

SYSTEM REQUIREMENTS ANALYSIS

Purpose

The purpose of **System Requirements Analysis** is to obtain a thorough and detailed understanding of the business need as defined in Project Origination and captured in the Business Case, and to break it down into discrete requirements, which are then clearly defined, reviewed and agreed upon with the Customer Decision-Makers. During System Requirements Analysis, the framework for the application is developed, providing the foundation for all future design and development efforts.

System Requirements Analysis can be a challenging phase, because all of the major Customers and their interests are brought into the process of determining requirements. The quality of the final product is highly dependent on the effectiveness of the requirements identification process. Since the requirements form the basis for all future work on the project, from design and development to testing and documentation, it is of the utmost importance that the Project Team create a complete and accurate representation of all requirements that the system must accommodate. Accurately identified requirements result from effective communication and collaboration among all members of the Project Team, and provide the best chance of creating a system that fully satisfies the needs of the Customers.

The primary goal of this phase is to create a detailed Functional Specification defining the full set of system capabilities to be implemented, along with accompanying data and process models illustrating the information to be managed and the processes to be supported by the new system. The Functional Specification will evolve throughout this phase of the SDLC as detailed business requirements are captured, and as supporting process and data models are created, ensuring that the eventual solution provides the Customers with the functionality they need to meet their stated business objectives.

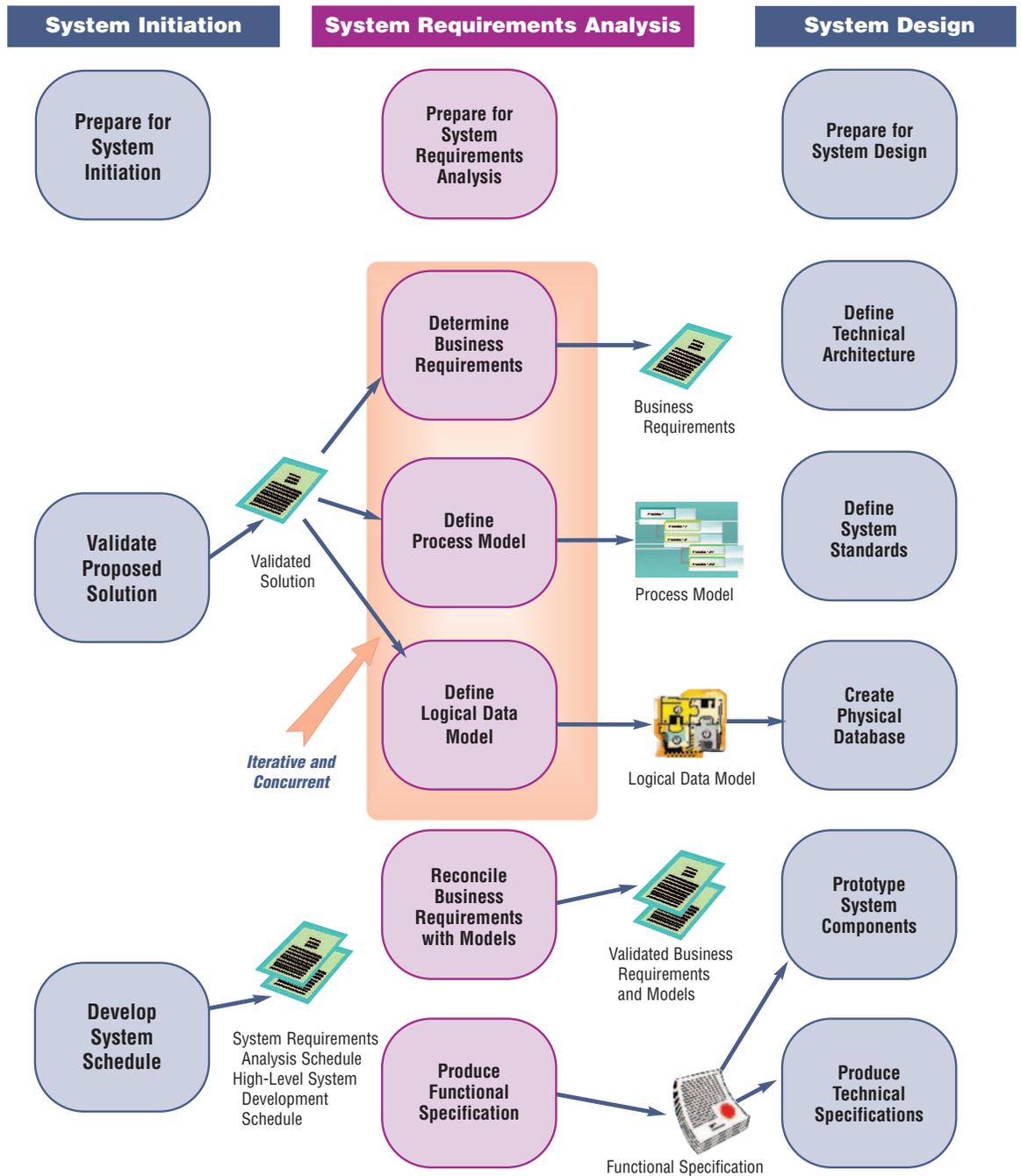
List of Processes

This phase consists of the following processes:

- ◆ **Prepare for System Requirements Analysis**, where steps are taken to ensure that the project environment and Project Team members are adequately prepared to both capture and analyze the system requirements;
- ◆ **Determine Business Requirements**, where in-scope and out-of-scope business requirements are identified, business rules are defined and documented, and interfaces to and from the new application are discussed;
- ◆ **Define Process Model**, where a pictorial top-down representation of the major business processes that interact with the system is diagrammed and decomposed into manageable functions and sub-functions until no further breakdown is feasible;
- ◆ **Define Logical Data Model**, where data that supports the processes and business rules is logically modeled, identifying entities and their relationships to other entities, and defining attributes with their business definitions;
- ◆ **Reconcile Business Requirements With Models**, where the Project Team ensures that the Process and Logical Data Models accommodate all requirements and business rules;
- ◆ **Produce Functional Specification**, where interfaces, processes and data are merged to describe systematically how the Consumer will use the application, and how data will be retrieved, processed and stored.

The following chart illustrates all of the processes and deliverables of this phase in the context of the system development lifecycle.

Figure 2-1



List of Roles

The following roles are involved in carrying out the processes of this phase. Detailed descriptions of these roles can be found in the Introductions to Sections I and III.

- ◆ Project Manager
- ◆ Project Sponsor
- ◆ Facilitator
- ◆ Business Analyst
- ◆ Database Administrator
- ◆ Data/Process Modeler
- ◆ Technical Lead/Architect
- ◆ Software Quality Assurance (SQA) Analyst
- ◆ Technical Services (HW/SW, LAN/WAN, TelCom)
- ◆ Information Security Officer (ISO)
- ◆ Technical Support (Help Desk, Documentation, Trainers)
- ◆ Customer Decision-Maker
- ◆ Customer Representative
- ◆ Consumer
- ◆ Performing Organization
- ◆ Stakeholders

List of Deliverables

The following table lists all System Requirements Analysis processes, some techniques available for use in executing these processes, and process outcomes and deliverables.

