Chapter: **5.0 Functional Design Stage**

Description: The functional design process maps the "what to do" of the Requirements Specification into the "how to do it" of the design specifications. During this stage, the overall structure of the product is defined from a functional viewpoint. The functional design describes the logical system flow, data organization, system inputs and outputs, processing rules, and operational characteristics of the product from the user's point of view. The functional design is not concerned with the software or hardware that will support the operation of the product or the physical organization of the data or the programs that will accept the input data, execute the processing rules, and produce the required output.

The focus is on the functions and structure of the components that comprise the product. The goal of this stage is to define and document the functions of the product to the extent necessary to obtain the system owner and users understanding and approval and to the level of detail necessary to build the system design.

Prototyping of system functions can be helpful in communicating the design specifications to the system owner and users. Prototypes can be used to simulate one function, a module, or the entire product. Prototyping is also useful in the transition from the functional design to the system design.

Input: The following work products provide input to this stage:

SEM Templates:

- Maintenance Plan (SEM-0301)
- Requirements Specification (SEM-0402)
- Requirements Traceability Matrix (SEM-0401)
- Software Configuration Management Plan (SEM-0302)

PMM Templates:

- Project Plan (PMM-03 or PMM-03 Exp)
- Quality Management Plan (PMM-07)

Other Inputs:

- Security Plan and Assessment (DIT-0170 or DIT-0170 Exp)
- Business Continuity Plan
- MDIT Hosting Solution Document

Testing Activities within Functional Design: Refer to the Testing Process Manual; Functional Design Stage 4.0 Section, for testing activities.

High-Level Activities:

The remainder of this chapter is divided into sections that describe specific high-level activities performed during this stage. These activities represent the minimum requirements for a large information systems engineering effort. Notes are provided, as applicable, to assist in customizing these lifecycle stage requirements to accommodate different sizes of systems engineering efforts.

The high-level activities are presented in the sections listed below.

5.1 Determine System Structure
5.2 Design Content of System Inputs and Outputs
5.3 Design User Interface
5.4 Design System Interfaces
5.5 Design System Security Controls
5.6 Build Logical Model
5.7 Build Data Model
5.8 Develop Functional Design
5.9 Select System Architecture

Touch Points:

The following touch points are involved in the Functional Design Stage:

- Contracts and Procurement
  - Contract liaison involvement in the process, if contract issues arise
- E-Michigan
  - Continue to work with E-Michigan's webmaster, as appropriate, to ensure ADA compliance and Michigan.gov look and feel standards.
- Infrastructure Services
  - Review and complete Hosting Solution document
- Security
  - Review MDIT and Agency security policies
  - Review State and Federal laws and regulations
  - Review existing or propose new security controls
  - Conduct preliminary risk analysis
  - Revise Infrastructure/Network and Data Flow Diagram

Output:

Several work products are developed during this stage. The work products listed below are the minimum requirements for a large systems project. Deviations in the context and delivery of these work products are determined by the size and

complexity of a project. Explanations of the work products are provided under the applicable activities described in the remainder of this chapter.

SEM Templates:

- Functional Design Document (SEM-0501) *[final]*
- Maintenance Plan (SEM-0301) *[revised]*
- Requirements Specification (SEM-0402) *[final]*
- Requirements Traceability Matrix (SEM-0401) *[revised]*
- Software Configuration Management Plan (SEM-0302) *[revised]*

PMM Templates:

- Project Plan (PMM-03 or PMM-03 Exp) *[revised]*

Other Outputs:

- Security Plan and Assessment (DIT-0170 or DIT-0170 Exp) *[revised]*
- Business Continuity Plan *[revised]*
- Data Dictionary *[final]*
- MDIT Hosting Solution Document *[final]*

Review the Project Plan for accuracy and completeness of all Functional Design Stage activities and make any changes needed to update the information.

