ADDENDUM 1

DATE: January 24, 2017

PROJECT: ITAMS Interface Rebuild

RFP NO: 744-R1804

OWNER: The University of Texas Health Science Center at Houston

TO: Prospective Bidders

I. The following are the University's responses to bidder's questions received by the Question Deadline - January 16, 2018, 2:00 PM CST.

1. How does UT define the new "user interface"? Is it related to functionality or the layout including the style?
   Reference:
   RFP Page 3:

   > UTHealth has a custom Information Technology Asset Management System (ITAMS) that is used for desktop asset management across the campus. Initially ITAMS was developed for a specific school as their desktop support Asset Management System, but it was later adopted as the primary tool for desktop asset management for the entire University. Due to the increase in scope, additional requirements, and existing bugs/issues in the current application, it has been determined that the ITAMS User Interface should be replaced. The scope of services provided by the Vendor under this RFP shall be the development of a new user interface which meets the requirements specified in this RFP to improve the overall user experience, streamline the business processes where possible, provide for accuracy and reliability of the data, and improve the stability of the ITAMS application.

   RFP Page 15:

   > The first part of Phase I of the project is to review and determine the requirements. This section covers the tasks involved for the Contractor to review and determine the necessary requirements for building the new ITAMS user interface. The UTHealth ITAMS project team has gathered much of the information for functional requirements of the application, including existing issues, and bugs as well as existing functionality that is acceptable to keep.

   RFP Page 16:

   > Generally speaking, UTHealth is happy with the layout of the existing ITAMS user interface. Aside from the critical issues identified, the overall base functionality of the application is acceptable.

   UTHealth is not unhappy with the aesthetics of the user interface but there are some serious bugs that need to be fixed with functionality as listed in the issues log and videos.

2. How much of the existing functionality will UT keep for the new application?

   All of the existing functionality should be kept however, many of the areas of the application do not work correctly or are cumbersome to achieve the task. For the scope of this project we are needing the

major bugs corrected. As discussed in the pre-bid conference and noted in the RFP, there are enhancements that are desired that would be determined in future work.

3. How much data volume is on the existing database?
   The database size is around 10G.
   There are total assets of 65,583. Active assets is 34,100.

4. Are there any workflow or procedure regarding the following status of an asset?
   ACTIVE, INACTIVE, ARCHIVED, DISABLE
   The only known workflow is that the system will automatically send a notification email to the administrator group.

5. There are a lot of '/portal.sph.uth.tmc.edu/assets/…' references in the coding, how do we locate them in the source code under ITAMS.Web project?

```
f="@Url.Content("//portal.sph.uth.tmc.edu/assets/plugins/bootstrap-daterangepicker/daterangepicker-bs3.css")" />
f="@Url.Content("//portal.sph.uth.tmc.edu/assets/plugins/bootstrap-datepicker/css/datepicker.css")" />
```

   This is a third party UI library called Metronic. The code is being provided and will be placed in the same location as the other source code.

6. There are some bundles in the ITMAS.Web project, that should be generated by Web Essentials, what is the version for Web Essentials?
   The current version of Web Essentials being used is v2015 utilized through Visual Studio 2015.

7. We cannot access the /ACL/Auth/Login due to missing the AclWcfServiceClient.dll file in ITAMS.Web project. Where can we find this file?
   This file will be provided in the same location as the other source code.

8. The price list states the following:

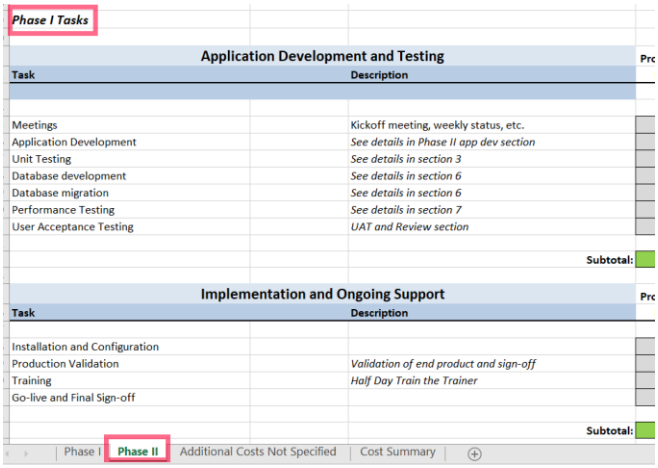
NOTE: UTHealth has estimated the personnel resources to require a Project Manager as well as an analyst and/or programmer.
This sheet has a predetermined amount of 4 personnel roles to allow for additional resources if needed. The first role has been p
Enter the title of the Personnel Role as well as the Hourly Rate below. Then, enter the estimated hours for each role and the cost
Any additional tasks not listed can be entered for Phase II on the "Additional Costs Not Specified" tab of the worksheet.

***PLEASE DO NOT ALTER ANY CALCULATED FIELDS OR ATTEMPT TO ENTER INFORMATION IN ANY FIELDS WITH A GREEN BACKGRO*

Personnel Role	Personnel Title	Hourly Rate (enter rate for each role assigned)
Role 1	Project Manager	
Role 2	Enter Title	
Role 3	Enter Title	
Role 4	Enter Title	

   Are the analyst and/or programmer to be considered as role 2 and 3 in the table?
   The Project Manager role is already listed since the project manager role is a required role as stated in the RFP. The additional role titles have been left blank to allow Contractors the ability to fill in the fields to bid on roles they see fit for the project such as analyst and/or programmer etc.

9. Would just like to confirm this refers to Phase 2 tasks?



Phase I Tasks

Application Development and Testing		Pro
Task	Description	
Meetings	Kickoff meeting, weekly status, etc.	
Application Development	*See details in Phase II app dev section*	
Unit Testing	*See details in section 3*	
Database development	*See details in section 6*	
Database migration	*See details in section 6*	
Performance Testing	*See details in section 7*	
User Acceptance Testing	*UAT and Review section*	
	Subtotal:	

Implementation and Ongoing Support		Pro
Task	Description	
Installation and Configuration		
Production Validation	*Validation of end product and sign-off*	
Training	*Half Day Train the Trainer*	
Go-live and Final Sign-off		
	Subtotal:	

Phase I | Phase II | Additional Costs Not Specified | Cost Summary

Yes. That is correct. That is a typo and should be corrected to say Phase II tasks.

10. Based on text in "Attachment+B_ITAMS-ArchDesign.docx" document there are two clients for the current implementation.
    - a hybrid ASP.NET MVC5 and AngularJS application
    - 1.2 client written in pure JavaScript

    Who are the users for "1.2 client written in pure JavaScript"? Is the vendor expected to work on only AngularJS client, only JavaScript client or both clients? Are both clients in sync with regards to system functionality?
    We should remove the "1.2 client written in pure JavaScript" as it was developed to a certain point but never implemented. There is only one client in production, which is the .NET MVC5 and AngularJS client that has been provided in the source code. This is only client that has been implemented.

11. Who is currently performing maintenance tasks for this application? Based on current support and maintenance requests, how many person hours/months of effort is spent on maintenance (site management, bug triaging, bug fixing, database management and development) in a month or year Please provide the ticket dump (If any) of issues resolved or estimated effort spent to maintain the current application?
    Very little time is spent supporting the existing application as there have been no changes for over a year and current bugs/issues are just logged but not being corrected.

12. During review as part of phase 1 do we need to provide for any multilingual or internationalization support? Please confirm if multilingual support of this application is in scope for the current engagement? If yes, which languages need to be supported?
    No

13. What version of AngularJs is being used for the current application? Is the vendor expected to use the same version of AngularJs?
    ITAMS references AngularJS library through bower components package, inside this package, the javascript file angular.js refers to version 1.3.15.
    Therefore, the AngularJS version should be v1.3.15. There is no expectation of the vendor to use the same version. In fact, the RFP states that the vendor is welcome to use any of the existing code they see fit or is free to re-write the entire user interface.

14. What version of .NET framework is being used for the current ASP.NET WebAPI development? Is the

vendor expected to use the same version?
.NET framework 4.5.2
Again, the source code and information of the existing application is being provided for informational purposes and to help with understanding what currently exists. There are no requirements of use of the ANY of the existing code.

15. What additional or supporting components, Microsoft frameworks/third party frameworks are being used for WebAPI development? For instance, does the current application use frameworks like Entity Framework, AutoMapper, Unit Framework or AutoFac for dependency injection or any others.
Multiple Microsoft or non-Microsoft framework(library) such as Entity framework, AutoMapper, etc. are being utilized. All libraries used by ITAMS are managed by NuGet, and Visual Studio will automatically download them into the packages folder at the first time you run the solution file, you can check them there.