Figure 2-2

| Processes | Techniques | Process Deliverables (Outcomes) |
|---|---|--|
| Prepare for System Requirements Analysis | Team Skills Assessment Site Walk-throughs Technology Needs Assessment Tool Needs Assessment | <i>Established Team and Environment for Requirements Analysis</i> |
| Determine Business Requirements | Interviews JAD Sessions Brainstorming Storyboarding Critical Success Factor Interviewing Context Diagramming Use Case Diagramming Prototyping Walk-throughs Potential Problem Analysis Expressing Logic: Pseudo Code, Structured English, Object Oriented Logic | Business Requirements |
| Define Process Model | Work Flow Diagramming Flow Chart Diagramming Process Modeling Customer Event Diagramming Use Case Diagramming Decision Trees Prototyping | Process Model |
| Define Logical Data Model | Entity Relationship Diagramming Data Normalization/ De-Normalization | Logical Data Model |
| Reconcile Business Requirements With Models | CRUD Matrices Gap Analysis | <i>Analysis Assessment</i> Validated Business Requirements and Models |
| Produce Functional Specification | Process Association and Grouping Logical Organization Work Flow Clustering Expressing Logic: Pseudo Code, Structured English, Object Oriented Logic | Functional Specification |

2.1 PREPARE FOR SYSTEM REQUIREMENTS ANALYSIS

Purpose

The purpose of **Prepare for System Requirements Analysis** is to position the Project Team and their working environment to ensure successful completion of System Requirements Analysis. This is the point at which the Project Team prepares to capture the detailed functional, technical, operational, and transitional requirements of the system.

Description

In preparing for this phase, the Project Manager must focus on the Project Team and the environment in which the team will work.

With each new project phase comes the need for new skills, experience, and, potentially, new Project Team members. The team needed during this phase must possess analytical skills that allow them to continually “peel the onion”, driving to continually deeper levels of requirements definition. Experience in effective interviewing, facilitation, various modeling techniques, requirements gathering, and gap analysis will be extremely beneficial.

Roles

- Project Manager
- Project Sponsor
- Business Analyst
- Facilitator
- Data/Process Modeler
- Technical Lead/Architect
- Customer Decision-Maker
- Customer Representative

In reviewing the Validated Solution all team members must share a clear and common understanding of the scope of this phase of the project, the Project Schedule, the deliverables to be produced, and their individual responsibilities relating to the creation of these deliverables.

Regardless of the size of the development effort being undertaken, System Requirements Analysis may place the greatest demand upon Customers in terms of resources and the extent of their required participation. During the preparation for this phase, the Project Manager should continue to manage the Customer’s expectations surrounding this participation. Less involvement typically leads to a less acceptable finished product. In addition, many

individuals earmarked to participate in the requirements gathering sessions may not have been privy to earlier project scope-setting sessions. This can lead to the possible perception of these upcoming sessions as opportunities to identify or request functionality and features that are beyond the original intent of the project. Since management of scope creep is an essential role of the Project Manager, this may be an appropriate time to review the established change management processes with the Customer.

At the start of the System Requirements Analysis phase, it is the Project Manager's responsibility to ensure that the environment in which the Project Team will work is properly established. Beyond the obvious need to ensure that team members have adequate equipment to perform their duties, there are additional elements of the environment that should not be overlooked. The project repository, a secure area for maintaining work products and deliverables that was established during Project Initiation, continues to evolve over subsequent phases of the project. Although the establishment of the repository itself is important, it is equally necessary to define the mechanisms and processes to be followed for creating and maintaining all System Requirements Analysis related materials.

2.2 DETERMINE BUSINESS REQUIREMENTS

Purpose

In **Determine Business Requirements**, information is gathered from a variety of project participants relating to the vision and scope of the system. From this, specific detailed requirements are identified and documented so that they address the stated business need. These requirements are then decomposed into a set of business rules.

Description

While this process specifically addresses the capturing of Business Requirements for the new system, the reality is that it may be necessary, and is often beneficial, for the Project Team to determine these requirements while simultaneously defining the supporting process and data models. By conducting these three processes (Determine Business Requirements, Define Process Model, and Define Logical Data Model) concurrently, as opposed to sequentially, the team can develop the process and data models as information and requirements are defined, and can update these models as a result of gathering new or changed information.

Roles

- Project Manager
- Project Sponsor
- Business Analyst
- Facilitator
- Data/Process Modeler
- Technical Lead/Architect
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Stakeholders

Because the three processes are performed not only concurrently, but also often iteratively, it is important for the Project Team to tightly manage the documentation for each process so that requirements are not lost, misunderstood or overlooked. The Project Manager may need to utilize techniques and/or tools to help document requirements and ensure that they are not missed. The Glossary contains a brief description of some of the techniques available to the Project Manager; examples include storyboarding, interviews, joint application design sessions (JAD), Unified

Modeling Language (UML), prototyping, data flow diagramming, process modeling, and entity-relationship diagramming.



Don't hesitate to use one of the commercially available tools to assist the Project Team in their documentation efforts. Through use of these tools, changes made to a process are automatically carried through to all other associated processes or business rules.

Determining Business Requirements requires eliciting, analyzing, specifying, prioritizing, verifying and negotiating business functions that the system must deliver and support. The results are captured in a Business Requirements deliverable (use Figure 2-5, Business Requirements template, as a guide). During this process it is important to have all of the Stakeholders involved. Since this is the process in which all business and processing requirements are determined and agreed to, it is critical that all parties understand the ramifications of including or excluding requirements from scope. This is an opportunity to work out business process issues as a group, in order to reach optimal performance and efficiency within an organization or even across organizations or functional areas. Decisions made will impact remaining phases, so all parties involved in the project lifecycle should be heard, and all areas of concern or question should be thoroughly addressed. Reaching consensus and agreement on the final deliverable from this phase will help to ensure that everyone gets the product to which they agreed.

