UC SANTA CRUZ

Systems Development Life Cycle (SDLC) Methodology

Information Technology Services

July 7, 2009
Version 1

Authors: Mel Barracliffe, Lisa Gardner, John Hammond, and Shawn Duncan

Document Control

Change Record

Date	Author	Version	Change Reference

Reviewers

Sign Off Date	Reviewer	Position	Sign Off

Distribution

Copy Number	Name

Table of Contents

1. Introduction

ITS provides a spectrum of information technology (IT) resources and services that support the instruction, research, administrative operations, and public service of the University

We provide information technology solutions to students, faculty and staff in the areas of:

- Instructional computing
- Administrative computing
- Voice and data services
- Information systems security
- Web services
- Media services
- Technical support and training

The successful provision of these IT solutions to the campus requires close cooperation between multiple groups and roles within ITS. In particular, the partnership between Applications and Project Management and Core Technologies is most critical in designing, building and implementing solutions that meet the needs of the campus and that are cost effective to maintain and evolve.

The degree of collaboration and integration of work effort required for successful systems delivery does not happen by accident; it must be deliberately designed and managed with diligence. The process of systems development must be agreed to and prescribed, roles and responsibilities defined, handoffs engineered, and deliverables properly specified to ensure smooth transitions at each stage of development.

The primary reasons for using a methodology are:

- Improve systems development productivity
- Improve quality of the system delivered
- Reduce cost of delivery
- Enable consistency and repeatability

The main benefits of adopting a methodology are:

- Use of a common language and better communication
- Clear definition of project goals and objectives and outputs
- Standardization of deliverables and techniques
- Better management of client expectations

Clear delineation of roles and responsibilities

- ITS roles
- Client roles
- Enable role specialization
- Improve development and leverage of junior staff
- Improve accountability

Other benefits that can be realized as an organization matures in its use of a methodology are:

- More accurate estimating of the work effort
- Improved management of schedule and costs
- Minimization of rework
- Capture and leverage past experience

    - Document templates and examples
    - Deliverables (Code, models, designs, etc.)
    - Estimating models
- Development of "Know how" and propagation of leading practices
- Resource reporting
- Improved control and predictability of the project

The **Systems Development Lifecycle (SDLC) Methodology** presented here is the result of a collaborative effort by Applications & Project Management (APM), Core Technologies (CT) and Client Services and Security (CSS). It provides a framework of principles, practices, and procedures to guide the systems development process. It is intended as an initial iteration of a methodology that will be refined and evolved over time as we develop better insight and practices on effective systems development in our environment.

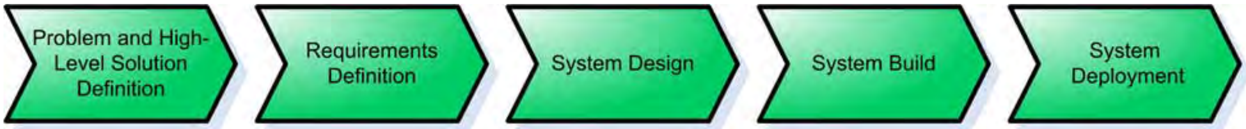
The SDLC Methodology is a companion methodology to the **ITS Project Management Methodology** already in use at ITS. Our long-term vision is to have a single methodological framework for the ITS function of which the SDLC, PM, EA, Service Management methodologies and future methodologies will be key components. For this reason we have standardized overlapping constructs wherever possible between these methodologies. For example, the standard methodology roles represent a harmonized set of roles across the SDLC, PM and Service Management.

In this document we use "system" interchangeably to include the application(s) and all components necessary to deliver the desired capabilities. There are many different lifecycle models for an SDLC, such as waterfall, spiral, iterative, agile, etc. Any one of these approaches may be selected depending on the complexity, priority, and/or cost of development and operations related to the desired solution. In this first iteration of the SDLC we have focused on the basic work breakdown structure and deliverables that would be common to all these lifecycle models. In a future narrative we will provide further guidance on how to adapt the framework presented here to specific life cycle models.

1.1. Methodology Work Breakdown Structure (WBS)

The methodology is organized into 5 phases of work:

- Problem and High-Level Solution Definition
- Requirements Definition
- System Design
- System Build
- System Deployment



These 5 phases of the SDLC are related to the 5 **PM Methodology** phases: **Define, Plan**, **Launch**, **Manage** and **Close**[1]. The work performed in the Problem/Solution Definition phase is integral to the Define and Plan phases of the PM methodology. A project cannot be estimated, planned or chartered if the scope of the solution is unknown. The PM methodology provides guidance on the conception of a project, the development of a Project Proposal or Charter and the approval and launch of the project. The SDLC

[1] For more details of the PM methodology please go to the ITS Web Site http://its.ucsc.edu/pmg/docs/pm_methodology.pdf

methodology guides the consideration of the client problem and the scope of a proposed IT solution to address it. If the project is approved, the remaining phases of the SDLC (Requirements Definition, Solution Design, Solution Build and Solution Deployment) will be performed as part of the project plan. This relationship is illustrated in Figure 1 .

Depending on the scale and nature of the problem and solution, the Requirements Definition, Solution Design, Solution Build and Solution Deployment phases may not progress in one linear sequence (waterfall approach) through to completion. In some circumstances, an iterative project approach may be employed in which repeated sequences of Requirements Definition, Solution Design and Solution Build activities will progressively define and build the solution (iterative or spiral approach).

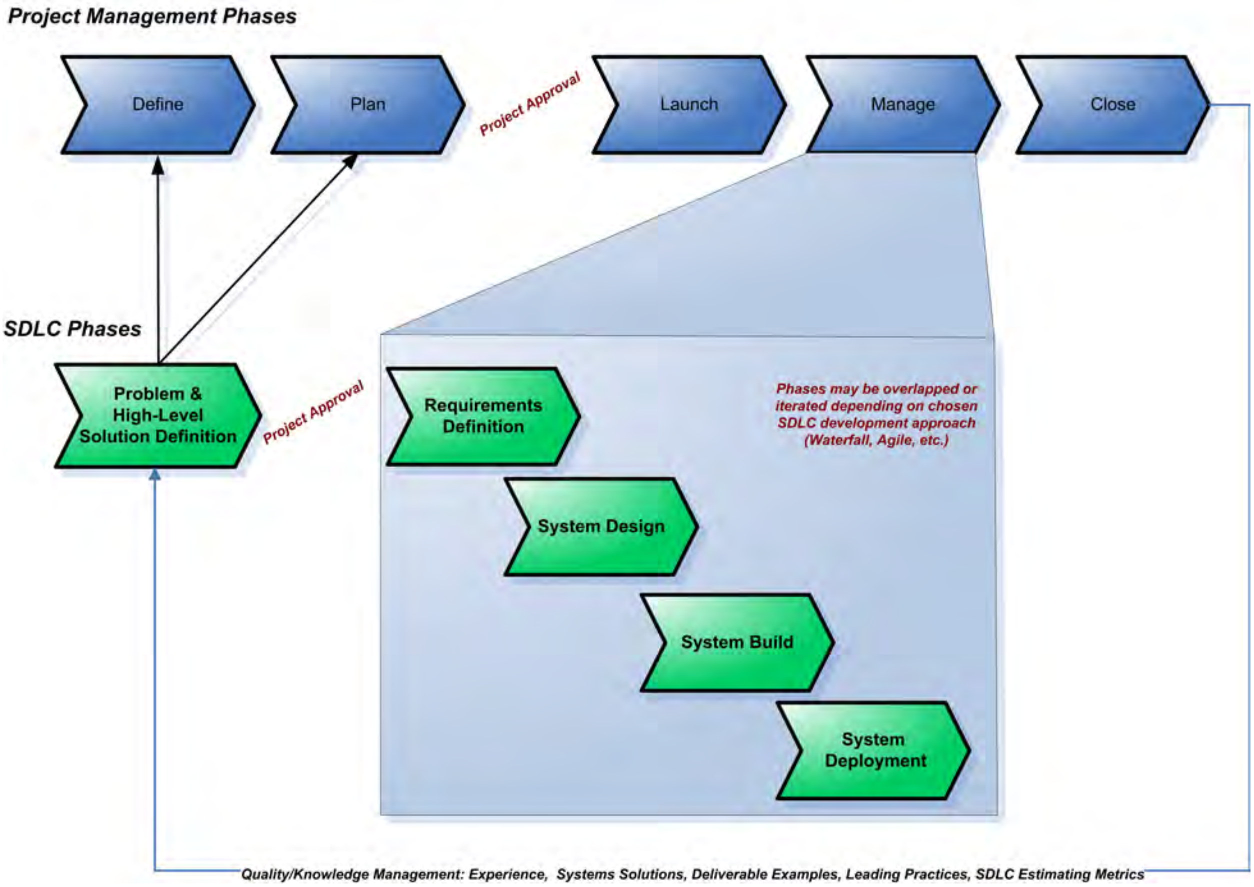
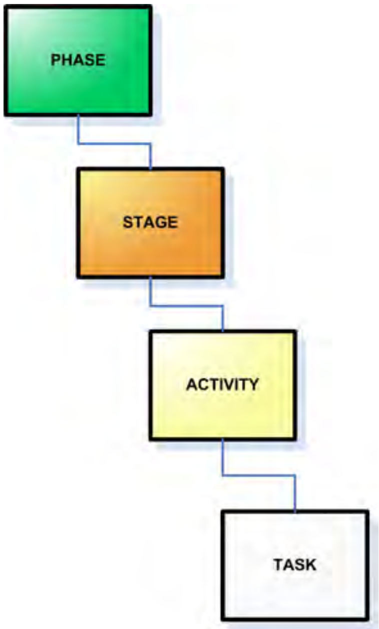

Project Management Phases
Define
Plan
Project Approval
Launch
Manage
Close
SDLC Phases
Problem & High-Level Solution Definition
Project Approval
Requirements Definition
Phases may be overlapped or iterated depending on chosen SDLC development approach (Waterfall, Agile, etc.)
System Design
System Build
System Deployment
Quality/Knowledge Management: Experience, Systems Solutions, Deliverable Examples, Leading Practices, SDLC Estimating Metrics

Figure 1 Relationship of PM and SDLC Methodologies

This SDLC Handbook is organized around a standard **Work Breakdown Structure (WBS)** has been developed to describe the typical work conducted during a systems development project (Figure 2). The WBS consists of 4 named levels of work decomposition: Phases, Stages, Activities and Tasks.

The WBS defines the work performed; the deliverables and work product represent the results or outputs of the WBS and structure. In this way, each **Phase** produces a corresponding set of major outputs or deliverables that move the development effort forward. These outputs may be actual physical artifacts such as System Design Proposal or major outcomes such as a “Go Live” event or a key sponsor approval. A table of all the major outputs for a phase (Deliverables plus any Work Products) is provided for each phase handbook section. In addition, the major deliverables for each phase are also listed in the detailed WBS diagram for each phase.

Each Stage generally corresponds to the work required to create a major deliverable, and Activities describe the more detailed steps to create a Work Product or component of a Deliverable. This WBS should be used as a starting point for developing a specific systems development Project Task Network in Microsoft Project or Merlin[2].



It should be noted that there is a distinction between a Project Task Network and a WBS. A WBS is a normalized description and classification of the work to be performed. For convenience the WBS may be described sequentially, as it is in this handbook, however a WBS sequence of tasks does not necessarily imply linear dependencies between those tasks. To build a realistic and robust plan, and calculate an accurate project critical path, the project manager must transform the WBS into a set of tasks in the project planning tool that accommodate the development approach. For example, the iteration of SDLC activities and tasks should be represented as well as their task dependencies (Start-to-Finish, Start-to-Start, etc.). The Project Manager must also add task duration estimates and assign project team roles to tasks (See Appendix A – RACI Matrices for guidance on Task role assignments). For more guidance on developing a project plan and please consult the ITS PM Methodology.

The actual project plan should combine tasks from the PM and SDLC methodologies so they are integrated into a single project network and plan. Additional project specific tasks should be added by the Project Manager as required at the **Activity** and **Task** levels below each major Stage. The major SDLC phases must be clearly delineated in the project plan either by using corresponding phase names as summary tasks or through the use of specific milestone tasks. This approach will allow us to compare and status plans from different ITS teams, and begin to collect measurements for estimating metrics in the future.

[2] Template SDLC Microsoft Project and Merlin Project plans can be found on the ITS web site.

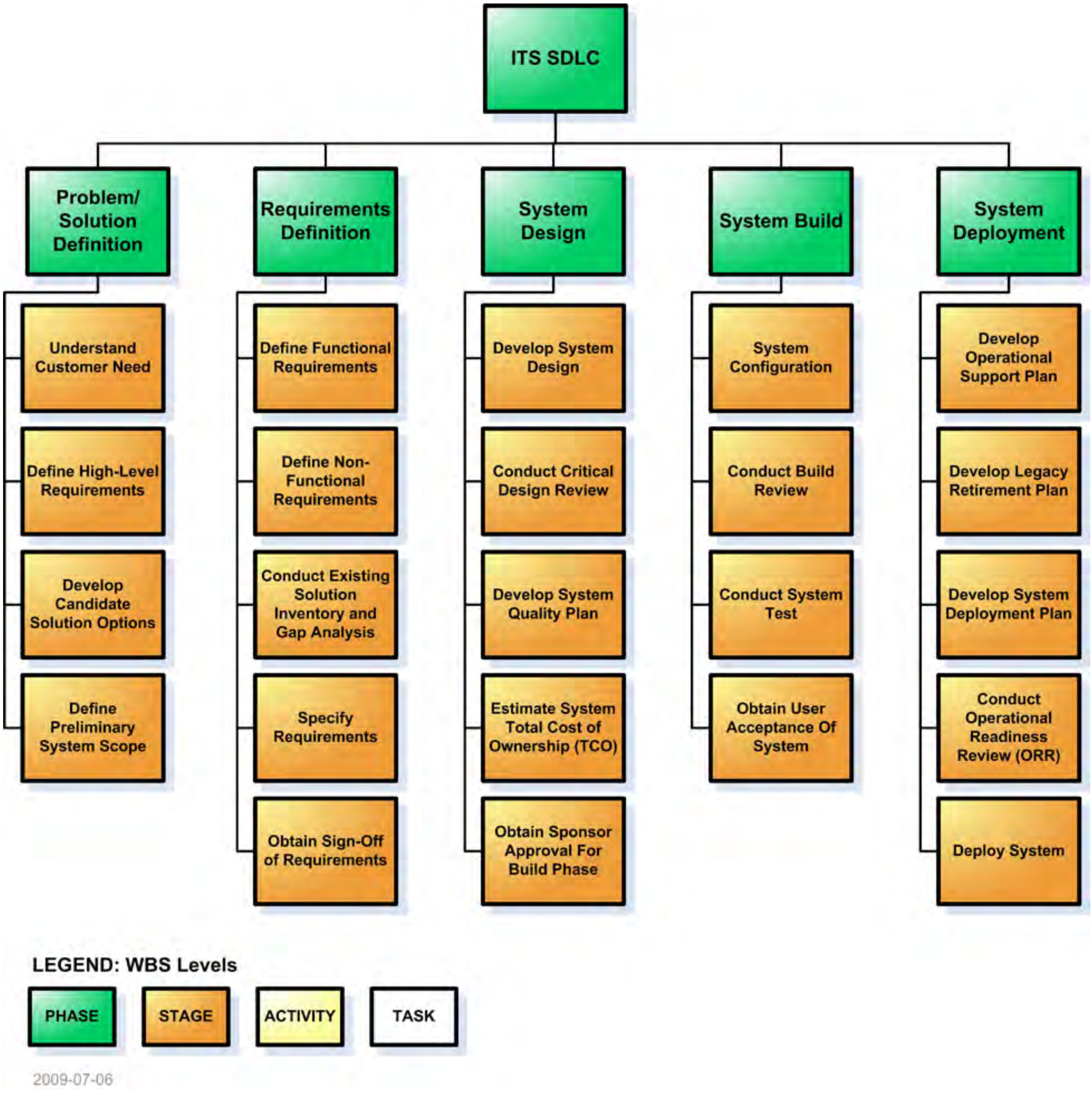
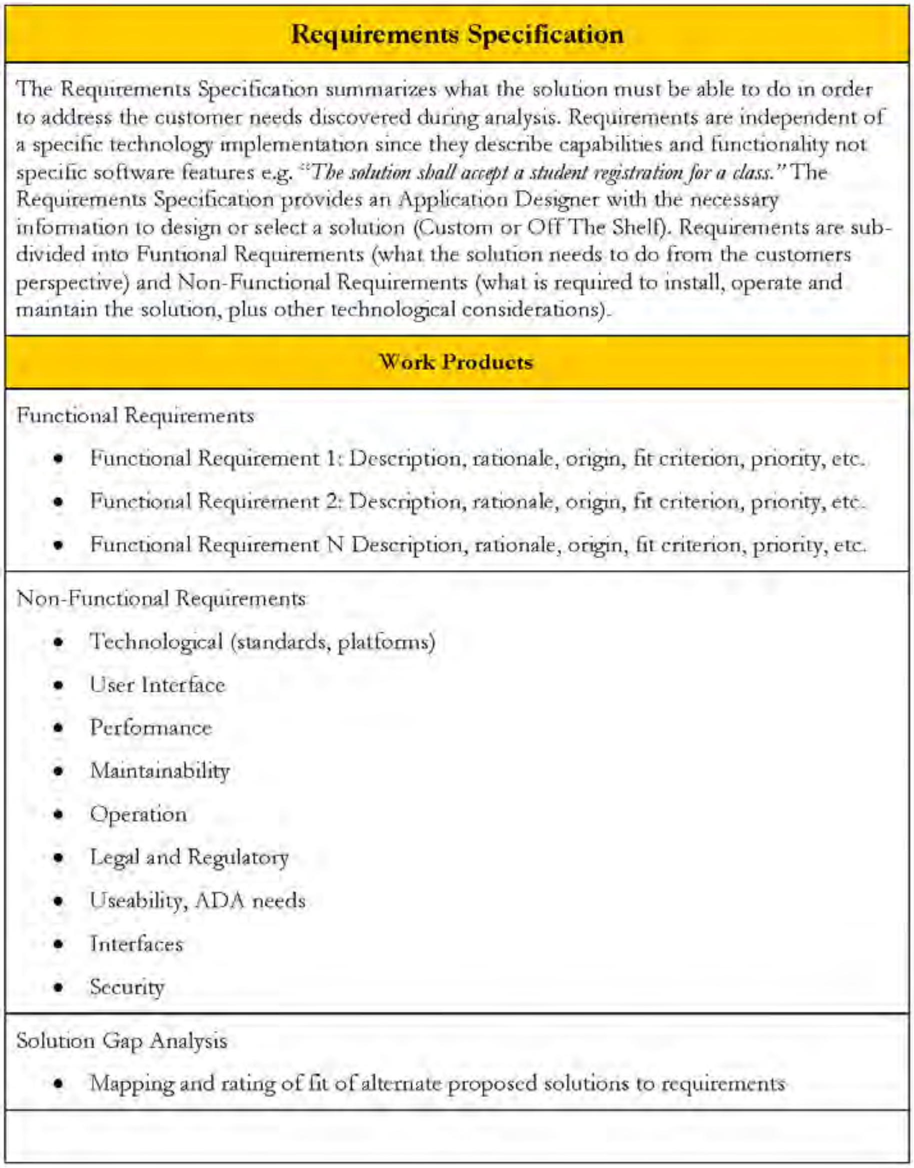

ITS SDLC
Problem/ Solution Definition
Requirements Definition
System Design
System Build
System Deployment
Understand Customer Need
Define High-Level Requirements
Develop Candidate Solution Options
Define Preliminary System Scope
Define Functional Requirements
Define Non-Functional Requirements
Conduct Existing Solution Inventory and Gap Analysis
Specify Requirements
Obtain Sign-Off of Requirements
Develop System Design
Conduct Critical Design Review
Develop System Quality Plan
Estimate System Total Cost of Ownership (TCO)
Obtain Sponsor Approval For Build Phase
System Configuration
Conduct Build Review
Conduct System Test
Obtain User Acceptance Of System
Develop Operational Support Plan
Develop Legacy Retirement Plan
Develop System Deployment Plan
Conduct Operational Readiness Review (ORR)
Deploy System
LEGEND: WBS Levels
PHASE
STAGE
ACTIVITY
TASK
2009-07-06

Figure 2 SDLC Methodology WBS

1.2. Standard Methodology Deliverables

Each level of the methodology WBS has a corresponding standard **Deliverable** or **Work Product** that it is expected to create. A Work Product is a component of a deliverable, or an interim product of the development process. Deliverables and Work Products can represent physical artifacts (System Configuration, Requirements Specification Document) or a decision or outcome (Approval For Build Phase).

