SECTION THREE

III

System Development Lifecycle

SECTION III: SYSTEM DEVELOPMENT LIFECYCLE
TABLE OF CONTENTS

Introduction 3

1. SYSTEM INITIATION 15

1.1 Prepare for System Initiation 18
1.2 Validate Proposed Solution 19
1.3 Develop System Schedule 23
Measurements of Success 25
Phase Risks/Ways to Avoid Pitfalls 26

2. SYSTEM REQUIREMENTS ANALYSIS 31

2.1 Prepare for System Requirements Analysis 36
2.2 Determine Business Requirements 38
2.3 Define Process Model 49
2.4 Define Logical Data Model 51
2.5 Reconcile Business Requirements with Models 54
2.6 Produce Functional Specification 56
Measurements of Success 63
Phase Risks/Ways to Avoid Pitfalls 64

3. SYSTEM DESIGN 71

3.1 Prepare for System Design 76
3.2 Define Technical Architecture 78
3.3 Define System Standards 85
3.4 Create Physical Database 92
3.5 Prototype System Components 94
3.6 Produce Technical Specifications 97
Measurements of Success 120
Phase Risks/Ways to Avoid Pitfalls 121

4. SYSTEM CONSTRUCTION 129

4.1 Prepare for System Construction 134
4.2 Refine System Standards 136
4.3 Build, Test and Validate (BTV) 137
4.4 Conduct Integration and System Testing 142
4.5 Produce User and Training Materials 147
4.6 Produce Technical Documentation 148
Measurements of Success 149
Phase Risks/Ways to Avoid Pitfalls 151

5. SYSTEM ACCEPTANCE 157

5.1 Prepare for System Acceptance 162
5.2 Validate Data Initialization and Conversion 163
5.3 Test, Identify, Evaluate, React (TIER) 165
5.4 Refine Supporting Materials 170
Measurements of Success 171
Phase Risks/Ways to Avoid Pitfalls 172

6. SYSTEM IMPLEMENTATION 177

6.1 Prepare for System Implementation 181
6.2 Deploy System 183
6.3 Transition to Performing Organization 187
Measurements of Success 188
Phase Risks/Ways to Avoid Pitfalls 189

Section III Introduction

There are currently many different methodologies employed for system development projects within New York State agencies. Many methodologies are driven by the application development tools, by the software architecture within which the application will operate, or by the "build versus buy" decision. There are standard phases and processes, however, that all system development projects should follow, regardless of environment and tools. This section describes the standard phases and major processes of the New York State System Development Lifecycle (SDLC), using a common language and in sufficient detail to enable a Project Manager to plan and manage a system development project.

System Development Lifecycle Overview

The material in this section is organized according to a generic system development lifecycle. While no two development efforts are exactly alike, all projects should progress through the same six phases:

1. **System Initiation** – in which the Business Case and Proposed Solution developed during Project Origination are re-examined to ensure that they are still appropriately defined and address an existing organizational need. This validation effort provides the Project Team with the basis for a detailed schedule defining the steps needed to obtain a thorough understanding of the business requirements and an initial view of staffing needs. In addition, a high level schedule is developed for subsequent system development lifecycle phases.

2. **System Requirements Analysis** – in which the needs of the business are captured in as much detail as possible. The Project Manager leads the Project Team in working with the Customers to define what it is that the new system must do. By obtaining a detailed and comprehensive understanding of the business requirements, the Project Team can develop the Functional Specification that will drive the system design.

3. **System Design** – which builds upon the work performed during System Requirements Analysis, and results in a translation of the functional requirements into a complete technical solution. This solution dictates the technical architecture, standards, specifications and strategies to be followed throughout the building, testing, and implementation of the system. The completion of System Design also marks the point in the project at which the Project Manager should be able to plan, in detail, all future project phases.