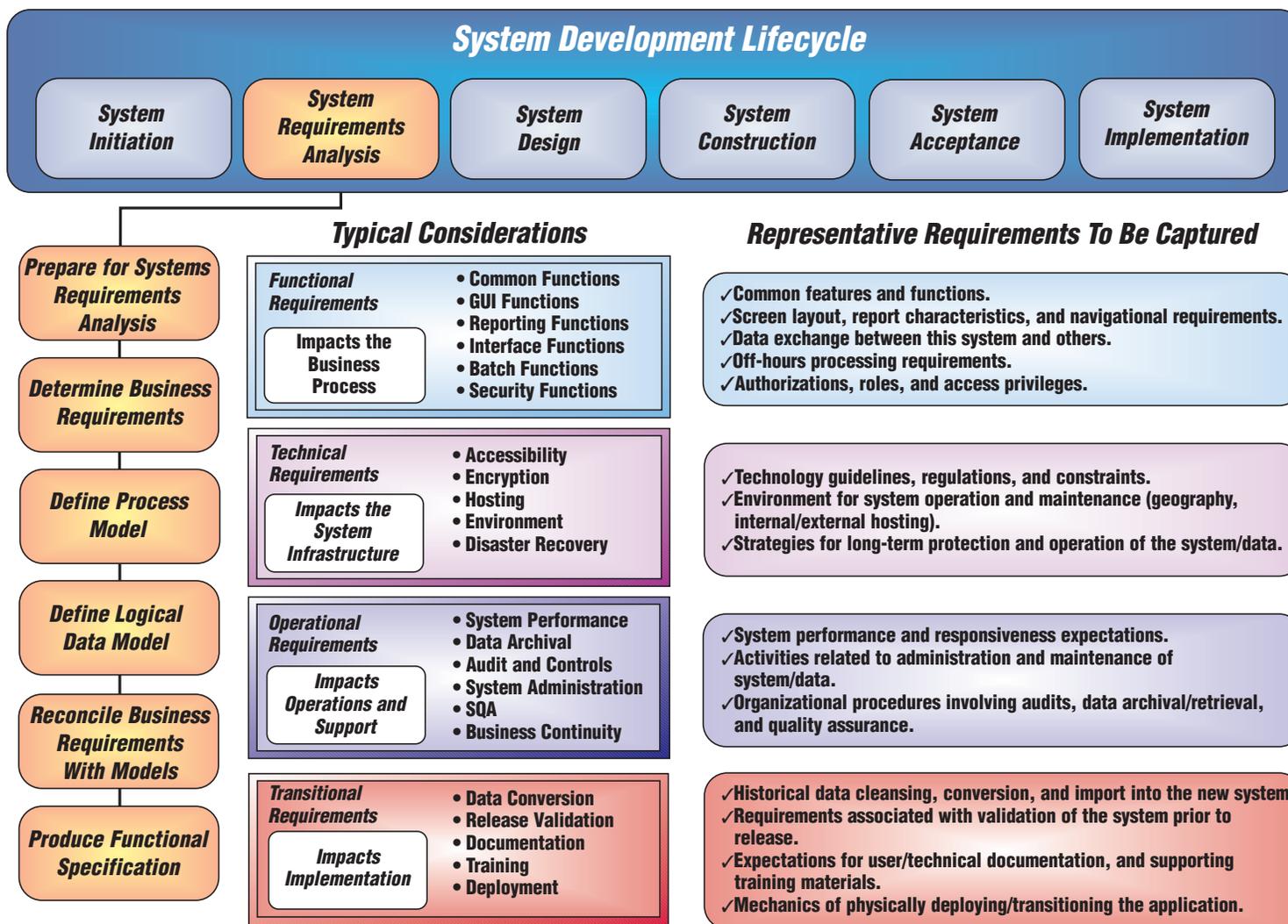
As stated in the SDLC Overview, requirements fall into multiple categories that, while related, are themselves separate and distinct. These categories are:

Figure 2-3

| Category | Description |
|----------------------------------|---|
| Functional Requirements | Requirements that define those features of the system that will specifically satisfy a Consumer need, or with which the Consumer will directly interact. |
| Technical Requirements | Requirements that identify the technical constraints or define conditions under which the system must perform. |
| Operational Requirements | Requirements that define those “behind the scenes” functions that are needed to keep the system operational over time. |
| Transitional Requirements | Requirements that define those aspects of the system that must be addressed in order for the system to be successfully implemented in the production environment, and to relegate support responsibilities to the Performing Organization. |

When capturing Business Requirements, the Project Manager must ensure that the Project Team addresses all of the categories above. Figure 2-4 illustrates the types of considerations and requirements that the Project Team must capture specific to System Requirements Analysis.

Figure 2-4 System Requirements Analysis Considerations



One approach to eliciting requirements from the Customer is to hold one or more JAD sessions. For these sessions, assembling individuals from both the program areas and IT into cross-functional groups can help clarify how proposed changes to a business process may impact operations. The benefit of having a session with multiple representatives from the program areas is that the pros and cons of business process changes are heard and discussed by all involved.

Requirements gathering, when properly facilitated, establishes a forum for everyone to be heard, for issues to be worked through, and for resolutions to be defined that meet the needs of all parties. Through this forum, multiple opinions may enhance the team's understanding of how certain processes are currently being performed, better defining how they should be structured within the context of the new application. This approach may also result in negotiations of functionality. There may need to be some trade-offs, and as a result processes may be reexamined and redefined. As the sessions progress, the Project Team must constantly assess and analyze the requirements.



A common mistake when coordinating the logistics of interviews or group requirements definition sessions is to hold these meetings where the majority of the participants represent only the Customer and Consumer populations. Doing so may set the stage for the Project Team to form a type of tunnel vision in which business requirements that focus primarily, or even exclusively, on the functional aspects of the system are captured. Just by the nature of their day-to-day responsibilities, Customers will often approach these requirements definition sessions from the perspective of, "What do I need this system to do in order for me to perform my duties?" Many operational, technical, and transitional requirements do not fall within the answers to that question. It is up to the Project Team member running these sessions (typically a Business Analyst or Facilitator) to make certain that these aspects of the system are also discussed, and that the individuals best positioned to provide this information are represented at the requirements definition meetings.

It is not unrealistic to assume that there may come a point at which negotiation or consensus building activities will break down, and that resolution of an issue may require insight or information not available to the Project Team. Ultimately, there must be a single decision making body responsible for resolving such issues. A key role of the Project Sponsor or Customer Decision-Maker is to make the final determination regarding these issues, and to communicate the decision to the Project Team. The Project Sponsor may or may not choose to share the

rationale for such decisions with the entire team, nor is it guaranteed that the team will agree with determinations that are made. The Project Manager should encourage the team to support the decisions and move forward.

The following steps should assist the team in building a useful and comprehensive Business Requirements document:

Absorb the requirements. Before the requirements can be analyzed, they must first be collected. Team members need to be sponges and take everything in, no matter how unimportant or inconsequential it may seem. There are many approaches to gathering these requirements, but all start with effective listening. Regardless of the technique used, the Project Team must remember that the goal of this process is to understand what the Customers need – not what the team members think they need.

Interpret the requirements. Now that the requirements have been captured, the Project Team must think about them. What have they heard? What was missed? Look for unanswered questions or contradictions. The requirements must be stated as clearly and concisely as possible, avoiding combining requirements and eliminating subjective wording. If the system must do A and B and C and D, there should be four distinct and verifiable requirements that can each be approved or rejected on its own merits. Avoid ambiguities and opinions. If the requirement is that some process should be “easy”, the team should go back and find out exactly what that means. What would make something about that process difficult or more complex than it should be?

Bind the requirements. While defining what the business requirements are, it is also necessary to determine what they are not! This is done to establish consensus on the Project Scope and to clarify any scope issues. At least some requirements that were captured will be labeled “out of scope.”

Categorize the requirements. Even for a relatively small system, you are likely to end up with scores of requirements. To understand how they relate to each other, and to effectively deal with them later on in the process, it is necessary to separate them into categories, logically grouping the requirements according to related business functions or organizational boundaries.

Prioritize the requirements. Regardless of how accurately the business requirements reflect the business need, it may not be feasible to implement them all at once (or even at all). In finalizing the requirements to be implemented, it will be necessary to prioritize them according to their criticality to the business. This classification into core, essential, and desirable will very likely involve both the Project Sponsor and Customer Decision-Makers.

