Appendix B

Business Requirements Document (BRD) Template

The rest of this page is left intentionally blank. The template follows on the next page.

Business Requirements Document (BRD)

Project No.   _____

Production Priority _____

Target date: _____

Approved by:

_____   _____

Name of user, department   Date

_____   _____

Name of user, department   Date

Prepared by: _____

Date: _____

Filename: _____

Version no.:   _____

Table of Contents

- Version Control
  - ▲ Revision History
  - ▲ RACI Chart
- Executive Summary
  - ▲ Overview
  - ▲ Background
  - ▲ Objectives
  - ▲ Requirements
  - ▲ Proposed Strategy
  - ▲ Next Steps
- Scope
  - ▲ Included in Scope
  - ▲ Excluded from Scope
  - ▲ Constraints
  - ▲ Impact of Proposed Changes
- Risk Analysis
  - ▲ Technological Risks
  - ▲ Skills Risks
  - ▲ Political Risks
  - ▲ Business Risks
  - ▲ Requirements Risks
  - ▲ Other
- Business Case
  - ▲ Cost/Benefit Analysis, ROI, etc.
- Timetable
- Business Use Cases
  - ▲ Business Use-Case Diagrams
  - ▲ Business Use-Case Descriptions (text and/or activity diagram)
- Actors
  - ▲ Workers
  - ▲ Business Actors
  - ▲ Other Systems
  - ▲ Role Map

- User Requirements
    - ▲ System Use-Case Diagrams
    - ▲ System Use-Case Descriptions
- State Machine Diagrams
- Nonfunctional Requirements
    - ▲ Performance Requirements
        - ◆ Stress Requirements
        - ◆ Response-Time Requirements
        - ◆ Throughput Requirements
    - ▲ Usability Requirements
    - ▲ Security Requirements
    - ▲ Volume and Storage Requirements
    - ▲ Configuration Requirements
    - ▲ Compatibility Requirements
    - ▲ Reliability Requirements
    - ▲ Backup/Recovery Requirements
    - ▲ Training Requirements
- Business Rules
- State Requirements
    - ▲ Testing State
    - ▲ Disabled State
- Static Model
    - ▲ Class Diagrams: Entity Classes
    - ▲ Entity Class Documentation
- Test Plan
- Implementation Plan
    - ▲ Training
    - ▲ Conversion
    - ▲ Scheduling of Jobs
    - ▲ Rollout
- End User Procedures
- Post Implementation Follow-Up
- Other Issues
- Sign-Off

Version Control

Revision History

Version #	Date	Authorization	Responsibility (Author)	Description

RACI Chart for This Document

RACI stands for Responsible, Accountable, Consulted, and Informed. These are the main codes that appear in a RACI chart, used here to describe the roles played by team members and stakeholders in the production of the BRD. The following table describes the full list of codes used in the table:

*	**Authorize**	Has ultimate signing authority for any changes to the document.
R	Responsible	Responsible for creating this document.
A	Accountable	Accountable for accuracy of this document (for example, the project manager).
S	Supports	Provides supporting services in the production of this document.
C	Consulted	Provides input (such as an interviewee).
I	Informed	Must be informed of any changes.

Name	Position	*	R	A	S	C	I

Executive Summary

(This is a one-page summary of the document, divided into the following subsections.)

Overview

(This one-paragraph introduction explains the nature of the project.)

Background

(This subsection provides details leading up to the project that explain why the project is being considered. Discuss the following where appropriate: marketplace drivers, business drivers, and technology drivers.)

Objectives

(This subsection details the business objectives addressed by the project.)

Requirements

(This is a *brief* summary of the requirements addressed in this document.)

Proposed Strategy

(This subsection recommends a strategy for proceeding based on alternatives.)

Next Steps

Action: (Describe the specific action to be taken.)

Responsibility

(State who is responsible for taking this action.)

Expected Date

(State when the action is expected to be taken.)

Scope

Included in Scope

(This is a brief description of business areas covered by the project.)

Excluded from Scope

(This subsection briefly describes business areas *not* covered by the project.)

Constraints

(These are predefined requirements and conditions.)