The required logical content of a deliverable is specified in a Deliverable Definition. As an example Figure 1 is a Deliverable Definition for a Requirements Specification. The exact format and style of a deliverable will be illustrated with Example Deliverables to be developed and stored on the Repository web site.



Requirements Specification

The Requirements Specification summarizes what the solution must be able to do in order to address the customer needs discovered during analysis. Requirements are independent of a specific technology implementation since they describe capabilities and functionality not specific software features e.g. *"The solution shall accept a student registration for a class."* The Requirements Specification provides an Application Designer with the necessary information to design or select a solution (Custom or Off The Shelf). Requirements are sub-divided into Funtional Requirements (what the solution needs to do from the customers perspective) and Non-Functional Requirements (what is required to install, operate and maintain the solution, plus other technological considerations).

Work Products

Functional Requirements

- Functional Requirement 1: Description, rationale, origin, fit criterion, priority, etc.
- Functional Requirement 2: Description, rationale, origin, fit criterion, priority, etc.
- Functional Requirement N Description, rationale, origin, fit criterion, priority, etc.

Non-Functional Requirements

- Technological (standards, platforms)
- User Interface
- Performance
- Maintainability
- Operation
- Legal and Regulatory
- Useability, ADA needs
- Interfaces
- Security

Solution Gap Analysis

- Mapping and rating of fit of alternate proposed solutions to requirements

Figure 3 Example Deliverable Definition

1.3. Methodology Roles and Responsibilities

A standard set of roles have been defined for describing the specific involvement of various participants in the systems development process e.g. Systems Analyst, User or System Architect. A complete set of standard role definitions can be found on the ITS web site in the Project Tools and Templates section. This set of roles augments the standard roles previously defined by the PMG Project Management Methodology e.g. Project Manager or Project Sponsor. These roles are not job descriptions or job classification levels, a given individual my play one or more roles in one project. They are simply a means of describing the responsibilities of team members on a project.

The responsibility of each role is specified in a RACI matrix that relates the roles to the activities and deliverables.

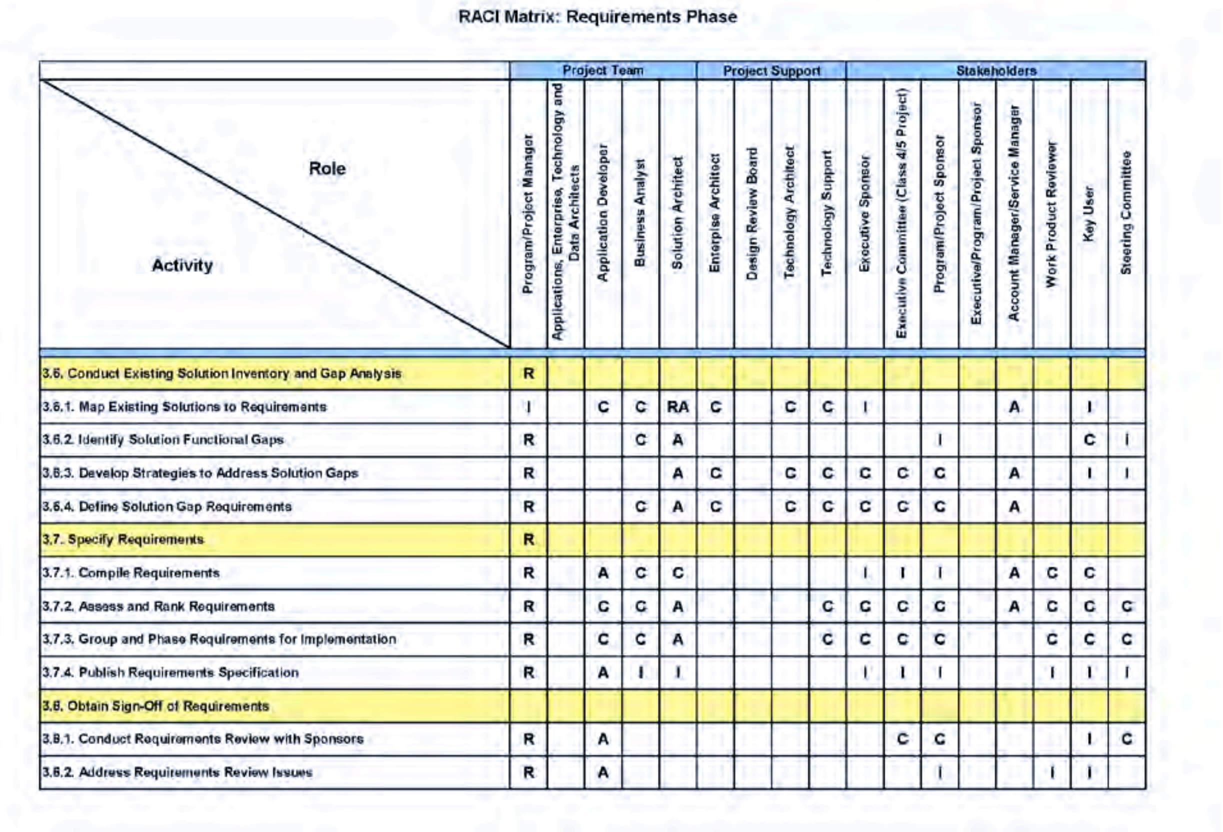
RACI Definitions:

Responsibility = person or role responsible for actually doing or completing the item

Accountable = person or role accountable for ensuring that the item is completed (typically one person)

Consulted = person or role whose subject matter expertise is required in order to complete the item

Informed = person or role that needs to be kept informed of the status of item completion



RACI Matrix: Requirements Phase

	Project Team					Project Support				Stakeholders							
Role Activity	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Solution Architect	Enterprise Architect	Design Review Board	Technology Architect	Technology Support	Executive Sponsor	Executive Committee (Class 4/5 Project)	Program/Project Sponsor	Executive/Program/Project Sponsor	Account Manager/Service Manager	Work Product Reviewer	Key User	Steering Committee
3.6. Conduct Existing Solution Inventory and Gap Analysis	R																
3.6.1. Map Existing Solutions to Requirements	I		C	C	RA	C		C	C	I				A		I	
3.6.2. Identify Solution Functional Gaps	R			C	A							I				C	I
3.6.3. Develop Strategies to Address Solution Gaps	R				A	C		C	C	C	C	C		A		I	I
3.6.4. Define Solution Gap Requirements	R			C	A	C		C	C	C	C	C		A			
3.7. Specify Requirements	R																
3.7.1. Compile Requirements	R		A	C	C					I	I	I		A	C	C	
3.7.2. Assess and Rank Requirements	R		C	C	A				C	C	C	C		A	C	C	C
3.7.3. Group and Phase Requirements for Implementation	R		C	C	A				C	C	C	C			C	C	C
3.7.4. Publish Requirements Specification	R		A	I	I					I	I	I			I	I	I
3.8. Obtain Sign-Off of Requirements																	
3.8.1. Conduct Requirements Review with Sponsors	R		A								C	C				I	C
3.8.2. Address Requirements Review Issues	R		A									I			I	I	

Figure 4 Example RACI Matrix

2. Problem and High-Level Solution Definition

2.1. Phase Goals

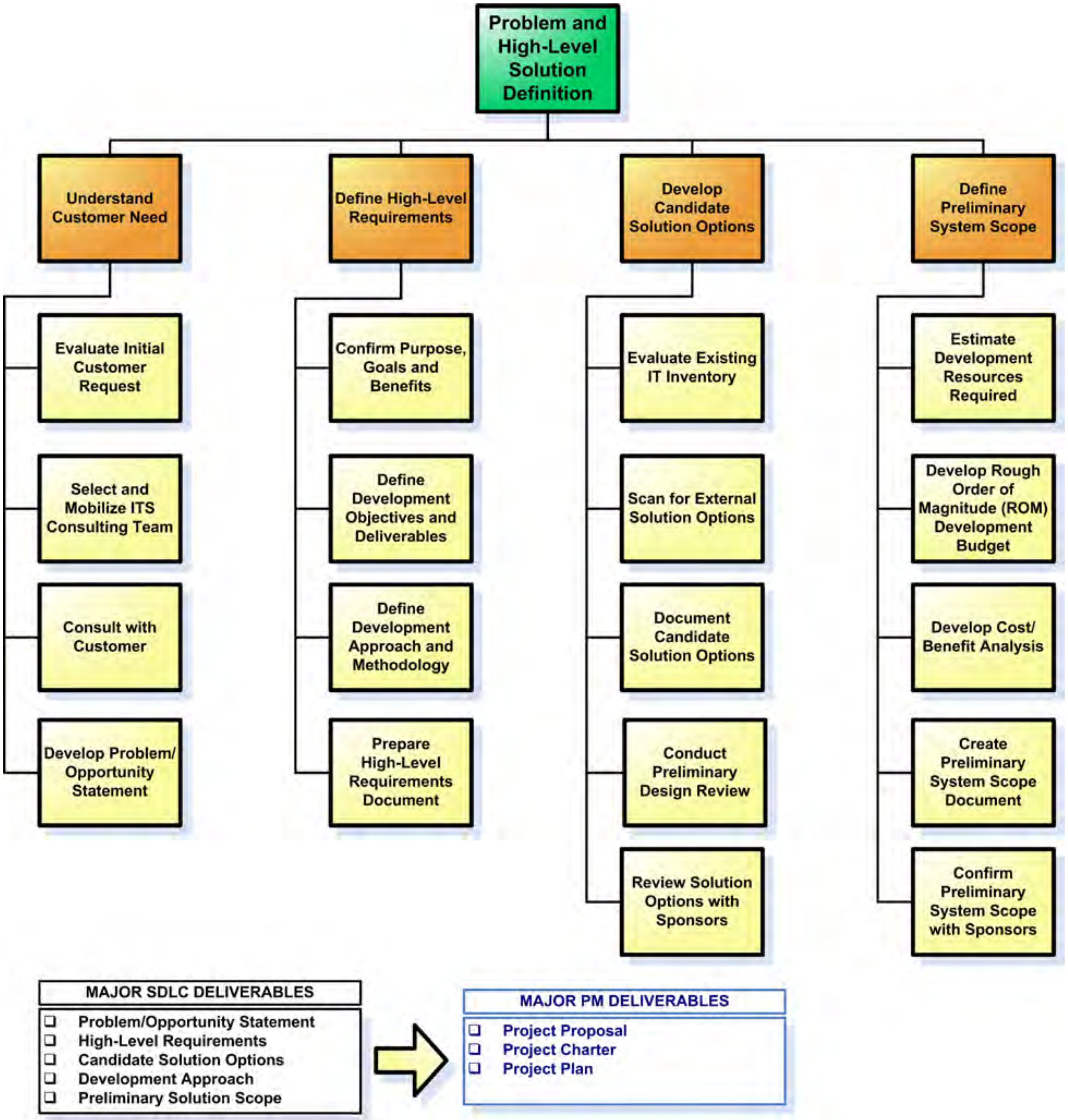
The goals of this phase are to:

- Understand the Customer Needs
- Develop the Problem/Opportunity Statement that confirm the problem to be solved or the opportunities to be enabled by an IT solution
- Define the High-Level Requirements
- Develop a high-level estimate of the solution development effort and costs for budget purposes
- Define and Scope a set of conceptual Candidate Solution Options to address the stated client needs
- Review and confirm validity of proposed solution(s) with all key project sponsors
- Provide the technical information for the development of a Project Proposal, Project Charter and a Project Plan using the Project Management Process

2.2. Phase Outputs

Stage	Deliverables/Work Products
Understand Customer Need	▪ Problem/Opportunity Statement
Define High-Level Requirements	▪ High Level Requirements ▪ Development Approach
Develop Candidate Solution Options	▪ Candidate Solution Options
Define Preliminary Solution Scope	▪ Preliminary Solution Scope

2.3. Phase WBS


Problem and High-Level Solution Definition
Understand Customer Need
Define High-Level Requirements
Develop Candidate Solution Options
Define Preliminary System Scope
Evaluate Initial Customer Request
Select and Mobilize ITS Consulting Team
Consult with Customer
Develop Problem/ Opportunity Statement
Confirm Purpose, Goals and Benefits
Define Development Objectives and Deliverables
Define Development Approach and Methodology
Prepare High-Level Requirements Document
Evaluate Existing IT Inventory
Scan for External Solution Options
Document Candidate Solution Options
Conduct Preliminary Design Review
Review Solution Options with Sponsors
Estimate Development Resources Required
Develop Rough Order of Magnitude (ROM) Development Budget
Develop Cost/ Benefit Analysis
Create Preliminary System Scope Document
Confirm Preliminary System Scope with Sponsors
MAJOR SDLC DELIVERABLES
❑ Problem/Opportunity Statement
❑ High-Level Requirements
❑ Candidate Solution Options
❑ Development Approach
❑ Preliminary Solution Scope
MAJOR PM DELIVERABLES
❑ Project Proposal
❑ Project Charter
❑ Project Plan

2009-05-05

2.4. Understand Customer Need

2.4.1 Evaluate Initial Customer Request

The purpose of this step is to understand the customer need and to be able to clearly express it as a Problem / Opportunity Statement that can be addressed by an IT Service. This Problem / Opportunity Statement is used with the Project Management Process to develop the ITS Project

Proposal. The execution of this step is the responsibility of the "Account Manager" for the client[3]. In most cases this will be either the designated Divisional Liaison (DL) or an IT Service Manager. If the request does not fit a specific DL or IT Service Manager it should be referred to the PMG (Portfolio Management Group).

The Account Manager should acknowledge the request with the customer and discuss the nature of the request within 5 working days of being contacted.

2.4.2 Select and Mobilize ITS Consulting Team

Once the Account Manager has spoken with the customer to gather initial information on the nature of the request they should evaluate what additional ITS expertise may be required to define and understand the problem/opportunity if any. Working with the Resource Managers (or their designates) the Account Manager should assemble an ITS consulting team that has the best combined expertise to discuss the customers problem or opportunity. The Account Manager should schedule a meeting with the client and the team and brief team beforehand on the situation. In most cases, this "ITS Consulting Team" will be an ad hoc team of 2-3 people drawn from Applications and Project Management, Core Technologies, Service Management or other Subject Matter Experts (SME). Under normal circumstances this is intended to be a very short phase of work completed within 1-2 weeks elapsed time at no cost to the client. Therefore the time commitment of the ITS team should be managed very carefully to allow the phase to be completed successfully for the client with the minimum consumption of resources. The Account Manager is responsible for managing the trade-offs of the quality of the problem/solution definition, the criticality of the client need, and the ITS effort expended.

2.4.3 Consult with Customer[4]

The IT team will meet with customers and interview them to better understand the problem/opportunity and the scope of possible solutions. This activity may consist of one or more meetings and interviews to:

- Understand the customer's situation and need
- Understand what solutions the customer may have already considered
- Understand the criticality and performance characteristics of the required solution
- Define the broad data, security and privacy needs of the solution in terms of confidentiality, integrity, and availability
- Identify any system interface needs
- Identify the user base, key customer SMEs and the candidate IT Project sponsor(s)
- Define the expected timeframe for the solution
- Identify any existing budget that the client has established

During these discussions with the customer, the team should also try to determine the required quality level of the final solution. Since no customer is going to expect or ask for a "less than quality solution", the answer to this question will have to be deduced from the customer's response to other questions about the longevity, availability, security and other desired aspects of the solution. Questions such as "What impact would there be on your operations is the system was unavailable?"

[3] A distinction is made throughout between "Customers" and the "Client". Clients use and receive services (in this case the IT Solution) and provide functional requirements to the ITS team. The "Customer" is the person who will be paying for the solution, has ultimate authority for the project, and is the business owner of the solution.

[4] For further guidance on the discussion with the customer see: https://collab.ucsc.edu/its-group-spaces/architecture/referance-library/reqs

are useful in this regard. The intent of this dialog is to gauge the feasibility of achieving the expected quality level within the timeframe and anticipated budget for the development effort.

2.4.4 Develop Problem/Opportunity Statement[5]

Based on the initial findings, the ITS team will construct a Problem/Opportunity statement and conduct some initial research of potential solutions.

2.5. Define High-Level Requirements

This stage defines and documents the high-level requirements discovered in the discussions with the customer in the previous stage and it represents a key integration point of the SDLC methodology with the ITS Project Management Methodology. Depending on the scale of the proposal or project, the information in the High-Level Requirements Document will either be incorporated into a Project Proposal or Project Charter or attached as an appendix or supporting document for the Project Charter.

If a Project Manager is assigned at this stage, they will be responsible for taking this input from the technical team and creating the project management deliverables. If no project manager has been assigned, the Account Manager would complete this stage of work and invoke the help of the PMG in initiating the PM Process.

2.5.1 Confirm Purpose, Goals and Benefits

The team should work closely with the client to document the clients' business rationale for the solution together with the goals and expected benefits. Both qualitative and quantitative benefits should be captured. This is a data gathering activity – not an approval. The Account Manager should conduct this analysis in a structured and objective way that best represents the client perspective. The necessary approvals will be obtained later through the IT Portfolio Management process with the involvement of the appropriate IT Governance entities.

This activity also contributes to the success of the project by providing a common understanding of the project and its goals between ITS and the client.

2.5.2 Define Development Objectives and Deliverables

The specific development objectives and the deliverables required to satisfy the objectives should be documented. Where possible, objectives should be measurable.

2.5.3 Define Development Approach and Methodology

Using the SDLC methodology and drawing from the experience of the ITS team, strategies for developing the solution should be identified, evaluated and discussed with the client. The results of this exercise may be captured in an initial Work Breakdown Structure (WBS), an approach schematic or a relevant extract from the SDLC methodology. This output will be used to develop a project plan as part of the PM methodology.

2.5.4 Prepare High Level Requirements Document

The work of the team to date is then documented and published as a high-level Requirements Document to support the Project Charter or Proposal. For smaller projects this information may simply be included in the Project Charter or Proposal.

[5] A Problem/Opportunity Statement should be documented using the Project Proposal word template. This template can be found at http://its.ucsc.edu/pmo/project_templates.php

2.6. Develop Candidate Solution Options

2.6.1. Evaluate Existing IT Inventory

In a significant number of cases, the client needs can be addressed by extending or refining an existing IT Service or solution. To avoid "re-inventing the wheel" the team should evaluate whether any existing IT service or solution could be used as a basis for a solution.

2.6.2. Scan for External Solution Options

If the client has already identified external solution options, these should be evaluated in this activity; otherwise the team should scan for off-the-shelf solutions (commercial or open source).

2.6.3. Document Candidate Solution Options

The pros and cons of alternate solution options are summarized and compared by the team. The team should also prepare a recommendation for the client on the best solution to meet the requirements and the client budget (if known).

2.6.4. Conduct Preliminary Design Review

It is strongly recommended that the project team confer with the Design Review Board (DRB) when considering candidate solution options. The intent at this stage is to use the DRB in a consultative capacity and not to conduct a formal Design Review

2.6.5. Review Solution Options with Sponsors

A formal review of the solution options and recommendations should be held with the user Sponsor and ITS sponsors. Based on this review, further refinement of approaches or options may be done or a particular solution will be selected.

2.7. Conduct Preliminary System Scope

The goal of this stage is to define the scope of the potential solution in enough detail to enable the client and ITS management to make a go/no go decision on whether to commit to a full ITS project. This is a "pre-feasibility" analysis based on the limited information that may have been collected up to this point. This analysis requires the right judgment to create an estimate with a sufficient level of accuracy for the decision to be made. Therefore it is recommended that all estimates developed during this stage should be qualified as broad ranges and assumptions clearly articulated.

2.7.1. Estimate Development Resources Required

Development resource effort should include ITS staff effort hours or FTE (Full Time Equivalents) by role for the duration of the project.

2.7.2. Develop Rough Order of Magnitude (ROM) Development Budget

The estimated development effort should be used to create a financial budget for the project. Any additional costs to establish the team and the development environment should also be included in this budget estimate (SW licenses, specific development tools, and equipment, vendor and consultant costs). It is also good practice to estimate the end user effort required, though this cost may not be included in the final project budget.

2.7.3. Develop Cost/Benefit Analysis

The Account Manager should work with the client to develop a cost/benefit analysis or "value proposition" for the project using the benefits identified by the client and the ROM development budget value. Established financial analysis techniques such as discounted cash flow analysis may be used to calculate a return on investment (ROI), Net Present Value (NPV) or Payback Period for the project[6].

2.7.4. Create Preliminary Solution Scope Document

The conceptual scope of the solution and estimated cost and benefit should be summarized and published in this activity. In most cases this information would be included and published as part of a project charter. If the project is small the scope may be summarized in the Project Proposal. If the project and the proposed solution is large an complex it may be necessary to create a standalone Solution Scope document that is cross-referenced by the Project Charter. For clarity, it is recommended that the solution scope be graphically depicted via a Conceptual Systems Design diagram or a Context Diagram in addition to a textual description of the desired functionality. These diagrams should also indicate key system boundaries, interfaces and data feeds.