16. Contractor is expected to migrate the database to an existing SQL Server 2014 cluster, or greater, as determined at the time of project implementation. Will the vendor take this on solely or provide guidance and support? Is the vendor responsible for implementing or configure SQL Always-On Failover Cluster Instances? Where will this cluster be hosted?
Contractor would work with UTHealth DBAs. Contractor is responsible for confirming and coding to ensure that the new system will successfully run in SQL Server 2014. Infrastructure components such as FCI and AGs would be configured by UTHealth personnel. Contractor would be responsible for ensuring compatibility with those components in a SQL Server 2014 environment and assist UTHealth db admins during implementation to the new SQL Server environment.

17. Are any unit testing frameworks being used currently? Is the vendor free to choose any unit testing framework/s?
Vendor is free to choose and should include in those details in the RFP.

18. Do you want a hybrid combination of onsite development as well as remote priced out? Or just onsite? Just remote? Just delivery center?
This is up to the Contractor. The development time is not being required to be onsite.

19. Minimum requirements were provided for expected site performance. Does the current implementation currently meet these thresholds?
Yes

20. Chrome and & IE are browser standards. Is there any desire to have a mobile component of the website?
This is not required for the scope of this project but would be desirable.

21. How is the AngularJs front-end consuming authentication tokens? Please provide architectural overview of how authentication and authorization are implemented across different layers of the current application.
It does not use any authentication token on the front end. All the authentication and authorization are controlled by the MVC controller. After the authentication step is passed, the valid authentication info will be maintained in the http session, within the same http session, all front-end to server-end requests/responses are valid.

22. When will we get answers back to questions asked? If answers go beyond 1/17 will you extend the deadline?
    The deadline has been extended to Thursday, 2/1/2018.

23. Requirements 5.3.26 states the following "Describe your proposed authorization method(s) for the application." Are you asking about our approach to authentication?
    In this context, the RFP is referring to Authorization, or access management. Meaning "What are your methods for handling authorization after a user has been authenticated?"

24. Will UT Health provide VPN access to environments for vendor resources that will be working remote?
    UTHealth utilizes SecureLink for vendor remote access.

25. For resources that are onsite, will work space be provide with computer and network access or does the vendor need to provide laptop and UTH will give network access and work space?
    It would be expected the vendor would provide their own computer resources, however, UTHealth will provide a work space and can accommodate computer resources for on-ste work if necessary, but would consist of a local computer that would not leave premises.

26. Will you provide an overview of the methodology that is currently used at UTH?
    See attached "System Development Methodology" document. It should be noted that this is a guideline document and is not to be assumed that this is a policy or procedure document that has to be followed specifically. However, it gives a good example of materials that should be included in the examples requested of the Contractor in the RFP.

27. Did any vendor/s support UTH with the creation of the RFP materials? If so, is that vendor allowed to respond? Has that vendor provided any support similar to what is being requested in this RFP?
    No. The RFP materials were created in whole with UTHealth resources only.

Information Technology
Project Support Office

System Development Methodology (SDM)

Please see the Preface on page 3 for applicability of this methodology.

Version 1.3
January 9, 2013

Version Control

Version	Description	Date
1.0	First draft	10/20/2005
1.1	Add system usability	12/12/2005
1.2	Add Applicability on P -3	2/2/2006
1.3	Applicability update	1/9/2013

Preface

The System Development Methodology (SDM) presented here is a ***guideline*** for the implementation of information technology at the University of Texas Health Science Center at Houston.

Like all system development methodologies, the purpose of this SDM is to reduce the risk of information technology implementation project failure to the university. Information technology implementation projects can be expensive, disruptive failures if not organized, funded and managed properly. System development methodologies seek to minimize the risk of failure by following a rigorous process designed to identify all of the requirements, define the critical success factors, engage all of the stakeholders and develop a realistic plan and budget.

There are many system development methodologies. They all seek to address the issues identified above in slightly different ways. As long as the issues above can be addressed satisfactorily, it matters less which SDM is used. However, not using any SDM creates an unacceptable risk to the university. In addition, spending a lot of time researching and learning another SDM when this SDM is available is wasteful.

There is no such thing as a 'one size fits all' SDM. It is expected that the SDM will be ***appropriately*** tailored to fit the implementation project being undertaken.

Applicability of the SDM

The SDM applies to information technology **Implementation Projects** that have a defined outcome and an associated beginning and ending date. With the exception of the types of information technology projects listed below, the SDM should be used.

Portions of the SDM can be useful in any type of technology project and the project manager should apply them as appropriate.

To obtain an exemption from the use of the SDM, please see the section below entitled **Exemptions to the Use of the SDM**.

The SDM *generally* does not apply to (see details below):

1. Research projects,
2. Proof of Concept (POC) projects,
3. Pilot Projects,
4. General support and maintenance or
5. Information Services Systems Requests (ISSR).

However, **where any of 1 – 5 above are part of a larger Implementation Project** which meets the criteria defined at the top of this section, the SDM for the larger Implementation Project should apply to **ALL** activities **including** the ones listed above (research, POC's, etc.). In other words, breaking projects into smaller pieces to avoid the use of an SDM is not allowed.

1. Research Projects

Research projects have only broadly defined goals and are focused on developing new technology that may or may not have specific applicability at the UT-H. Success of the project is

not guaranteed. Research projects should follow the SDM if at the outset of the project it is known, or reasonably anticipated, that an output of the project will be an operational system upon which the university will rely. If there is any doubt about the applicability of the SDM, ask the CIO at CIO@uth.tmc.edu. See **Exemptions to the Use of the SDM** below.

2. Proof of Concepts

A proof of concept is a small project designed to test a certain technology to determine if it will work for a specific application. While there is a goal in mind with a specific applicability, the tasks to meet the goal are undefined. Once again, portions of the SDM may be helpful but are generally not required. If there is any doubt about the applicability of the SDM, ask the CIO at CIO@uth.tmc.edu. See **Exemptions to the Use of the SDM** below.

3. Pilot Projects

Pilot projects are small implementations that, if successful, precede a full implementation. Pilot projects should, by definition, be small and manageable. Since they hopefully precede a wider implementation, it is recommended that key elements of the SDM should be followed so that this work will be done prior to the full implementation which probably will require the SDM. If there is any doubt about the applicability of the SDM, ask the CIO at CIO@uth.tmc.edu. See **Exemptions to the Use of the SDM** below.

4. General Support and Maintenance

General support and maintenance are operational support activities like problem resolution (HEAT tickets), small changes, technology upgrades, supporting end users, etc. General support and maintenance rarely, if ever, follow the SDM due to the short and transient nature of these activities. If a longer term, more comprehensive solution (like a system upgrade) is required to resolve an issue, the result may be that a implementation project is initiated that will follow the SDM.

5. Information Services Systems Requests

ISSR's are small to medium projects to modify information technology. The major phases of the SDM apply but are sized to fit the nature of the work. Signoffs and documentation are required although the number of signoffs and amount of documentation is smaller. The project, or system, manager should be capable of identifying those parts of the SDM are most applicable.

Use of the SDM

As stated above, the SDM is a ***guideline*** for the implementation of information technology. It is to be applied by appropriately experienced IT professionals to reduce the risk of technology implementations. An appropriately knowledgeable professional will understand the terms and concepts in the SDM and know how to appropriately apply them to a project. The forms and formats are not universally applicable but should be tailored to fit the needs of the project.

It is **not** a:

- 'Cookbook' on how to implement information technology,
- Substitute for good judgment,
- Substitute for experienced, knowledgeable project management,
- Training manual for new managers,
- Project management guide, or
- Guarantee for success.

If you are undertaking a technology implementation project and you do not understand the concepts presented in the SDM please call the Information Technology Project Support Office at 713-486-2273 for assistance in determining if the SDM and an IT provided Project Manager can be of value to you.

Exemptions to Use of the SDM

Exemptions to the use of the SDM are granted only by the Vice President, Information Technology and Chief Information Officer. The project sponsor can request an exemption by contacting the CIO at CIO@uth.tmc.edu. The CIO will require enough information to determine the applicability of the SDM to the project for which the exemption is being requested. A brief description of the project, including the desired end result, will serve as the starting point for a short discussion. It will be most expeditious for the CIO to discuss the project by phone, or in person, to determine whether or not the SDM should be used and to what extent. In certain cases, the proposed project may require additional data gathering which could be assigned to the Project Support Office. If desired, the Project Support Office is available to assist in the process prior to the request for exemption.