Impact of Proposed Changes

Business Use Case	New?	Desired Functionality	Current Functionality (If a Change)	Stakeholders/ Systems	Priority

Risk Analysis

(A risk is something that could impact the success or failure of a project. Good project management involves a constant reassessment of risk.)

For each risk, document:

- Likelihood
- Cost
- Strategy: Strategies include:
  - *Avoid*: Do something to eliminate the risk.
  - *Mitigate*: Do something to reduce damage if risk materializes.
  - *Transfer*: Pass the risk up or out to another entity.
  - *Accept*: Do nothing about the risk. Accept the consequences.

Technological Risks

(This subsection specifies new technology issues that could affect the project.)

Skills Risks

(This subsection specifies the risk of not getting staff with the required expertise for the project.)

Political Risks

(This subsection identifies political forces that could derail or affect the project.)

Business Risks

(This subsection describes the business implications if the project is canceled.)

Requirements Risks

(This subsection describes the risk that you have not correctly described the requirements. List areas whose requirements were most likely to be incorrectly captured.)

Other Risks

Business Case

(Describe the business rationale for this project. This section may contain estimates on cost/benefit, return on investment (ROI), payback [length of time for the project to pay for itself], market share benefits, and so on. Quantify each cost or benefit so that business objectives may be measured after implementation.)

Timetable

Business Use Cases

(Complete this section if the project involves changes to the workflow of end-to-end business processes. Document each end-to-end business process affected by the project as a business use case. If necessary, describe existing workflow for the business use case as well as the new, proposed workflow.)

Business Use-Case Diagrams

(Business use-case diagrams describe stakeholder involvement in each business use case.)

Business Use-Case Descriptions

(Describe each business use case with text and/or an activity diagram. If you are documenting with text, use an informal style or the use-case template described in the "User Requirements" section below.)

Actors

Workers

(List and describe stakeholders who act within the business in carrying out business use cases.)

Department/ Position	General Impact of Project

Business Actors

(List and describe external parties, such as customers and partners, who interact with the business.)

Actor	General Impact of Project

Other Systems

(List computer systems potentially impacted by this project. Include any system that will be linked to the proposed system.)

System	General Impact of Project

Role Map

(The role map describes the roles played by actors [users and external systems] that interact with the IT system.)

User Requirements

(Describe requirements for automated processes from a user perspective.)

System Use-Case Diagrams

(System use-case diagrams describe which users use which features and the dependencies between use cases.)

System Use-Case Descriptions

(During Initiation, only short descriptions of the use cases are provided. During Analysis, the following template is filled out for each medium to high-risk use case. Low-risk use cases may be described informally. This template may also be used to document the business use cases included earlier in the BRD.)

Use-Case Description Template

1. *Use case*: (The use-case name as it appears on system use-case diagrams.)
   Perspective: Business use case/system use case
   Type: Base use case/extension/generalized/specialized

   1.1 Brief Description
        (Briefly describe the use case in approximately one paragraph.)

   1.2 Business Goals and Benefits
        (Briefly describe the business rationale for the use case.)

   1.3 Actors

        1.3.1 Primary Actors
               (Identify the users or systems that initiate the use case.)

        1.3.2 Secondary Actors
               (List the users or systems that receive messages from the use case.
               Include users who receive reports or on-line messages.)

        1.3.3 Off-Stage Stakeholders
               (Identify non-participating stakeholders who have interests in this
               use case.)

   1.4 Rules of Precedence

        1.4.1 Triggers
               (Describe the event or condition that "kick-starts" the use case:
               such as *User calls Call Center*; *Inventory low*. If the trigger is time-
               driven, describe the temporal condition, such as *end-of-month*.)

        1.4.2 Preconditions
               (List conditions that must be true before the use case begins. If a
               condition *forces* the use case to occur whenever it becomes true, do
               not list it here; list it as a trigger.)

   1.5 Postconditions

        1.5.1 Postconditions on Success
               (Describe the status of the system after the use case ends success-
               fully. Any condition listed here is guaranteed to be true on success-
               ful completion.)

        1.5.2 Postconditions on Failure
               (Describe the status of the system after the use case ends in failure.
               Any condition listed here is guaranteed to be true when the use case
               fails as described in the exception flows.)