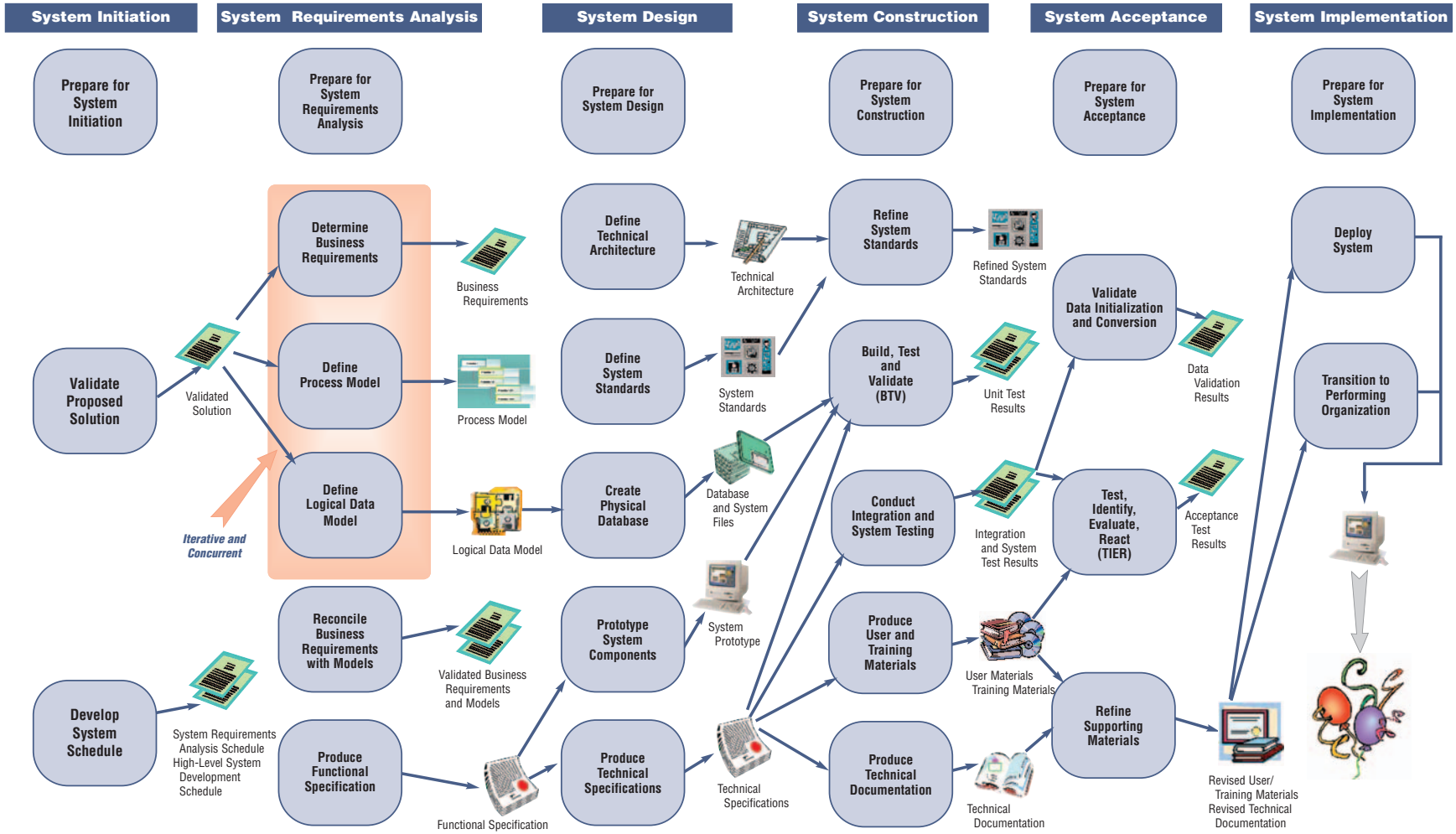
4. **System Construction** – throughout which the Project Team builds and tests the various modules of the application, including any utilities that will be needed during System Acceptance and System Implementation. As system components are built, they will be tested both individually and in logically related and integrated groupings until such time as a full system test has been performed to validate functionality. Documentation and training materials are also developed during this phase.

5. **System Acceptance** – during which the focus of system validation efforts shifts from those team members responsible for developing the application to those who will ultimately use the system in the execution of their daily responsibilities. In addition to confirming that the system meets functional expectations, activities are aimed at validating all aspects of data conversion and system deployment.

6. **System Implementation** – the final phase of the lifecycle, which comprises all activities associated with the deployment of the application. These efforts include training, installation of the system in a production setting, and transition of ownership of the application from the Project Team to the Performing Organization.

The following diagram illustrates every phase, process and deliverable in the system development lifecycle.

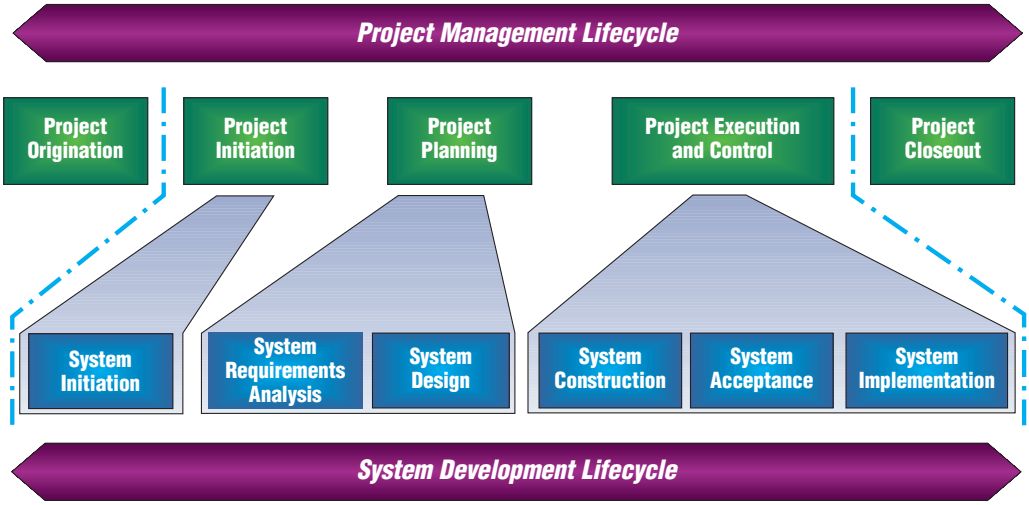
Figure 0-1 *NYS Project Management Guidebook* The System Development Lifecycle



Mapping the Project Management and System Development Lifecycles

The phases of the system development lifecycle generally align with the phases of the project management lifecycle; however, SDLC phases do not correspond one-to-one with the project management phases. One of the challenges for system development projects is aligning the SDLC with the project management lifecycle. The following diagram demonstrates how the phases of the two lifecycles may be integrated.

Figure 0-2



In reality, each phase of the SDLC can be thought of as a mini-project in itself, requiring planning, execution, and analysis. As the Project Team proceeds through the project, they will need to create a clear and detailed plan for the phase immediately in front of them, along with a higher-level view of all remaining phases. As the team executes each phase, they will collect additional information that will enable the detailed planning of subsequent phases. Some of this information will be a natural by-product of having performed the processes associated with the current phase (e.g., as the detailed technical design evolves throughout the System Design phase, the team will have a much better understanding of the modules that will need to be built during construction, and will therefore be able to refine any prior estimates and plans for System Construction). Additional information can be obtained through a focused

analysis effort, performed at the completion of each phase. This assessment is analogous in many respects to conducting the Post-Implementation Review as described in Section I, Project Closeout, although it is typically conducted in a less formal fashion. The responsibilities of the Project Manager include assessing how closely the phase met Customer needs, highlighting those aspects of the phase that worked well, identifying lessons learned and best practices in an attempt to derive ways to improve upon processes executed throughout the project, and, most importantly, communicating results.