The following guidelines may be used: “Core” requirements are the ones without which the system may as well not be developed at all; it will be of no use to most Customers without these. “Essential” requirements are those for which a short-term work-around could be developed (or for which an old process can hobble along for a little while longer) but over the long run, they have to be there. “Desirable” requirements are the “bells and whistles” that may be precious to certain constituencies, but without which the system will function just fine.

To put it another way, the system must go into production with all Core and a good portion of Essential requirements represented, and with a plan to implement the remaining Essential requirements in the subsequent phase.

Deliverable

- ◆ **Business Requirements** – A document containing detailed requirements for the system being developed. These requirements define the functional, technical, operational, and transitional capabilities, restrictions, and features that must be provided by the new system.

Figure 2-5 Business Requirements Document

< Name of Agency >
Business Requirements Document
< System Name >

| | |
|-----------------|--|
| Agency | |
| Project Name | |
| Project Sponsor | |
| Project Manager | |
| Document Date | |
| Prepared By | |

*Enter the name of the **Agency** for which the system is being developed.
Enter the **Project Name**, and the names of the **Project Manager** and the **Project Sponsor**.
Enter the **Date** as of which this document is current.
Enter the names of the Project Team members by whom the document was **Prepared**.*

Figure 2-5 (Continued)

| |
|---|
| <p style="text-align: center;">Business Requirements Document</p> <p style="text-align: center;">TABLE OF CONTENTS</p> <p><i>The Table of Contents should be at least two levels deep.</i></p> <p>1.0 DOCUMENT SCOPE</p> <p><i>Document Scope describes the goal of this document in a narrative.</i></p> <p><i>Example:</i></p> <p><i>To define a consistent set of business requirements for the <XYZ> system and to identify what is in and out of scope.</i></p> <p><i>The narrative should also provide an overview of the efforts conducted to gather business requirements.</i></p> <p><i>Example:</i></p> <p><i>This document summarizes the requirements gathered in a series of <X> Joint Application Design (JAD) sessions that were conducted by <members of the Project Team> with <DEF> and <GHI> business units between <date> and <date>.</i></p> <p>2.0 GENERAL REQUIREMENTS</p> <p><i>The General Requirements section lists high-level business requirements that apply to the whole system.</i></p> <p><i>Example:</i></p> <p><i>This system will provide a central repository for all <XYZ> data.</i></p> <p><i>It should also include those requirements that are common to all Customer groups.</i></p> <p><i>Example:</i></p> <p><i>This system will provide ad hoc reporting capabilities to each Consumer business unit.</i></p> <p>NOTE: <i>By default, all requirements listed in this section are deemed to be Core to the system. Those general requirements that do not meet these criteria should be listed below under "4.0, Business Requirements Not Being Implemented".</i></p> |
|---|

Figure 2-5 (Continued)

Business Requirements Document

3.0 SPECIFIC REQUIREMENTS

The **Specific Requirements** section lists business requirements specific to each Customer group. A concise and specific listing of **Business Requirements by Business Function** should be provided. Requirements should be categorized, bulleted, detailed, and prioritized. These requirements should encompass the multi-dimensional aspects of the system (i.e., the functional, technical, operational, and transitional requirements).

3.1 Business Unit

Description

The **Description** identifies the purpose and main functions of the **Business Unit**.

3.1.1 Business Function 1

Description

In addition to **Business Function Description**, the narrative focuses on the desired business processes that will be in place when the new system is implemented as opposed to the current state of the business function, which can be documented, if necessary, in the Appendix.

- Business Requirement 1 (Priority)
- Business Requirement 2 (Priority)
- Etc.

3.1.2 Business Function 2

Description

- Business Requirement 1 (Priority)
- Business Requirement 2 (Priority)
- Etc.

This format is repeated for all Customer and Consumer groups and their respective functions that require interaction with the system.

Example:

<XYZ> Unit

Description:
This unit is responsible for developing the <ABC> deliverable, maintaining the <DEF> function and executing the <GHI> process.

Business Function:
Developing the <ABC> Deliverable

Description:
The <XYZ> unit personnel will utilize the system reports to compile and produce the <ABC> deliverable (see attached).

Business Requirements (Priority)

1. The System will provide a report detailing <JKL> expenditures that will be used for Page 2 of the <ABC> Deliverable. (Essential)
2. Etc.

Figure 2-5 (Continued)

Business Requirements Document

4.0 BUSINESS REQUIREMENTS NOT BEING IMPLEMENTED

This section specifies requirements that will NOT be part of the new system.

Example:

Director of Contracts and Director of Claims have determined that the following functions are outside the scope of this system:

- 1. <ABC> Process*
- 2. <DEF> Deliverable*
- 3. Etc.*

APPENDIX A – Requirements Definition Supporting Details

All published work products (meeting notes, session results, etc.) of individual interviews and group facilitated sessions held to determine business requirements should be included in this section.

A table detailing dates, times, topics and participants of all interviews and sessions should precede the compilation.

2.3 DEFINE PROCESS MODEL

Purpose

The purpose of the **Define Process Model** process is to create a pictorial representation of the functions and operations (i.e., the processes) that will eventually be performed by the system being developed.

Description

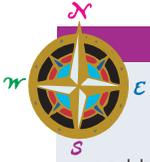
The second of the three concurrent processes within System Requirements Analysis, Define the Process Model may begin at any time after the Project Team has started collecting specific

business requirements. The resulting Process Model of the system, also often referred to as the “To Be” model, illustrates the system processes as they are envisioned for the new system. Over time, this pictorial top-down representation of the major business processes will be decomposed into manageable functions and sub-functions until no further breakdown is possible. When combined with the detailed set of Business Requirements and the supporting Logical Data Model, this Process Model should completely address not only the full list of business needs to be satisfied by the new system, but also the vision for how the new system will provide and support this functionality.

Roles

- Project Manager
- Business Analyst
- Facilitator
- Data/Process Modeler
- Technical Lead/Architect
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Stakeholders

During the Determine Business Requirements process, a picture of the *current* business processes and practices will begin to evolve. This can be a useful tool in confirming that all current processes have been identified, and can be used by the Project Team as a means of ensuring that their Process Model has not neglected any existing functionality. There is a risk, however, that too much focus on current business processes may cause Customers to take a myopic view of their true business needs, ultimately defining a system that provides little value over the system that is already in place.



A key to successful process modeling is to find a way to get your Customers to look beyond “what they do and how they do it”, and to instead describe “what they need and how it could best be accomplished” if they were not forced to perform their duties within the constraints or limitations of existing systems and processes. One of the most common mistakes made during this phase of the development lifecycle is to automate a bad process, simply because that’s the way the business has always operated.

Remembering that much of System Requirements Analysis is iterative, the Project Manager must ensure that as requirements are updated as a result of continued efforts to Determine Business Requirements, the Project Team also refines the Process Model to accommodate those changes.

The Project Manager must ensure that the Stakeholders and Customers periodically validate the Process Model as it is being developed. It is important that they understand that the Process Model is a representation of the proposed business solution, an attempt to meet everyone’s needs. As part of validating the final Process Model deliverable with the Customer, it may be beneficial to conduct walk-throughs to map the defined business requirements to the diagrammed Process Model. A walk-through helps to identify any requirements missed by both the Project Team and the Customer, and helps to further validate that the requirements and processes are accurately decomposed.