Review Process: Quality reviews are necessary during this stage to validate the product and associated work products. The activities that are appropriate for quality reviews are identified in this chapter and Chapter 2.0, Lifecycle Model. In addition, a Preliminary Design Review will be conducted. This review is an important milestone in the design process. The time and resources needed to conduct the walkthroughs and Functional Design Review should be reflected in the project resources, schedule, and work breakdown structure.

Structured Walkthrough (SWT)

Requirements for a peer review or a more formal structured walkthrough are documented under *Review Process* at the end of each Task, Subtask, or Activity section in this stage. The State of Michigan guide titled *Structured Walkthrough Process Guide* provides a procedure and sample forms that can be used for SWTs. This document is available on the MDIT SUITE website.

Stage Exit

Schedule a Stage Exit as the last activity of the Functional Design Stage to enable the project approvers to review project deliverables and provide a concur/non-concur position to the project manager. The State of Michigan guide titled *Stage Exit Process Guide* provides a procedure and sample report form that can be used for stage exits. This document is available on the MDIT SUITE website.

References: Chapter 2.0, Lifecycle Model, *Quality Reviews* provides an overview of the Quality Reviews to be conducted on a project.

Bibliography: The following materials were used in the preparation of the Functional Design Stage chapter.

1. Barker, Richard, *CASE*METHOD,* Tasks and Deliverables, 1990. pp. 5-10 and 5-11.
2. Gane, Chris and Sarson, Trish, *Structured Systems Analysis: Tools and Techniques,* Prentice-Hall, Inc., Englewood Cliffs, 1979.
3. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Guide to Software Design Descriptions,* IEEE Std 1016.1-1993, New York, 1993.
4. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Guide to Software Requirements Specifications,* ANSI/IEEE Std 830-1998, New York, 1998.
5. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Developing Software Lifecycle Processes,* IEEE Std 1074-1991, New York, 1992.
6. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Software Verification and Validation Plans,* ANSI/IEEE Std 1012-1998, New York, 1998.
7. U.S. Department of Defense, Military Standard, *Technical Reviews and Audits for Systems, Equipments, and Computer Software,* MIL-STD-1521 B, 1985. pp.5, 33-52.
8. U.S. Social Security Administration, Office of Systems, *Software Engineering Technology (SET) Manual*, Volume 1, 1990.

Activity: **5.1 Determine System Structure**

Responsibility: Project Team Analysts

Description: A hierarchical approach is useful for determining the structure and components of the product. System decomposition is one hierarchical approach that divides the system into different levels of abstraction. Decomposition is an iterative process that continues until single purpose components (i.e., design entities or objects) can be identified. Decomposition is used to understand how the product will be structured, and the purpose and function of each entity or object.

The goal of the decomposition is to create a highly cohesive, loosely coupled, and readily adapted design. A design exhibits a high degree of cohesion if each design entity in the program unit is essential for that unit to achieve its purpose. A loosely coupled design is composed of program units that are independent or almost independent.

Several reliable methods exist for performing system decomposition. Select a method that enables the design of simple, independent entities. Functional design and object-oriented design are two common approaches to decomposition. These approaches are not mutually exclusive. Each may be applicable at different times in the design process.

Tasks: The system decomposition activity includes the following tasks.

5.1.1 Identify Design Entities
5.1.2 Identify Design Dependencies

Task: **5.1.1 Identify Design Entities**

Description: Design entities result from a decomposition of the product requirements. A design entity is an element (or object) of a design that is structurally and functionally distinct from other elements and is separately named and referenced. The number and type of entities required to partition a design are dependent on a number of factors, such as the complexity of the product, the design method used, and the development environment. The objective of design entities is to divide the product into separate components that can be coded, implemented, changed, and tested with minimal effect on other entities.

Attributes: A design entity attribute is a characteristic or property of a design entity. It provides a statement of fact about an entity. The following are common attributes that should be considered for each design entity.