The SDLC defined in this section may appear to have characteristics of a classic "waterfall" approach, which assumes that each phase and process is completed and agreed upon before the next phase begins. The reality is, however, that phases generally overlap, with each successive phase introducing changes to the work of the prior phase, resulting in an iterative process.

This SDLC is also consistent with newer techniques for system development, such as Rapid Application Development (RAD). RAD allows users to participate in an iterative design and development process. Conceptually, the project "loops" through the Design, Construction, and Acceptance phases, followed by re-Design, revised Construction, Acceptance, and so on. Project management deliverables such as the Project Scope Statement, Project Schedule, and budget estimates are refined to reflect increasing clarity of scope and requirements with each iteration.

While there is the potential to compress Requirements Analysis, Design, and Construction in RAD approaches, compression introduces increased risks. It is important, therefore, to include risk analysis in each iteration of the design, build, and evaluate loop. When a prototype is presented, Project Managers must actively and diligently address the management of Customer expectations and the maintenance of current documentation.

The RAD approach has advantages, since it usually achieves results quickly, the design is less abstract, and users have assurance that up-to-date requirements are considered. Its disadvantages include difficulty in controlling the process and ensuring the creation of an acceptable product.

Many factors may impact your choice of approach to follow when developing a system. The better you know your Customers and Stakeholders, and the better you understand the factors that may influence their assessment of the project, the more likely it will be that your approach will suit their personalities, preferences, vision, and needs.

The key is to pick the approach that you believe will provide the best complete solution, balancing the preferences of your Customers, the abilities of your Project Team, and the overall business drivers (legislated timeframes, overall time to market, etc.).

In any approach, the basic SDLC processes must be performed – what differs is the timing of their execution. As with the project management methodology, if processes or deliverables are skipped, the Project Manager must record the reasons why, and must describe how the objectives of that process/deliverable will otherwise be met.

Understanding the Breadth of System Development Projects

When assessing the scope of a system development project, it is important that the needs, goals, and challenges of the project are understood from many perspectives. The ***business requirements***, which define the high-level Customer objectives and vision for the system, are used to determine the scope of the system. When capturing the business requirements, it is essential that the Project Team look at all aspects of the system, including:

- **Functional Requirements** – describing processes and tasks that the Consumer must be able to accomplish through the use of the system. These can typically be categorized as processes that require action on the part of Consumers (data entry, selection of a system command, etc.), and those that are not directly related to human interaction with the system (for example, off-hours processing or the automated exchange of information between systems).
- **Technical Requirements** – identifying technical aspects and constraints that must be considered when defining the new system. Considerations may include accessibility needs of Consumers, whether or not the storage and

handling of data must follow specified encryption regulations, or whether the system will operate on internal agency hardware or will be hosted at either an internal or external data center.

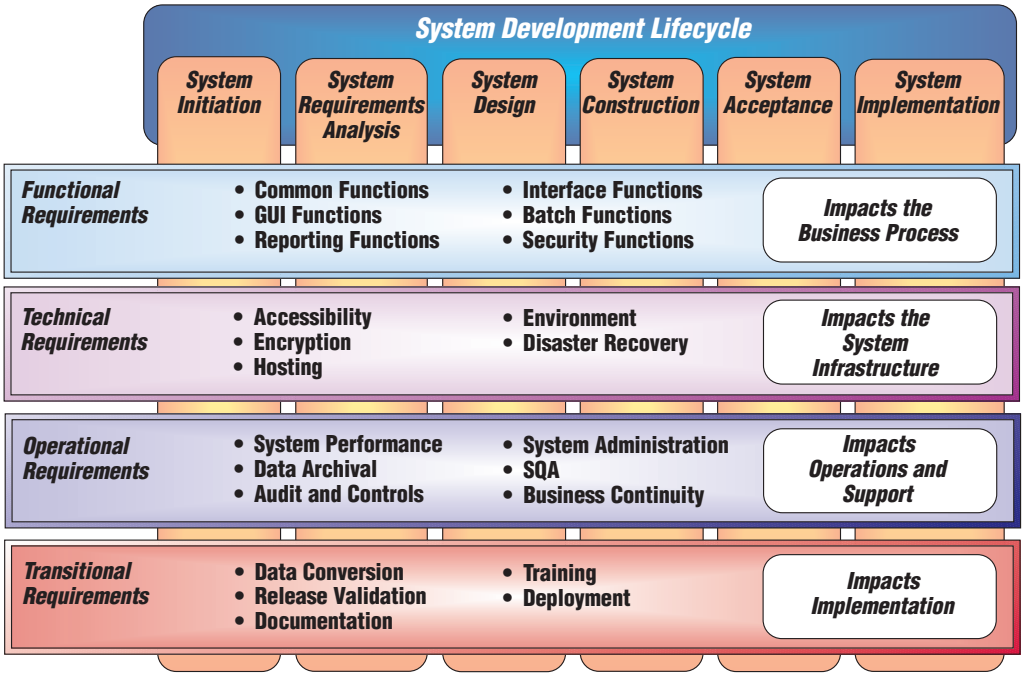
- **Operational Requirements** – specifying any administrative constraints or expectations that must be supported by the system. These requirements may include system performance expectations, technical infrastructure constraints, security mechanisms that must be followed, the need to regularly archive data, and any mandated audit and control processes.
- **Transitional Requirements** – defining the realm of conditions that must be satisfied prior to physically implementing the system in a production environment, or to relegating support responsibilities to the Performing Organization. Data conversion requirements and development and delivery of Consumer training programs and materials fall into this category.

Formally organizing thoughts along these four dimensions will drive the identification of tasks to be performed beginning in System Initiation and continuing throughout the lifecycle. The Project Manager is responsible for creating this broad view of requirements and communicating it to the Project Team, establishing a pattern that should be carried throughout all project phases.