Before the Customer accepts the final deliverable for this process, ensure that he or she understands the ramifications of acceptance. For instance, if a process critical to the application was overlooked in a JAD session, and therefore not modeled, and the deliverable has been approved, change control may be necessary. Ensuring that the processes have been identified and decomposed will make it more likely that the data model built to support the process is adequate. If a process has been overlooked, there will most likely be an impact to the design of the data model, and therefore to the database itself. This could definitely warrant change control downstream. This applies to the Business Requirements and Logical Data Model deliverables as well as the Process Model. You may need to validate these three deliverables incrementally during this phase, and provide the Customer with a final walk-through at the conclusion of this process, before proceeding to the development of the Functional Specification.

Deliverable

- ◆ **Process Model** – A graphical representation of the decomposition of all business processes that interact with the system.

2.4 DEFINE LOGICAL DATA MODEL

Purpose

The purpose of **Define Logical Data Model** is to identify all uniquely distinguishable objects either used or produced by the system (the data entities), to capture all of the characteristics that help define those objects (the data attributes), and to describe the relationships between the entities.

Description

Like process modeling, definition of the data model can start as soon as the interviews or JAD sessions begin. A Data Modeler is most often responsible for designing the logical representation of the data to support the business need. Typically, this model will evolve throughout the iterations of capturing and documenting the business requirements.

Roles

- Project Manager
- Business Analyst
- Facilitator
- Database Administrator
- Data/Process Modeler
- Technical Lead/Architect
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Stakeholders

The Data Modeler may begin to work on one of two paths: the first assumes that the application is brand new, and that the Data Modeler is working from a blank slate. In this case, informational requirements are captured as they are identified during the JAD sessions or interviews. As sessions are held, a view of potential entities and attributes is constructed and organized.

The second path assumes that the new application is going to replace an existing system. In this case, the Data Modeler may work with the current Data Base Administrator (DBA) to reverse engineer the existing database or file structures, often using a modeling tool. This enables the Project Team to use the existing structures as a starting point for the new Logical Data Model, and as a means

of validating that all informational needs of the system are being accommodated for in the Process Model.

It is important to define the entities and attributes in business English to facilitate Customer consensus and to ensure consistency with the organizational nomenclature forming the framework for the design of the application and the database. As additional requirements are flushed out during the interviews or sessions, the informational needs of the system are continually re-analyzed and re-applied to the data model. It is important to keep in mind, when identifying data sources, that consideration must be given to enterprise data and standard data that may be maintained in external systems (e.g., County Code Table, OSC code tables, etc.)



A key to successful data modeling is to ensure that the logical data model is not dependent on how the system processes the data. This ensures that data is grouped and organized based strictly on the informational needs of the system, and not based on an implied or assumed usage of the data by the system. The benefit is that the integrity of the data model will remain intact even if future business needs change the functionality of the system.

Defining the data model also helps to define the business rules by establishing the data entities (tables) and identifying attributes (fields). With the requirements and business rules known, and the Process Model outlined, the Project Team can begin to establish relationships between the data entities. This becomes the foundation of the data repository (or physical data model). As attributes are identified, the Data Modeler begins to build the Data Dictionary – again, in business English. Data normalization, a process in which complex relationships are simplified, is important once the Data Dictionary has been established. This eliminates redundancy, creates stable data structures, prevents anomalies, and simplifies data maintenance. The Logical Data Model is the basis for the DBA to create the physical database, so it is important that the Data Dictionary is clear in its definitions, and that all the data has been modeled appropriately.



The experience that a seasoned Data Modeler can bring to the Project Team can often make the difference between a successful project and one that encounters multiple setbacks or surprises. Understanding how to identify entities and attributes, establish relationships between the entities, normalize the attributes and define the Data Dictionary are important to developing a high performing application. These activities lay the groundwork for the technical team to build the physical database.

The Project Manager's goal for this process is to ensure that the Project Team accurately reflects the data requirements as they are defined, and as they relate to the Business Requirements and Process Model. The Project Manager should ensure that a process exists for the various Project Team members to share information and refine the requirements and models without risk of losing information, or jeopardizing the consistency and inter-dependencies of these deliverables.

The evolution of the Logical Data Model is analogous to the creation and confirmation of the Process Model in that it requires frequent interaction and validation by the Customer. While many of the deliverables in System Requirements Analysis present information in formats and terminology with which the Customer is familiar, this may not always be the case with data models. As a result, the Project Manager should anticipate the need for additional interaction during these reviews to ensure that the Customer can accurately interpret the output of this process. In addition, walk-throughs must be conducted in conjunction with, or in close proximity to, the reviews of the Process Model to be most effective.

Deliverable

- ◆ **Logical Data Model** – Diagrams and Data Dictionary information defining data elements, their attributes, and logical relationships as they align within a business area, with no consideration yet given to how data will be physically stored in actual databases.

2.5 RECONCILE BUSINESS REQUIREMENTS WITH MODELS

Purpose

The purpose of **Reconcile Business Requirements With Models** is to ensure that all business requirements and rules that have been captured have been accurately reflected and accommodated for in the Process Model and the Logical Data Model.

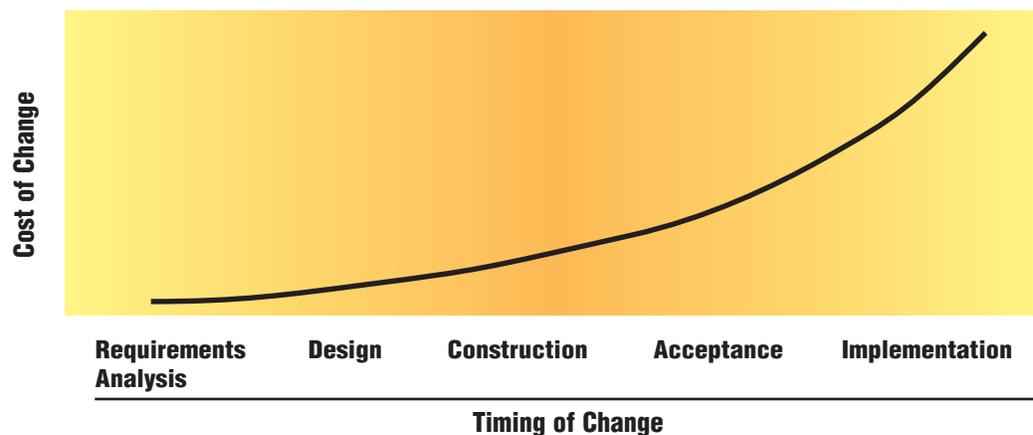
Description

Since the Process and Logical Data Models will ultimately form the basis of the system design, it is critical to invest the time here in System Requirements Analysis to ensure that these models are complete and accurate. If business requirements have been identified that are not reflected in these models, or if discrepancies exist between these models, then it is almost certain that this step will be revisited at some point later in the project. As Figure 2-6 indicates, the further out in the project that deficiencies in the business requirements are identified, the more costly the effort required to correct these deficiencies.