If the project sponsor and the CIO cannot agree on the applicability of the SDM to a particular project, the Information Technology Governance Council will make the final decision after evaluating the information presented by the project sponsor and the CIO.

Exemptions must be documented and retained with the project files by the project sponsor.

It is rare that an Information Technology project will be exempted entirely from the SDM since the SDM is designed to provide maximum flexibility for tailoring to fit the complexity of the project (see the Complexity Matrix). It is more likely that the project sponsor will request modifications to the SDM that are not recommended, or acceptable, based on the Complexity Matrix or in the opinion of the Project Support Office. In this case, the project sponsor should:

1. work with the Project Support Office to determine what is appropriate given the unique circumstances of the project,

2. appeal the recommendation of the Project Support Office to the Vice President, Information Technology and Chief Information Officer or

3. appeal to the Information Technology Governance Committee.

It is anticipated that exemption requests will be rare since the SDM can be tailored to meet the needs of the individual project.

System Development Methodology
Table of Contents

Introduction

Abstract

ORACLE Corporation's **CASE*Method** has been chosen as a guide to implement the University of Texas Health Science Center's (**UTHSC-H**) standard System Development Methodology (***SDM***). It provides a framework for developing software applications by partitioning the development process into cohesive stages and activities in which proven techniques are employed to produce specified deliverables. The model methodology is outlined in the book "*CASE*Method: Tasks and Deliverables," Addison Wesley 1992, by Richard Barker*.

The UTHSC-H Information Services **SDM Guide** outlines the standards, conventions, techniques, methods, and tools to be employed throughout the life cycle of an applications development project. The purpose of this document is to provide a consistent guideline for experienced project managers to follow for the implementation of information technology.

Overview

This **SDM Guide** provides a list of activities that need to be performed during the development of an application, who should perform these tasks, and during which stage of development these tasks should be performed. Also included are descriptions for each of the deliverables that must be produced. As a supplement to this document, a template will be provided, in **Microsoft Word** format, for each of these required deliverables. These templates can be tailored as required to meet the needs of the project.

The Project Manager is responsible for following the SDM throughout the project. All customizations to the system development approach are made in advance by the Project Manager, based upon scoring the project on the complexity matrix. Customizations to the development approach must be approved by management prior to the start of the project, and documented in the Project Plan. Customization may entail omission of individual tasks based on attributes of the specific project. However, key deliverables are expected of every applications development project.

Following are the Stages and Key Deliverables of the SDM:

Stages	Key Deliverables
Strategy Stage	Strategy Study
Analysis Stage	Requirement Specification
Design Stage	Design Report

or Rapid Application Development Prototyping Stage	or Prototype Documentation
Build and User Documentation Stage	Implementation Documentation User's Guide
Transition Stage	System and Control Documentation
Production Stage	Production Documentation

Scope

The **SDM Guide** addresses the following:

- Complexity Matrix Evaluation Criteria
- SDM Tasks and Deliverables
- Automated Tool Support
- Quality Assurance Activities
- Issue Resolution and Problem Reporting
- System Usability

References and Source Documents

- Barker, Richard, "CASE*Method Tasks and Deliverables", Addison-Wesley, 1992
- Barker, Richard, "CASE*Method Entity Relationship Modeling", Addison-Wesley, 1990
- Barker, Richard**,** "CASE*Method Function and Process Modeling", Addison-Wesley, 1990
- Oracle's CASE*Designer Reference Guide
- Billings, Chris & Maria, "Rapid Development with Oracle CASE", Addison-Wesley, 1993
- Bureau of information Technology, Department of Environmental Protection, Common Wealth of Pennsylvania

Roles and Responsibilities

A successful application development project requires bringing together knowledge of the business procedures and policies to be addressed by the system with knowledge of efficient computer systems design and development techniques. To ensure this:

- A successful project team needs to include users familiar with the requirements together with experienced systems development professionals.

- The user participants need to be responsible for clearly identifying and specifying their business requirements and reviewing all deliverables very carefully to ensure that these requirements are being addressed.
- The developers need to be responsible for all technical design and implementation decisions. Project leaders from both sides need to be designated to be responsible for the project's success.

A **Project Sponsor** must be designated from the user side to serve as coordinator and final decision maker for user participants. This should be a single individual even in instances where projects involve user participants from different organizations. The user sponsor needs to have the authority to obtain information, decisions and resources from the program areas involved in a timely fashion as needed to keep the project on track. The project sponsor is responsible for ensuring that business requirements are correctly identified and truly reflect the needs of the business areas that are to use the proposed system. This involves the following:

- identify sources of information about business requirements including:
  - key users
  - decision makers
  - documentation
- management support
  - obtain decisions from management as needed
  - authority to make detail decisions
- authority to obtain resources
  - staff time to participate in project
  - funding necessary for implementation
- responsibility for
  - correctly and completely identifying requirements up front
  - limiting changes and controlling project scope
  - carefully and completely reviewing draft documentation
  - correct errors/misconceptions up front

A **Technical Sponsor** authorizes the project and provides technical personnel and resources; reviews the project to ensure it meets institutional technical goals.

A **Project Manager** needs to be assigned from the development team. This needs to be an individual experienced with applications development project management. This person is responsible for the management of both the functional and technical aspects of the project. The project manager generally reports to the project sponsors. In large projects, the Project Manager may have a technical team leader and a functional team leader. In smaller projects, these three roles may be collapsed into either one or two roles based on the capabilities of the individuals. For complex projects, a professional project manager may be appointed.

The **Technical Team Leader** is the responsible party for managing the technical activities of the project including all technical implementation decisions.

The **Functional Team Leader** is responsible for managing the functional activities of the project including all functionality issues.

Both team leaders are responsible for identifying and resolving issues where functional and technical tradeoffs must be made. The project manager and sponsors can be involved as necessary.

Project Complexity Matrix

The SDM is intended as a guide to ensure a quality development project that meets the users' needs. Specific tasks and supporting documentation are required to assure proper communication and agreement on the application to be developed and to ensure appropriate coordination of effort among all team participants. Documentation will largely be reflective of project tasks deemed as 'required' through utilization of the project complexity matrix outlined below. The larger and more complex the application, the greater the number of tasks and amount of documentation required.

Project documentation also provides for efficient future application maintenance.

The SDM is intended to be customized according to the scale of the project. Certain tasks and documentation will be required for all application development projects. Others will be optional depending upon the size or type of project. Experienced project managers in the Information Services department, along with key user management, will make determinations of required tasks and documentation for their projects. The Chief Information Officer (CIO) will review all project complexity recommendations for concurrence.

Project Complexity Measures

Measuring size and complexity of an application and its associated documentation requirements will be dependent upon a variety of factors. The following is intended as a guide to estimating the complexity of a project. These numbers used are subjective and the formulas should not be taken as absolute rules. They are intended as guidelines for project managers while allowing flexibility for the development process.

It should be noted that much of the factors identified below would be unknown until a certain amount of analysis is performed. For this reason, any ISSR, user request, new business requirements, etc labeled as a 'development effort', whether a new system or a large effort on an existing system, a Strategy study must be performed. Through this process, the items outlined below will be accurately scored. This in turn will identify the true magnitude of a project and those tasks and deliverables that will be required.

I. Project Complexity Considerations:

A. Development Costs: The greater the amount of taxpayer money being expended upon the development project the greater the need for formal documentation procedures to assure wise expenditures.

Labor Costs:

Identify all internal and external staff required. This should include but not be limited to just the developers and project managers, but also include estimates for all parties to be contacted to provide information during analysis and requirements studies as well as those involved in reviewing deliverables and testing the application.

For each type of staff required perform the following calculations and sum them for all staff types.

Number of staffers * Staff Salaries (including recovery cost) * number of person work hours or workdays

Hardware/Software Costs:

- Software Purchase/License Fees and maintenance cost
- Hardware Purchase/License Fees and maintenance cost
- other purchase requirements

After identifying and totaling all costs score the Development Costs using the following guidelines.

Scoring:

	Very Low	Low	Moderate	High	Very High
< $100K	**1**				
>$100K and < $400K		**2**			
>$400K and < $700K			**3**		
>$700K and < $1M				**4**	
>$ 1M					**5**

B. New Technology Risk: Implementing new technologies introduces additional risks into a project. The chances for cost overruns are significant. Even more significant is the possibility of miscommunication and conflicting expectations. The need for clear documentation to manage expectations is therefore considerable.