- Assign a unique name to each entity.
- Classify each entity into a specific type. The type may describe the nature of the entity, such as a subprogram or module; or a class of entities dealing with a particular type of information.
- Describe the purpose or rationale for each entity. Include the specific functional and performance requirements for which the entity was created.
- Describe the function to be performed by each entity. Include the transformation applied to inputs by the entity to produce the desired output.
- Identify all of the external resources that are needed by an entity to perform its function.
- Specify the processing rules each entity will follow to achieve its function. Include the algorithm used by the entity to perform a specific task and contingency actions in case expected processing events do not occur.
- Describe the data elements internal to each entity. Include information such as the method of representation, format, and the initial and acceptable values of internal data. This description may be provided in the data dictionary.

Work Product: Maintain a record of all design entities. The records will be integrated into the Functional Design Document.

Review Process: Schedule structured walkthroughs to verify that the design entities are correct, complete, and possess the required attributes.

Task: **5.1.2 Identify Design Dependencies**

Description: Design dependencies describe the relationships or interactions between design entities at the module, process, and data levels. These interactions may involve the initiation, order of execution, data sharing, creation, duplication, use, storage, or destruction of entities.

Identify the dependent entities of the system design, describe their coupling, and identify the resources required for the entities to perform their function. Also define the strategies for interactions among design entities and provide the information needed to perceive how, why, where, and at what level actions occur.

Dependency descriptions should provide an overall picture of how the product will work. Data flow diagrams, structure charts, and transaction diagrams are useful for showing the relationship among design entities.

The dependency descriptions may be useful in producing the system integration plan by identifying the entities that are needed by other entities and that must be developed first. Dependency descriptions can also be used to aid in the production of integration test cases.

Work Product: Add specific dependency information to the design entity records. The records will be integrated into the Functional Design Document.

Review Process: Schedule structured walkthroughs to verify that the design entities and dependencies are correct, complete, and possess the required attributes.

Activity: **5.2 Design Content of System Inputs and Outputs**

Responsibility: Project Team Analysts

Description: Design the content and format for each of the product inputs and outputs based on the system input and output requirements identified during the Requirements Definition Stage. Involve the system owner and users in the design process to make certain that their needs and expectations are being met.

Procedure: Use the following procedure to implement the design process.

- Identify the types of electronic and printed input that will be accepted by the product, such as data entered manually from source documents and files or records extracted from other systems.
- Identify the types of electronic and printed output that will be produced by the product; such as data, records, or files; screen displays; and printed reports. Also, identify the output that will be exported to other systems.
- Identify the specific input and output items that already exist and the items that will be created for input or output as part of the product.
- Assign a name to each type of input and output and describe each item from a functional perspective.
- Identify the owner/originator of each type of input and output.
- Identify the frequency of each type of input and output.
- Design the content and format for each new input and output item or modify the format of existing items that must be changed to accommodate the new product.

Work Product: Document the design for the system inputs and outputs in accordance with the project design standards. Discuss the designs with the system owner and users and submit completed designs for their review and approval. The approved designs will be incorporated into the Functional Design Document.

Review Process: Schedule a structured walkthrough to verify that the system input and output designs are correct and complete and describe the designs in a manner that can be understood by the system owner and users.

Activity: **5.3 Design User Interface**

Responsibility: Project Team Analysts

Description: Design a user interface that is appropriate for the users, content, and operating environment for the product. Determine interface levels for all categories of users. For interactive user environments, prototype the user interface. Arrange for users to experiment with the prototypes so that design weaknesses in the interface can be identified and resolved early. Use prototypes to gain user acceptance of the interface.

MDIT standards for e-government applications and relevant State of Michigan standards should be used to specify the user interface for every product developed.

Basic Principles: The following basic principles can help improve the product user interface when there is graphical, command-based, menu-driven, or block mode features.

- Give users control. Let them choose actions to perform.
- Give users feedback and progress reports. Tell them when the system is working and when an action is completed.
- Be consistent in the format and wording of text.
- Keep it simple. White space is as important on the screen as on the printed page. Reduce screen clutter.
- Put information where it can be easily seen; avoid information in corners or borders.
- Limit the amount of information users must know. Offer choices instead of making users remember and manually enter information. Provide defaults, and make sure they are logical and satisfy a large number of users.
- Offer shortcuts. Keyboard shortcuts (e.g., hot keys) and command abbreviations help experienced users work more quickly.
- Help users get out of trouble. Provide messages that are understandable and that offer solutions.
- Let users reverse their actions. If an action will destroy something, identify the object of destruction and wait for a response.