Roles

- Project Manager
- Business Analyst
- Facilitator
- Database Administrator
- Data/Process Modeler
- Technical Lead/Architect
- SQA Analyst
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Stakeholders

Figure 2-6 Impact of Change on Project Costs



A typical technique at this point in the SDLC is to perform an analysis assessment, which validates and cross-references all requirements to the process and data models, and which continues until all gaps have been identified, resolved, or recognized as an out of scope item.

One technique to reconcile the Business Requirements, Process Model and Logical Data Model is for the Business Analyst to create a gap analysis checklist or matrix that may be used to display the interactions among the requirements, data entities and the processes. This will help to ensure that all the requirements have been captured and modeled appropriately.

It is helpful to walk Customers through this exercise so that they understand how all requirements have been captured and modeled. These reviews are often iterative, and any gaps identified are corrected through subsequent revisions to the Business Requirements, the Process Model, or the Logical Data Model. It may be necessary to hold several review sessions to go over the reconciliation with different sets of Customers, remembering that the more people who review the output, the less likely it will be that key elements have been missed.

The Project Manager must ensure that the Customer understands the ramifications of overlooking a process, or of failing to decompose and model data appropriately. By understanding the potential impacts on both schedule and cost, the Customer is more likely to dedicate the appropriate staff to participate in these reviews.

Deliverable

- ◆ **Validated Business Requirements and Models** – An updated set of Business Requirements, Process and Logical Data Models that have been modified to address any gaps or weaknesses identified as a result of a gap assessment performed on these documents as a single unit.

2.6 PRODUCE FUNCTIONAL SPECIFICATION

Purpose

Produce Functional Specification maps the Logical Data Model and Process Model to the organizations and locations of the business. This process also produces the final deliverable for the phase – the Functional Specification.

Description

The ultimate goal of this process is to derive a comprehensive representation of the application that logically organizes related business processes, functions, data, and workflows. This provides a detailed roadmap from which the Customer Representatives can fully envision the final solution, and from which the Project Team can progress into the Design and Construction phases of the project lifecycle. Whereas all System Requirements Analysis efforts up to this point have been focused on continually decomposing information into discrete requirements or processes that can each be reviewed and validated on their own merits, this final process now builds a broader view of the system that groups the individual pieces of the solution into logically related business functions. The final result, the Functional Specification, defines and illustrates how each requirement of the system will eventually be satisfied in terms of business processes (or transactions).

Roles

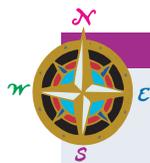
- Project Manager
- Project Sponsor
- Business Analyst
- Facilitator
- Data/Process Modeler
- Technical Lead/Architect
- SQA Analyst
- Technical Services
- Information Security Officer
- Technical Support
- Customer Decision-Maker
- Customer Representative
- Stakeholders

Deliverables resulting from this phase of the lifecycle must capture the full set of requirements for the new system in a way that is completely independent of any development approach, methodology, or organizational constraints. Much like System

Initiation defined **why** the new system was needed, the System Requirements Analysis deliverables must define **what** the system must do without making any assumptions regarding **how** the system will be built.

The Functional Specification will present many views of the system, from different perspectives and at different levels of detail. For example, the System Context Diagram shows how the new system fits into the larger picture of the performing organization's application portfolio. The Business Flow Diagram shows how Customer and Consumer business units will interact with the new system from the business process and data flow perspectives. And the System Interface Diagram will present a view of the system from a perspective of Consumer interface, depicting menu structures and navigation paths of online system components, and organization and distribution of reports and other batch interfaces.

Within the Functional Specification, each business process or transaction will correlate to the set of Business Requirements that it satisfies and a representation of the corresponding data elements. The reports associated with each process, business constraints (such as related security or controls), interfaces to other systems and business functions, and any related administrative operations required to support the system should also be identified.



When identifying the new vision of the system with its proposed sets of related processes and functions, organizational or operational changes may be introduced. Ultimately, the Customer must be comfortable with these changes, and be willing and able to institute them. The Project Team must take this into account when establishing this framework. Both the Project Manager and the Stakeholders must find the appropriate balance between the potential need for the change, and the likelihood that it will be embraced throughout the organization.

As discussed in Determine Business Requirements, requirements gathering sessions frequently result in features or functions above and beyond those initially envisioned during Project Origination being identified. One advantage of a well-defined

Functional Specification is that it provides the Project Team with a vehicle to assist the Customer with decisions on trade-offs in functionality and scope, should the situation arise that sufficient budget is not available to support the development of the full set of capabilities.

To ensure that the Customer agrees with the final deliverable, the Project Manager should schedule a walk-through to review the concepts and flow of the Functional Specification, and to achieve consensus that the proposed grouping of processes defines a solution that will satisfy the Customer's needs.

Deliverable

- ◆ **Functional Specification** – Document describing the logical grouping of related processes and functions within the new system, along with the mapping of these processes to both the business requirements that they satisfy and the data items with which they interact.

Figure 2-7 Functional Specification Template

< Name of Agency >
Functional Specification
< System Name >

| | |
|-----------------|--|
| Agency | |
| Project Name | |
| Project Sponsor | |
| Project Manager | |
| Document Date | |
| Prepared By | |

*Enter the name of the **Agency** for which the system is being developed.
Enter the **Project Name**, and the names of the **Project Manager** and the **Project Sponsor**.
Enter the **Date** as of which this document is current.
Enter the names of the **Project Team** members by whom the document was **Prepared**.*

Figure 2-7 (Continued)

Functional Specification

TABLE OF CONTENTS

The **Table of Contents** should be at least two levels deep.

1.0 DOCUMENT SCOPE

Document Scope describes the goal of this document in a narrative.

Example:

To define a comprehensive set of functional specifications for the <XYZ> system and to identify what is in and out of scope.

The narrative also provides an overview of the efforts conducted to gather business requirements, derive process and logical data models, and to reconcile business requirements with these models.

2.0 GENERAL FUNCTIONAL SPECIFICATIONS

The **General Functional Specifications** section details those specifications that are common to all aspects of the system (e.g., the menu structure, security, accessibility, overall performance requirements, etc.).

Three graphical representations of the overall system should be included:

- a System Context Diagram, showing how this system will interact with other existing systems,
- a Business Flow Diagram, showing how Customer and Consumer business units will interact with the system; and
- a System Interface Diagram, showing the application structure (menu structure and navigation of online components), organization of reports and other batch interfaces, and utilities.

3.0 DETAILED FUNCTIONAL SPECIFICATIONS

The **Detailed Functional Specifications** section lists functional specifications for each aspect of the system. 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.

It may also be useful to provide more detailed versions of the Business Flow and System Interface diagrams for each sub-system.

Figure 2-7 (Continued)

Functional Specification

3.1 Sub-system

The **Sub-system Description** describes the sub-system in a narrative.

3.1.1 Component Type

Depending on system structure (and Functional Specification document), it may be useful to organize system components by **type** (such as screens vs. reports, or tracking vs. auditing). If that is the case, **Component Type Description** would provide a rationale for such structural breakdown, and describe common elements of all components within that type.