Scoring:

0. Not applicable - No new technologies involved. All parties familiar with technologies involved.

1. Very Low - The technology is new to these users, but not to the organization itself.

2. Low - Developers have had no experience with the technology but have received training and others in their organization with successful experience implementing it are available on site

3. Moderate - Developers have had no experience or training with the technology but have received training and others in their organization with successful experience implementing it are available on site.

4. High - The technology has been used successfully in similar circumstances in other organizations but none of the projects' participants and their organizations has experience.

5. Very High - Technologies are bleeding edge. None of the team members have experience with the technology, nor do other support staff in their organizations. There is little industry background to draw from.

C. Variety of Business Functions Performed: The more different individual business functions performed and the greater the variety of the ways in which those functions may be performed, the greater the need to ensure through proper documentation that all of these functions and their variations are fully identified.

Scoring:

1. Very Low - Single function application. No variety in that function.

2. Low - Single function application but with some variety in how that function is performed, or one or two functions with little variety in how these functions are performed.

3. Moderate - Narrow range of functions with moderate variety.

4. High - Wider range of functions and/or variety.

5. Very High - Full range of functions with lots of variety in how each may be performed.

D. Integration Considerations: Integration can involve either data sharing or interoperability between applications. With data sharing it may be that multiple applications share a common set of data in a single repository or transfer data from one to the other or record data on the same subject without automated integration (i.e. need to be manually kept in sync). Interoperability refers to applications that have some functional relationship to one another. This may include applications that handle different parts of a set of related functions. These systems have a need to inter-operate in some fashion (automated or manual) so that the latter process is aware of those facilities which have been constructed.

Scoring:

0. None - No integration

1. Very Low - Data sharing with applications maintained by the same organization on the same technology platform.

2. Low - Data sharing with applications maintained by different organizations on the same technology platform.

3. Moderate -Data transfers between applications maintained by different organizations on different technology platforms.

4. High - Data Sharing and interoperability with applications maintained by different organizations.

5. Very High -Tightly coupled integration with applications maintained by different departments or institutions on different technology platforms. This includes data sharing

E. Business Process Changes: When the application is required to automate processes that are new or significantly changed, it is imperative that all of the steps in the process be fully identified and analyzed to determine that they are complete and thorough. They need to be communicated thoroughly to avoid misunderstandings or ambiguity. Often with new process the analysis involved in attempting to analyze them for development purposes discloses gaps in the process such as unforeseen contingencies.

Scoring:

1. Very Low - Upgrading existing application from one software version to another.

2. Low - Automating an existing manual process that is well established and well understood.

3. Moderate - Minor process changes.

4. High - Revising business processes as part of a process improvement initiative or reorganization.

5. Very High - Implementing a new business function that has never been performed before, e.g. legislation requiring a new regulatory program that does not fit the model of other previous programs. Policies and procedures are being drawn up concurrently with the application development project.

F. Criticality and Strategic Impact: Applications which are critical to the department's primary mission need to be subject to the greatest amount of review for quality assurance. Extensive documentation is therefore necessary to ensure these business processes are being adequately modeled.

Scoring:

1. Very Low

2. Low

3. Moderate

4. High

5. Very High

G. Security/Audit Requirements: Any time the potential for misuse of information arises; special care must be taken to ensure that the system is designed to prevent misuse. Any of the following information will require special security/auditing documentation: Confidential Information (e.g. personnel, HIPPA, FERPA, legally sensitive information).

Scoring:

1. Very Low

2. Low

3. Moderate

4. High

5. Very High

H. Revenue Producing, Fee Collection, Payments: Due to the potential costliness of errors and the potential for theft and fraud, applications involving funds received or paid out need to be extremely well documented. Special design considerations need to be included to prevent fraud or theft and audit trails established to track system transactions. Representatives from the Financial Services' office will need to be involved in all phases of the project to ensure adequate procedures are included. Everything must be documented to facilitate audits of system integrity during the application's life.

Scoring:

0. Not Applicable - No funds involved.

1. Very Low

2. Low

3. Moderate

4. High

5. Very High - Funds paid/received in excess of $1 million per year

I. Program Areas Affected / Organizational Boundaries Crossed: The more program areas or organizations involved the greater the need for thorough documentation to ensure common understanding of the business processes involved, work flows, requirements etc.

Scoring:

1. Very Low: one group or a department

2. Low: Single department

3. Moderate: multiple departments

4. High: Single school or operating unit

5. Very High: Across campus

J. Number of Users: The greater the number of affected customers (students, faculty, patients and staff) the greater the need for formal documentation procedures to assure the application will function as required.

Scoring:

1. Very Low - < = 20

2. Low - >20 and <= 150

3. Moderate - >=150 and <=500

4. High - >=500 and <= 1000

5. Very High - >= 1000

Scalability Worksheet for Determining Project Complexity

		Complexity	Adjusted
Complexity Factors	***Factor Weight***	***Score***	***Score***
Development Costs	5	1	5
New Technology Risk	1	0	0
Number of Business Functions	4	1	4
Integration Considerations	5	0	0
Business Process Change	5	1	5
Criticality and Strategic Impact	4	1	4
Security/Audit Concerns	4	1	4
Revenue Considerations	4	0	0
Program Areas Affected	3	1	3
Number of Users	2	1	2
	Total Complexity Ranking Score		27
Minimum Reasonable Score	27		
Maximum Reasonable Score	175		

Assessing Situation: Applications measuring low on all factors require the minimum level of documentation described below. Applications measuring high in multiple areas require the highest level of documentation and should not take shortcuts.

Minimum Requirements: All projects no matter how small must have a Strategy Study. The Strategy Study identifies the nature and scope of the proposed project. Costs and benefits of the proposed system need to be identified and the project's feasibility and Return on Investment (ROI) evaluated. The strategy study must be reviewed and a go/no-go decision made before any further work is to be completed.

SDM SCALABILITY MATRIX

Individual tasks and deliverables identified in the SDM may not be required for less complex projects. Having assessed project complexity using the above worksheets, use the following as a guideline for required vs. optional tasks and deliverable components.

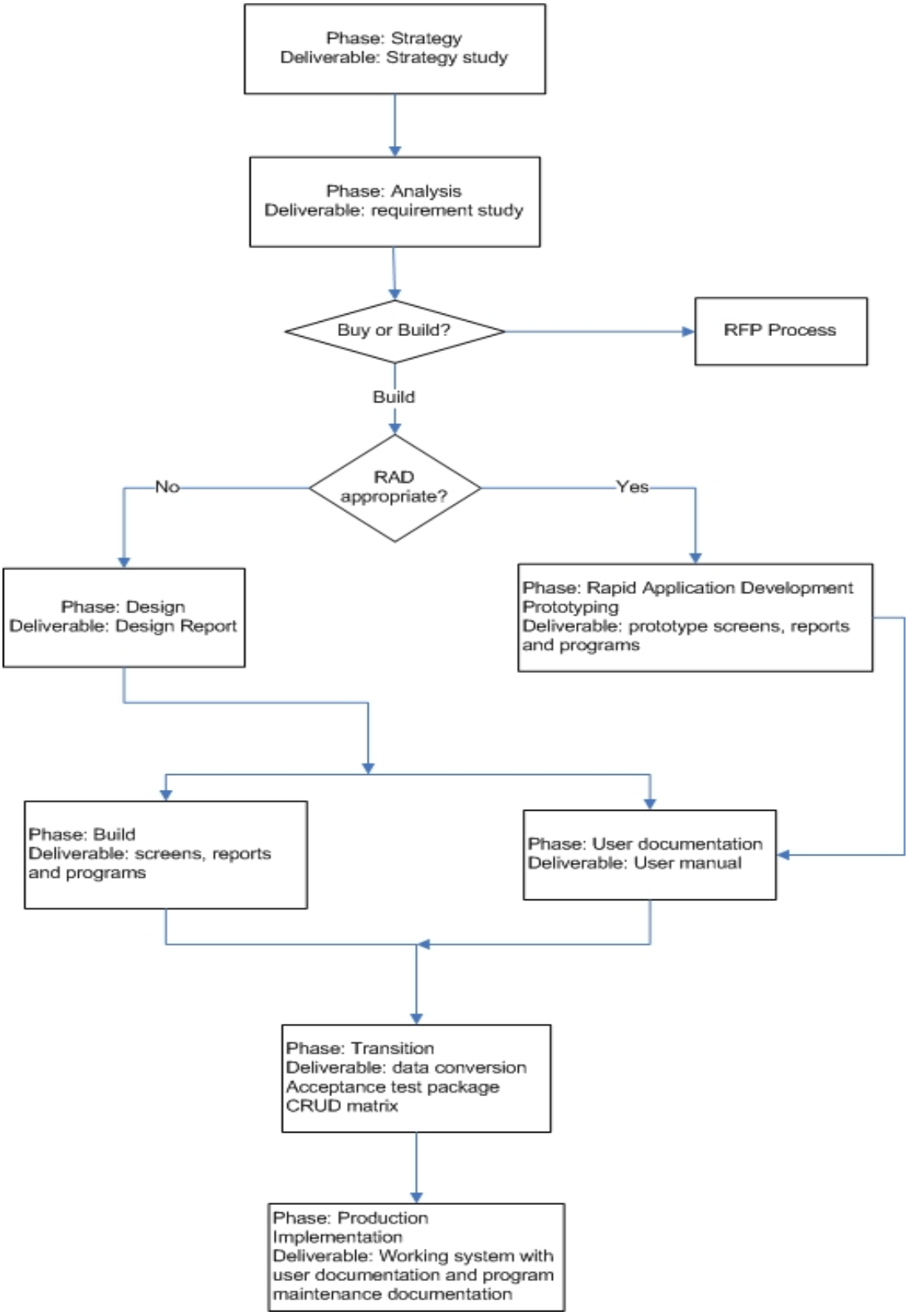
Phase	Deliverable	Component or Task	Required vs. Option
Strategy	Strategy Study report	-Complexity Matrix	-Always required
		-Key Signoff	-Always required
		-Mgmt Summary	-Always required
		-Function Model	-Required for any development project > 1 month
		-Entity Relationship Model	-Required for any development project > 1 month
		-Dataflow diagram	-Optional,