Tasks: The following tasks are involved in specifying the user interface.

5.3.1 Design Menu Hierarchy
5.3.2 Design Data Entry Screens
5.3.3 Design Display Screens
5.3.4 Design Online Help
5.3.5 Design System Messages

Task: **5.3.1 Design Menu Hierarchy**

Description: Use the following guidelines to improve the design of menu hierarchies.

- Choose an organizing principle for the menu options, such as:
  - Expected frequency of use
  - Logical sequence of operations
  - Alphabetical order (should be used for horizontal word menus with five or more words)
- Put a meaningful title at the top of every menu.
- For full-screen menus, provide symmetric balance by centering the title and the menu options around the center axis of the screen.
- To facilitate scanning, put blank lines between logical groupings of menu options and after about every fifth option in a long list.
- Limit the number of menu choices to one screen.
- Select icons that are intuitive to the function they represent.
- Use a menu option selection method that is consistent with the technology available at the user's workstation and the size of the product being designed, such as:
  - Numbers
  - Letters or letter combinations
  - Cursor movement
- Provide a way for the user to leave the menu without performing any action. Be sure that the option to leave the menu describes the consequences of its selection.
- Words used for menu options should follow these rules:
  - Use words that clearly and specifically describe what the user is selecting.
  - Use common English words rather than computer or technical jargon. When space permits, spell out words completely.
  - Use simple, active verbs to tell users what actions will result from their choice. Try to start each option with a verb.
  - Use parallel construction to describe the options.
- Minimize the highlighting used on a menu. Highlighting should be limited to situations where the user needs to know that there is an exception to the normal practice.
- Do not require the user to enter leading or trailing blanks or zeros, and do not include a default value on a menu.
- Display the menu options in mixed letters (i.e., upper and lower case).
- Organize menu hierarchies according to the tasks users will perform, rather than the structure of the modules.

Work Product: Document the design for the menu hierarchy in accordance with the project design standards. Discuss the design with the system owner and users and submit the completed design for their review and approval. The approved design will be

incorporated into the Functional Design Document.

Review Process: Conduct a structured walkthrough to ensure that the menu hierarchy design is complete, logical and describes the design in a manner that can be understood by the system owner and users.

Task: **5.3.2 Design Data Entry Screens**

Description: Use the following guidelines to improve the design of data entry screens.

- When the user must transcribe data directly from a source document to the screen, the layout of the screen should be similar to the layout of the source document.
- Group data fields into logical categories on the screen; provide a header that describes the contents of each category.
- Make areas of the screen that are not needed for data entry or commands inaccessible to the user.
- Do not require the user to enter information that is already available to the software or can be computed by it.
- Do not require the user to enter dimensional units, leading or trailing blanks, or zeros.
- Allow the user to enter data by character replacement.
- Put a caption describing the data to be entered adjacent to each data field; incorporate memory joggers into the caption.
- Justify data entries automatically.
- Display default values in data fields when appropriate.
- Provide context-sensitive help for data entry fields.

Work Product: Document the designs for the data entry screens in accordance with the project design standards. Discuss the design with the system owner and users and submit the completed designs for their review and approval. The approved designs will be incorporated into the Functional Design Document.

Review Process: Conduct a structured walkthrough to assure that the data entry screen designs are consistent, complete, and logical and describe the designs in a manner that can be understood by the system owner and users.

Task: **5.3.3 Design Display Screens**

Description: Use the following guidelines to design display screens that are easy to use and understand.