2.7.5. Confirm Preliminary Solution Scope with Sponsors

The Preliminary Solution Scope should be review and confirmed with all key project sponsors. The Preliminary Solution Scope Document should be updated with any review feedback and re-published if necessary.

Finally, the information created in this phase should be incorporated into the Project Management process as input to the Project Charter or Project Proposal. Once this is accomplished, this signifies the end of the Problem and High-Level Solution Definition Phase.

[6] Engineering Economy by William G. Sullivan, Elin M. Wicks, C. Patrick Koelling, Prentice Hall 2008.

3. Requirements Definition

3.1. Phase Goals

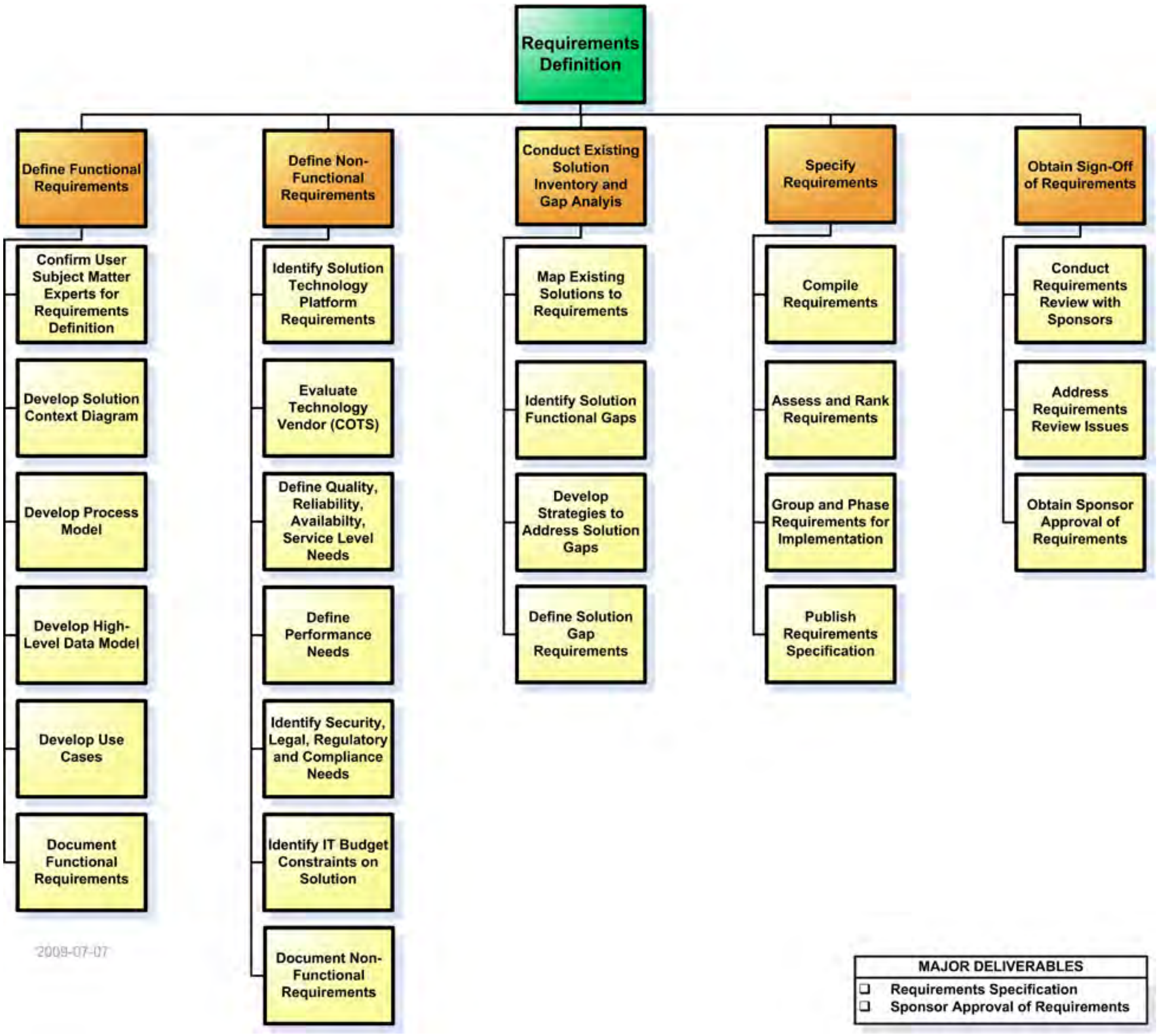
The goals of this phase are to:

- Specify the detailed formal requirements for a system to address the problem or opportunity identified by the customer
- Verify the work of the Problem/Solution Definition Phase (Candidate Solution Options, Preliminary System Scope) and if significant gaps exist revisit the prior phase

3.2. Phase Outputs

Stage	**Deliverables/Work Products**
Analyze Functional Requirements	▪ Functional Requirements
Analyze Non-Functional Requirements	▪ Non-Functional Requirements
Conduct Existing Solution Inventory and Gap Analysis	▪ Inventory Gap Analysis
Specify Requirements	▪ Requirements Specification
Obtain Sign-Off of Requirements	▪ Sponsor Approval of Requirements

3.3. Phase WBS



Requirements Definition
Define Functional Requirements
Confirm User Subject Matter Experts for Requirements Definition
Develop Solution Context Diagram
Develop Process Model
Develop High-Level Data Model
Develop Use Cases
Document Functional Requirements
Define Non-Functional Requirements
Identify Solution Technology Platform Requirements
Evaluate Technology Vendor (COTS)
Define Quality, Reliabilty, Availabilty, Service Level Needs
Define Performance Needs
Identify Security, Legal, Regulatory and Compliance Needs
Identify IT Budget Constraints on Solution
Document Non-Functional Requirements
Conduct Existing Solution Inventory and Gap Analyis
Map Existing Solutions to Requirements
Identify Solution Functional Gaps
Develop Strategies to Address Solution Gaps
Define Solution Gap Requirements
Specify Requirements
Compile Requirements
Assess and Rank Requirements
Group and Phase Requirements for Implementation
Publish Requirements Specification
Obtain Sign-Off of Requirements
Conduct Requirements Review with Sponsors
Address Requirements Review Issues
Obtain Sponsor Approval of Requirements
2009-07-07
MAJOR DELIVERABLES
❑ Requirements Specification
❑ Sponsor Approval of Requirements

3.4. Define Functional Requirements

Requirements Definition is an art and a science. It requires analysts to work very closely with customers and to draw on their communication and technical skills to surface underlying business needs that might be addressed by a system solution. One of the major pitfalls is to "leap to a solution" with an inadequate understanding of the operating problems and fundamental needs of the customer. Requirements analysis is also by nature an explorative and iterative process. Frequently, customers cannot adequately state what they really need until they see what they have asked for in previous requirements iterations with an IT team. As stated earlier in the introduction to this handbook, an iterative project approach may need to be used in which repeated sequences of Requirements Definition, System Design and System Build phases will progressively define and build the solution in iterative or spiral approach.

When developing a project specific work plan, it should be noted that the Work Breakdown Structure for this phase does not necessarily reflect the sequence of how the stages and activities should be performed on a given project. Due to the iterative and exploratory nature of the Requirements Definition Phase it is recommended that stages 1-4 (Define Functional Requirements, Define Non-Functional Requirements, and Conduct Existing Solution Inventory and Gap Analysis) be overlapped and executed concurrently with a finish-to-finish dependency on the publication of the Requirements Specification.

Solution Requirements may be divided into "Functional" and "Non-Functional" requirements. Functional requirements specify what the solution must do. Non-Functional requirement refers to additional Technical and Operational and Performance requirements for the solution to be implemented and supported, but do not by themselves characterize the underlying business problem or opportunity to be solved. In order to clearly understand the customer's problem domain before devising a solution, functional requirements are defined in a way that makes them independent of a particular technology implementation. They describe the capabilities and functionality of the required solution, not specific software features. For example, a Functional requirement might be: "The system shall accept a student registration for a class" or "The system shall automatically route the approval of a purchase requisitions to a pre-designated signing authority for a particular unit".

Requirements should also be prioritized during the analysis process. This is typically achieved by categorizing and weighting each requirement with customer input and the cost-benefit ratio of the requirement. For example:

Requirement Priority	**Description**
Must Have	A "show stopper" requirements, without which the solution is of little use or benefit
Preferred	Demonstrable efficiency, cost savings, or income producing requirement
Optional	Intangible or small benefits only; cost >= benefit)
Cancel	Too complex or costly, or no value add

IT solutions for a particular customer problem or opportunity will have three major aspects that need to be considered during analysis: Process, Data and Behavior. Process refers to the work activity and steps that the solution needs to support. Data refers to the information and facts to enable the process. Behavior relates to the animation of the process and the interaction with the actors in the process. These aspects exist and are intertwined in every IT solution, but their weight will depend on the specific nature of the customer need. For example, a Faculty Ladder Ranking process, a Student Registration process, a facility monitoring process and a Data Mart or Decision Support System will all vary in the relative degree of data, process and behavior involved. In considering the activities in this phase the analyst must use their best judgment to build a project specific approach that addresses this variation in the problem domain.

A final consideration for the analyst is to select the data gathering approach that will be used to capture requirements. Customers may be interviewed individually, in small groups, in facilitated work sessions or some combination. The exact approach and sequencing of work needs to be determined and agreed to with the client so as to minimize disruption of their daily operations but maximize involvement of key stakeholders and subject matter experts. Since the Solution Context Diagram provides a boundary and baseline for the analysis effort, it is strongly recommended that this be developed in a facilitated work session with the broadest group of customer representatives and the Project Sponsor present.

3.4.1 Confirm User Subject Matter Experts for Requirements Definition

The first crucial step in defining requirements is identifying those persons who understand the problem domain can speak with some authority on the needs of the client and the customer constituency. In cases where a solution will serve a large and varied number of customers it will be necessary to work with client leadership to determine a subset of customers who can adequately represent the larger set of customers who will use the IT solution.

3.4.2 Develop Solution Context Diagram

A graphical presentation of the interfaces and boundaries of the problem area or process to be analyzed should be developed. The context diagram must show all known and relevant external agents or actors (organizations, systems, roles, external processes etc.) and the major data flows in and out of the area to exchange information or to respond to events.

3.4.3 Develop Process Model

Excellent treatises have been written to describe the development of process models[7] and a full description is beyond the scope of this handbook. The purpose of developing a process model is to understand the work being done and the roles of the various actors in the process. In those cases where the process flow aspect of the problem domain is significant, Functional Decomposition Diagrams and Cross-Functional Workflow models should be developed with the customer subject matter experts. In simpler cases a Functional Decomposition Diagram may be sufficient to partition, decompose, and describe the activities of the area under study. The scope of the potential IT solution can be determined through examination of the current state process model and the future state or "To Be" process model that the client anticipates the new IT solution will support.

3.4.4 Develop High-Level Data Model

A conceptual level entity relationship diagram (ERD) should be created to define the data needs of the customer and major business rules. This should include major fundamental and associative entity types, their major attributes and relationships. This should be a conceptual level ERD that is independent of a specific physical database implementation. Its primary purpose is to enable validation of business data needs, terms and business rules and to provide input to the designers and data base administrators who will build more detailed logical and physical data models for implementation. If the current situation involves an existing IT solution then existing conceptual data models can be reused or reverse engineered from the solution. The data model can also be used to capture new business rules and data needs for the envisioned solution.

3.4.5 Develop Use Cases

Use Cases provide a transition into system design from the process model to the human-system interaction that will be supported by the IT solution. High-level "business" Use Cases should be defined that characterize the major interactions between the users and the solution. If a cross-functional workflow model has been developed, candidate Use Cases can be identified from those points at which the customer roles interact with the system to perform a task. Judgment must be applied to ensure that the Use Cases can be validated directly with the customer and yet form a basis for the development of more detailed Use Cases by the Application Developers in the System Design Phase.

3.4.6 Document Functional Requirements

Functional requirements will be uncovered as a result of the preceding activities in this stage, through a combination of conversations with the customers and by analyzing the context model, process models, data model and use cases. During this activity, the Business Analyst simply documents these

[7] Workflow Modeling: Tools for Process Improvement and Application Development, Alec Sharp and Patrick McDermott, Artech House, 2001

Functional Requirements in preparation for more formal requirements specification, grouping and ranking later in the stage.

3.5. Define Non-Functional Requirements

Non-Functional requirements encompass all the qualities that the solution must have for it to be usable, fast, reliable, secure, and attractive to users[8]. During this stage the team will identify, group and prioritize this wide range of requirement types.

3.5.1. Identify Solution Technology Platform Requirements

If known at this stage, the technology platform required to support the solution should be specified. It is anticipated that in many cases the solution will be an extension of an existing IT Service and so the technology platform requirements will already be known. Prevailing IT Architecture standards should also be considered during this activity.

3.5.2. Evaluate Technology Vendor (COTS)

If a **Commercial Off-The-Shelf (COTS)** solution is being evaluated, its technology platform needs to be evaluated against functional requirements in this activity and the IT Architecture. In addition, the financial viability, and qualifications (references, etc.) of the vendor should also be evaluated.

3.5.3. Define Quality, Reliability, Availability, Service Level Needs

The team should document the expectations of the solution in terms of system quality, reliability, availability windows, and the broad IT Service level needs.

3.5.4. Define Performance Needs

The team should capture and estimate any system performance and volumetric information (response time, transaction volumes and cycles, concurrent users, dataset size and growth rate, archival needs, etc) that can be used to define the performance needs and metrics for the solution.

3.5.5. Identify Security, Legal, Regulatory and Compliance Needs

Any security, legal, regulatory and compliance needs should be documented and verified with the client and checked against Federal, State, UC and UCSC IT compliance policies. The project team should refer to the Security Design Package for guidance on policies that will guide classification of the system and data, and thus identify security requirements that must be met within the system design.

3.5.6. Identify IT Budget Constraints on Solution

If there are any existing budget constraints on the solution these should be identified in this activity to guide decision making and the generation of solution options. The team should also evaluate the feasibility of meeting the solution requirements against the financial constraints and expectations of the client.

[8] Mastering the Requirements Process, Suzanne Robertson and James Robertson, Addison-Wesley, 2006

3.4.7 Document Non-Functional Requirements

The Non-Functional requirements uncovered in this stage are documented by the Business Analyst.

3.6. Conduct Existing Solution Inventory and Gap Analysis

3.6.1. Map Existing Solutions to Requirements

The functional and non-functional requirements should be mapped against existing IT services and solutions to determine if an existing solution may be reused, partially or completely, to serve the client needs.

3.6.2. Identify Solution Functional Gaps

If an existing solution is found that partially addresses the requirements, the shortcomings or gaps in the solution should be noted and weighted for importance.

3.6.3. Develop Strategies to Address Solution Gaps

In this activity the team will develop strategies and approaches to address the gaps in functionality between the requirements and the existing solution. These strategies may include 3rd party "bolt-ons" or additional modules for software package solutions, use of other existing solution components or custom development and interface development.

3.6.4. Define Solution Gap Requirements

The solution functional gaps and the strategies to address them should be refined into a set of solution gap requirement statements to be included in the final Requirements Specification document.

3.7. Specify Requirements

This stage gathers all the functional and non-functional requirements discovered during the Requirements Definition Phase and assembles them into a ranked and phased set of requirements. This information is then published as the Requirements Specification deliverable for review and sign-off by the client.

3.7.1. Compile Requirements

The number of requirements captured, even in a medium sized project can be burdensome to manage and evaluate with the client team. To aid in this process, the requirements should be specified in a uniform, structured way for ease of management and evaluation. In order to facilitate this effort a requirements template has been developed based on the Volere Requirements Specification Template from the Atlantic Systems Guild[9].

3.7.2. Assess and Rank Requirements

The set of requirements should be evaluated and ranked in terms of user priority and criticality.

[9] See the ITS SDLC Repository.

3.7.3. Group and Phase Requirements for Implementation

Some requirements will have logical dependencies and affinity with each other. The team should use this information, together with the requirement ranking information to group and phase the requirements for implementation sequence and perform a trade-off analysis when considering the scope of the initial solution to be implemented. At this stage, all known requirements should be captured, grouped and phased – even if they are beyond the current planned scope for the system. This is important since some cases a desired requirement in a later planned phase may influence the design of the earlier system designs.

3.7.4. Publish Requirements Specification

In this activity, the Requirements Specification published and distributed to the client and key stakeholders for review. Document version control is important in this and later stages as feedback is received from multiple stakeholders and corrections are made. Therefore it is recommended that a single team member be assigned to manage the version control for the Requirements Specification document and recording of the approval process for agreed changes.

3.8. Obtain Sign-Off of Requirements

3.8.1. Conduct Requirements Review with Sponsors

A formal review meeting should be held with the client, key stakeholders and sponsors to capture feedback on the Requirements Specification deliverable.

3.8.2. Address Requirements Review Issues

Issues raised in the Requirements Review should be captured in the Project Issue log by the Project Manager and systematically addressed by the team. To avoid gridlock and ‘analysis paralysis”, the issues should be ranked and weighted in relationship to the prioritized set requirements. The project team and sponsor should aim to resolve all major issues before moving to the next phase. Minor issues may be deferred and addressed in solution release planning.

3.8.3. Obtain Sponsor Approval of Requirements

Formal written approval of the Requirements Specification should be obtained from the client, ITS and executive sponsors.

4. System Design

4.1. Phase Goals

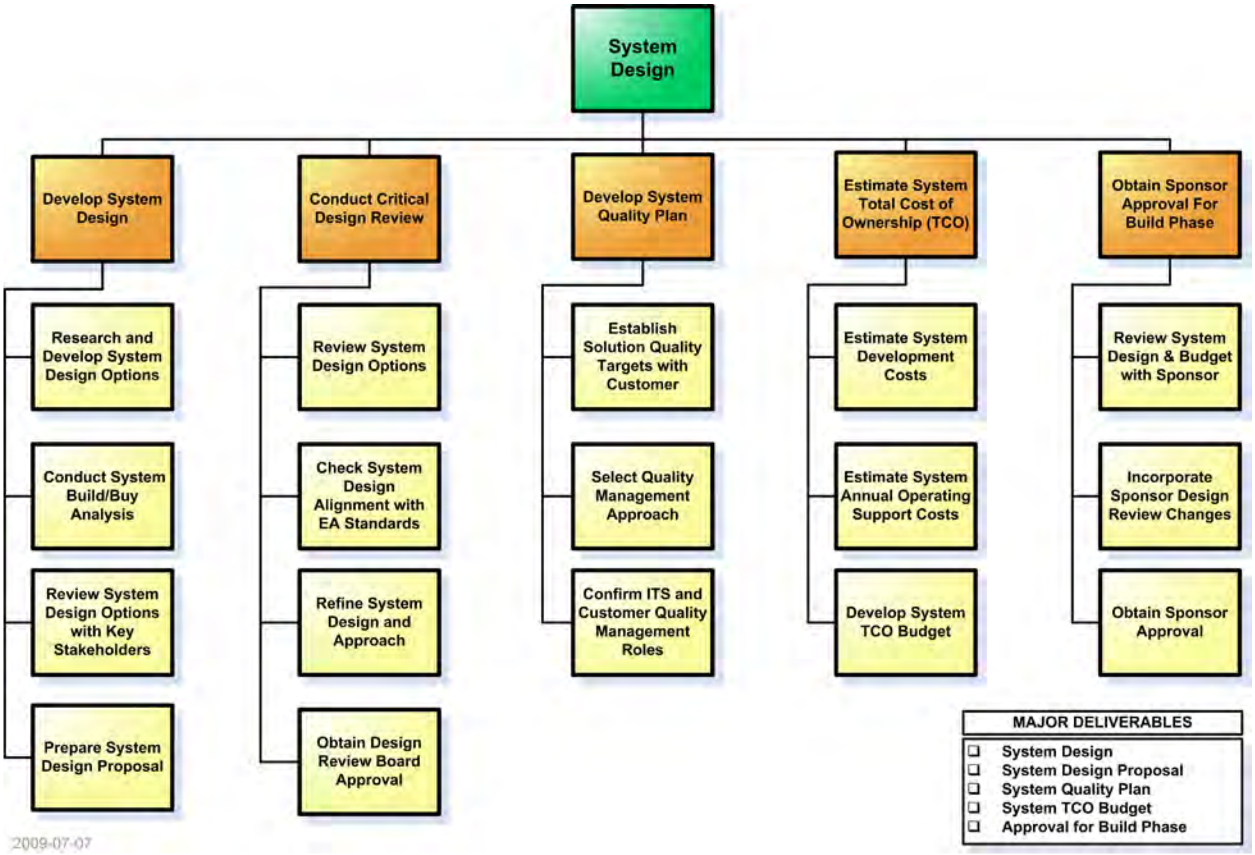
The goals of this phase are to:

- Create the System Design and a System Design Proposal for the Customer
- Ensure that the IT System design Review Package complies with prevailing ITS architecture and standards
- Obtain approval of the System Design from Sponsors, Stakeholders and Clients

4.2. Phase Outputs

Stage	**Deliverables/Work Products**
Develop System Design	▪ System Design ▪ Security Design Package ▪ System Design Proposal
Conduct Critical Design Review	▪ Critical Design Review Package
Develop System Quality Plan	▪ System Quality Plan
Estimate Total Cost of Ownership (TCO) for System	▪ System TCO Budget
Obtain Sponsor Approval For Build Phase	▪ Sponsor Approval

4.3. Phase WBS


System Design
Develop System Design
Conduct Critical Design Review
Develop System Quality Plan
Estimate System Total Cost of Ownership (TCO)
Obtain Sponsor Approval For Build Phase
Research and Develop System Design Options
Conduct System Build/Buy Analysis
Review System Design Options with Key Stakeholders
Prepare System Design Proposal
Review System Design Options
Check System Design Alignment with EA Standards
Refine System Design and Approach
Obtain Design Review Board Approval
Establish Solution Quality Targets with Customer
Select Quality Management Approach
Confirm ITS and Customer Quality Management Roles
Estimate System Development Costs
Estimate System Annual Operating Support Costs
Develop System TCO Budget
Review System Design & Budget with Sponsor
Incorporate Sponsor Design Review Changes
Obtain Sponsor Approval
MAJOR DELIVERABLES
□ System Design
□ System Design Proposal
□ System Quality Plan
□ System TCO Budget
□ Approval for Build Phase
2009-07-07

4.4. Develop System Design

System Design is by nature an iterative process and often multiple viable designs will emerge that can meet the stated system requirements. Sometimes the project team members may have quite disparate opinions on the right design. It is therefore important that the project team reach consensus on the design options they wish to propose, before the team presents its design to the Design Review Board. If no consensus can be reached, it is the responsibility of the Project Manager to summarize the alternate design arguments by the project team members and work with the DRB to resolve the issue.