			recommended for fee-related systems
		-Data Source Identification	-recommended
		-Business Unit Identification	-Always required
		-System architecture	-Always required
		-Term definition	-recommended
		-Forward plan recommendation	-Always required, detail will vary
		-Gantt Chart	-Required for project lasting more than 2 months. However, a project timeline is always required
		-DBA review for ensuring data consistency/sharing	-recommended
		-CIO review and approval	-Always required
		-CFO review and approval	-Required only when fee/revenue are involved
Analysis	Requirements Specification	-Key signoff	-Always required
		-**All** Strategy Stage tasks mentioned above	-Required as stated in Strategy Study section
		-Matrix Diagrams	-optional
		-Volumetric	-Required
		-Event Definitions	-Optional, recommended for complex systems
		-Function Logic	-Optional, recommended for complex systems
		-Input/Output Descriptions	-Optional, recommended for complex systems
		-Security/Auditing Requirements	-Optional, recommended for complex systems
		-Backup/Recovery	-Required, however, degree of detail will vary
		-Testing Requirements	-Required, however, degree of detail will vary
		-Acceptance Criteria	-Required
		-Business Process Model	-Required
		-Training Plan	-Optional except when new business processes are being introduced (e.g. a program newly

			created due to new legislation) or business processes and workflows are being significantly modified.
		-Data Conversion Plan -Cut Over Plan -Processes being Converted Prototype Documentation -Forward Plan Recommendation	-Recommended, required for average to complex projects -Optional, required for systems involving integration -Recommended -Optional -Always Required
Design	Design Report		**Optional when complexity score is less than < 100 Prototyping approach may be used instead**
		Function Hierarchy	Optional Unless Policy/Procedure Change Score is 4 or Higher
Prototyping			
Build and User Documentation		Validation Logic at Database Level	Required for any global tables. Required for any tables to be shared among applications or used for electronic commerce or web input.
	Fully tested system		-Required
	User and Operations documentation		-Required
	Disaster Recovery Plan		-Required
Transition	Trained users and operations staff		-Required
	Installed and fully operational system		-Required
	Converted data		-Optional
	Completed System		-Required

	Documentation		
	System Signoff		-Required
Production		-Back-up, recovery and archive files -Performance statistics -Change control log -Problem reports -System audit results -New requirements	-Required - Required - Required - Required - Optional - Optional

SYSTEM DEVELOPMENT METHODOLOGY AT A GLANCE



Strategy Stage

The purpose of the Strategy Stage is to define the goals, priorities, and high-level business needs of the end-user organization. This includes development of a set of business models to accurately depict client organization needs, identify potential implementation alternatives and tradeoffs, quantify and assess the scope of the effort required to complete a detailed analysis, and create a forward development plan for proceeding with the project. This Systems Development Methodology is based upon the principles discussed by Richard Barker from Oracle Corporation in his book: CASE*METHOD Tasks and Deliverables. Task numbers are as they appear in Richard Barker's book.

Strategy Stage	Activities and Deliverables
Roles & Responsibilities • Management • Project Manager • Lead Analyst • Database Administrator • Analyst • Sponsoring User • End User	Strategy Tasks (Outlined by Richard Barker)
Quality Assurance Activities	

Strategy Stage Activities and Deliverables

Activities

Task #	Description	Check
10	Project Administration and Management	
20	Scope the study and agree to Terms of Reference	
30	Plan a strategy study	
40	Briefings, interviews and other information gathering	
50	Model the business	

60	Prepare for feedback session	
70	Consolidate results of feedback session	
90	Complete documentation of the business model	
100	Evolve information system architecture and make other recommendations	
110	Determine forward system development plan	
120	Prepare verbal report	
130	Report to senior management	
140	Prepare and deliver written report	

Deliverables

☐	Strategy Study Report
☐	Complexity Matrix
☐	Entity Relationship Diagram (Optional)
☐	Functional Hierarchy Diagram (Optional)
☐	GANTT chart
☐	Critical Issues Discussion (meeting(s))

A Strategy Study Report document template for the Strategy Study Report is available as a word document. The template is to be used as a guide for preparation of the Strategy Study Report. The exact requirements for this report may vary depending upon the complexity of the project envisioned. For further information see SDM Scalability Matrix.

Strategy Stage Roles and Responsibilities

Management: Their primary objective is making sure that the clients invest enough of their resources into the project to make it a success. They are responsible for defining project goals and resolving conflicts.

Project Manager: Responsible for planning and control, putting the plan into action, keeping all parties informed of plans, progress and issues, and managing the project team. The Project Manager should also serve as a working manager by providing technical leadership as required. It is important that the project leader maintain a short time frame for the strategy stage so as not to lose momentum. The project leader must also keep in mind that the most cost effective and

efficient solution may not include automated technology. The Project Manager will take the lead at user committee meetings to ensure the project is meeting the business needs.

Lead Analyst: Responsible for setting up the CASE Application, and through interviews with the users, learn the business functions, procedures and tasks in detail enough to develop entity relationship diagrams, system boundary definitions and possible system architecture which clearly depict the area of business to be analyzed. Be sure to include all entities, manual, as well as automated. While conducting interviews be sure to separate interviewee "needs" from "likes." Provide technical leadership to other team members as needed.

Database Administrator: Responsible for reviewing strategy study and planning appropriately for providing adequate capacity. Review strategy study and identifying opportunities for data sharing or reuse and compliance with departmental data standards.

Analyst: Assist lead analyst with project activities and deliverables. Responsible for analyzing the business and identifying the terms of reference. Also, through interviews with the users, learn the business functions, procedures and tasks in detail enough to develop function hierarchy, system boundary definition and possible system architecture which clearly depict the area of business to be analyzed. Be sure to include all functions manual, as well as automated. While conducting interviews be sure to separate interviewee "needs" from "likes."

Sponsoring User: Identify individuals who can provide details necessary to perform the tasks. Ensure the users are participating in a cooperative manner and are allocated the required time necessary to actively participate.

User: The users at this stage should be a selective sample of individuals who will serve as a true representation of the organization as a whole. These users must provide analysts with information necessary to clearly model the area of business and verify that the analyst interpretation is correct and true to form.

Quality Assurance Activities

- Conduct technical walkthroughs with project team at logical intervals to establish consensus and commonality.
- Review all deliverables in draft form with key users in structured walkthroughs to assure requirements are accurately reflected.
- Cross check entities to functions to assure completeness.
- Take minutes of all meetings and forward copies to attendees for confirmation.

Analysis Stage

The analysis stage begins with the business models developed during the initial strategy stage. The preliminary analysis findings are verified and expanded into sufficient detail to describe exactly **"what" will and will not** be built into the system. The information obtained during the analysis stage is documented in the requirement specification. The requirement specification establishes the foundation for all-subsequent design and development work.