   1.6 Extension Points
        (Name and describe points at which extension use cases may extend this use
        case.)
        Example of extension point declaration:

        1.6.1 Preferred Customer: 2.5-2.9

   1.7 Priority

   1.8 Status

Your status report might resemble the following example:
Use-case brief complete: 2005/06/01
Basic flow + risky alternatives complete: 2005/06/15
All flows complete: 2005/07/15
Coded: 2005/07/20
Tested: 2005/08/10
Internally released: 2005/09/15
Deployed: 2005/09/30

   1.9 Expected Implementation Date

  1.10 Actual Implementation Date

  1.11 Context Diagram

(Include a system use-case diagram showing this use case, all its relationships [includes, extends, and generalizes] with other use cases and its associations with actors.)

2. Flow of Events

Basic Flow

   2.1 (Insert basic flow steps.)

Alternate Flows

  2.Xa (Insert the alternate flow name.)

(The alternate flow name should describe the condition that triggers the alternate flow. "2.X" is step number in basic flow where interruption occurs. Describe the steps in paragraph or point form.)

Exception Flows

  2.Xa (Insert the exception flow name.)

(The flow name should describe the condition that triggers the exception flow. An exception flow is one that causes the use case to end in failure and for which "postconditions on failure" apply. "2.X" is step number in basic flow where interruption occurs. Describe the steps in paragraph or point form.)

3. Special Requirements

(List any special requirements or constraints that apply specifically to this use case.)

   3.1 Nonfunctional Requirements

(List requirements not visible to the user during the use case—security, performance, reliability, and so on.)

   3.2 Constraints

(List technological, architectural, and other constraints on the use case.)

4. Activity Diagram
    (If it is helpful, include an activity diagram showing workflow for this system use case, or for select parts of the use case.)

5. User Interface
    (Initially, include description/storyboard/prototype only to help the reader visualize the interface, not to constrain the design. Later, provide links to screen design artifacts.)

6. Class Diagram
    (Include a class diagram depicting business classes, relationships, and multiplicities of all objects participating in this use case.)

7. Assumptions
    (List any assumptions you made when writing the use case. Verify all assumptions with stakeholders before sign-off.)

8. Information Items
    (Include a link or reference to documentation describing rules for data items that relate to this use case. Documentation of this sort is often found in a data dictionary. The purpose of this section and the following sections is to keep the details out of the use case proper, so that you do not need to amend it every time you change a rule.)

9. Prompts and Messages
    (Any prompts and messages that appear in the use case proper should be identified by name only, as in *Invalid Card Message*. The *Prompts and Messages* section should contain the actual text of the messages or direct the reader to the documentation that contains text.)

10. Business Rules
    (The "Business Rules" section of the use-case documentation should provide links or references to the specific business rules that are active during the use case. An example of a business rule for an airline package is "Airplane weight must never exceed the maximum allowed for its aircraft type." Organizations often keep such rules in an automated business rules engine or manually in a binder.)

11. External Interfaces
    (List interfaces to external systems.)

12. Related Artifacts
    (The purpose of this section is to provide a point of reference for other details that relate to this use case, but would distract from the overall flow. Include references to artifacts such as decision tables, complex algorithms, and so on.)

State Machine Diagrams

(Insert state machine diagrams describing the events that trigger changes of state of significant business objects.)

Nonfunctional Requirements

(Describe across-the-board requirements not covered in the use-case documentation. Details follow.)

Performance Requirements

(Describe requirements relating to the system's speed.)

Stress Requirements

(This subsection of performance requirements describes the degree of simultaneous activity that the system must be able to support. For example, "The system must be able to support 2,000 users accessing financial records simultaneously.)

Response-Time Requirements

(This subsection of performance requirements describes the maximum allowable wait time from the moment the user submits a request until the system comes back with a response.)

Throughput Requirements

(This subsection of performance requirements describes the number of transactions per unit of time that the system must be able to process.)

Usability Requirements

(Describe quantitatively the level of usability required. For example, "A novice operator, given two hours of training, must be able to complete the following functions without assistance…." Also, refer to any usability standards and guidelines that must be adhered to.)

Security Requirements

(Describe security requirements relating to virus protection, firewalls, the functions and data accessible by each user group, and so on.)

Volume and Storage Requirements

(Describe the maximum volume [for example, the number of accounts] that the system must be able to support, as well as random access memory [RAM] and disk restrictions.)