4.4.1 Research and Develop System Design Options

This activity creates the System Design which in turn is the basis for a System Design Proposal that will be presented to the customer. The System Design is a technical IT deliverable. The System Design Proposal is a decision document that summarizes the design from a customer perspective.

The effort in this activity is the culmination of several preceding activities to define the Problem/Opportunity, a high-level solution(s), functional, and non-functional requirements. The requirements identified during the **Requirements Definition** phase are examined and transformed into a physical System Design and evaluated for their efficacy in addressing the customer needs. If a software package is being considered, this activity should be conducted concurrently with the activity “Conduct Solution Build/Buy Analysis” to confirm the viability of using a Commercial Off The

Shelf (COTS) solution or open source approach versus a custom programming approach to create the solution.

The solution design activity can vary greatly depending on whether the solution will be custom developed in a programming language, implemented using COTS software, or some combination of both. For example, more detailed procedural logic, specifications and design guidance will be needed in a custom development scenario as opposed to a software package configuration scenario. Given the large variety of development options currently available at ITS, it is beyond the scope of this methodology to prescribe a specific design approach. However, irrespective of the particular technology employed to build the solution, a number of transformations must occur in this activity from logical requirements to physical design:

Requirements Source	**to**	**Physical Solution Design**
Use Cases and Process Model	➔	Program procedural logic or COTS Module mapping/configuration
Data Flows	➔	Reports and Screen Designs
Conceptual Data Model	➔	Physical Database Design Form Design Report and Screen Design
Triggers/Events	➔	System Navigation and Behavior Design
Non-Functional Requirements	➔	User Interface Design Software Services and Middleware Design Hardware and Operating System Design Security Controls and Compliance Design Storage and Archiving Design Network Design Interface Design BC/DR[10] Compliance (Adjustments to HW, System configuration, etc)
Users/Actors/Roles	➔	Required User groups, roles, access rights and privileges

This is not a purely technical exercise since cost constraints and other non-technical factors will have a bearing on the solution design. This activity should also create a set of alternate designs or design variations. These Solution Options should be evaluated against the customer requirements and the project team should prepare their specific recommendation on the best choice of design.

Finally, the team should work with IT Service management to draft the IT service definition that the solution would support.

[10] BC/DR Business Continuity/Disaster Recovery planning are processes that help organizations prepare for disruptive events and how they will continue to operate should such an event occur. In this context the project team should conduct a business impact analysis to examine what would happen if a disruptive event made the system unavailable. This insight is then used to adjust specific aspects and parameter of the physical design to address these needs e.g. System recovery, etc.,

4.4.2 Conduct System Build/Buy Analysis

The goal of this activity is to confirm whether the solution is best addressed through a COTS system or a custom built system. Although at first glance a custom solution may be appealing, there are enormous total cost of ownership, quality and maintenance advantages of acquiring a commercially available or shared source solution, as well as speed to solution advantages. This activity ensures that these options are diligently explored.

Note: If the decision is made in this activity to pursue a COTS solution, the work breakdown structure should be adjusted to describe the appropriate purchasing process as this is not included in the SDLC methodology.

Note: Project teams must consult the Data Center Hosting Advisory Council (DCHAC) before acquiring technology and establishing the development environment (IDE etc).

4.4.3 Review System Design Options with Key Stakeholders

The project team should prepare a summary of the proposed design options and review them with the key project stakeholders. The focus of this review is on how well the designs address the requirements. Although some of the Stakeholders may be ITS functional leaders – this is not an IT technology review. With this purpose in mind, the team should prepare a conceptual summary of the design that is suitable for users to understand the nature of the solution, how it will work (impact on their operations etc), and how it will meet their needs. The team should then modify the design based on the feedback gained through this Stakeholder review.

4.4.4 Prepare Critical Design Review Package

The final System Design is documented and packaged for submission to the ITS Design Review Board (DRB). The project manager is responsible for requesting a Critical Design Review with the ITS Design Review Board (DRB).

The charge of the Design Review Board (DRB) is to:

- Provide a broad base of subject matter expertise
- Identify, evaluate, and make recommendations on technology architecture standards, principles, and guidelines that are submitted for periodic review
- Review the Security Design Package and incorporate applicable policies and guidelines for designing the security model of the system
- Critique project specific proposal designs for consistency with enterprise architectures; advise on any requests for exception; identify potential design flaws; make recommendations for necessary corrections; sign off on final designs

4.5. Conduct Critical Design Review

4.5.1 Review System Design Options

The Solution Design and Design Options are reviewed for technical quality. In preparation for this activity the Application Developer and Business Analyst should consult with the ITS Enterprise Architects to create a Design Review Checklist. This checklist is tailored for the specific project from a master Design Review Checklist maintained by the Enterprise Architecture team. The solution is then evaluated against this checklist by the project team and the appropriate responses are prepared for Design Review Board approval[11].

4.5.2 Check System Design Alignment with EA Standards

The solution is evaluated against the current ITS Enterprise Architecture Standards with the Enterprise Architects. Any exceptions are noted by the team. This activity also provides an opportunity for the evolution of EA standards and the adoption of the System Design as "Solution or Architectural Building Block" that may be reused by other developers and designers.

4.5.3 Refine System Design and Approach

Based on the review of the design checklist and alignment with EA standards, the System Design may be modified or refined in this activity prior to submission for Design Review Board approval.

Also depending on the nature of the solution, the project team should confirm the best design and development approach for creating the solution. This would include consideration of Application Prototyping versus a traditional development approach.

4.5.4 Obtain Design Review Board Approval

The Project Manager is responsible for requesting a Design Review but the Design Review Board is responsible for scheduling and holding the Design Review Board (DRB) Meeting and granting approval of the solution design. Design Review Board meetings are conducted with a formal agenda with explicit objectives, content coverage, and specific actions for follow-up by the project team. EA Compliance and Technology Feasibility can be an iterative process. Depending on the complexity of the solution, more than one DRB meeting may be held before sign-off to proceed to solution design proposal, and Build (pending outcome of proposal approval).

A typical Design Review Board meeting agenda will include:

- Minutes of Preview DRB meeting
- Requests for Change
- Architecture Compliance Review (scrutiny of the compliance of a specific project against established architectural criteria, spirit, and business objectives.)
- Dispensations (The dispensation is the mechanism used to request a change to the existing architectures, principles, and standards outside of normal operating parameters).
- Dispute Resolution (Disputes that have not been resolved through the Architecture Compliance and dispensation processes are identified for further action)
- Actions Assigned (Reference, Priority, Action description, Action owner etc.)
- Contract Documentation Management (Formal acceptance of updates and changes to architecture documentation for onward publication as versioned Adobe Acrobat PDF files)
- Any Other Business (Description of issues not directly covered under any of the above.) These may not be described in the agenda but should be raised at the beginning of the meeting. Any supporting documentation must be managed as per all architecture governance documentation.

[11] See ITS Document "Design Review Process" V1.3 for more details and the master copy of the design review checklist.

4.6. Develop System Quality Plan

It should be noted that although this activity indicates the development of a specific quality plan for the project, it is our intent to develop a standard ITS Quality Plan deliverable and template that will be adapted by various project teams to their specific circumstances.

Defining the quality of a delivered IT Solution can be elusive, but regardless of the exact definition of quality, quality and satisfaction are determined ultimately by the customer's perception of the total product's value or service. If we define quality from a customer's perspective, quality is much more than simply conformance to specifications. A flawlessly designed, defect-free solution that does not meet the customer's needs cannot be considered to be of high quality. A high quality IT solution is one that conforms to customer requirements as well as technical specifications.

4.6.1 Establish Quality Targets with Customer

The final quality of an IT System cannot be defined independently of the cost and timeframe to develop it. Ultimately the delivered system quality is a trade-off between these variables. As much as it is possible to do so, this trade-off must be made explicitly with the full involvement of the customer. At times customer expectations around cost or time to solution may be misaligned with their expectation of the quality of the system. However, even in these circumstances it is the responsibility of the project manager to bring these expectations into alignment and establish realistic targets and expectations through an open and objective dialog with the client.

An explicit mapping of customer requirements to the defined solution should be established as a baseline quality target for the system and for User Acceptance Testing in later phases of the project. Traceability of requirements to design is a critical success factor in ensuring that customer expectations are managed and met. As described in the **Requirements Definition** phase, customer requirements should be ranked and prioritized. The Project Manager should also agree with the Project Sponsor who, from a user perspective, will have final authority to sign-off on the quality of the solution.

In addition to the baseline quality targets around requirements, the Project Manager should also discuss the following expected measures of quality with the customer:

- Robustness of the solution in terms of acceptable uptime and availability
- Solution warranty (Break/Fix expectations)
- Special security needs
- Flexibility of the solution in terms of its ability to scale and adapt to future needs

This can be one of the most important conversations with the customer in terms of setting expectations against budget and timeline for the solution. Finally, it should be remembered that the quality of the customer experience in working with ITS is a significant factor in perceived solution quality

4.6.2 Select Quality Management Approach

The project team should determine an appropriate quality management approach for the development project. Although inspections such as the Design Review can identify quality issues, consistent development quality comes from the application of a consistent, repeatable process that is continuously improved through experience. This is one of the primary goals of the SDLC described here. An unstable process produces unstable products. The methodology work breakdown structure provides a starting point for a Quality Management Approach, but this must be tailored to the quality targets set with the customer.

Depending on the scale and complexity of the project, the following items should be considered in constructing the Quality Management Approach:

- Structure and frequency of user reviews and sign-offs
- Formal Requirements Mapping
- Rigor of unit and system test plans
- Frequency of System Builds
- Stress testing
- Use of automated test tools
- Methods and processes for migration of software between the Development, Test, Stage, and Production environments.
- Formal User Acceptance Testing with pre-defined UAT Scripts and test cases
- System and internal control audits
- Independent or external solution reviews

The quality management approach should also include decision criteria and subsequent actions that should be agreed to in advance for User Acceptance Testing (UAT). As a guide, if client/customer expectations are unmet during acceptance testing, the client must:

- Accept deficiency with no further expectation to meet; or
- Reject deficiency and therefore ITS must fix (cycle back through appropriate SDLC activities); or
- Transfer the deficiency to another phase or another project.

4.6.3 Confirm ITS and Customer Quality Management Roles

Once a Quality Management approach has been determined, the Project Manager should confirm the roles that ITS and users should play in the quality management process. In some cases, this may require a fairly significant commitment of time by users and sponsors. In other cases, additional resources and tools may need to be acquired. These arrangements should be agreed to in advance and incorporated into the Project Charter and budget by the Project Manager.

4.7. Estimate System Total Cost of Ownership (TCO)

The goal of this stage is to create a comprehensive cost profile for the system that the customer and ITS management can evaluate against the expected benefits and make a sound investment decision. Total Cost of Ownership means that the team must not only consider the cost of acquiring and building the system but also the ongoing costs to operate and support the system.

4.7.1 Estimate System Development Costs

These costs should include all expenditures from the Design phase through to Solution Deployment. Typical costs would include:

Software:

- Software license purchase costs
- Software license tracking costs
- Upgrades and User Scaling Costs
- Testing costs
- Software maintenance costs

Hardware

- Server hardware and software costs
- Network hardware and software costs
- End-user hardware and peripheral purchase costs

- Infrastructure (floor space) costs
- HVAC Costs
- Hardware warranties and maintenance costs
- Operations Infrastructure Costs
- Consumables and supplies

Personnel

- IT Personnel costs
- Technology training costs of users and IT staff.
- Consultants and Contractor Costs
- Management and User Time costs

Life Cycle Management Costs

- Deployment costs
- Decommissioning costs
- Replacement costs
- Migration costs
- Audit and Controls Costs

4.7.2 Estimate System Annual Operating Support Costs

In collaboration with ITS operations and other advisors (vendors, consultants), the team should estimate the annual support costs for the solution including:

- IT Service Management (Life Cycle Management)
- Incident Management
- Change Management
- Failure or outage costs
- Backup and Recovery Process costs
- Hosting Costs

4.7.3 Develop System TCO Budget

The estimated costs should be assembled into a TCO budget for review by ITS and the Project Sponsor.

4.8. Obtain Sponsor Approval For Build Phase

4.8.1 Review System Design & Budget with Sponsor

Formal approval to proceed should be obtained from the project sponsor using the appropriate IT governance process. If the Project Sponsor considers that all the solution proposals are too costly, the project team must cycle back to requirements definition and re-prioritize the requirements so that they can be brought in line with customer expectations on costs.

4.8.2 Incorporate Sponsor Design Review Changes

Any design changes as a result of Sponsor review my be incorporated into the budget before final approval

4.8.3 Obtain Sponsor Approval

The Project Manager should obtain final approval from the Sponsor to proceed with the Build Phase of the project.

Sponsor Approval documentation includes concurrence on changes to budget and requirements:

Accept - changes are accepted and can be acted upon.

Transfer - changes are accepted but transferred to another project, or deferred to subsequent phases, or...

Reject - changes are not acceptable and must return to prior SDLC phase (re-design, requirements, etc.)

5. System Build

5.1. Phase Goals

The goals of this phase are to:

- Build, configure and test the System
- Gain user acceptance that the developed system meets their need
- Finalize the IT Service Definition for the system (OLA and or/SLA)

The major challenge of describing the System Build Phase is to provide sufficient guidance to a particular development team on the set of tasks and activities required to conduct the System Build Phase using their particular software platform or technology set. Of necessity, the System Build Phase must encompass a wide range of technologies and software for developing an IT Solution for the customer. These can range from installation and configuration of a commercial software package such as PeopleSoft ERP; use of an open source software platform like Moodle; use of a SaaS[12] solution; and use of databases, scripting and programming languages (MySQL, Oracle, PERL, PHP, Java, C++, .Net etc.) to develop "custom" applications, or a combination of any of these methods. The system in question may also involve the configuration of a set of hardware components, such as a firewall or server or any combination of hardware and software elements to meet the requirements specified in earlier phases.

In spite of this diverse range of solution types, we feel it is still useful to provide some level of common work breakdown structure that can be used by project teams as a starting point for building their specific project plans and provide a set of common milestones and deliverables that can aid the project management process and the management of customer expectations. Therefore, the stage and activity descriptions for this phase have been deliberately written at level of abstraction that could apply to a wide range of IT development technologies. It is assumed that this guidance is no substitute for the knowledge of the development team, and that the developers will work with the Project Manager to develop a more detailed WBS and work plan that uses their experience with a specific software platform.

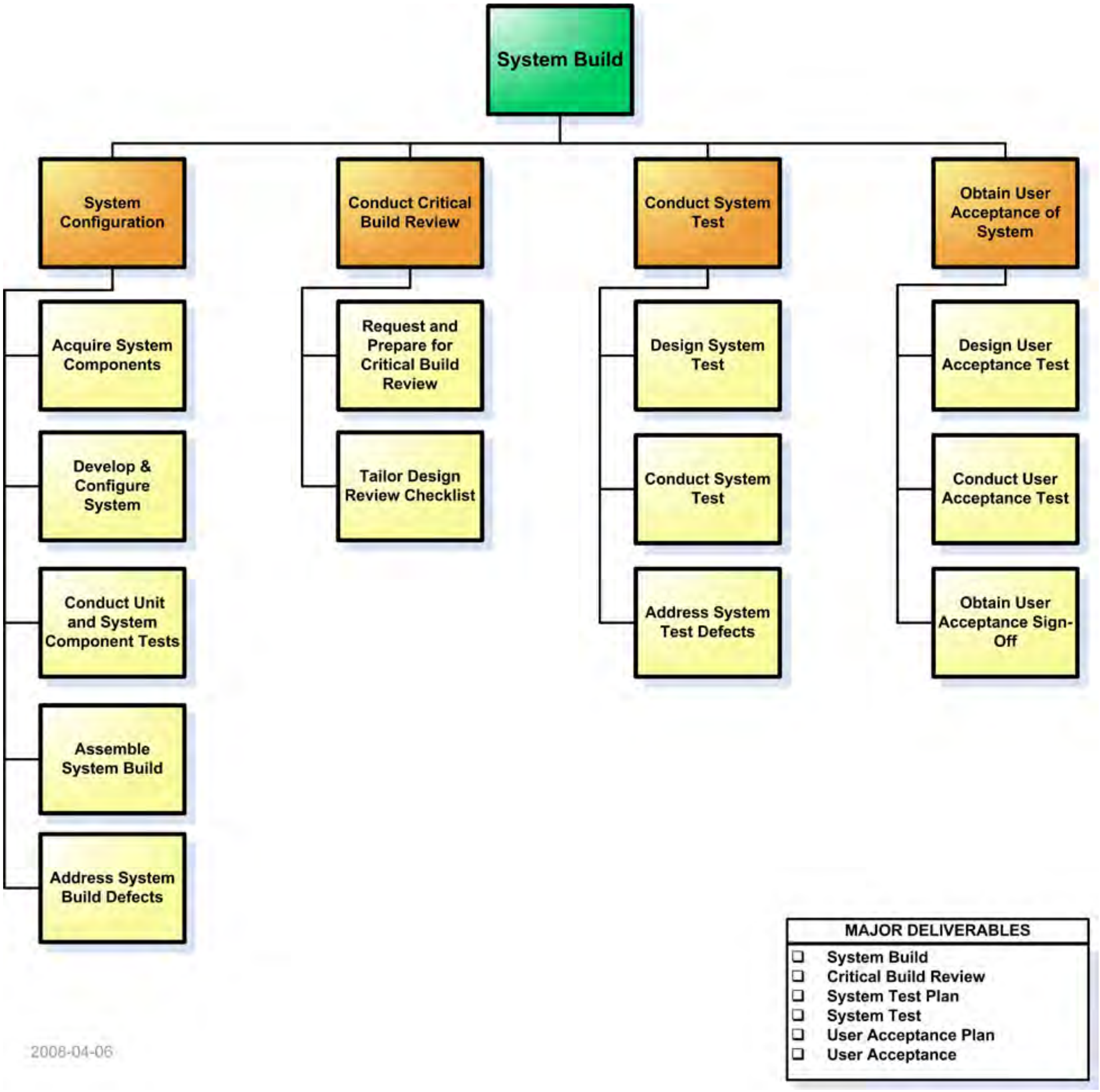
5.2. Phase Outputs

Stage	**Deliverables/Work Products**
System Configuration	▪ System Build ▪ Unit Test Results
Conduct Critical Design Review	▪ Critical Build Review
Conduct System Test	▪ System Test Plan ▪ System Test ▪ System Test Results

[12] Software As A Service (SAAS). A software deployment method in which a provider licenses an application to customers for use as a service on demand through a hosted environment.