Analysis Stage	Activities and Deliverables
Roles & Responsibilities • Management • Project Manager • Analyst/Developer • Sponsoring User • User	Analysis Tasks (Outlined by Richard Barker)

Analysis Stage Activities and Deliverables

Activities

Task	Description	Check
10	Project Administration and Management	
20	Plan Detailed Analysis	
30	Review Standards, Constraints and Potential Design Issues	
40	Investigate Detailed Requirement	
50	Review Findings Against Terms of Reference to Confirm Approach	
60	Provide Detailed Specification	
70	Provide Initial Transition Strategy	
90	Define Audit/Control Needs	
90	Define Back-Up/Recovery Requirements	

100	Perform Outline Sizing and Predict Performance	
110	Review Results of Detailed Analysis	
120	Obtain Stage End Commitment	

Deliverables

☐	Requirements Specification
☐	Entity Relationship Diagram
☐	Function Hierarchy Diagram (Optional)
☐	Function/Business Unit Matrix (Optional)
☐	Entity/Business Unit Matrix (Optional)
☐	Description of Work flows and manual procedures
☐	Data Volumes and Transaction Frequencies
☐	Performance Expectations
☐	Audit/Control and Backup/Recovery Requirements
☐	Constraints and Assumptions
☐	Approach to Further Development: “Buy vs. Build”, Traditional Build or RAD approach
☐	Revised Gantt Chart
☐	Transition Plan, Data Conversion Plan

A document template for the Requirements Specification report is available as a word document.

Analysis Stage Roles and Responsibilities

Management: Their primary objective is making sure that the clients invest enough of their resources into the project to make it a success. They are responsible for defining project goals and resolving conflicts.

Project Manager: Responsible for managing the project team, project planning, control and implementation, communicating the progress and issues to the team members. It is also important that the Project Manager ensures that the project is going in the right direction and anticipates any needs or problems.

Analyst/Developer: Responsible for performing detailed analysis with the users to extend and crosscheck the Entity Relationship Diagrams and Function Hierarchies, assist users in obtaining volumes and frequencies, and identify transition issues.

Sponsoring User: Identifies individuals who can provide details necessary to perform the tasks. Ensure the users are participating in a cooperative manner and are allocated the required time necessary to actively participate.

User: Provide analysts with information necessary to clearly model the area of business and verify that the analyst interpretation is correct and true to form. Determine how they will review progress: this involves the role of the sponsoring user and/or committee, and the content and frequency of reports.

The Testing Strategy

An initial testing strategy is developed upon completion of the Analysis Stage and may be refined and updated during the Design and Build stages by the project manager, the project team, and the end-user organization.

Proper testing of the software is critical to the overall quality of the end-product application. Historically, it is also the most neglected or over-looked component of the development process. As such, the testing process must be properly planned and methodically executed to ensure acceptable quality standards.

A testing strategy is formed during the Analysis Stage after system requirements have been refined and stabilized. Test specifications are developed for each software module based on the detailed module design specifications. A test specification outlines test objectives, provides test cases and scenarios, the test data to be used, and expected test results.

At a minimum, the Testing Strategy should address the following:

- **Module Testing**: *(also referred to unit testing)* Focuses on verification of the smallest unit of software design -- the module. Using the detailed design specification as a guide, important control paths are tested to uncover errors within the boundary of the module. The man to machine interfaces are tested to assure that information properly flows into and out of the module, allowable boundary values are verified, and module-data structure interface is tested to assure that data is properly stored according to established integrity rules. Module testing is performed during the Build Stage.
- **System Testing**: *(also referred to as integration testing)* ensures that the system as a whole satisfies input/output specifications and that interfaces between modules/programs/subsystems are correct. Emphasis is placed on system access, security, performance, and recovery capabilities. System Testing may be performed at the closing of the Build Stage.
- **Acceptance Testing**: Acceptance testing is performed by the client organization with support from the project team to ensure that the application satisfies established acceptance criteria and that both manual procedures and automated functions perform according to stated requirements. Acceptance testing is performed during the Transition Stage. A detailed acceptance test package should be prepared identifying what is to be tested and providing space for signoff and notation of problems. A sample is available among the document templates.

Design Stage

The design stage focuses on **"how"** the system will be implemented to achieve the requirements specified in the requirements specification. The overall system architecture is defined, the logical and physical database design is derived from the entity- relationship models, and business functions are translated into detailed module design definitions describing specific screens, reports, menus, algorithms, and utilities. All system design information gathered during the design stage is documented in the system design report. The activities and deliverables that may comprise the design stage are outlined in the Design Stage Activities and Deliverables section.

Design Stage	Activities and Deliverables
Roles & Responsibilities • Management • Project Manager • System Analyst • Programmer • Sponsoring User • User	Design Tasks (Outlined by Richard Barker)

Activities and Deliverables

Activities

Tasks	Description	Check
10	Project administration and management	
20	Design application	
30	Design and build database	
40	Produce network/communication design	
50	Design audit/control needs	
60	Design back-up/recovery needs	
70	Review outline design and produce program specifications	

80	Complete system test plan	
90	Complete transition strategy	
100	Review results of design stage	
110	Obtain stage-end commitment	

Deliverables

☐	Design Report
☐	System Test Plan
☐	Transition Plan

A document template for the Design report is available as a word document.

Design Stage Roles and Responsibilities

Management: Their primary objective is making sure that the clients invest enough of their resources into the project to make it a success. They are responsible for defining project goals and resolving conflicts.

Project Manager: Responsible for managing the project team, project planning, control and implementation, communicating the progress and issues to the team members. While each group keeps focused on their specific area, the Project Manager must ensure that dependencies between groups are addressed.

System Analyst: Responsible for transforming specifications into actual system design. Work closely with analysts and users to ensure that the design meets the intended requirements and takes usability into consideration. Provide recommendations for the development and testing cycles and the current database design.

Programmer: Responsible for ensuring that the design meets the intended specifications and requirements. Work with the designers to ensure that the right trade-off decisions are made in tuning.

Database Administration: Review and enforce the naming conventions that have been developed by the project team. Review volumetric information, sizing limits and data typing. Review all referential integrity rules as developed by the Project Team. Review entity- relationship diagrams, key-based models, and entity/attribute definitions, provided by the Project Team. Assist in creating the initial physical database design including appropriate indices. Consider module definitions, event definitions, physical system limitations, and identifiable bottlenecks. Provide recommendations to avert any problems. Provide advice regarding security issues.

Sponsoring User: Identify individuals who can provide details necessary to perform the tasks. Ensure the users are participating in a cooperative manner and are allocated the required time necessary to actively participate. Ensure that users understand, and agree, their part in working on and reviewing the project is vital. Identify 'trainers' and users who need

training. With the assistance of the development team and project leader, begin to develop the training plan.

User: Assess the usability of the design and determine the required procedures to support it. Engage in system testing where required and verify that test plans cover all conditions perceived in the business.

Rapid Application Development [RAD] Approach

The traditional Systems Development Lifecycle Methodology was developed in an era where programming tools were limited and consequently the time and costs of programming the application screens and reports consumed a major part of the overall project costs. The methodology was intended to reduce total costs by making sure that everything was designed and specified on paper before any programming was initiated. Traditionally the costs of changing things on paper were far less significant than changing actual programs. So the greatest cost savings on a project were ensured if everything was fully specified in great detail on paper in the Design Specifications before a single line of code was written.

Today's software development tools such as Oracle Developer have greatly changed the cost ratios that this traditional methodology was based upon. Today changes to screen and report programs can often be made as rapidly and cheaply as changes to written specifications. Consequently, in certain circumstances with certain projects, the traditional methodology actually adds unnecessary costs to the project. In these instances the traditional Design report can be dispensed with and the Design/Build phases can be completed in tandem through iterative prototyping.

In the RAD-Prototyping approach, instead of preparing written design specifications which include drawing screen designs and report layouts on paper prior to actually building them with the development tools, they are actually painted via the development tools. Thus the two processes are combined into one.

An iterative process is used to refine the design during the build process. A prototype module is defined at a high level. The screen/report/module is then built. The user reviews the prototype module and indicates changes or additional features needed. These are incorporated into the next iteration of the prototype and the process continues.

A Rapid Application Development approach in lieu of formal Design specifications is appropriate for projects or portions of projects having the following characteristics.