Component Type Description

3.1.1.1 Component 1

- Component Description
- Component Mockup (where appropriate)
- Component Business Flow
 - Cross-reference to Business Requirement(s), Logical Data and Process Models
 - Flowchart
 - Detailed Business Rules for each Flowchart element

The **Component Description** should identify the appropriate Customer or Consumer group, and provide a description of how their needs are being met by this component.

Where appropriate, a mockup of the component should be included in the document.

Component Business Flow details how the system supports the business process.

A **Cross-Reference** is provided to all prior deliverables.

A **Flowchart** details the system component's interaction with the business process. Every shape and arrow on the Flowchart is annotated with detailed descriptions of **Business Rules** governing that particular interaction or transformation.

3.1.1.2 Component 2

- Component Description
- Component Mockup (where appropriate)
- Component Business Flow
 - Cross-reference to Business Requirement(s), Logical Data and Process Models
 - Flowchart
 - Detailed Business Rules for each Flowchart element

Figure 2-7 (Continued)

Functional Specification

4.0 OTHER SPECIFICATIONS

Besides functional aspects of the Business Requirements, the specifications for the system should also enumerate technical, operational and transitional aspects of the system.

4.1 Technical Specifications

This section documents in detail the technical specifications, regulations and existing constraints that must be considered in relation to business requirements. These include considerations such as accessibility, encryption, security, disaster recovery, and other technical areas.

4.2 Operational Specifications

This section should document in detail the operational specifications that must be considered in relation to business requirements. These include considerations such as system performance, data archival, audit and controls, system administration, software quality assurance and business continuity. The narrative should specify how these operational requirements may affect the organization and its current business processes.

4.3 Transitional Specifications

This section documents in detail the transitional specifications that must be considered in relation to business requirements. These include considerations such as data conversion, system testing, documentation, training and deployment. The narrative should describe how historical data will be cleansed, converted and imported into the new system, how expectations must be set for the deployment of and support of user and technical documentation and training, and what approach may be employed to physically deploy and transition the system into the organization.

5.0 BUSINESS REQUIREMENTS NOT BEING IMPLEMENTED

This section specifies those requirements that will NOT be part of the new system. If this list is identical to the one published in the Business Requirements document, a simple reference to the prior document may be substituted.

APPENDICES – SUPPORTING DOCUMENTS

The Appendices should contain all relevant documents provided by Customers and Consumers during System Requirements Analysis, as well as documents supporting decisions made while compiling the Functional Specification.

Measurements of Success

The immediate measurement of success for System Requirements Analysis is the acceptance of all deliverables by the Customer, while the ultimate measurement is whether or not the Project Team has created solid groundwork for the upcoming design and development of the application. Each process in this phase builds towards the final deliverable: Functional Specification. It is necessary to validate that certain steps have been successfully executed to ensure that the Functional Specification has been derived appropriately.

The Project Manager can assess how successfully this phase is proceeding by utilizing the measurement criteria outlined below. More than one “No” answer indicates you may have serious risk to the Project.

Figure 2-8

| Process | Measurements of Success | Yes | No |
|--|---|-----|----|
| Prepare for System Requirements Analysis | Have all new team members participated in project orientation sessions? | | |
| | Is the team comfortable with the process defined for managing the deliverable repository? | | |
| | Do all team members have experience with (or training on) the tools that will be used in this phase? | | |
| Determine Business Requirements | Do the business requirements state what is in scope as well as what is out of scope? | | |
| | Have the business requirements been reviewed and approved by the Customer? | | |
| | Are the requirements stated in such a way as to allow for easy validation of their existence in the final solution (i.e., a Yes/No determination of whether or not the requirement has been satisfied by the new system)? | | |
| | Are requirements prioritized within the Business Requirements deliverable? | | |
| | Do the requirements consider the technical, operational and transitional aspects of the system, including elements such as: <ul style="list-style-type: none"> • Security/access needs; • Existing technical standards (accessibility, encryption, etc.); • Application hosting; | | |

Figure 2-8 (Continued)

| Process | Measurements of Success | Yes | No |
|---|--|-----|----|
| Determine Business Requirements (Continued) | <ul style="list-style-type: none"> • Disaster recovery; • Archiving, audit and regulatory needs; • Performance requirements by all Customers and Consumers for all aspects of the system; • Business continuity; • Data conversion? | | |
| Define Process Model | Have all known business requirements and associated business rules been mapped and accommodated for in the Process Model? | | |
| | Has the Process Model deliverable been reviewed and approved by the Customer? | | |
| Define Logical Data Model | Have all known business requirements and associated business rules been mapped and accommodated for in the Logical Data Model? | | |
| | Has the data model been normalized? | | |
| | Has the Logical Data Model deliverable been reviewed and approved by the Customer? | | |
| Reconcile Business Requirements With Models | Does the Customer agree that all aspects of the requirements and rules have been accommodated for in the process and data models? | | |
| Produce Functional Specification | Has the Functional Specification deliverable been reviewed and approved by the Customer? | | |
| | Has a system validation and testing approach been formulated? | | |
| | Have training needs been identified? | | |
| | Has an approach for the implementation and transition of the application been developed? | | |

Phase Risks / Ways to Avoid Pitfalls

PITFALL #1 – IF YOU BUILD IT (TOO SOON), WILL THEY COME?



Your Project Team is top-notch and you can't believe your luck in securing their stellar capabilities. You have the guru of gurus as your development lead. After the first JAD session he tells you he already has envisioned the application – it's a financial module and similar to one he has previously developed. He convinces you that he needs to skip the rest of the JAD sessions so he can begin to prototype immediately. He is so enthusiastic and convincing in his argument to you, that you bless him to go off and start creating. You have already re-forecast your

Project Schedule mentally and foresee an early delivery date. How sweet it is!

Four weeks into the JAD sessions, you haven't yet seen any output from him. He tells you he's almost there, just a few more tweaks ... you remind him it is just a prototype, not a full-fledged construction effort yet. He assures you it is just what the Customer wants. But you realize that the Customer hasn't been involved in the prototyping exercise. How can your lead developer be creating anything without the Customer vision?

You arrange a meeting with the developer and the Customer to review the prototype as it is so far, and to elicit feedback. Your developer is giddy with anticipation, convinced that what he has already built will make their dreams come true. Then, the moment of truth: the developer begins to walk the Customer through the first screen, and it is already apparent that there is a huge disconnect between the developer's and the Customer's visions of the application. Productive dialogue quickly turns into heated debate.

It all could have been avoided if the developer's vision really was the Customer's vision. If only he had gone to the interviews and JAD sessions and heard the requirements, instead of making assumptions as to what HE thought the Customer should have. If you build it, you'd better make sure they can see it, or they won't come.....

PITFALL #2 – LET'S CROSS THAT BRIDGE WHEN WE COME TO IT



During your JAD sessions the Project Team has collected and prioritized a set of issues that need to be resolved by Customer Decision-Makers or a Steering Committee. However, some of the issues appear to be so huge that no one thinks they'll be able to resolve them. Instead of using the defined escalation process outlined in your Communication Plan, group consensus based on speculation and gut feeling is that by the time the Project Team is ready to start development, the issue will have been resolved by the 'higher ups'. Thus, the old saying, "Let's cross that bridge when we come to it."