Stage	Deliverables/Work Products
Obtain User Acceptance of Solution	▪ User Acceptance Plan ▪ Users Acceptance

5.3. Phase WBS



System Build
System Configuration
Acquire System Components
Develop & Configure System
Conduct Unit and System Component Tests
Assemble System Build
Address System Build Defects
Conduct Critical Build Review
Request and Prepare for Critical Build Review
Tailor Design Review Checklist
Conduct System Test
Design System Test
Conduct System Test
Address System Test Defects
Obtain User Acceptance of System
Design User Acceptance Test
Conduct User Acceptance Test
Obtain User Acceptance Sign-Off
MAJOR DELIVERABLES
❑ System Build
❑ Critical Build Review
❑ System Test Plan
❑ System Test
❑ User Acceptance Plan
❑ User Acceptance
2008-04-06

5.4. System Configuration

5.4.1 Acquire System Components

The first step for the project team is to examine the System Design and develop a manifest or "bill of materials" of all the required system components. These components then need to be examined to determine which are currently available and which must be acquired (provisioned from existing IT assets or procured). If the solution set and software platform is not already in place, the project team will consult the Data Center Hosting Advisory Council (DCHAC) before acquiring the technology and establishing the development environment (IDE etc). If this technology is new to the team, this step will include any required vendor training. In some cases, temporary specific expertise may be required that is best provided by the vendor, this should also be considered in this activity.

5.4.2 Develop & Configure System

The physical System is constructed by the project team. During this activity the development team will work with the chosen software and hardware platforms to transform the design into a working system. Some example transforms from the system design to the physical design are shown in the table below.

Physical System Design	**to**	**System**
Program procedural logic or COTS Module mapping/configuration	➔	Programs and Scripts (Physical code) Configured Software Package modules
Reports/Forms/Screens Design Layouts	➔	Reports, Screens
Physical Database Design	➔	Database Tables (Instantiated with test data)
User interface design System navigation and behavior	➔	User interface and dialogs
Software Services and Middleware Design Hardware and Operating System Design Security Controls and Compliance Design Storage and Archiving Design Network Design Interface Design BC/DR[13] Compliance (required adjustments to HW, System configuration, etc)	➔	Solution Software Build + Configured Supporting Technology Components
User groups, roles, access rights and privileges	➔	Configured User Accounts

[13] BC/DR Business Continuity/Disaster Recovery planning are processes that help organizations prepare for disruptive events and how they will continue to operate should such an event occur. In this context the project team should conduct a business impact analysis to examine what would happen if a disruptive event made the system unavailable. This insight is then used to adjust specific aspects and parameter of the physical design to address these needs e.g. Recovery, failsafe provisions. etc.

As described earlier, the sequencing of these tasks will vary depending on the nature of the development effort, its scale and the approach (Traditional Waterfall, Prototyping/Spiral development, or Agile.)

5.4.3 Conduct Unit and System Component Tests

As the programming progresses each software developer will be responsible for assuring the functionality and quality of their assigned modules through Unit Testing. Each hardware component will also be tested by the project team.

5.4.4 Assemble System Build

On a periodic basis a software build should be created. In a software system the System Build links the source code components together into a Build version that can be tested by the development team. For a hardware solution, the individual components will be assembled and tested as a complete system. Depending on the complexity of the system, test cases and scenarios derived from the Use Cases will then be used to check the functionality of the system against requirements. This activity would also involve the creation of any test data and the migration of software components from the Development environment to the Test environment.

For software development it is recommended that 4 separate but synchronized software environments be employed to manage the software release process from source code development to implementation, namely:

- **Development Environment**: Programmer Analysts will use this environment for check-out/check-in of software components and to develop and unit test components
- **Test Environment**: This environment will be used for software builds and system tests by the development team and some degree of user testing. The purpose of this environment is to test software quality.
- **Staging Environment:** Release ready software is staged to this pre-production environment to replicate and test the software deployment process and to conduct User Acceptance Testing (UAT), performance and vulnerability testing.
- **Production Environment:** After an Operational Readiness Review has been conducted, the software will be staged to the production environment for live running and deployment to end users.

For a hardware system development only a Test and Production environment are recommended.

5.4.5 Address System Build Defects

Any system defects detected during system testing should be investigated and corrected in this activity.

5.5. Conduct Build Review

ITS management has established a Design Review Board (DRB) to improve the quality and efficiency of project solution design implementations and details of the ITS Design Review Process have also been documented in a separate ITS publication[14]. An earlier design review should have been conducted during the System Design Phase. The goal of this design review is to assess the quality of the "as built" design.

The project manager is responsible for working with the assigned DRB advisor and to check the design against the original system design that was approved by the DRB. If the technology stack of the "as built"

[14] ITS Design Review Process Version 1.3

solution is substantially different from what was approved by the DRB in the System Design Phase a more formal design review must be conducted. This judgment will be made by the DRB advisor. If it is deemed that a formal DRB review is required, the WBS steps from the System Design review should be repeated (See 4.5) otherwise the team may proceed with the sign-off from the DRB Advisor.

5.5.1 Request and Prepare for Design Review

The Project Manager should request a Design Review from the Design Review Board and work with the project team to assemble the design review materials. The Design Review Board will assign one of its members as advisor to work with the project team and prepare them for the review.

5.5.2 Tailor Design Review Checklist

The Design Review process uses a master review design checklist that covers a wide range of technology solutions. The project team should work with their assigned DRB advisor to customize the master design checklist to the scale, nature and technology of their project.

5.6. Conduct System Test

5.6.1 Design System Test

"Testing is the process of executing a program with the intent of finding errors[15]". The goal of this activity is to construct a system test plan that will maximize the number of system defects uncovered by the team and thereby improve the team's ability to improve product quality. As with other activities in this phase, the scope of this activity can vary greatly with the complexity of the solution under development and to what degree the system to be tested is a pure software system, hardware system, or some combination of both. In the case of software testing, many textbooks have been written on the subject of Software Testing and Testing techniques. For now, the reader is encouraged to review these sources in constructing a system test plan. In the future we hope to provide more specific guidance as we compile our own experiences and leading practices at ITS.

5.6.2 Conduct System Test

The solution is staged to the Test environment and the system test is conducted by the project team.

5.6.3 Address System Test Solution Defects

The Project Manager summarizes and ranks the priority of defects uncovered during testing, assigns a development team member to fix each defect and provides the necessary follow-up to ensure that all major defects are remedied. The System Test activity is then repeated until all major defects have been addressed adequately. The determination of whether all major defects have been addressed is sometimes a difficult trade-off that must be made by the project manager in consultation with the project sponsor.

[15] "The Art of Software Testing", Glenford J. Myers, Tom Badgett, Todd M. Thomas, Corey Sandler p 6, Edition: 2, John Wiley and Sons, 2004.

5.7. Obtain User Acceptance of System

5.7.1 Design User Acceptance Test

The goal of User Acceptance Testing (UAT) to test the solution with actual end users in conditions that simulate the real world operating environment as closely as possible. The degree and formality of UAT will vary greatly depending on the complexity and criticality of the solution.

At a minimum the User Acceptance Test design should involve the users in a structured way to test the application in the context of their real world situation and to allow the project team to capture any design flaws, functional gaps and software defects. Functional gaps are identified by systematically testing each Functional and Non-Functional Requirement defined earlier in the SDLC against the capabilities of the actual solution. In the case of a more complex system the UAT design should be formalized through the design of User Scripts and test cases. These scripts can be derived from the process or workflow models developed during the Requirements Definition Phase.

5.7.2 Conduct User Acceptance Test

The project Manager should work with the Project Sponsor to ensure that that UAT minimizes disruption to the daily work of the affected users but maximizes the quality of the testing. The user acceptance tests should be conducted over a sufficient period of time to allow adequate participation of all key users. Testing may take place in a specially designated Conference Room UAT location or at the user locations. In either case, key members of the project team should participate in these work sessions and be available to provide support. The project team is responsible for collecting and compiling the UAT results and summarizing them for the user group and the Project Sponsor.

5.7.3 Obtain User Acceptance Sign-Off

The compiled results of the UAT are reviewed with the Project Sponsor to obtain sign-off that the solution adequately meets the stated requirements. Once this approval has been obtained, the Project Team should begin work with the designated Service Management Team to finalize the service definition for the system. Although in this case, the Service Development process is invoked by the SDLC process, it is a separate but related critical process in the creation of a solution for the customer. The Service management process includes the definition of a Service Level Agreement (SLA) and/or Operations Level Agreement (OLA), where appropriate. More information on the Service Management process and methodology may be found on the ITS web site[16].

[16] See the Client Services and Security Group web page http://its.ucsc.edu/it_services/index.php

6. System Deployment

6.1. Phase Goals

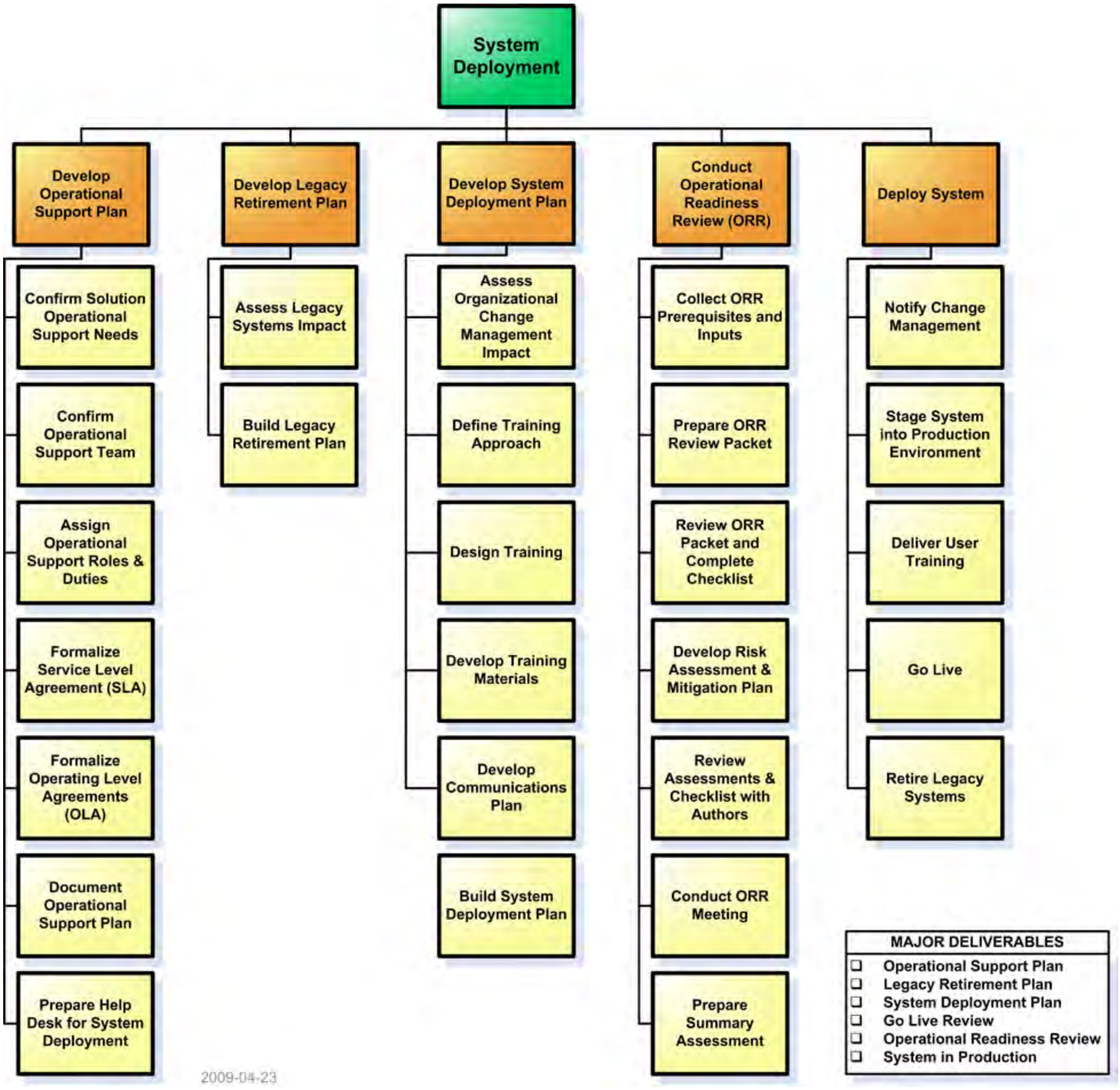
The goals of this phase are to:

- Confirm the operational readiness of the solution for deployment
- Migrate the system to the production environment
- Deploy the system to its end users

6.2. Phase Outputs

Stage	**Deliverables/Work Products**
Develop Operational Support Plan	▪ Operational Support Plan
Develop Legacy Retirement Plan	▪ Legacy Retirement Plan
Develop System Deployment Plan	▪ System Deployment Plan
Go Live Check	▪ Go Live Review
Conduct Operational Readiness Review	▪ Operational Readiness Review
Deploy Solution	▪ System in Production

6.3. Phase WBS


System Deployment
Develop Operational Support Plan
Confirm Solution Operational Support Needs
Confirm Operational Support Team
Assign Operational Support Roles & Duties
Formalize Service Level Agreement (SLA)
Formalize Operating Level Agreements (OLA)
Document Operational Support Plan
Prepare Help Desk for System Deployment
Develop Legacy Retirement Plan
Assess Legacy Systems Impact
Build Legacy Retirement Plan
Develop System Deployment Plan
Assess Organizational Change Management Impact
Define Training Approach
Design Training
Develop Training Materials
Develop Communications Plan
Build System Deployment Plan
Conduct Operational Readiness Review (ORR)
Collect ORR Prerequisites and Inputs
Prepare ORR Review Packet
Review ORR Packet and Complete Checklist
Develop Risk Assessment & Mitigation Plan
Review Assessments & Checklist with Authors
Conduct ORR Meeting
Prepare Summary Assessment
Deploy System
Notify Change Management
Stage System into Production Environment
Deliver User Training
Go Live
Retire Legacy Systems
MAJOR DELIVERABLES
□ Operational Support Plan
□ Legacy Retirement Plan
□ System Deployment Plan
□ Go Live Review
□ Operational Readiness Review
□ System in Production
2009-04-23

6.4. Develop Operational Support Plan

The purpose of an Operational Support Plan is to ensure that the developed system will be:

- Properly integrated into the IT Service Catalog and ITS Service Management process
- Adequately supported by ITS as a production system[17] on an ongoing basis in line with an agreed client Service Level Agreement (SLA) and Operating Level Agreements (OLA)

[17] Our definition of production system is any system that is delivered to the client for business use (purpose/intent). This definition has a client dependency since some delivered systems may be flagged as "beta", and thus under reduced or voluntary operational support plan as agreed to by the customer.

The activities in this stage assume that no Operational Support Plan exists and that it will have to be created for the system. If the system under consideration is already well established as an ITS service with an assigned Service Team then most of the Operational Support needs should already be in place. In this case the Project Team simply needs to work with the Service Team to consider any changes in these arrangements that need to be made due to the development project.

6.4.1 Document System Operational Support Needs

The team should define the scope of the Operational Support Plan by identifying all key resources, operating activities and tasks that will be required to support the system. Example components are:

- Hours of operation and scheduled maintenance windows (This should be integrated with the ITS Maintenance Calendar)
- System Backup schedule
- System environmental considerations (physical, hardware, operating environments, facilities, networks and platforms)
- Relation to other systems and services (Is the system a standalone system, a subsystem of a larger system or a replacement of an existing system?)
- System and network monitoring needs (performance measurements, statistics, and system logs)
- Documentation of Operating Procedures
- Acquisition and storage of consumable supplies (i.e. paper, toner, tapes, removable disk);
- Physical security needs, designated personnel access
- Disaster recovery procedures
- Software and Hardware licensing and maintenance agreement information, client and vendor contacts

6.4.2 Identify Operational Support Team

The roles (responsibilities and competencies) and functional groups responsible for ongoing support of the system should be identified.

6.4.3 Assign Operational Support Roles & Duties

Specific personnel should be formally assigned to the identified Operational Support Team roles and provided with the necessary orientation to perform their support role. The Service Team roster should be updated to reflect this information.

The assigned Operational Support team will then begin to create the components identified in 6.4.1.

6.4.4 Finalize Service Level Agreements (SLA)

The Project Manager should work with the Account Manager, Project Sponsor and IT Service Management to develop a formal Service Level Agreement or SLA for the system. The IT Service Development process is a separate but related process. This activity represents a key integration point between these processes but for more details of SLA standards and development please refer to the ITS Web site[18].

[18] http://its.ucsc.edu/service_catalog/sla/

6.4.5 Finalize Operating Level Agreements (OLA)

Operating Level Agreements (OLA) should also be developed to establish the linkages and responsibilities between internal IT Service providers (Help Desk, Applications Development, etc.) to deliver the service to the end customers. For an OLA template please refer to the ITS web site[19].

6.4.6 Document Operational Support Plan

The Project Manager should ensure that the various components of the Operational Support Plan are documented, compiled and provided to the designated IT Service Manager.

6.4.7 Prepare Help Desk for System Deployment

Final arrangement and preparations should be made with the IT Help Desk to prepare them for System Deployment.

6.5. Develop Legacy Retirement Plan

6.5.1 Assess Legacy Systems Impact

In most cases the impact of the new system on any legacy systems it will replace will have been identified earlier in the design phase. This analysis needs to be documented in this activity in preparation for the development of a legacy retirement plan.

6.5.2 Build Legacy Retirement Plan

The team should build a legacy retirement plan that phases out the old system as the new solution is implemented and adopted. In some cases this may include a "parallel running" stage in which both systems, old and new, are operated concurrently until final user acceptance.

6.6. Develop System Deployment Plan

The System Deployment Plan is a holistic implementation plan that considers the people, processes and technology that need to be in place for the system to be successfully installed, adopted by the user community, and the benefits of the system to be realized.

6.6.1 Assess Organizational Change Management (OCM) Impact

The implementation of every major system will require some change in the behavior and practice of the organization to be successful.

Depending on the nature and scale of the system being implemented the Project Team may encounter significant organizational resistance to change. If this "soft side" of implementation is not properly addressed the project will fail to achieve its goals. It is beyond the scope of this SDLC methodology to fully address the implications of Organizational Change Management (OCM) but at the very least the team should assess impact of the new system in terms of new tasks, changes in procedures, policies and

[19] ITS OLA Template

processes, new skills and competencies required, and new or modified job roles on the user organization. This analysis should then be used to shape the development of the deployment approach, training and communications for the system implementation and to work with the Project Sponsor in rolling out the system.

It should also be noted, that although this activity is positioned in the deployment phase of the SDLC this OCM analysis and conversation with the Project Sponsor should be begun as early as possible in the SDLC process.

6.6.2 Define Training Approach

The training approach should be determined in terms of audiences (Users, IT Help Desk, etc), learning objectives, training modes (personal coaching, self study, classroom, Computer based Training, On-line, etc.), use of specialist instructional designers and trainers, and sources of training e.g. vendor classes etc.

6.6.3 Design Training Curriculum

The Project Manager or a specialist Trainer/Instructional Designer will identify, design and source the training offerings required to support the various users of the system. These Users may be End Users, Sponsors, ITS support staff etc.

6.6.4 Develop Training Materials

Training materials are developed or acquired. It is recommended that the training materials also be reviewed by the ITS Help Desk staff. This will enable the Help Desk staff to gain advance knowledge of the capability and support needs of the system before the Go Live date.

6.6.5 Develop Communication Plan

If the system to be implemented is complex, effective communications is a critical success factor for the deployment process. Using the insight gained from the OCM analysis, the Project Manager should work with the Project Sponsor to develop a communication plan that identifies and addresses:

- Target stakeholder groups and individuals and their communication needs (e.g. awareness, sponsorship, etc.)
- Communication roles of the Project Sponsor and the project team
- Communication channels (Regular staff meetings, webinars, Town Halls, etc)
- Messaging and development of communications collateral
- Timing and sequence of communication events
- Creation of any additional system user feedback process beyond the normal IT Incident Management process

6.6.6 Build System Deployment Plan

All the previously identified aspects of the System Plan should be interpreted into a consolidated deployment project plan with appropriate tasks, role and milestones. This plan should encompass all the major activities of the deployment phase through to system "Go Live".

6.7. Conduct Operational Readiness Review (ORR)

Operational Readiness refers to the explicit acknowledgement that all the necessary requirements for production support are in place for the system and that the user system steward and the ITS Operations team accept responsibility for the system in production. An Operational Readiness Review must be initiated by the project manager to conform to the **ITS Operational Readiness Review process**. The project manager is responsible to follow the process through to operational readiness sign off by the external contributors and primary stakeholders. The primary work products are the ORR Review Packet and the Summary Assessment.