- Put a title on every display screen. The title should clearly and specifically describe the contents of the screen.
- Display only information that the user needs to know.
- Display data to the user in directly usable form.
- Provide symmetric balance to displays by centering titles and headings and by placing information on both sides of the center axis.
- Every display should indicate how to exit from the screen. Use consistent exit procedures.
- When the display continues over multiple screens, the screen should indicate where the user is in the display (e.g., Screen 1 of 3).
- Data fields need to be grouped into logical categories or according to the structure of a source document (when there is one).
- Be consistent in the use of words and special characters.
- Display text conventionally in mixed letters (i.e., upper and lower case) and with appropriate punctuation. Avoid all uppercase letters. Put a blank line between paragraphs.
- Left justify text, and leave a ragged right margin.
- Avoid hyphenation of words between lines.
- Use abbreviations and acronyms only when they are significantly shorter than the full text and when they will be understood by the user.
- Be consistent with the format of information being displayed.
- Consider the skills of the users and the information they will manipulate when information is displayed in multiple windows.

Table and List Guidelines: Use the following guidelines to improve the design of online tables and lists.

- Put a meaningful label on the columns and, if appropriate, the rows of tables and lists. Continue the labels when a table or list extends over more than one screen.
- If data items are scrolled, the labels should be fixed on the screen and not be part of the scrolled area (they remain in place as the body of the table or list changes).
- If data items are continued on subsequent screens, the labels should be added to each screen.
- Arrange the items in a table or list in some recognizable order to facilitate scanning.
- Put items in a multiple column list in vertical columns that are read from left to right on the screen.
- Left justify columns of alphabetic data; right justify columns of numeric data or align them by the decimal point or other delimiter.
- Insert a blank line after about every fifth row in a long column.
- Insert a minimum of two spaces between the longest item in a column and the beginning of the next column.
- Start with a one (1) not a zero (0) when listed items are labeled by number.

Work Product: Document the design for the display screens in accordance with the project design standards. Discuss the designs with the system owner and users and submit the completed designs for their review and approval. The approved designs will be incorporated into the Functional Design Document.

Review Process: Conduct a structured walkthrough to ensure that the display screen designs are consistent, complete, logical complete, logical and describe the designs in a manner that can be understood by the system owner and users.

Task: **5.3.4 Design Online Help**

Description: Online help is typically requested by users when they want to perform a new, complex, or infrequently used procedure, or when they do not know what else to do. The text of online help messages needs to be planned, drafted, and evaluated as carefully as print documentation. In addition, the layout and format of online help must be designed to deal with the special constraints imposed by the video screen.

Use online help to explain concepts, procedures, messages, menu choices, commands, words, function keys, and formats. Work with the users to identify the level of detail needed for online help. Determine whether the users need a one-line message at the bottom of the screen or a full online explanation with successive levels of detail.

Effective online help messages tell users what the product is doing, where they are in the sequence of screens, what options they have selected, and what options are available.

Guidelines: The following guidelines can improve the design of online help.

- Write online help messages in plain english.
    - Straightforward and reads as if it were spoken.
    - Clear, direct, and simple.
    - Effectively organized with a concern for what users need to know.
- Address the user directly as "you"; use the active voice.
- Use simple action verbs to describe procedures. Do not use nouns to replace pronouns, verbs, and adjectives.
- Describe procedures in logical order.
- Avoid computer terms or other jargon, such as:
    - Terms that are unique to the computer profession or to a particular company.
    - Terms that have a common meaning outside of the data processing environment, but a special meaning within it, such as *boot*, *abort*, *default*, and *utility*.
    - Terms that are created to describe some special function, such as *ungroup* and *de-archive*.
- Avoid humor in online documentation.
- Write in short complete sentences and paragraphs and use proper punctuation.
- Write sentences in the positive or simple negative. Avoid the passive voice and do not use double negatives.
- Use bullets, numbered lists, and tables to make it easier to find the most important information. Leave ample open space.
    - Use bulleted lists to explain options. Whenever a sentence lists

options with commas between them, consider breaking up the text into a bulleted list.
    - Use numbered lists to show the steps in a process.
    - Use a table to explain two or more categories of information.
- Use examples to show users what they should enter and what the results will look like.
- Do not expect users to read more than about three screens of help at one time.
- Provide an orientation to the structure of the product.
- Whenever possible display help text on the screen with the function or task that is being performed.
- Provide a direct route back to the function or task being performed.