Unfortunately, everyone's crystal ball that day was cloudy and/or cracked, and assumptions were proven wrong. The issue did not go away and it has suddenly become a high priority need because legislation was passed to 'make it so'.

Because the team did not address or resolve the issue, a major piece of functionality for the application is missing, and has a negative downstream effect on other modules. Why oh why didn't you escalate the issue when it arose, so you wouldn't be in the boat you're in now?

Taking the time NOW, during those JAD sessions, to address all issues, assess their impact and develop resolutions to them, is critical to achieving success in the ultimate design and development of the application. You will also save yourself hours of regret and heartache later.

PITFALL #3 – DESIGNING ON THE FLY



By nature, most technicians – including system developers – are natural problem-solvers. They are the kind of folks that disdain reading the twenty pages of directions, but immediately start fitting the parts together by eye; the kind of people who never read the manual, but just start pushing the buttons to see what happens.

When you sit with them at a requirements gathering session, you may see their eyes glaze over and their faces assume this far-away look. This is by no means due to a lack of interest, but because they are already designing – and may be even programming – the new system in their heads, based on some initial snatches of conversation.

Tell them to “Snap out of it!”

If the system requirements are not sufficiently defined and understood, the Project Team may experience “expectation gaps” – for example, the developers may build a Taj Mahal, while the Customer wanted a simple privy. At best, this may result in frustration and friction between the Project Team and the Customers – at worst, it can mean significant cost and schedule overruns that can impact many aspects of the Customer's business operations.

The trick to successfully capturing business requirements is to make sure that the Project Team does not get ahead of itself. The Business Requirements deliverable is a detailed and concise list that, without passing judgment and without in any way

indicating a solution, identifies the full set of business requirements that must be met by the system. If the list has been done correctly, the Project Team should be able to walk through it at any future point in the lifecycle and determine whether the emerging solution satisfies each requirement.

PITFALL #4 – GOT PICTURES?



Contrary to what the SDLC preaches, the wording used in Business Requirements can end up high-level, leaving many of the actual requirements open to interpretation. As this chapter has discussed, the decomposition of these requirements involves translating the business need into discrete, well-defined components that collectively provide the desired functionality. Therefore, you need to be creative in the ways in which these requirements are captured. Whenever possible, do not limit your understanding and representation of the requirements strictly to the written word.

As we all know, the English language is open to misinterpretation. How many of us have walked in for a haircut, described exactly what we're looking for, and walked out 30 minutes later wondering how "Take a little off the sides" turned into "I'd like you to make me as physically repulsive as possible, so that young children cry at the sight of me, and junkyard dogs turn and flee for safety"? In this case, a picture would have been worth more than a thousand words (at least to the children and junkyard dogs of the world).

The same is true when capturing system requirements. Words like "automate", "process", and "calculate" may mean different things to different people, and it is essential that everyone involved in the project share the same view of these requirements. Moving forward with an incomplete or incorrect vision of the system can be more terrifying than the haircut, except that this time, the ones running away will be your Customers. Never underestimate the power of prototyping, diagrams, illustrations, and modeling when trying to fully represent and validate the requirements that your Customer is communicating to you. These techniques, and the pictures that result from them, can be worth 1000 words, and many times that in dollars.



Frequently Asked Questions

What do you do if you don't have a Project Team schooled in the art of early life cycle techniques?

The skills required for the requirements definition and analysis activities are very different from those utilized in other phases of the System Development Lifecycle. The best strategy is to get an experienced Facilitator, if only for a few initial sessions, to transfer the skills to the team. However, if no early life cycle expert is available, you need to identify candidates among your team who are best suited to an interactive mode of communications, get them into some training, and have them study all available materials (including this chapter of the Guidebook). Then, make sure the team follows the SDLC methodology, and you, the Project Manager, follow the PM methodology to the letter, and rely on Project Sponsor and Customer Decision-Maker feedback to make sure your team is doing the right things.

How does your Project Team know when done is done?

In the normal course of events, if you have identified ALL Customers, Consumers and Stakeholders, gathered ALL their requirements as they relate to this system, built your Process and Logical Data Models, and then validated, cross-checked and verified everything, obtaining Customer and Project Sponsor approvals all along the way – then you are done! However, you are probably talking about cases when Customers keep changing their minds and you are trapped in an endless cycle of revisions and clarifications. Or cases when you keep chasing Customers who just won't spare a moment to talk to you. In either instance, think Time Box. With the help of your Project Sponsor, declare a deadline, communicate it to all participants, and end the game there.

What if you don't know which tools to select to help you through this phase?

While many tools exist that can simplify the creation of documents, illustrations, and other materials that support requirements analysis efforts, it should be stated that a pretty chart does not a good system make. People used to draw flowcharts by hand – and their systems did not come out any worse. The important thing is to get all the requirements, understand how

they relate to each other and to current and proposed business processes, and to get agreement and consensus on what will be delivered and when. And if in the process you can use some nifty tool and generate some neat documents – great!

What is the risk of just compiling all the process deliverables into one big deliverable, and foregoing the final deliverable: Functional Specification?

The Functional Specification deliverable has many aspects to it that its constituent parts miss (look at the annotated templates earlier in this chapter to see what they are). It is not enough to slap a bunch of work products together, tie them with a rubber band, and declare victory. “The whole is greater than the sum of its parts.” The Functional Specification is a document that supplies the background for the effort and organizes the work products in a logical sequence that makes it easy to understand the process and verify the result.

How do I deal with a Customer who is afraid to commit?

We’ve all dealt with folks who believe that putting their signature on that acceptance form will forever doom them to dealing with an inadequate and cumbersome system; and whether that attitude comes from having such forms flung back into their faces in the past in response to reasonable requests or from genuine personality disorders is beside the point: you need to get the Customer OK, and he won’t have any part of it! What to do?

For starters, don’t say, “Just trust me!” (or equivalent). Like love, trust takes a while to form, and you can’t really force it. Try to figure out what the underlying concern is. Is it fear of being locked into a particular design? Mistrust of the process used to gather the requirements? Lack of confidence in the players? Misunderstanding of the nature or purpose of the deliverable? Lack of knowledge about what will come next?

Education is the key here. Try to have a reasonable, calm conversation with the Customer Decision-Makers. Go over the process used to gather the requirements and prepare the deliverable. Explain the methodology, and the intent behind it: to support the Customer’s business process with the best darn system a bunch of chip-heads can come up with.

And if they are still holding up the process without a good reason, pull out your big gun – the Project Sponsor – and have him earn his keep.

Where in the lifecycle do I define that a phone number is ten digits with dashes and parentheses?

Yes, it's a lot easier to record vague functionality requests ("the system should produce the required reports accurately and on time") than to get to brass tacks and figure out exactly what is going to happen, when and how.

When you are gathering business requirements, you should document every data source the Customer mentions, and mock up every interface the Customer requests. The data elements thus captured are formalized in the Data Dictionary (part of the Logical Data Model deliverable), and then further refined during requirements reconciliation, development of the Functional Specification deliverable, and creation of the prototype. By the time technical specifications are created, your data definitions should be set in concrete.