level Business Requirement, business rule, industry standard, or government regulation. Requirements matrices are useful techniques for completing this task. 2. Identify the user groups that will benefit from each feature. 3. If the origin of a requirement is unclear, question whether it is really needed.
Analysis Paralysis	
Symptoms	• The development of requirements seems to drag on forever. • There is a prevailing viewpoint between all parties that development cannot begin until all Requirements are documented and approved. • All requirements must be modeled “six ways from Sunday,” the entire system must be prototyped, and development will be held up until all requirement changes cease.
Solutions	1. Identify the key decision-makers early in the project; they can resolve issues and help the project move ahead with development. 2. Make sure the scope of the effort is not too broad to address a single project. 3. Look for sets of requirements that are clear and cohesive and then drive them to closure. Then work through the remaining requirements for subsequent deliveries.

Scope Creep	
Symptoms	• New requirements are continually added during development. This symptom usually occurs when the product scope was never clearly defined. • New requirements are proposed, rejected, and then resurface later—with ongoing debates about whether they belong in the system—the scope definition is probably inadequate.
Solutions	1. Review the requirements research process to make sure no requirements or user types were overlooked. 2. The use of effective requirements gathering methods early on helps control scope creep. 3. Establish a meaningful process for standardizing your requirements specifications.
Inadequate Change Process	
Symptoms	• The project does not have a specific process for dealing with requirements changes, resulting in new functionality becoming evident only during system or beta testing. • It is unclear who makes decisions about proposed changes. • Change decisions are not communicated to all those affected, and the status of each change request is not known at all times.
Solutions	1. Define a practical change-control process for your project. 2. Set up a Change Control Board (CCB) to consider proposed changes at regular intervals and make binding decisions to accept or reject them.
Insufficient Change Impact Analysis	
Symptoms	• Developers or project managers agree to make suggested changes without carefully analyzing the implications. The change may be more complex than anticipated, take longer than promised, be technically or economically impossible, or conflict with other requirements. • Developers keep finding more affected system components as they implement the change.
Solutions	1. Analyze the impact of each proposed change. 2. Understand the implications of accepting a change on affected systems, identify all associated tasks, and estimate the effort and schedule impact. 3. Provide estimates of the costs and benefits of each change proposal to the Change Control Board before they make commitments.
Inadequate Version Control	
Symptoms	• Approved changes are not incorporated periodically into the Requirements document; project participants are unclear as to what is in the Requirements baseline. • Using a date to distinguish a document's different versions. Different versions may have been drafted on the same date.

Solutions	1. Incorporate approved changes into the Requirements document periodically and communicate the revised document to all stakeholders. 2. Adopt a versioning scheme for documents that clearly distinguishes drafts from baseline versions. 3. Store requirements documents in an automated version control tool. Restrict read-write access to a few authorized individuals, but make the current versions available in read-only format to all project stakeholders.
Inadequate Estimates for Gathering Requirements	
Symptoms	• The project has missed major milestones. • There have been cost overruns. • There have been incomplete or inadequate requirements developed.
Solutions	1. Carefully review all time allocations and budgetary planning. 2. Rework and review all requirements descriptions to improve all written specifications.

Appendix B: Business Requirement Templates

Refer to the company's templates for a Requirements Template

Project ID	Business Area	Business Owner	Contact Information

Functions: What business functions, such as billing or scheduling, are included in the project scope?

Function Name	Function Requirement	User Name	Type of User	Priority	Time frame
Name of Function	Description of each specific requirement	Name	See list	H, M, L,	When needed

For each function, describe its required details.

Function ID	Function Details
Name of Function	Description of Function Detail
	Frequency (of execution)?
	What information is needed?
	Who originates information?
	What is the output of the Function?
	Who needs access to output?