The following diagram illustrates some representative categories of requirements that the team should consider when defining tasks and activities throughout the system development project.

Figure 0-3

Representative SDLC Considerations



It should be noted that not all considerations may be applicable to every project, and additional categories may be discovered that are not represented in the diagram. The fundamental point is that for those considerations that do apply to the project, there will be corresponding activities required throughout each and every phase of the SDLC, all contributing to the eventual implementation of the system. Regardless of whether the Project Team is performing System Initiation, Requirements Analysis, Design, Construction, Acceptance, or Implementation activities, they will need to understand and address the full realm of functional, technical, operational, and transitional requirements to ensure a successful project. In an attempt to reinforce this point, this diagram will be revisited in each of the individual SDLC phases that follow, drawing specific references to the processes relevant to each phase of the lifecycle.

Software Quality Assurance

In the way that project management provides the umbrella under which all project activities are directed, software quality assurance provides the foundation on which all system development activities should occur so that the highest quality system possible will be delivered. According to the IEEE Standard Glossary of Software Engineering Terminology, quality is defined as the degree to which a system, component, or process meets specified requirements and Customer needs and expectations.

As will be stressed throughout the following chapters, it should be noted that simply meeting requirements is not enough to guarantee a successful system development effort. Ultimately, Customer needs and expectations can be met only if the requirements are fully and correctly captured in the first place.

Analogous to the Quality Assurance Plan associated with the project management lifecycle, software quality assurance programs should be comprised of three components – quality standards, quality assurance processes, and quality controls.

Software Quality Standards define the programming standards, and development/testing standards to be followed throughout the project.

Software Quality Assurance Processes define practices and procedures to be used by the Project Team to meet the quality standards, and to provide management with evidence that these procedures are being followed.

Software Quality Controls comprise a series of reviews and audits that evaluate deliverables with respect to defined standards and acceptance criteria. These controls include software testing techniques and peer reviews.

The key to these SQA efforts is that they must be performed throughout all phases of the project. In addition, all SQA efforts should ideally be performed by a third party, independent from the team members responsible for delivering the system. Availability of staff and budget are two factors that must be considered in determining the feasibility of applying an independent SQA Analyst or team to the project. In developing the

overall system development plan, the Project Manager needs to allocate sufficient time and resources to perform the appropriate level of SQA activities, and must obtain management commitment to providing these resources as called for in the Project Schedule.

Project Roles and Responsibilities

As presented in the Section I Introduction, Project Roles and Responsibilities, there are many groups of people involved in both the project and project management lifecycles. When staffing system development projects, there are a number of roles that should be considered. It should be noted that the SDLC only provides details to the phase and process level, whereas the PM lifecycle further decomposes activities down to individual tasks. As a result, while the roles identified within the SDLC are representative of those that are typically required in a system development effort, the function of the role as it relates to a given SDLC process may not be specifically described within that process narrative.

The **Project Team** consists of a Project Manager and a variable number of Project Team members who are responsible for planning and executing the project. Team members specific to the System Development Lifecycle are described below.

The **Facilitator** leads sessions to identify business requirements and issues, keeps sessions focused and productive, draws out issues and ideas from all participants, and maintains clear and open communications within the session.

The **Business Analyst** effectively leads discussions with the Customers to determine the business requirements, participates in preparing the data and process models, prepares module specifications, test data, and user documentation materials, assists in prototyping activities, and develops strategies for testing and implementation.

The **Database Administrator** is responsible for providing and maintaining database administration policies and procedures, approving and executing database scripts, performing database tuning activities, and transforming a pictorial representation of the system data (the Logical Data Model) into physical database tables that support the final system.

The **Data/Process Modeler** develops and maintains data and process models to represent the business information needs in the area under study, develops and defines the data dictionary, validates models with the Customers, and participates in prototyping.

The **Technical Lead/Architect** drives the logical process and data models into an application architecture, establishes architecture guidelines, and develops strategies for the creation and distribution of applications.

Application Developers include all those responsible for developing prototypes, technical specifications, and application code, and for executing test scripts.

The **Software Quality Assurance (SQA) Analyst** is responsible for establishing and executing the Quality Assurance Plan, for assisting in the preparation of test scripts and test data, and for participating in integration and acceptance testing efforts.

Technical Services (HW/SW, LAN/WAN, TelCom) include all those responsible for the ordering, installation and maintainence of hardware and software components, LAN/WAN components and telecommunications components.

The **Information Security Officer (ISO)** is responsible for identifying and enforcing security standards and processes.

Technical Support (Help Desk, Project Administration, Documentation, Trainers) includes all those responsible for supporting the development of the new system. Support includes the documentation of user, training, operation materials, and help files, training for Customers, responding to technical and business questions forwarded to the Help Desk, and supporting the project and associated administrative processes.

Figure 0-4 New York State System Development Life Cycle Templates

Phase	Template	Description	Page in Text	Page in Appendix
System Requirements Analysis	Business Requirements Document	A document containing detailed functional, technical, operational and transitional requirements for the system being developed	45	99
System Requirements Analysis	Functional Specification	A document describing the logical grouping of related processes and the mapping of those processes to business requirements and data items.	59	103
System Design	Technical Architecture	A document describing the system architecture in terms of hardware, software, tools and peripherals, and the distribution of system components and processes across this architecture.	81	109
System Design	System Standards	A document detailing the standards to be applied and adhered to throughout the project.	87	115
System Design	Technical Specifications	A compilation of system diagrams, module specifications, and test plans that serve as a detailed, comprehensive blueprint for the system.	109	121
System Construction	Defect Log	A document used to log defects encountered when performing integration, system, data validation or acceptance testing, and track their resolution.	145	131

1 SYSTEM INITIATION

Purpose

The purpose of **System Initiation** is to validate the Proposed Solution developed during the Project Origination phase of the Project Management Lifecycle, and to estimate the system development effort in greater detail. In this phase, the broad parameters of the new system are defined, and applicable system development activities are identified.