- Low Development costs due to small project team and limited duration
- Low Development turnover risk
- Low number of business functions and entities
- Low Integration Considerations
- Low Policy and procedure change

When all or most of these factors are present, the iterative build costs are low and the risk of cost overruns is also low. In these instances the costs of drafting formal design documentation before beginning building prototypes is excessive relative to the total project cost, and the RAD approach is indicated.

A document template for the Prototype Documentation is available as a word document.

Build and User Documentation Stage

The purpose of the build and documentation stage is to develop and test the component modules described in the design stage. This includes execution of a multi-tiered test strategy, refinement of the system transition plan, installation of production hardware and software (if applicable); development and refinement of required manual procedures, development of end-user training materials, and development and refinement of user and operations manuals. The primary activities that may comprise the build and user documentation stage are outlined in Build and User Documentation Stage Activities and Deliverables.

Build and User Documentation Stage	Activities and Deliverables
Roles & Responsibilities • Management • Project Manager • Database Administrator • System Analyst • Programmer • Operations • Sponsoring User • User	Build Tasks (outlined by Richard Barker)

Build and User Documentation Stage Activities and Deliverables

Activities

Tasks	Description	Check
10	Project Administration and Management	
20	Prepare For Build Stage	
30	Review Designs and Estimates with Programmer	
20	Complete User Documentation	
30	Provide Operations hand-over Documentation	
40	Produce Programs	
50	Prepare, Perform and Review System Test	
60	Review Test Results	

70	Obtain Stage-End Commitment	

Deliverables

☐	Completed & tested modules
☐	Detailed Module Specifications
☐	Fully tested system
☐	Installed development hardware/software and early indications of performance
☐	End-user documentation and Operations (hand-over) documentation

A document template for the Implementation Documentation and template for the User's guide are available as a word document.

Build Stage Roles and Responsibilities

Management: Their primary objective is making sure that the clients invest enough of their resources into the project to make it a success. They are responsible for defining project goals and resolving conflicts.

Project Manager: Responsible for managing the project team, project planning, control and implementation, communicating the progress and issues to the team members. It is also important that the Project Manager ensures that the project is going in the right direction and anticipates any needs or problems.

System Analyst: Responsible for transforming analysis specifications into actual system design. Work closely with analysts and users to ensure the design meets the intended requirements.

Database Administrator: Creating/altering all test database objects according to the initial physical database design, including table/dataset definitions, index definitions, and space requirements. Determine the physical implications. Establish a standard "look and feel" for all forms and reports to be utilized throughout the system through Application Preferences. Review all design for compatibility and adherence to standards of the design.

Programmers: Responsible for coding the modules into programs following the design developed in the Design Stage. Rigorously test any limits, exceptions or other aspects of the programs.

Operations: Responsible for developing operation procedures to support the system. Provide input to system design.

Sponsoring User: Identify individuals who can provide timely details necessary to perform the tasks. Ensure the users are participating in a cooperative manner and are allocated the required time necessary to actively participate. Ensure that users understand, and agree, their part in working on and reviewing the project is vital. Ensure that, where feasible, design issues have

been resolved prior to coding. Finalize training plan, with the assistance of the development team and the project leader. '*Trainers'* identified in the plan should receive training from the development team*. 'Trainers'* can begin training the users.

User: Assess the usability of the modules and determine the required procedures to support it. Effectively convey patterns of work so the design provides for efficient interface. Engage in system and module testing and verify that system covers all procedures perceived in the business. Ensure the quality of both tests and results.

Transition Stage

The transition stage focuses on data conversion, data cutover, system acceptance testing, end-user training, and implementation of operational procedures to support day-to-day usage of the system. This includes establishing a support mechanism *(i.e. help desk)* and a process for identification and resolution of system-related problems. The primary activities and deliverables that may comprise the transition stage are outlined in Transition Stage Activities and Deliverables.

Transition Stage	Activities and Deliverables
Roles & Responsibilities • Management • Project Manager • Project Team • Database Administrator • Operations • Sponsoring User • User	Transition Tasks (Outlined by Richard Barker)

Transition Stage Activities and Deliverables

Activities

Tasks	Description	Check
10	Project Administration and Management	
20	Train Users	
30	Prepare for Acceptance Testing	
40	Support Acceptance Test	
50	Perform Data Take-on	
60	Carry Out Installation of Hardware, System Software, and Other Components of the Production Configuration	
70	Perform Any Other Pre-Implementation Trials	
80	Prepare for Cut-Over	
90	Perform Cut-Over	

100	Support System During the Critical Period	
110	Perform Post-Implementation Review	

Deliverables

	Installed, operational, and accepted system
	Completed and approved documentation
	Trained users and support staff
	Help desk and end-user support
	Operations procedures
	Acceptance test results
	Change request and problem reporting procedures

A document template for the System and control report is available as a word document.

Transition Stage Roles and Responsibilities

Management: Their primary objective is making sure that the clients invest enough of their resources into the project to make it a success. They are responsible for defining project goals and resolving conflicts.

Project Manager: Responsible for managing the project team, project planning, control and implementation, communicating the progress and issues to the team members. While each group keeps focused on their specific area, the Project Leader must ensure that dependencies between groups are addressed.

Project Team: Advise users in determining the scope of the acceptance testing. Update documentation as required to reflect changes identified during testing and training. Provide system overview and training to users and managers. Prepare for and perform move to production. Provide analysis and programming skills where required to support system transition.

Database Administration: Create database objects within the test environment. Assist in the diagnosis of problems to the extent of determining database issues and providing co-ordination of database resources necessary for resolution of data and/or logic problems.

Resources include test work areas, database utilities, SQL tuning, and Oracle support. Submit any software or hardware issues discovered during the testing cycle to the Project Team. Prepare a final Impact Assessment and/or Project Plan for final database implementation jointly with the Project Team.

Perform the actual production database creation from scripts provided by the Project Team.

Operations: Assist users in the planning and preparation of the acceptance test. Maintain log of system faults. Carry out installation of hardware, system software and other components of the production configuration.

Sponsoring User: Identify individuals who can provide time and knowledge necessary to evaluate and test system and provide feedback. Ensure users are participating in a cooperative manner and are allocated the required time necessary to actively participate. Ensure that testing and training reflect actual usage of system.

User: Responsible for developing the acceptance test criteria and determining whether the scope of the planned acceptance test is satisfactory. Test operational aspects of the system and document or log all tests and associated results. Assist in data conversions and verify the integrity of the data.

Production Stage

The production stage (*or maintenance stage)* focuses on on-going support for the newly delivered system. It provides for system monitoring and performance tuning, fixing any latent "bugs" that were not uncovered during the testing process, and implementing changes requested by the end-users. Also included during the production stage are basic operations procedures such as data backup, archiving, and recovery.

The primary activities and deliverables that may comprise the production stage are outlined in Production Stage Activities and Deliverables.

Production Stage	Activities and Deliverables
Roles & Responsibilities • Management • Project Manager • Project Team • Database Administrator • Operations: • Sponsoring User • User	Production Tasks(Outlined by Richard Barker)

Production Stage Activities and Deliverables

Activities

Tasks	Description	Check
10	Provide Operational Service	
20	Respond to User Requests	
30	Monitor/Review Performance	
40	Assess the Future of the System	

Deliverables

	Production Documentation
	Change requests/problem reports
	Performance reports

	On-going training and support

A document template for the Production Documentation is available as a word document.

Production Stage Roles and Responsibilities

Management: Their primary involvement is in making sure that the right people in the business give enough of their time to the project to make the Production a success.

Project Manager: Maintain close liaison with users, development, training and support staff. Develop plan to include co-existence with existing systems, integration with automated office systems and any other operational aspects when changes are implemented. Ensure that changes to the system are done in a manner that minimizes impact on the business.

Project Team: Respond to user requests; determine impact of requested changes. Examine requested changes and determine if they are within the scope of the system objectives. Provide analysis and programming skills where required to support production. Ensure other components have not been affected by changes to the system.

When testing is complete and the change(s) is determined to be desired upon the production database, prepare a request detailing the modification(s) and submit it to the Database Administrations Section.

Update documentation and training where required.

Database Administration: Prepare an Impact Assessment or Section Project Plan to implement the change(s). Implement the desired modification(s) in conjunction with the Project Team and Operations.

Sponsoring User: Identify individuals who can provide time and knowledge necessary to evaluate system and provide feedback. Ensure users are participating in a cooperative manner and are allocated the required time necessary to actively participate. Ensure that changes to the system are done in a manner that minimizes impact on the business. Ensure that the assessment criteria for the system audit are appropriate and measurable.

User: Control system parameters and monitor usage and performance.