6.8.1 Collect ORR Prerequisites and Inputs

The Project Manager should consult with the IT Operations team, impacted IT Functional managers, and the Project Sponsor to identify, assess and summarize all outstanding issues, implementation concerns and unfinished project work that could impede the transition of the system into production.

6.8.2 Prepare ORR Review Packet

An ORR Review Packet is created for distribution to the IT Functional Group leads. This review packet provides an overview of the system and its operating requirements.

6.8.3 Review ORR Packet and Complete Checklist

Each Functional Group Leads uses the information in the ORR Review Packet to create a specific ORR Checklist for their group area and develop any required risk assessment and mitigation plans for system cutover.

6.8.4 Develop Risk Assessment & Mitigation Plan

The Project Manager compiles the input from the Functional Group Leads into an overall Risk Assessment and Mitigation Plan.

6.8.5 Review Assessments & Checklist with Authors

The Project manager schedules the ORR meeting and reviews and compiles the ORR Checklists from each Functional Group Leader.

6.8.6 Conduct ORR Meeting

The ORR meeting is held.

6.8.7 Prepare Summary Assessment

The Project Manager created a Summary Assessment report as a result of the ORR Meeting. This document is then shared with the external contributors and primary stakeholders to inform their decision to proceed or reschedule based on the outcome of the ORR Meeting.

6.8. Deploy Solution

6.8.1. Notify Change Management

The Change Coordinator must notified of the Go Live timing at least two weeks in advance.

6.9.2 Stage Solution into Production Environment

The system is migrated into production.

6.9.3 Deliver End User Training

The training plan is executed.

6.9.4 Go Live

The Project Manager should confirm the roles and responsibilities of all project team members, Project Sponsor and Key Users for orchestrating the system go live event. The ITS Help Desk should be also represented in this team. Team members are responsible for completing their assigned tasks and reporting their status at the Go Live meeting.

A final Go Live checklist should be developed and a Go Live Checklist Review meeting scheduled[20].

The Project Manager and Project Sponsor chair the Go Live meeting. The goal of the meeting is to determine that:

- The Client is ready
- The Users are ready
- The Implementation and Operational technicians are ready
- The system is ready
- An outage schedule has been negotiated and/or an RFC has been submitted/approved through the Change Management Process

The Go Live checklist is reviewed and action items assigned for follow-up. The Project Manager will oversee the completion of all outstanding Go Live Checklist issues.

After resolution of all checklist issues, the system is put into production. The Project Manager must then notify Change Management that the system is now live. This is the final notification that deployment has been completed, or that the System (full or portions of) failed and was backed out of the Production environment.

6.9.5 Retire Legacy Systems

The Project Team executes the Legacy Retirement Plan.

[20] See the ITS Website for a master ORR Checklist that may be used as a starting point: https://collab.ucsc.edu/its-group-spaces/applications-development-architecture/artifacts/orr-checklist/

7. Appendix A – RACI Matrices

This Appendix shows the recommended involvement of the standard methodology roles with the methodology work breakdown structure for each phase of the SDLC. The interaction of each role with a specific activity is codified using a conventional RACI matrix format for each phase of the SDLC.

The responsibility of each role is specified in a RACI matrix that relates the roles to the activities and deliverables with an intersecting letter code:

RACI Code Definitions:

Responsibility = role is responsible for actually doing or completing the item

Accountable = role is accountable for ensuring that the item is completed. Usually only one person.

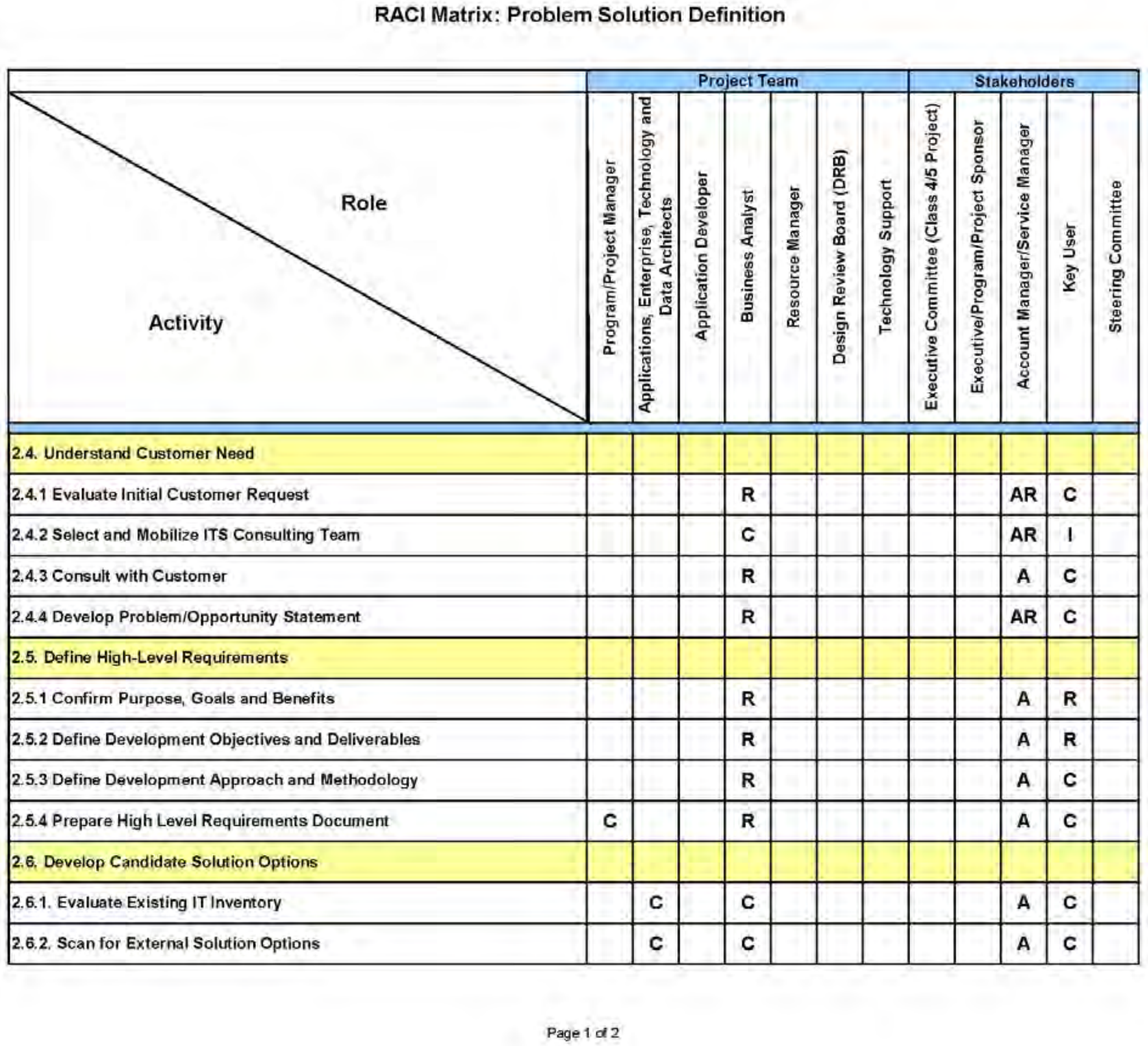
Consulted = role whose subject matter expertise is required in order to complete the item

Informed = role that needs to be kept informed of the status of item completion

In some cases multiple roles have been combined in one column to make the RACI more compact and to address those circumstances where the group of have has identical responsibilities but each contributes specific subject matter knowledge to the activity e.g. Applications, Enterprise, Technology and Data Architect roles.

The Account Manager/Service Manager combination is a special case that warrants further explanation. This role combination assumes that all clients will have a designated ITS single point of contact for coordination of delivery of IT services (As originally recommended by the CRSP Project and endorsed by SMT). In most cases this role is performed by the Divisional Liaisons (DLs) acting as an Account Manager[21]. However, as the appointment of IT Service Managers progresses, clients may be directed to a specific Service Manager for a particular ITS global service such as CRUZMail. At the time of writing, DLs continue to play the role of Service Manager for locally provided IT services.

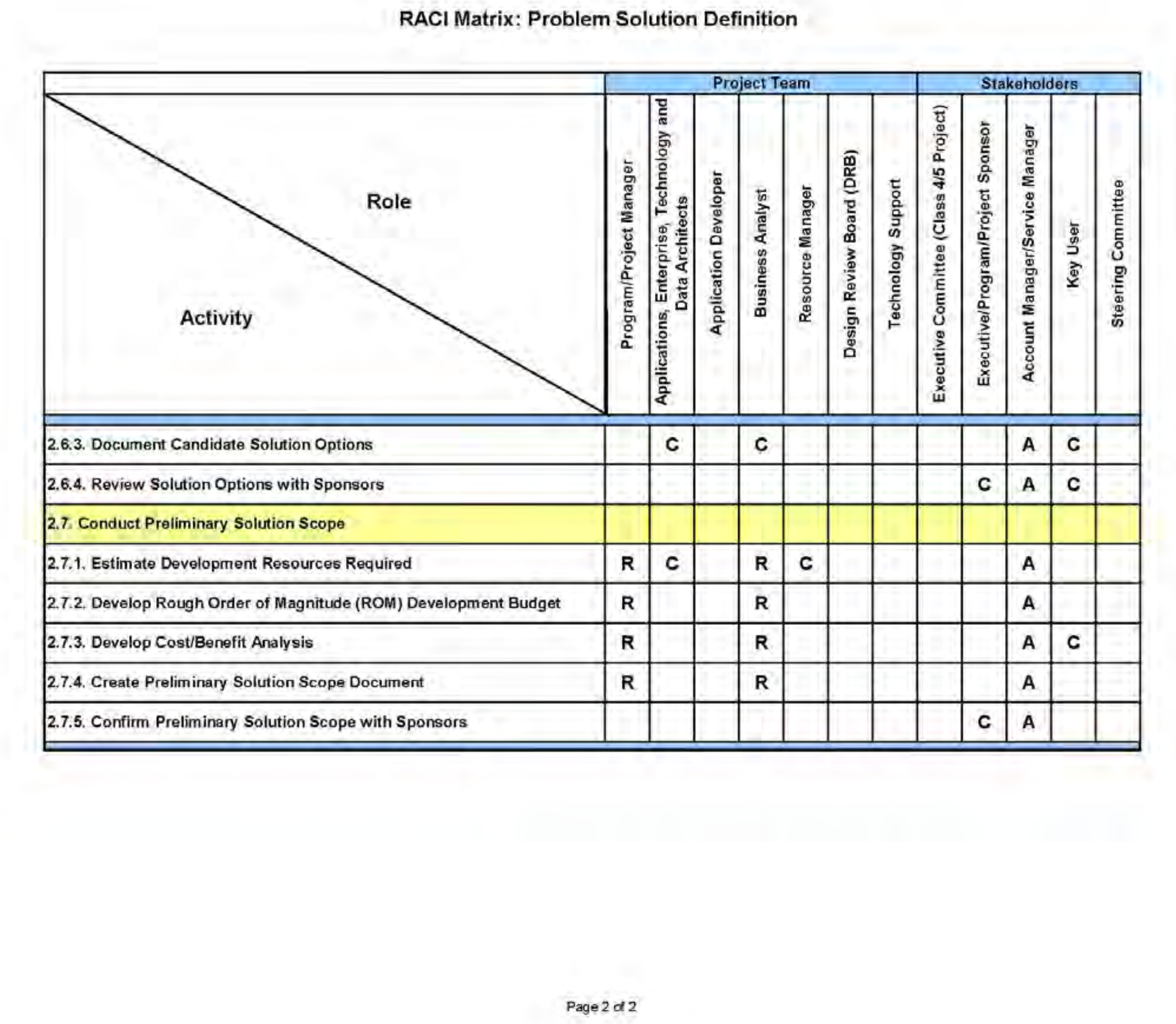
[21] CRSP (Customer Request to Solution Provided) Process Executive Summary Report 2007-05-13



RACI Matrix: Problem Solution Definition

Role / Activity	Project Team							Stakeholders				
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Resource Manager	Design Review Board (DRB)	Technology Support	Executive Committee (Class 4/5 Project)	Executive/Program/Project Sponsor	Account Manager/Service Manager	Key User	Steering Committee
2.4. Understand Customer Need												
2.4.1 Evaluate Initial Customer Request				R						AR	C	
2.4.2 Select and Mobilize ITS Consulting Team				C						AR	I	
2.4.3 Consult with Customer				R						A	C	
2.4.4 Develop Problem/Opportunity Statement				R						AR	C	
2.5. Define High-Level Requirements												
2.5.1 Confirm Purpose, Goals and Benefits				R						A	R	
2.5.2 Define Development Objectives and Deliverables				R						A	R	
2.5.3 Define Development Approach and Methodology				R						A	C	
2.5.4 Prepare High Level Requirements Document	C			R						A	C	
2.6. Develop Candidate Solution Options												
2.6.1. Evaluate Existing IT Inventory		C		C						A	C	
2.6.2. Scan for External Solution Options		C		C						A	C	

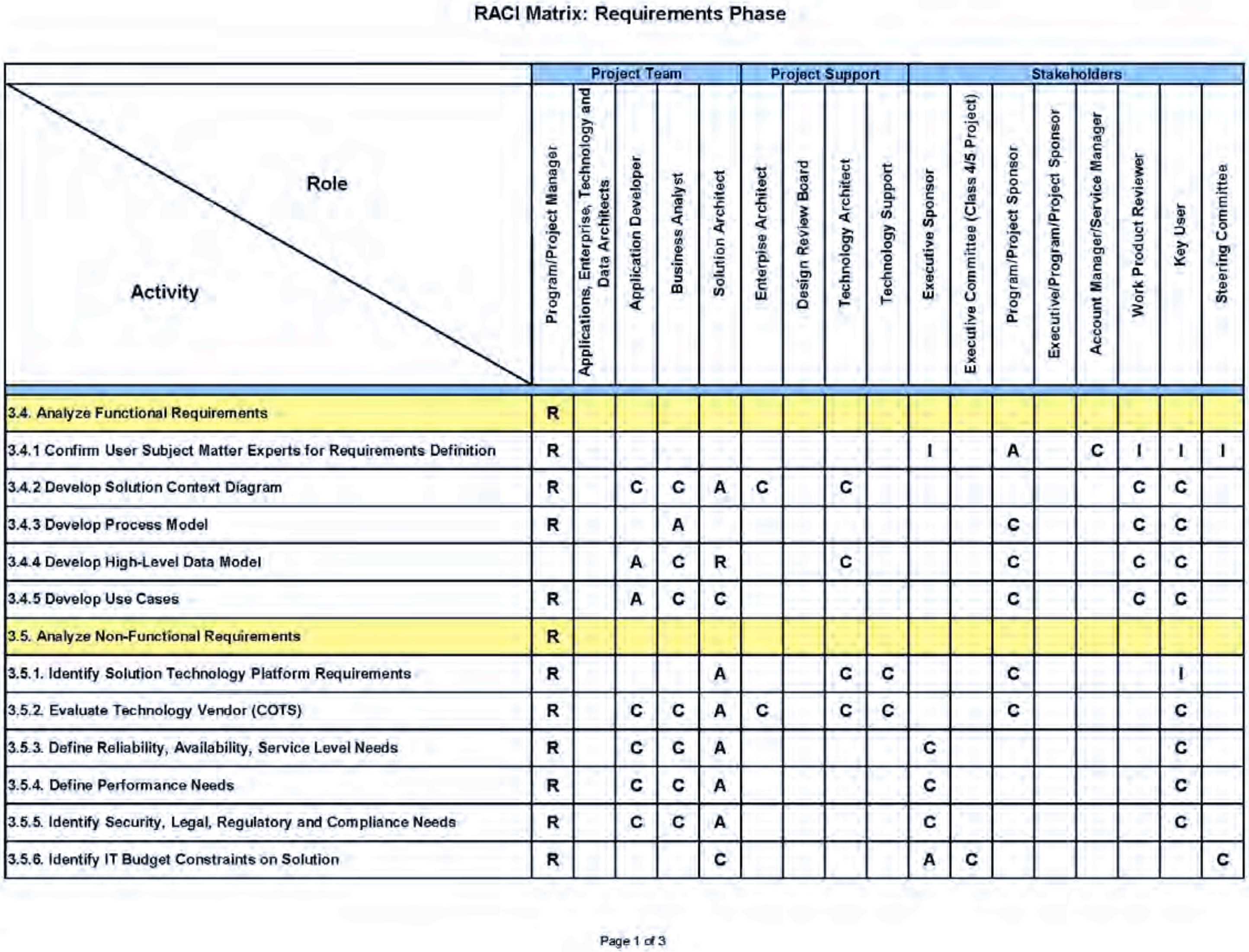
Page 1 of 2



RACI Matrix: Problem Solution Definition

Role / Activity	Project Team							Stakeholders				
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Resource Manager	Design Review Board (DRB)	Technology Support	Executive Committee (Class 4/5 Project)	Executive/Program/Project Sponsor	Account Manager/Service Manager	Key User	Steering Committee
2.6.3. Document Candidate Solution Options		C		C						A	C	
2.6.4. Review Solution Options with Sponsors									C	A	C	
2.7. Conduct Preliminary Solution Scope												
2.7.1. Estimate Development Resources Required	R	C		R	C					A		
2.7.2. Develop Rough Order of Magnitude (ROM) Development Budget	R			R						A		
2.7.3. Develop Cost/Benefit Analysis	R			R						A	C	
2.7.4. Create Preliminary Solution Scope Document	R			R						A		
2.7.5. Confirm Preliminary Solution Scope with Sponsors									C	A		

Page 2 of 2

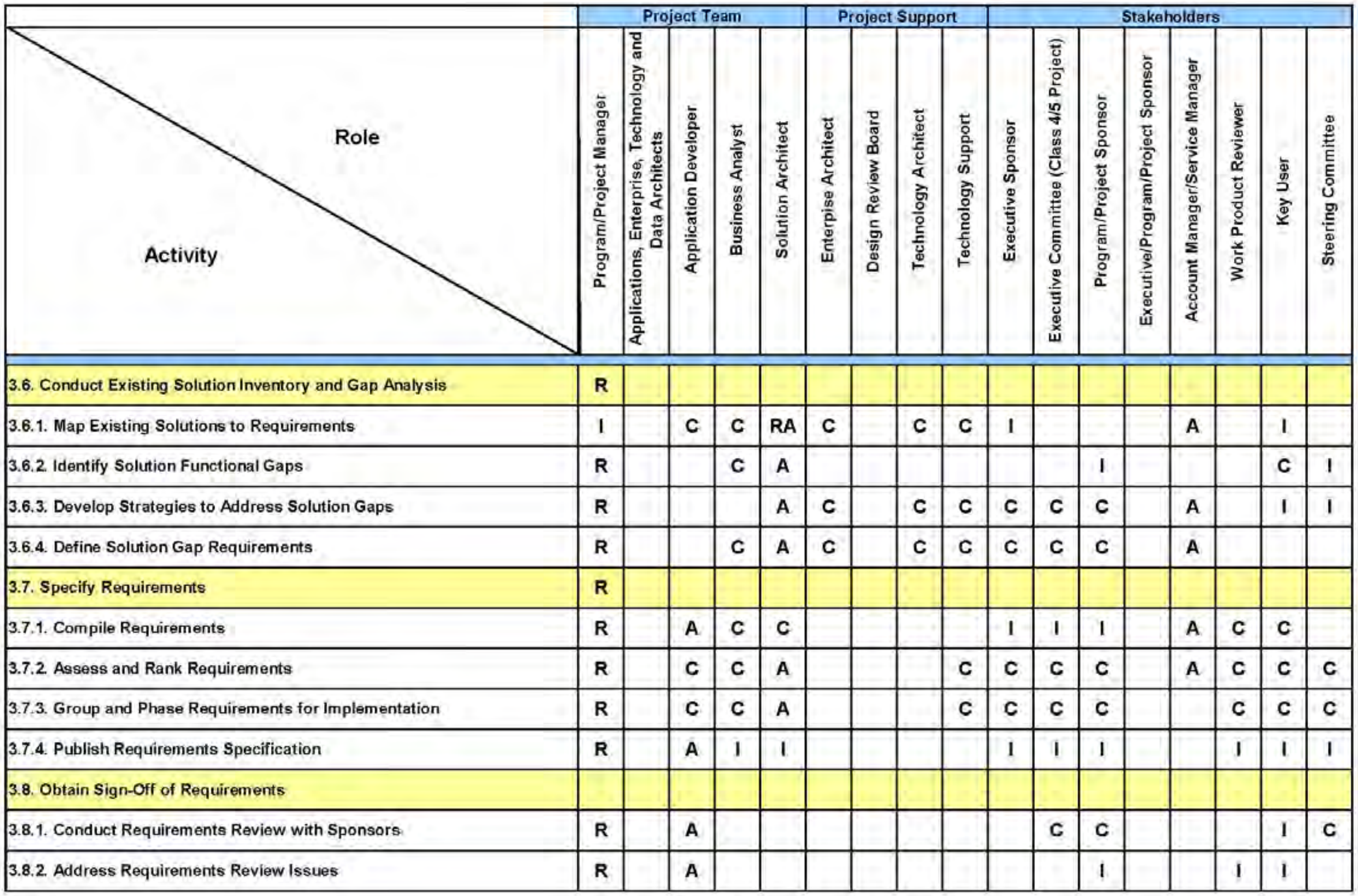


RACI Matrix: Requirements Phase

Role / Activity	Project Team					Project Support				Stakeholders							
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Solution Architect	Enterpise Architect	Design Review Board	Technology Architect	Technology Support	Executive Sponsor	Executive Committee (Class 4/5 Project)	Program/Project Sponsor	Executive/Program/Project Sponsor	Account Manager/Service Manager	Work Product Reviewer	Key User	Steering Committee
3.4. Analyze Functional Requirements	R																
3.4.1 Confirm User Subject Matter Experts for Requirements Definition	R									I		A		C	I	I	I
3.4.2 Develop Solution Context Diagram	R		C	C	A	C		C							C	C	
3.4.3 Develop Process Model	R			A								C			C	C	
3.4.4 Develop High-Level Data Model			A	C	R			C				C			C	C	
3.4.5 Develop Use Cases	R		A	C	C							C			C	C	
3.5. Analyze Non-Functional Requirements	R																
3.5.1. Identify Solution Technology Platform Requirements	R				A			C	C			C				I	
3.5.2. Evaluate Technology Vendor (COTS)	R		C	C	A	C		C	C			C				C	
3.5.3. Define Reliability, Availability, Service Level Needs	R		C	C	A					C						C	
3.5.4. Define Performance Needs	R		C	C	A					C						C	
3.5.5. Identify Security, Legal, Regulatory and Compliance Needs	R		C	C	A					C						C	
3.5.6. Identify IT Budget Constraints on Solution	R				C					A	C						C