Configuration Requirements

(Describe the hardware and operating systems that must be supported.)

Compatibility Requirements

(Describe compatibility requirements with respect to the existing system and external systems with which the system under design must interact.)

Reliability Requirements

(Describe the level of fault-tolerance required by the system.)

Backup/Recovery Requirements

(Describe the backup and recovery facilities required.)

Training Requirements

(Describe the level of training required and clearly state which organizations will be required to develop and deliver training programs.)

Business Rules

(List business rules that must be complied with throughout the system. For example, an inventory system might have a rule that whenever inventory falls below a trigger level that an automatic order is placed with the supplier. If an external rules engine is being used, this section should refer the reader to the location of these rules.)

State Requirements

(Describe how the system's behavior changes when in different states. Describe the features that will be available and those that will be disabled in each state.)

Testing State

(Describe what the user may and may not do while the system is in the test state.)

Disabled State

(Describe what is to happen as the system goes down. Clearly define what the user will and will not be able to do.)

Static Model

(During Initiation, only strategic classes are modeled.)

Class Diagrams: Entity Classes

(Insert class diagrams representing classes of business objects and relationships among the classes. This section centralizes rules that govern business objects, such as the numerical relationships among objects, the operations associated with each object, and so on.)

Entity Class Documentation

(Insert documentation to support each of the classes that appear in the class diagrams. Not every class needs to be fully documented. First do a risk analysis to determine where full documentation would most benefit the project.)

Class Name

Alias: (List any other names by which the class is known within the business domain.)

Description:

Example: (Provide an example of an object of this class.)

Attributes: (These may be documented in a table, as follows.)

Attribute	Derived?	Derivation	Type	Format	Length	Range	Dependency

(When your requirements are complete up to this point and approved by the appropriate people, submit them to developers. You can then work on the test plan, implementation plan, and end user procedures.)

Test Plan[1]

(To standardize the testing, you should develop a test plan document for analysts to follow when constructing projects test plans. Although every project is different, the following may be used as a guideline. Each project should consider the following stages during testing):

1. Submit the requirements to the technical team. The technical team completes development. Concurrently, the BA builds numbered test scenarios for requirements-based testing. Consider using decision tables to identify scenarios and boundary value analysis to select test data. The technical team conducts white-box testing, to verify whether programs, fields, and calculations function as specified. The BA or technical team specifies the required quality level for white-box testing, such as multiple-condition coverage.

2. Perform requirements-based testing. The BA or dedicated QA (Quality Assurance) staff administers or supervises tests to prove or disprove compliance with requirements. Ensure that all formulae are calculated properly. Describe principles and techniques to be used in black-box testing, such as structured testing guidelines and boundary value analysis.

3. Conduct system testing. Ensure that the integrity of the system and data remain intact. For example:
    - *Regression test*: Retest all features (using a regression test bed).
    - *Stress test*: Test multiple users at the same time.
    - *Integration tests*: Make sure that the changes do not negatively affect the overall workflow across IT and manual systems.
    - *Volume test*: Test the system with high volume.

4. Perform user acceptance testing. Involve the end users at this stage. Choose key users to review the changes in the test environment. Use the testing software as a final check.

Implementation Plan

Training

(Specify who is responsible for training.)

(Specify who is to be trained.)

(Specify how training will be done.)

---

[1]These requirements are often described in a separate test plan. If they are not addressed elsewhere, describe them here in the BRD.

Conversion

(Specify existing data that must be converted. Promote programs to new release. Grant privileges to the users.)

Scheduling of Jobs

(Advise Information Systems [IS] operations which jobs to add to the production run. Specify the frequency of the run: daily, weekly, monthly, quarterly, semi-annually, or annually. Ensure that the job is placed in the correct sequence. Advise IS operations of the reports to be printed and the distribution list for reports and files.)

Rollout

(Advise all affected users when the project is promoted.)

End User Procedures

(Write up the procedures for the affected departments. Distribute this document to them in addition to providing any hands-on training.)

Post Implementation Follow-Up

(Follow up within a reasonable time frame after implementation to ensure that the project is running successfully. Determine whether any further enhancements or changes are needed to ensure success of the project.)

Other Issues

Sign-Off