Automated Tool Support

The system development approach is predicated on the use of automated software tools, documentation templates, development checklists, and procedures to automate, manage, and support the software development process. Integration of software tools into the development process facilitates development of high-quality software systems in a timely and cost-effective manner. The following tools are recommended to be utilized to support all aspects of a project.

Microsoft Project: The project planning and management tool might support task definition, scheduling, resource loading, cost monitoring, and project status assessment and reporting activities, if available.

Application Designer: The front-end analysis tool for developing graphical models of the system being developed. Depends on the project, either the PeopleSoft Application designer or Oracle Designer can be used. This tool provides the capability to develop detailed data and function models through an easy-to-use graphical user interface *(GUI)*. This tool also allows navigation of the central repository for all analysis, detailed design, and implementation-related information pertaining to each system. This tool supports all stages of the SDM, from strategy through production. An Application Designer can also be used to automate the build stage *(coding stage)* of the development process to the extent possible.

Oracle Forms Builder: The general-purpose tool for developing and executing interactive forms-based applications *(screens).* Forms Builder is a 4GL application development tool that provides an overall application structure and framework for creating, displaying, and updating data residing in an Oracle database. Forms Builder screens are the primary user interface to data residing in the Oracle database.

Oracle Report Builder: a database reporting tool that enables application developers to create fully formatted, multi-part reports. Report Builder is the primary tool for developing standard formatted reports.

TOAD: TOAD is a database reporting tool for producing formatted reports and writing command procedures to manage information on an ORACLE database.

Oracle PL/SQL: PL/SQL is an application development tool, which allows the use of procedural techniques, such as looping and branching to process data. PL/SQL combines the data manipulating power of SQL with the data processing power of procedural languages.

HEAT: Problem tracking and reporting tool, which allows users to report problem via WEB interface. It also has the ability to notify assigned technical staff and escalate an issue. UTHSC-H also uses this tool to record changes.

Quest Stat: Application change management tool, which allows version control and version roll back. It has application lifecycle migration management, environment synchronization and comparison features. It supports central repository for auditing and reporting.

SDM Document Templates: Document templates are created for each of the SDM deliverable documents to be produced during the course of each system development project. These templates are electronic files containing a standard document format, detailed descriptions of what is to be placed in each respective section, and quality control checklists to help ensure that all tasks are complete and concise. Document templates expedite the document *"writing"* process by providing a standard document format and supporting boilerplate text.

System Development Methodology Guide: The SDM Guide available via Information Services WEB page is a "*desk-top reference*" provided to each project developer. The guide outlines details of the system development methodology *(SDM)* and specifies project standards, conventions, and guidelines to be adhered to throughout each stage of the development process. You are currently reading a page of the SDM Guide.

Quality Assurance Activities

The term "***software quality"*** is used extensively in the world of applications development. Software quality is an elusive term that has different meanings to different people.

For the purposes of this document, a high-quality application is one that:

- Carries out the purpose intended by the user.
- Is delivered on time and within budget.
- Has no error.
- Can be maintained and modified to meet changing conditions at a minimal cost.

Quality cannot be tested into an application; it must be built into it. Software Quality Assurance *(SQA)* is an *"umbrella activity"* applied throughout the development process to ensure that proper quality controls are built into the development process. Software Quality Assurance encompasses:

- Standard analysis, design, and development techniques.
- Formal technical reviews and walkthroughs.
- Informal technical reviews and walkthroughs.
- Testing strategies.
- Configuration management and change control procedures.
- Assessment and reporting mechanisms.

Software Quality Assurance begins with proven technical methods and techniques. These *"methods and techniques"* are the tasks and deliverables comprising the System Development Methodology. It is the responsibility of the Project Manager to ensure that all project activities and work products are performed and developed in accordance with the System Development Methodology. It is also there responsibility for supporting standards and guidelines for activities and products.

Structured Walkthroughs and Technical Reviews outline the quality assurance practices to be applied throughout the development process.

Structured Walkthroughs and Technical Reviews

The two most effective techniques for assuring software quality are Structured Walkthroughs and Technical Reviews. These project reviews occur many times throughout the development process. The principle objective of these reviews is to use participants' *"fresh look"* at the product *(deliverable)* to find as many errors, deficiencies, or problems as possible, so that they can be fixed before additional work is performed. This technique is based on the fact that the earlier an error is found, the cheaper it is to fix.

A **Technical Review** is an informal review by the project team of an intermediate *deliverable (i.e. data model, function hierarchy, procedure logic, etc.).* The product is scrutinized for completeness, correctness, consistency, technical feasibility, efficiency, and adherence to established standards and guidelines by the Client Organization. Technical Reviews should be performed for all deliverables outlined in CASE*Method.

The **Structured Walkthrough** is a review of the formal deliverables *(i.e. those deliverables which are reviewed and signed-off by the client organization)* produced by the project team. Participants of this review typically include end-users and management of the client organization, management of the development organization, and sometimes auditors, as well as members of the project team. As such, these reviews are more formal in nature with a predefined agenda, which may include presentations, overheads, etc. The Project Manager determines the degree of formality.

The materials to be reviewed should be distributed to participants at least two days prior to the review date to ensure adequate time for review and preparation. Structured Walkthroughs are major milestones in the project and should be represented as such in the project plan and schedule.

Issue Resolution and Problem Reporting

During the course of a project, many questions, problems, and recommendations are encountered that can affect the cost, schedule and deliverables of the project. By resolving these matters in a consistent and disciplined manner, the quality of the project can be significantly improved while maintaining mutually agreeable delivery schedules and cost. The issue resolution process ensures that differences, questions, and unplanned requests are properly defined, raised to management attention, and resolved quickly. This process should be defined during the Strategy Stage by the Project Manager and enforced throughout the project.

An issue is a circumstance that prevents *(or limits the effectiveness of)* a team member or an end-user from performing their task on time or within established quality standards. Issues include technical problems, awaiting answers for unanswered questions regarding desired functionality, and people/organizational problems.

The process for identifying, documenting, and resolving issues is established at the beginning of the project and must be communicated to project team members, management, and the client organization. The issue resolution/escalation process should be documented in the Project Plan.

Issues that do arise during the course of the project should be communicated to management in the Project Status Reports and documented in the Project Notebook. Issues must be closely monitored to ensure that appropriate action is taken to bring them to closure.

Information that should be tracked for each issue includes:

Name of person who identified the issue and date recorded
Description of the issue
The impact *(or potential impact)* of the issue
The corrective course of action
Disposition of the Issue/Resolution date.

A standard Change Control form is available in WORD format. This form should be used to document issues and issue status-related information in a consistent manner.

Security and Access Policy

Application Security:

Application security should be dictated by the needs of the application, while recognizing the principles of the §TAC 202's presumption that all records are public unless there is some compelling reason to the contrary. Protecting the integrity of the data through protecting the integrity of the application and the data input/validation procedures will require maintaining some access control.

For each application developed by or under the direction of UTHSC-H's CIO, the §TAC 202 and UTHSC-H security policy have to be followed closely.

Access to Program/Database Objects:

Program and database objects will include: database tables and schema, grants and roles, DML scripts, executables, libraries, source files, etc. Access to these objects should be limited to the application development team to assure data integrity. The application DBA should also take steps to ensure that there is no conflict between developers in the use of these objects.

The application which originally created these objects should retain ownership of the objects through its application DBA. However, in the interests of promoting reuse of objects in order to improve productivity, application code and libraries should be shared when appropriate. Sharing of these objects with other development teams should be coordinated through the application DBA having original ownership of these objects. In order to ensure proper application functioning these objects are not accessible to end users.

Access to Data in Database Tables:

Access to database tables is controlled by roles. Some roles will have update privileges others will have only query capabilities. Individual users can be granted roles by application owners which determine their access to the data.

Access to production application programs, reports, and queries:

Access to application programs is controlled by roles assigned by the application owners. Determination of what roles are to be granted to what individuals is based on discussion between the application DBA and the systems coordinator for that application. The operating principle in accessing report/query data, however, is that unless contraindicated by the sensitive nature of some data or system performance constraints, most report data is assumed to be accessible to department users.

Access to ad hoc reporting tools:

Due to resource constraints posed by improper use of such tools, restrictions on the number of users having access to these tools must be limited. Users must have appropriate training in both the tools themselves and the data structures that they plan to access to assure accurate reports and reasonable use of system resources. Such access must be approved by the user's management to assure reasonableness of request.

System Usability

This section presents practical guidelines for managing the system usability as it pertains to program development.