Once the overall approach has been confirmed, it is necessary to estimate the effort and resources required for the next phase in elemental detail, and to provide high-level estimates for subsequent phases, to the extent necessary to support the project management lifecycle deliverables and activities of Project Initiation.

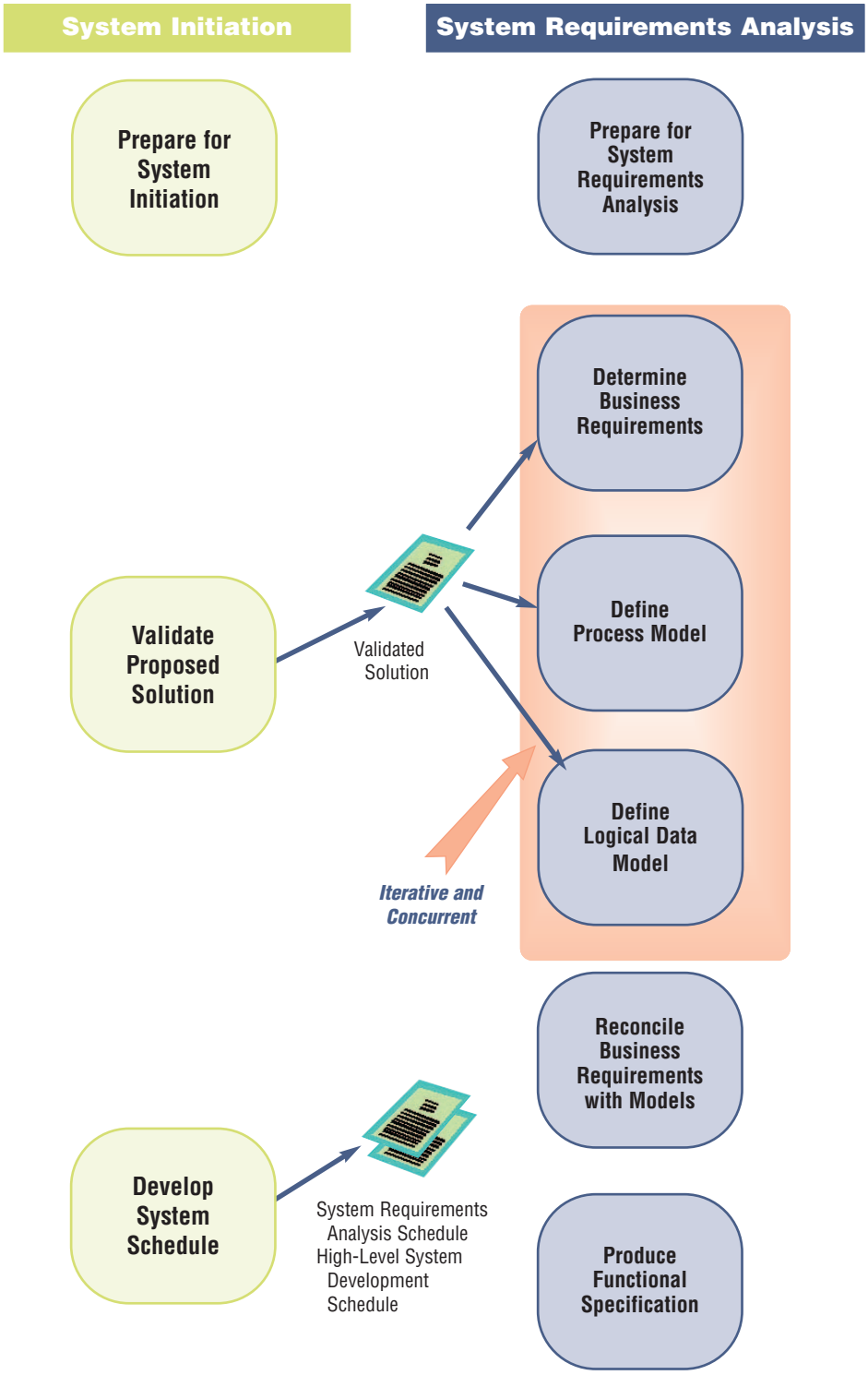
List of Processes

This phase consists of the following processes:

- **Prepare for System Initiation**, where the initial members of the Project Team familiarize themselves with the project's defining documents and plan the activities for the rest of the phase;
- **Validate Proposed Solution**, where the original technology direction and system development approach are validated;
- **Develop System Schedule**, where a detailed System Requirements Analysis schedule is developed, and a high-level system development schedule is produced.

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

Figure 1-1


System Initiation
System Requirements Analysis
Prepare for System Initiation
Prepare for System Requirements Analysis
Determine Business Requirements
Define Process Model
Define Logical Data Model
Validate Proposed Solution
Validated Solution
Iterative and Concurrent
Reconcile Business Requirements with Models
Develop System Schedule
System Requirements Analysis Schedule
High-Level System Development Schedule
Produce Functional Specification

List of Roles

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

- Project Manager
- Project Sponsor
- Business Analyst
- Technical Lead

List of Deliverables

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

Figure 1-2

Processes	Techniques	Process Deliverables *(Outcomes)*
Prepare for System Initiation	Interviews Document Gathering and Reviews	*Established Team and Environment for System Initiation*
Validate Proposed Solution	Brainstorming Research	Validated Solution
Develop System Schedule	Brainstorming Research Estimating	System Requirements Analysis Schedule High-Level System Development Schedule

1.1 PREPARE FOR SYSTEM INITIATION

Purpose

The purpose of **Prepare for System Initiation** is to ensure that the Project Team, and the environment in which it will operate, are ready for successful completion of this phase.