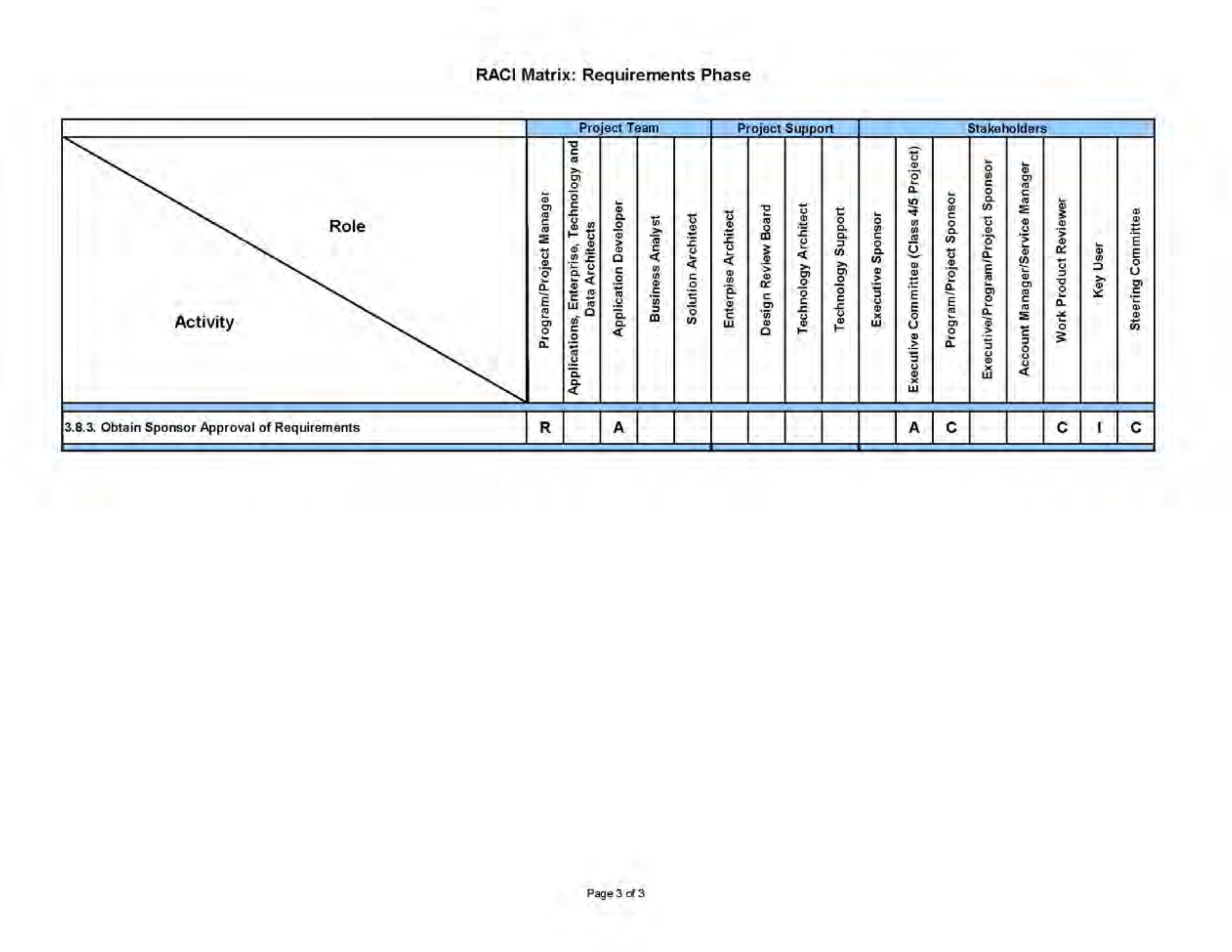
Page 1 of 3

RACI Matrix: Requirements Phase



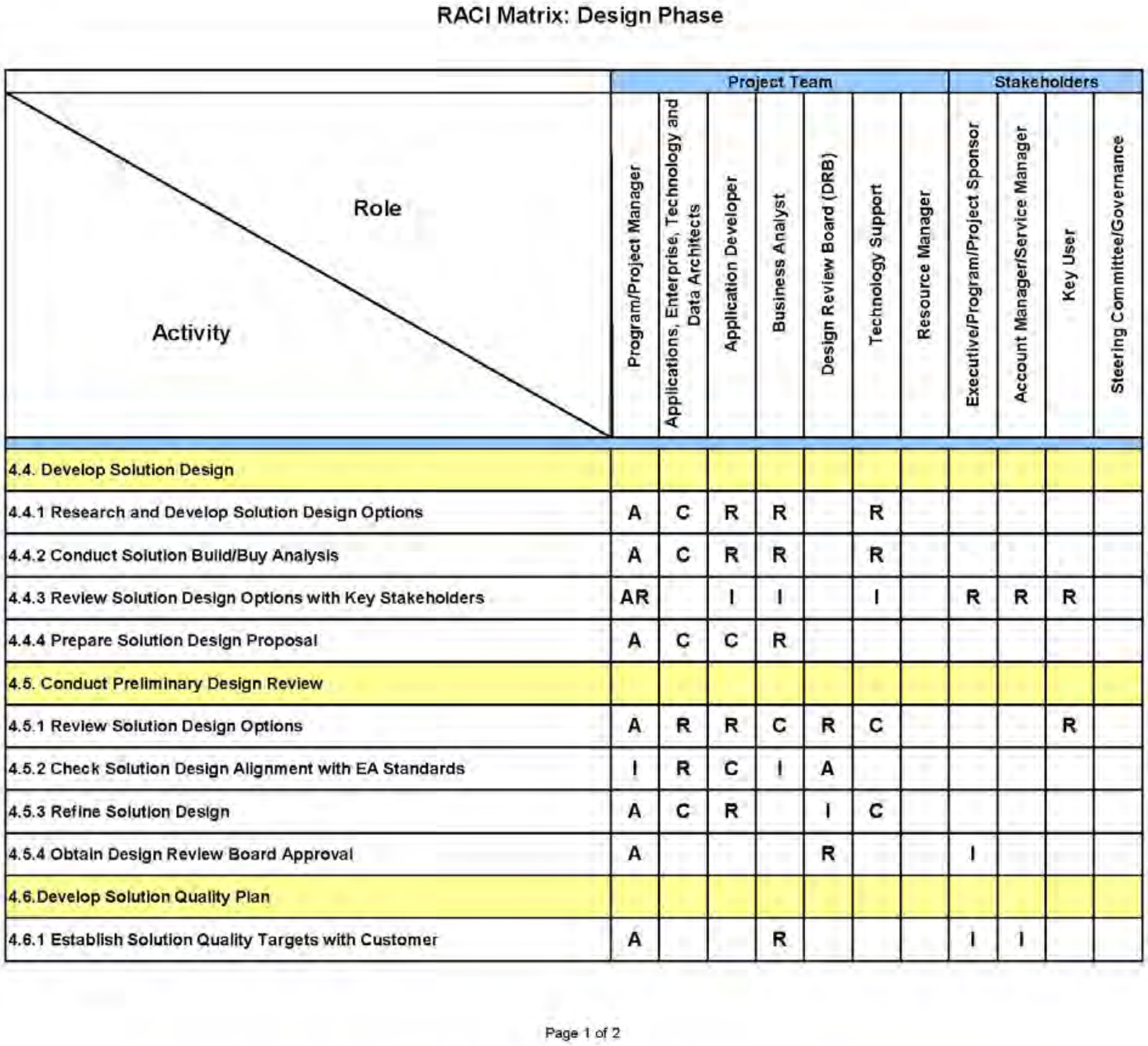
Role / Activity	Project Team					Project Support				Stakeholders							
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Solution Architect	Enterprise Architect	Design Review Board	Technology Architect	Technology Support	Executive Sponsor	Executive Committee (Class 4/5 Project)	Program/Project Sponsor	Executive/Program/Project Sponsor	Account Manager/Service Manager	Work Product Reviewer	Key User	Steering Committee
3.6. Conduct Existing Solution Inventory and Gap Analysis	R																
3.6.1. Map Existing Solutions to Requirements	I		C	C	RA	C		C	C	I				A		I	
3.6.2. Identify Solution Functional Gaps	R			C	A							I				C	I
3.6.3. Develop Strategies to Address Solution Gaps	R				A	C		C	C	C	C	C		A		I	I
3.6.4. Define Solution Gap Requirements	R			C	A	C		C	C	C	C	C		A			
3.7. Specify Requirements	R																
3.7.1. Compile Requirements	R		A	C	C					I	I	I		A	C	C	
3.7.2. Assess and Rank Requirements	R		C	C	A				C	C	C	C		A	C	C	C
3.7.3. Group and Phase Requirements for Implementation	R		C	C	A				C	C	C	C			C	C	C
3.7.4. Publish Requirements Specification	R		A	I	I					I	I	I			I	I	I
3.8. Obtain Sign-Off of Requirements																	
3.8.1. Conduct Requirements Review with Sponsors	R		A								C	C				I	C
3.8.2. Address Requirements Review Issues	R		A									I			I	I	

Page 2 of 3



RACI Matrix: Requirements Phase

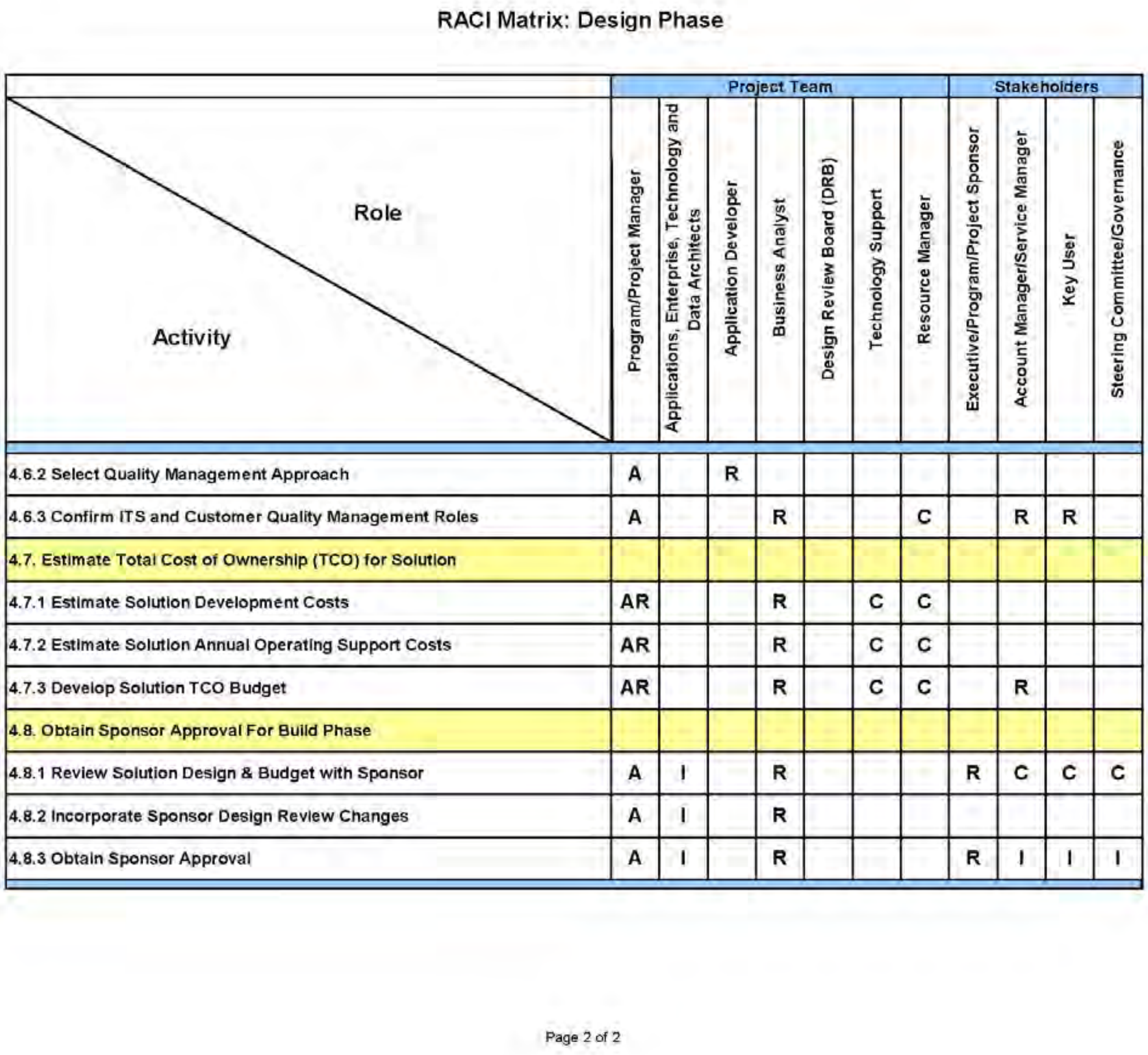
Role / Activity	Project Team					Project Support				Stakeholders							
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Solution Architect	Enterprise Architect	Design Review Board	Technology Architect	Technology Support	Executive Sponsor	Executive Committee (Class 4/5 Project)	Program/Project Sponsor	Executive/Program/Project Sponsor	Account Manager/Service Manager	Work Product Reviewer	Key User	Steering Committee
3.8.3. Obtain Sponsor Approval of Requirements	R		A								A	C			C	I	C



RACI Matrix: Design Phase

Role / Activity	Project Team							Stakeholders			
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Design Review Board (DRB)	Technology Support	Resource Manager	Executive/Program/Project Sponsor	Account Manager/Service Manager	Key User	Steering Committee/Governance
4.4. Develop Solution Design											
4.4.1 Research and Develop Solution Design Options	A	C	R	R		R					
4.4.2 Conduct Solution Build/Buy Analysis	A	C	R	R		R					
4.4.3 Review Solution Design Options with Key Stakeholders	AR		I	I		I		R	R	R	
4.4.4 Prepare Solution Design Proposal	A	C	C	R							
4.5. Conduct Preliminary Design Review											
4.5.1 Review Solution Design Options	A	R	R	C	R	C				R	
4.5.2 Check Solution Design Alignment with EA Standards	I	R	C	I	A						
4.5.3 Refine Solution Design	A	C	R		I	C					
4.5.4 Obtain Design Review Board Approval	A				R			I			
4.6.Develop Solution Quality Plan											
4.6.1 Establish Solution Quality Targets with Customer	A			R				I	I		

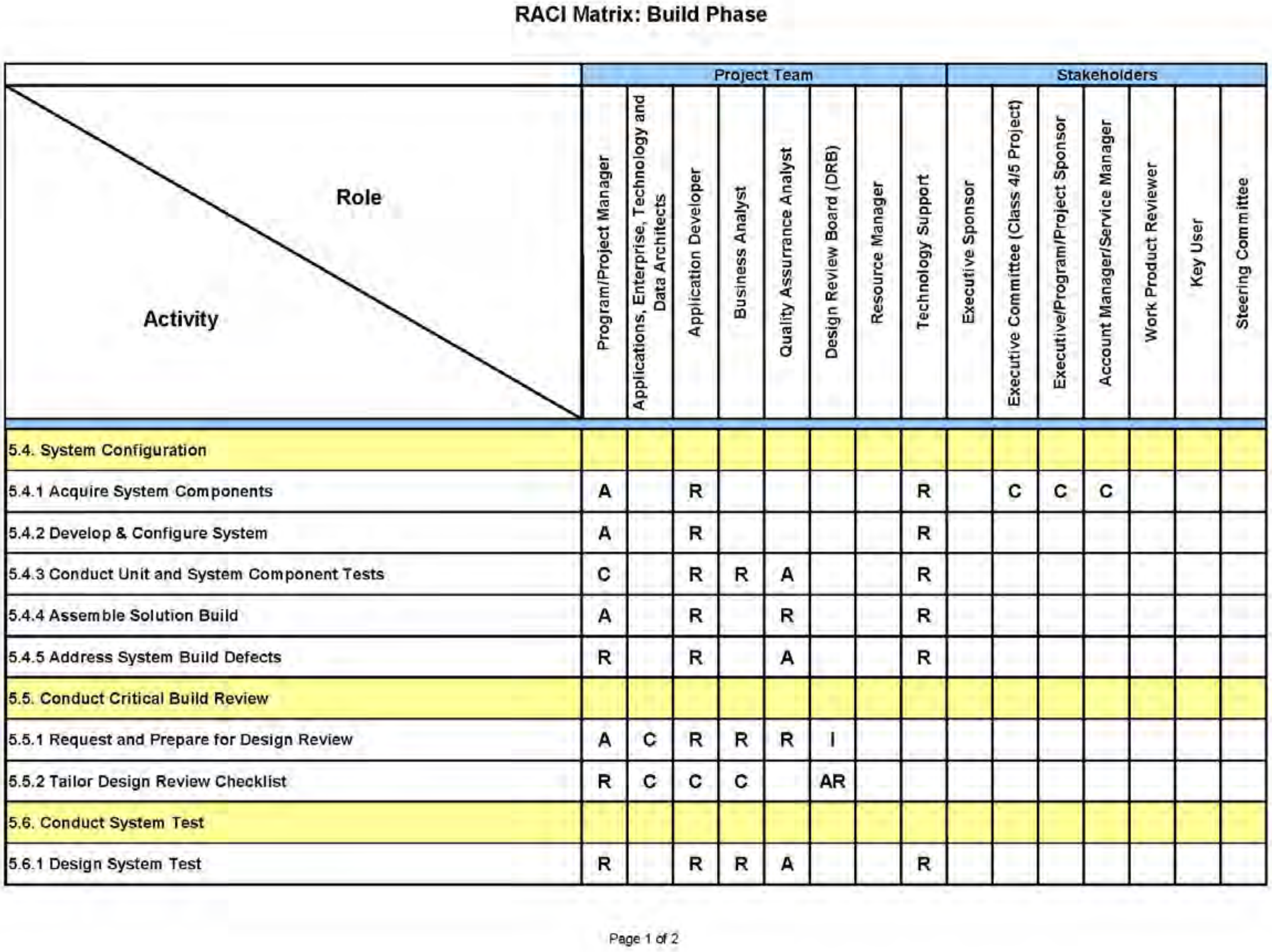
Page 1 of 2



RACI Matrix: Design Phase

Role / Activity	Project Team							Stakeholders			
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Design Review Board (DRB)	Technology Support	Resource Manager	Executive/Program/Project Sponsor	Account Manager/Service Manager	Key User	Steering Committee/Governance
4.6.2 Select Quality Management Approach	A		R								
4.6.3 Confirm ITS and Customer Quality Management Roles	A			R			C		R	R	
4.7. Estimate Total Cost of Ownership (TCO) for Solution											
4.7.1 Estimate Solution Development Costs	AR			R		C	C				
4.7.2 Estimate Solution Annual Operating Support Costs	AR			R		C	C				
4.7.3 Develop Solution TCO Budget	AR			R		C	C		R		
4.8. Obtain Sponsor Approval For Build Phase											
4.8.1 Review Solution Design & Budget with Sponsor	A	I		R				R	C	C	C
4.8.2 Incorporate Sponsor Design Review Changes	A	I		R							
4.8.3 Obtain Sponsor Approval	A	I		R				R	I	I	I