Roles

- Project Manager
- Project Sponsor
- Business Analyst
- Technical Lead

Description

In addition to the Project Manager, Business Analyst and Technical Lead roles are required to complete the team for this phase.

Identification, assignment and orientation of new team members required for this phase are activities in Project Initiation in the project management lifecycle.

Environment preparation includes gathering all relevant project and historical documentation, and placing it in the document repository. At a minimum, the Project Team should have available the Project Proposal (consisting of the Business Case and Proposed Solution), and any relevant evaluation and decision documentation. Any historical data, such as best practices or performance statistics from similar earlier efforts, can also serve to guide the Project Team towards – or away from – certain solutions.

A record of prior efforts to develop a similar system (including current system documentation, if it is a replacement) can also be very helpful, although you should take care not to pre-judge the approach either because of, or in spite of, prior experiences.

1.2 VALIDATE PROPOSED SOLUTION

Purpose

The purpose of **Validate Proposed Solution** is to make sure that the original technology decision and system development approach still represent the optimal solution for the identified business need.

Description

Considerable time may have elapsed since the Project Proposal (and its constituent Proposed Solution) was developed, due to the vagaries of the budget process or to other procedural delays. In the interim, the Performing Organization may have changed its course and the state-of-the-art technology may also have changed significantly. With rapid advances in technology, it is certain that the longer the period of time between the original proposal and the commencement of System Initiation, the more likely it is that the chosen technology is no longer supported, has become obsolete, or commands a vanishing talent pool.

Roles

- Project Manager
- Project Sponsor
- Business Analyst
- Technical Lead

If System Initiation closely follows the development of the Project Proposal, it may not be necessary for the Project Team to perform all parts of this validation process.

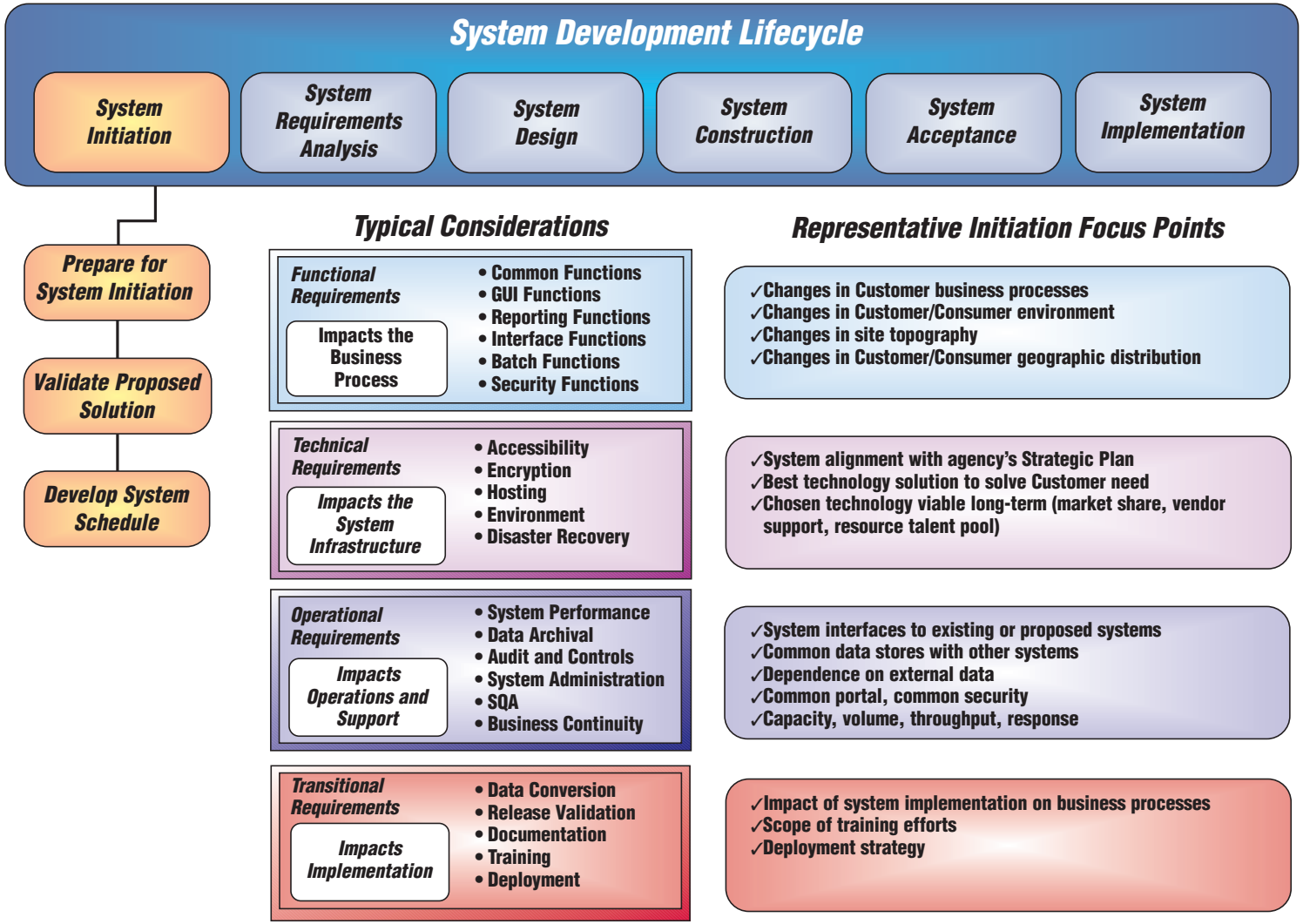
To validate the Proposed Solution, the Project Team must:

- understand the agency's current Strategic Plan, and how the new system fits into it;
- assess the proposed technology solution in view of Customer needs, the Performing Organization's long term technology direction and constraints, and state-of-the-art technology; and
- confirm feasibility of the proposed system development approach.

In assessing the Proposed Solution, the team must consider how it fits into the Performing Organization's application portfolio of existing or proposed systems. A System Context Diagram can be used to illustrate how the new system will make use of, or add to, existing major data stores, and how it will interface with other systems identified in the Strategic Plan (extract, update, data entry, etc.).

The next step is to confirm that the Proposed Solution is still the best option for the current set of business needs and conditions (as they are reflected in the Project Charter project management deliverable). For example, reorganization may have dispersed Consumers over a large geographical area, necessitating a re-evaluation of the originally proposed technology that was best suited to many Consumers in close physical proximity to one another.

Figure 1-3 System Initiation Considerations



In deciding whether the proposed technology direction represents an industry trend or a dead end, there are numerous professional journals available by subscription or free on the Web. There are forward-looking reports by organizations such as Gartner, Inc. or the Meta Group, Inc., and many consulting companies can offer valuable advice. The NYS Office for Technology can also serve as an authoritative point of reference.

Once it has been determined that the proposed technical solution fits into the Performing Organization's Strategic Plan, any lingering questions may be resolved through formal reviews, or directed to the Project Sponsor.

If it becomes apparent that the original Proposed Solution is no longer the optimal one, the team should propose an alternative solution. The Project Sponsor will then direct the team to proceed with the original solution, to take the alternative proposal, or to take action such as terminating the project or repeating the project management lifecycle starting at an earlier process or phase.

Finally, the system development approach needs to be validated against the latest understanding of both the business needs and the technology solution. Certain decisions must be considered even if they cannot yet be made. Among them are:

- Should the system be developed in-house or acquired as a Custom Off-The-Shelf (COTS) solution?
- Are there available resources to develop/customize the system, or is it necessary to contract for additional resources?
- Is the choice of technology platform predicated on the existing environment, or does the system offer an opportunity to upgrade the infrastructure?

If the Proposed Solution involves using infrastructure and/or services provided by the NYS Office for Technology, early notification to OFT is necessary to ensure smooth integration with planned service upgrades and other service demands. The Office for Technology may also be able to provide valuable contacts in other agencies where systems similar in function or technical components have been developed.

Deliverable

- **Validated Solution** – The team should update the original Project Proposal, or recreate the Proposed Solution using the template from Section I, Project Management Lifecycle.

1.3 DEVELOP SYSTEM SCHEDULE

Purpose

The purpose of **Develop System Schedule** is to create a detailed schedule for System Requirements Analysis and a high-level schedule for the remaining phases.

Description

After the technical solution has been validated, it is possible to decide how the rest of the System Development Lifecycle will be applied to this particular system development effort.

Roles

- Project Manager
- Project Sponsor
- Business Analyst
- Technical Lead

Using a project scheduling tool of choice and referring to the chart of the System Development Lifecycle, the Project Manager should populate the project with the phases and processes and the suggested roles. Then the Project Team, including the Business Analyst and the Technical Lead, should walk through and brainstorm each process, attempting to answer as well as possible the following questions:

- What are the system development deliverables in this process?
- What are the tasks necessary to produce this deliverable?
- Is there a logical way to organize the tasks associated with this process for this system? That is, are the modules already defined? Will they be worked on in parallel or serially?
- How complex is each task likely to be?
- What skills are required to perform each task?
- How many resources with each identified role/skill set will be needed to produce the deliverables?

In addition to thinking through the deliverables necessary for implementing the desired functionality, the team should consider the technical, operational and transitional requirements of the system (refer to Figure 1-3). These additional requirements will influence the definition of necessary tasks and the tasks' order or complexity.

This scheduling process needs to be performed to an elementary level of detail for the next phase, but only at a high level for the subsequent phases. The goal of the high-level estimating process is not premature precision, but rather a coherent, purposeful system development strategy that will form the basis for subsequent efforts (understanding that it will nevertheless be changed, augmented and enhanced throughout the lifecycle.)

At the end of System Initiation, all system-related materials gathered and produced by the Project Team should be consolidated and prepared for use as input for the next phase.

Deliverables

- **System Requirements Analysis Schedule** – Task-level schedule for the System Requirements Analysis phase of the System Development Lifecycle.
- **High-Level System Development Schedule** – Process-level schedule for the remaining System Development Lifecycle phases.

Both the detailed and high-level schedules produced in this process are part of, and are integrated into, the High-Level Project Schedule deliverable of the Project Initiation phase of the Project Management Lifecycle.

Measurements of Success

The success of this phase is measured by how readily the team can perform the next phase. The schedule for System Requirements Analysis should be immediately executable by the team.

The Project Manager can assess how successfully the project is proceeding by utilizing the measurement criteria outlined below. More than one "No" answer indicates a serious risk to the next phase and the eventual success of the system.

Figure 1-4

Process	Measurements of Success	Yes	No
Prepare for System Initiation	Have you obtained the materials that describe (1) the business needs and benefits, by functional unit; (2) the proposed solution; (3) the reasons that this project and this solution were selected for initiation?		
Validate Proposed Solution	Does the head of the information technology organization (or designate) agree that the system solution fits into the Strategic Plan?		
Develop System Schedule	Have the standard development procedures been customized for the specific system components defined for this system, including functional, technical, operational and transitional requirements?		
	Do you have management commitments for team member availability for the next phase?		
	In the High-Level System Development Schedule, do you know if the effort allocated to system development phases correlate to industry-accepted norms?		

Phase Risks / Ways to Avoid Pitfalls

PITFALL #1 – PLAYING WITH TOYS

You read the original Proposed Solution, and it seems very astute and on the ball: it promotes the latest technology trend, and is full of impressive-sounding concepts.

You consult with your technology folks, and they enthusiastically endorse the approach: it's the latest rage, and they just can't wait to get their hands on those great new packages. In fact, they will use your system as the showcase of the new technology, and allocate all kinds of resources to make it succeed. All of them will just be delighted to work on your system.