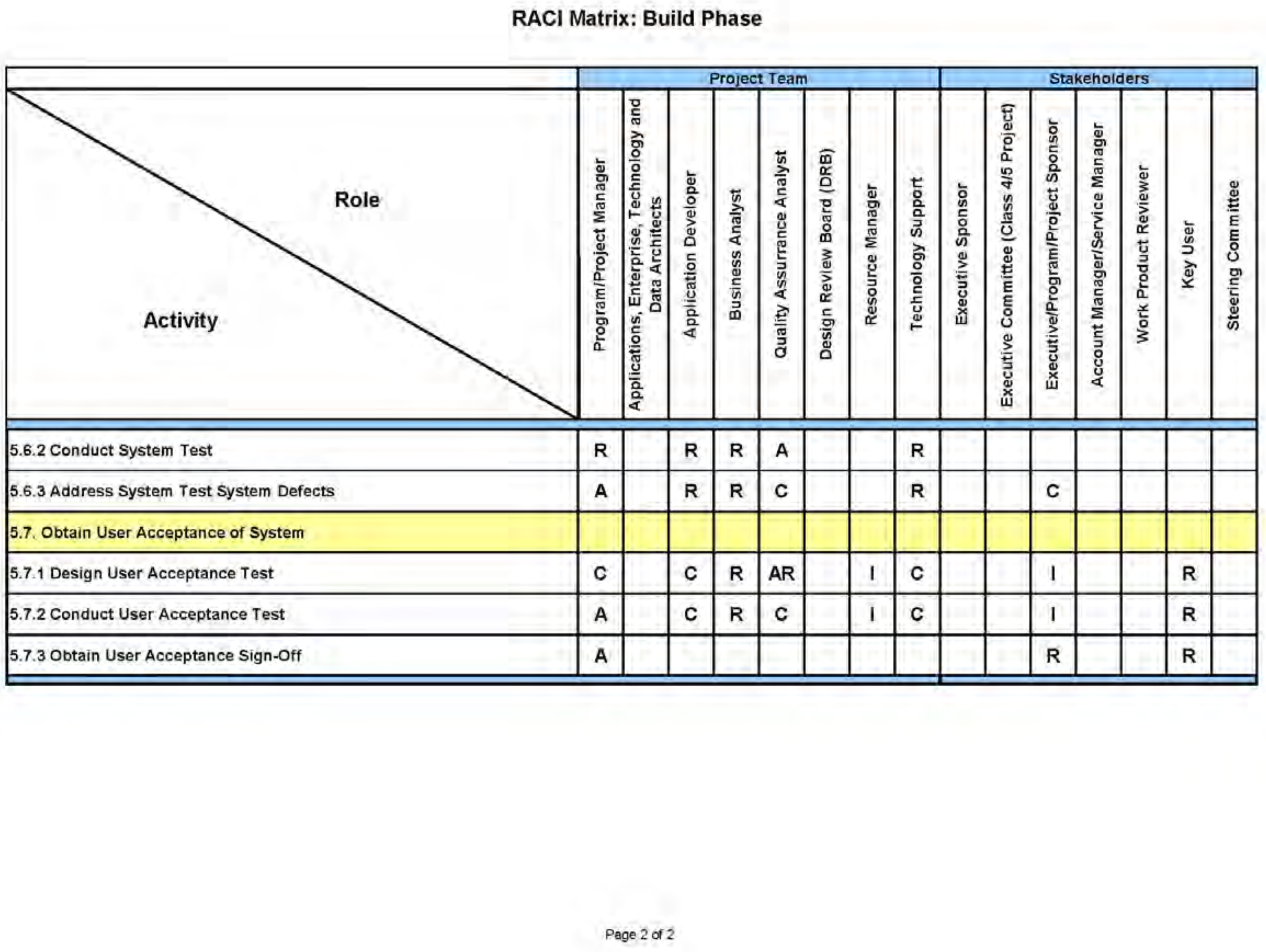
Page 2 of 2



RACI Matrix: Build Phase

Role / Activity	Project Team								Stakeholders						
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Quality Assurrance Analyst	Design Review Board (DRB)	Resource Manager	Technology Support	Executive Sponsor	Executive Committee (Class 4/5 Project)	Executive/Program/Project Sponsor	Account Manager/Service Manager	Work Product Reviewer	Key User	Steering Committee
5.4. System Configuration															
5.4.1 Acquire System Components	A		R					R		C	C	C			
5.4.2 Develop & Configure System	A		R					R							
5.4.3 Conduct Unit and System Component Tests	C		R	R	A			R							
5.4.4 Assemble Solution Build	A		R		R			R							
5.4.5 Address System Build Defects	R		R		A			R							
5.5. Conduct Critical Build Review															
5.5.1 Request and Prepare for Design Review	A	C	R	R	R	I									
5.5.2 Tailor Design Review Checklist	R	C	C	C		AR									
5.6. Conduct System Test															
5.6.1 Design System Test	R		R	R	A			R							

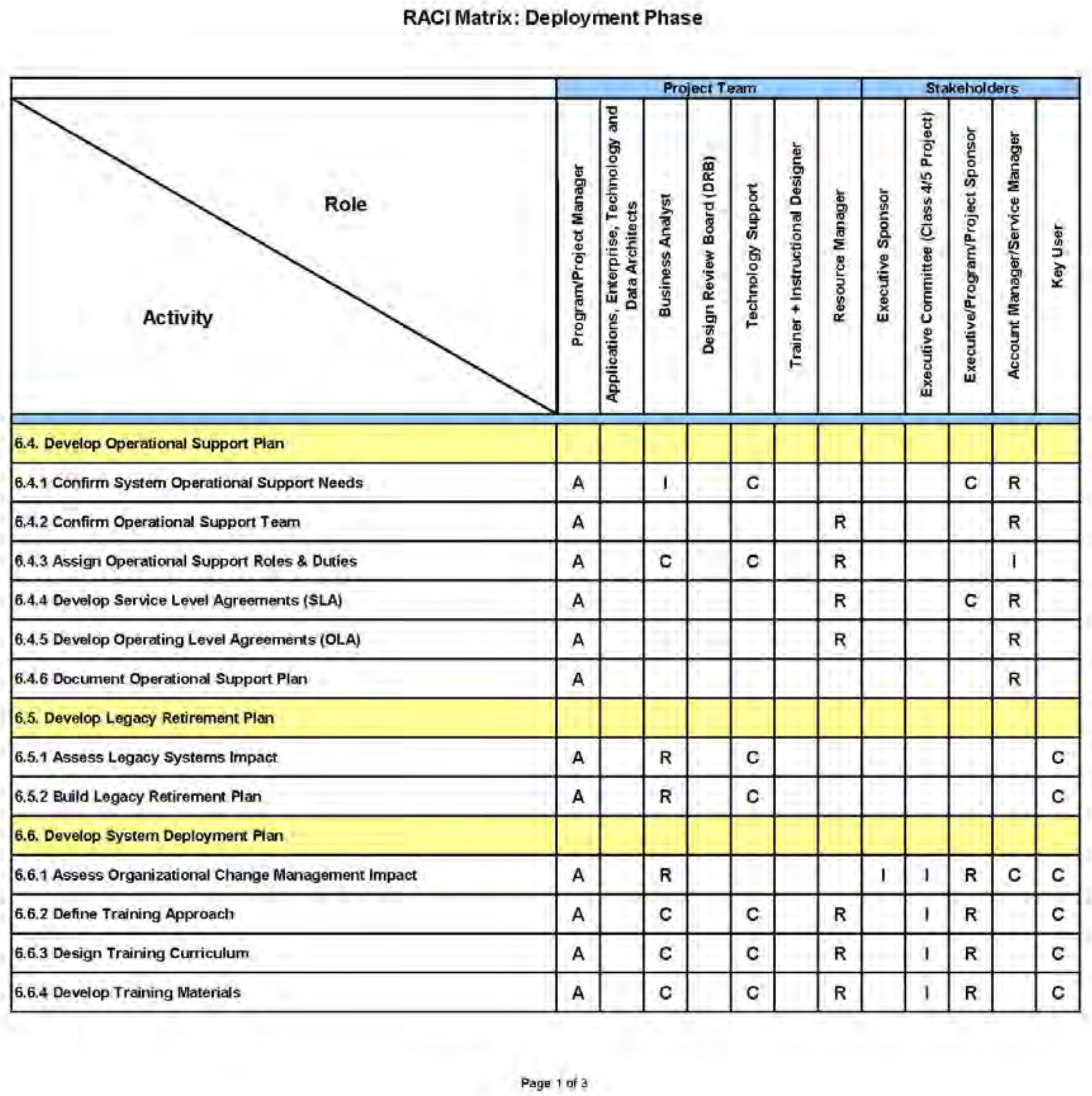
Page 1 of 2



RACI Matrix: Build Phase

	Project Team								Stakeholders						
Role Activity	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Application Developer	Business Analyst	Quality Assurrance Analyst	Design Review Board (DRB)	Resource Manager	Technology Support	Executive Sponsor	Executive Committee (Class 4/5 Project)	Executive/Program/Project Sponsor	Account Manager/Service Manager	Work Product Reviewer	Key User	Steering Committee
5.6.2 Conduct System Test	R		R	R	A			R							
5.6.3 Address System Test System Defects	A		R	R	C			R			C				
5.7. Obtain User Acceptance of System															
5.7.1 Design User Acceptance Test	C		C	R	AR		I	C			I			R	
5.7.2 Conduct User Acceptance Test	A		C	R	C		I	C			I			R	
5.7.3 Obtain User Acceptance Sign-Off	A										R			R	

Page 2 of 2

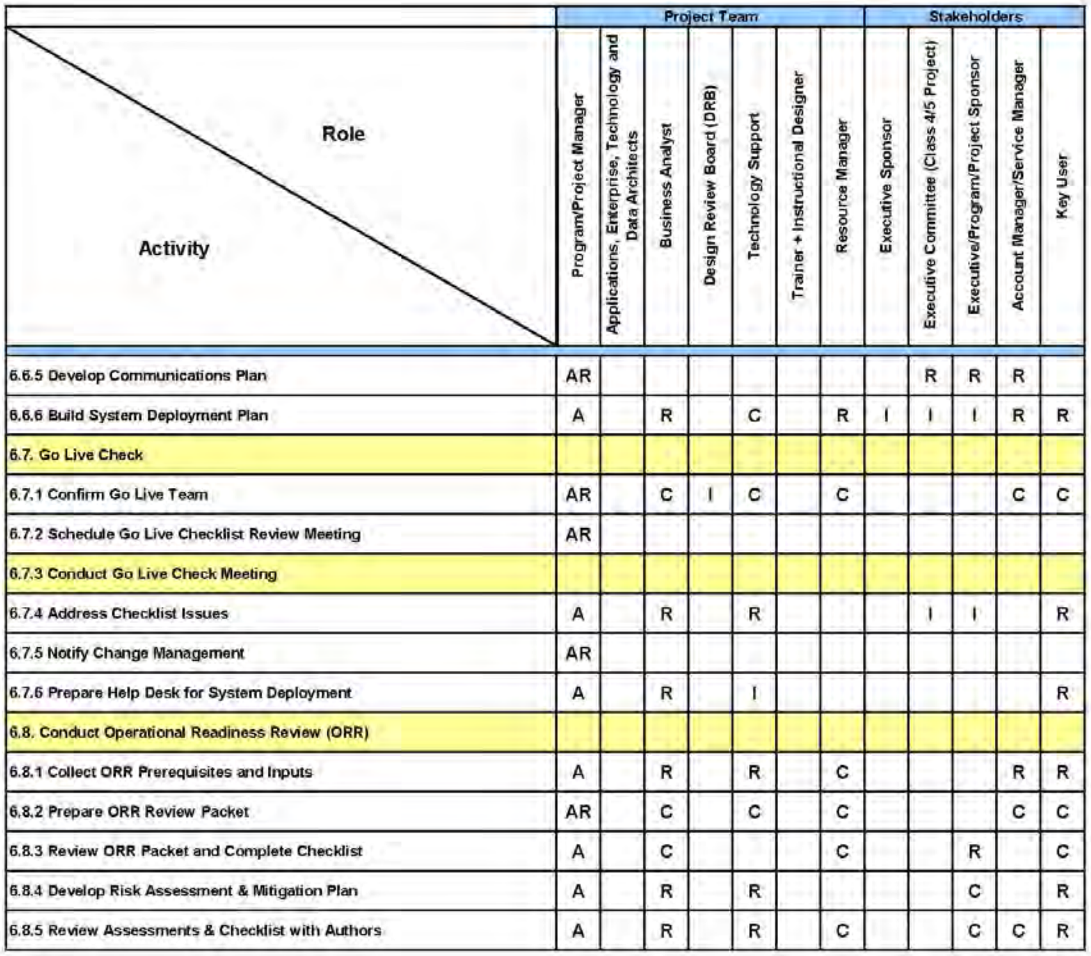


RACI Matrix: Deployment Phase

	Project Team							Stakeholders				
Role Activity	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Business Analyst	Design Review Board (DRB)	Technology Support	Trainer + Instructional Designer	Resource Manager	Executive Sponsor	Executive Committee (Class 4/5 Project)	Executive/Program/Project Sponsor	Account Manager/Service Manager	Key User
6.4. Develop Operational Support Plan												
6.4.1 Confirm System Operational Support Needs	A		I		C					C	R	
6.4.2 Confirm Operational Support Team	A						R				R	
6.4.3 Assign Operational Support Roles & Duties	A		C		C		R				I	
6.4.4 Develop Service Level Agreements (SLA)	A						R			C	R	
6.4.5 Develop Operating Level Agreements (OLA)	A						R				R	
6.4.6 Document Operational Support Plan	A										R	
6.5. Develop Legacy Retirement Plan												
6.5.1 Assess Legacy Systems Impact	A		R		C							C
6.5.2 Build Legacy Retirement Plan	A		R		C							C
6.6. Develop System Deployment Plan												
6.6.1 Assess Organizational Change Management Impact	A		R					I	I	R	C	C
6.6.2 Define Training Approach	A		C		C		R		I	R		C
6.6.3 Design Training Curriculum	A		C		C		R		I	R		C
6.6.4 Develop Training Materials	A		C		C		R		I	R		C

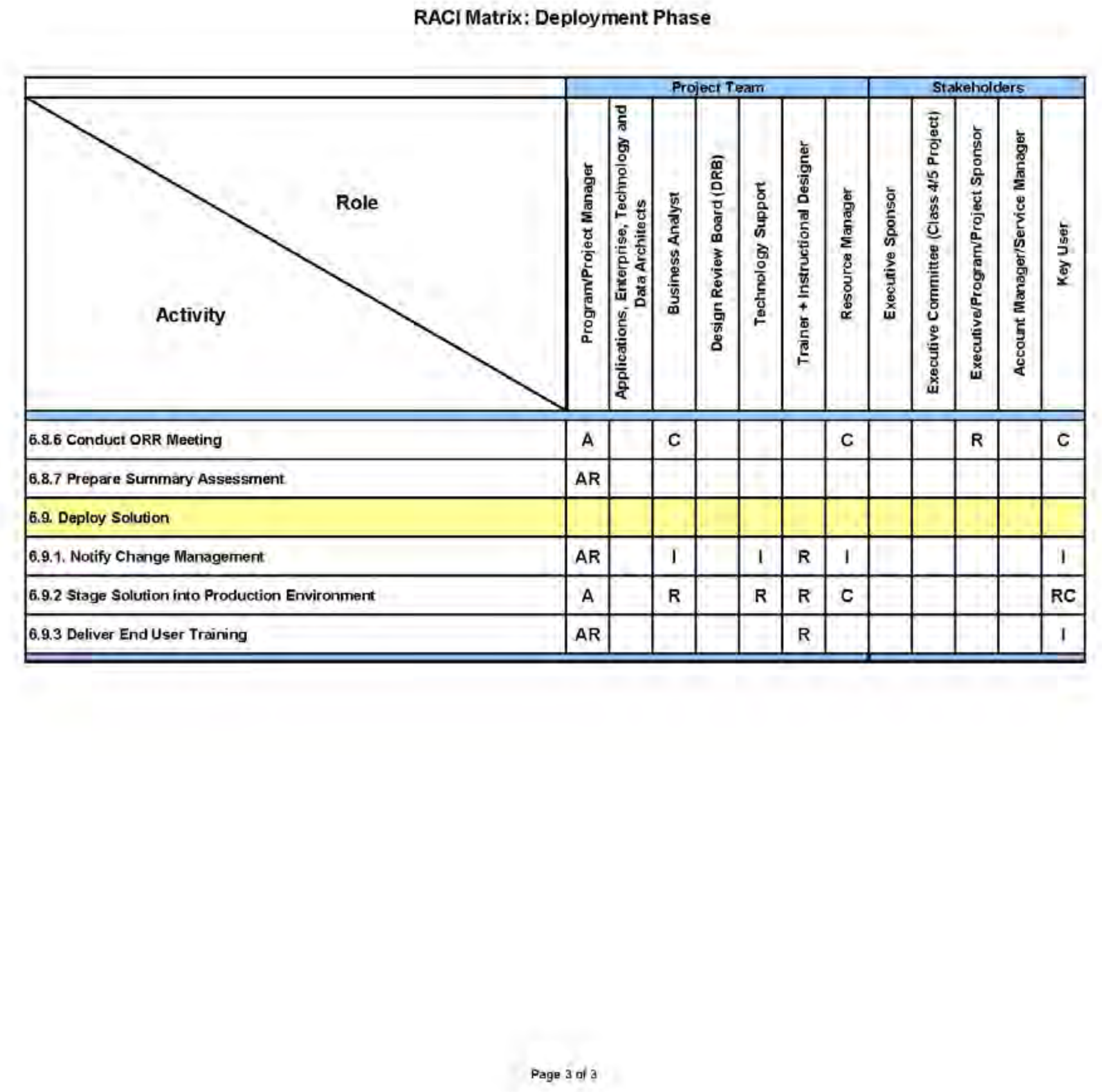
Page 1 of 3

RACI Matrix: Deployment Phase



Role / Activity	Project Team							Stakeholders				
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Business Analyst	Design Review Board (DRB)	Technology Support	Trainer + Instructional Designer	Resource Manager	Executive Sponsor	Executive Committee (Class 4/5 Project)	Executive/Program/Project Sponsor	Account Manager/Service Manager	Key User
6.6.5 Develop Communications Plan	AR								R	R	R	
6.6.6 Build System Deployment Plan	A		R		C		R	I	I	I	R	R
6.7. Go Live Check												
6.7.1 Confirm Go Live Team	AR		C	I	C		C				C	C
6.7.2 Schedule Go Live Checklist Review Meeting	AR											
6.7.3 Conduct Go Live Check Meeting												
6.7.4 Address Checklist Issues	A		R		R				I	I		R
6.7.5 Notify Change Management	AR											
6.7.6 Prepare Help Desk for System Deployment	A		R		I							R
6.8. Conduct Operational Readiness Review (ORR)												
6.8.1 Collect ORR Prerequisites and Inputs	A		R		R		C				R	R
6.8.2 Prepare ORR Review Packet	AR		C		C		C				C	C
6.8.3 Review ORR Packet and Complete Checklist	A		C				C			R		C
6.8.4 Develop Risk Assessment & Mitigation Plan	A		R		R					C		R
6.8.5 Review Assessments & Checklist with Authors	A		R		R		C			C	C	R

Page 2 of 3



RACI Matrix: Deployment Phase

Role / Activity	Project Team							Stakeholders				
	Program/Project Manager	Applications, Enterprise, Technology and Data Architects	Business Analyst	Design Review Board (DRB)	Technology Support	Trainer + Instructional Designer	Resource Manager	Executive Sponsor	Executive Committee (Class 4/5 Project)	Executive/Program/Project Sponsor	Account Manager/Service Manager	Key User
6.8.6 Conduct ORR Meeting	A		C				C			R		C
6.8.7 Prepare Summary Assessment	AR											
6.9. Deploy Solution												
6.9.1. Notify Change Management	AR		I		I	R	I					I
6.9.2 Stage Solution into Production Environment	A		R		R	R	C					RC
6.9.3 Deliver End User Training	AR					R						I

Page 3 of 3