Their enthusiasm is contagious, and you consider yourself one lucky Project Manager. Until, that is, you realize that none of them actually knows how the new technology works; not that it stops them from trying to use it, all in their different ways, until they make a total mess of your "sandbox".

People like gadgets and gismos, buzzwords and catch phrases, and the technology folks like them most of all. They are like a bunch of little kids, going all googly-eyed at a new shiny toy.

So before you agree to play with a new solution, make sure that somebody convinces you that the tried-and-true really won't work here, and that you need to take on all the risks of the new technology. Do your own research, consult with outside experts, talk to the managers who've implemented technology projects successfully in the past, do the risk vs. reward analysis, and dull that "bleeding edge".

PITFALL #2– UNDERESTIMATING REQUIREMENTS ANALYSIS

If you have worked in your organization for a long time, you may have dealt with your Customers long enough to know what they want better than they know it themselves. In your mind, it all seems so clear – they need access to this data, here's how they will want to see it broken down, and this is the type of interface they are most comfortable with. In fact, you are probably already constructing the screens and reports in your mind, visualizing their reaction to this great new technological advance. The real challenge seems to be in the development, while the requirements analysis effort is often seen as a

straightforward exercise – get together with some key individuals, tell them what the new system will do for them, and you are done.

The reality, of course, could – and should – be very different. The estimated effort includes not only identifying the requirements of ALL interested parties, but also documenting and reconciling them, to a point where the Functional Specification could be passed to another team altogether and still result in an accepted product.

Frequently Asked Questions

How do I map the Project Team roles identified in the Section I, the Project Management Lifecycle, to the people working on the project? Do I need that many people on the project?

First of all, you need to understand that a role does not necessarily translate to an FTE. Depending on the size and complexity of the system you are trying to develop, the size and geographical distribution of your Customer and Consumer base, and finally on the versatility of your Project Team, you may need many people to fill one role, or you may have one person fill many roles. The following steps should help you map available people to required roles:

1. Determine Project Team requirements. Using the High-Level Project Schedule, estimate which roles are going to be required throughout the duration of the project, and to what extent.
2. Understand Stakeholder landscape. Using the Description of Stakeholder Involvement (contained in the Project Plan), refine your list of required roles, adjusting for approach (individual vs. group requirements meetings, individual vs. classroom training), geography (all Customers in one building vs. multiple facilities throughout the state) and size (a handful of Customers who are also Consumers vs. scores of representatives from competing Customer and Consumer groups.)
3. Research your team capabilities. Understand people's skills, interests and proclivities. Document their work hours and availability to the project.

4. Map roles to people using all information available. Document and address any apparent gaps.

I have a few skeptical Customers who, because they've been around for a while, have the SME (Subject Matter Expert) status on my project. However, they've never participated in a formal system development effort before. How do I get them to contribute in a productive fashion?

The good old carrot and stick approach should work: on the one hand, appeal to their self-interest and flatter them immoderately; on the other, make sure their manager is on board with the project and is aware of their expected contributions.

You should be able to convincingly demonstrate to any Customer (and most Consumers) why and how the new system will benefit them; in addition, you should genuinely appreciate (and praise!) their knowledge of the business process.

Having an experienced Facilitator and/or Business Analyst is extremely helpful as well, to engage all participants in a constructive dialog and set realistic expectations for future contributions.

What are the go/no go points in the SDLC? How do they integrate with decision points in the project management lifecycle?

The decisions to proceed with, or to halt, the system development effort properly belong in the project management lifecycle. To that end, the final process in each of its first three phases (Project Origination, Project Initiation and Project Planning) contains an explicit go/no go decision point. However, even before the project comes to those points, it is entirely possible that the system development lifecycle will necessitate project go/no go decisions.

The first such event may occur in System Initiation. If it is determined that the original Proposed Solution is no longer adequate or desirable, it may be prudent to halt the system development process (and the project) altogether, or re-initialize the project back at the Project Origination phase.

The next event may occur during System Requirements Analysis. If the business requirements gathered during this phase push the scope of the project way beyond the initial estimate, it may be necessary to halt Project Planning activities until the amended scope, schedule and budget are approved, or the decision is made to terminate the project.

The same is true for System Design. Insurmountable technical difficulties or irreconcilable differences over the prototype may jeopardize the success of the project, disrupting the flow of Project Planning and forcing a go/no go decision.

Finally, the controls put in place for Project Execution and Control may be triggered by difficulties with System Construction, Acceptance and even Implementation activities, and an abrupt end (or reiteration) of the Project Management Lifecycle may occur as a result.

How do I estimate the System Requirements Analysis phase?

The first thing to remember is not to do it in a vacuum. Use your entire team to flesh out the answer.

Start by decomposing the SDLC and creating the Work Breakdown Structure for the System Requirements Analysis phase. Consider the Stakeholder landscape and your team's capabilities and availability (see the first question above) and map out to whom you will be talking, when, how many times, and for how long. Then, based upon your (and your team's) knowledge of the business environment, estimate how much effort – and time – you will need to come up with all the System Requirements Analysis deliverables. Be as precise as possible, decomposing each process into elementary tasks, and each deliverable into its constituent (or pre-requisite) work products.

Is this SDLC appropriate for outsourced/contracted out engagements? How do I know what system development methodology my vendors will use?

The whole point of creating common project management and system development methodologies for New York State is to have a consistent approach to system development on ALL engagements. Not only does it streamline planning and execution and enable state Project Managers (and Project Team

members) to move within and among state agencies with a minimal learning curve, it also provides a standard for agency staff to use when contracting with private vendors. The state can now provide the methodology for its contractors, and direct them to adhere to it, instead of requiring New York State staff to adjust to the different development methodologies of each firm with whom they contract.

This SDLC is designed to be generic enough for virtually all system development efforts, and allows utilization of nearly all platforms, tools and techniques.