Work Product: Document the design for online help in accordance with the project design standards. Discuss the design with the system owner and users and submit the completed design for their review and approval. The approved design will be incorporated into the Functional Design Document.

Review Process: Conduct a structured walkthrough to ensure that the online help design is consistent, complete, and logical and describe the design in a manner that can be understood by the system owner and users.

Task: **5.3.5 Design System Messages**

Description: System messages are the various types of information that the system provides to the user such as status messages, user prompts, and error messages.

Status Messages: Status messages are important for giving users the feeling they are in control of the system. They tell users what the system is doing, where they are in the sequence of screens, what options they have selected, and what options are available.

User Prompts: Prompts inform the user to type data or commands or to make a simple choice.

- Use prompts to ask the user to make a simple choice or to enter data or commands. Be as specific as possible.
- Include memory aids in the prompt to help users type a response in the proper format and order, initiate infrequently used processes, or clearly identify exceptions to normal practice.
- When defaults are allowed with prompts, indicate clearly which default value will be initiated.

Error Messages: Error messages should allow users to recover from mistakes by making it clear what the mistake was and how to correct it. Error messages need to be specific about why a mistake was made.

- Design the product to check for obvious errors.
- Be as specific as possible in describing the cause of an error. Do not use error codes.
- Do not assign blame to the user or the product in an error message. Use a neutral tone.
- Whenever possible, the error message should indicate what corrective action the user needs to take.
- Be consistent in the format, wording, and placement of messages.
- Consider describing error messages at more than one level of detail.

Work Product: Document the design for the system messages in accordance with the project design standards. Discuss the designs with the system owner and users and submit the completed designs for their review and approval. The approved designs will be incorporated into the Functional Design Document.

Review Process: Conduct a structured walkthrough to ensure that the system message designs are consistent, complete, and logical and describe the designs in a manner that can be understood by the system owner and users.

Activity: **5.4 Design System Interfaces**

Responsibility: Project Team Analysts

Description: Develop a design depicting how the product will interface with other systems based on the system interface requirements identified in the Requirements Definition Stage. Submit the applicable interface designs for review by the system owner or system administrator for each system that will interface with the product. Any incompatibilities with the interfaces will be identified early in the design process and corrective actions can be initiated to assure each interface is properly designed and coded.

Sample Issues: The following list provides some of the issues that should be considered when designing the system interfaces.

- System inputs and outputs
- Method of interface
- Volume and frequency of data
- Platform of interfacing system
- Format of data
- Automatic or manual initiation of interface
- Need for polling device(s)
- Verification of data exchange
- Validation of data

Work Product: Document the design(s) for the system interfaces in accordance with the project design standards. Discuss the designs with the system owner and users and submit completed designs for their review and approval. The approved designs will be incorporated into the Functional Design Document.

Review Process: Schedule a structured walkthrough to verify that the system interface designs are correct, complete, and logical and describe the design in a manner that can be understood by the system owner and users.

Activity: **5.5 Design System Security Controls**

Responsibility: Project Team Analysts and MDIT Office of Enterprise Security liaison

Description: Design the security controls that will be incorporated into the product based on the security and access requirements identified during the Requirements Definition Stage. Design the security controls in conjunction with the site or system owner and the MDIT Office of Enterprise Security liaison.

Procedure: Use the following procedure to implement the design process.

- Identify the users and organizations that will have access to the product. Indicate what access restrictions they will have. All persons in a work area may not have the same security access level. Measures should be taken to assure that sensitive materials and systems requiring protection are not accessed by unauthorized individuals.
- Identify controls for the product, such as the user identification code for system access and the network access code for the network on which the product will reside.
- Identify whether access restrictions will be applied at the system, subsystem, transaction, record, or data element levels. Sensitive information must be protected in accordance with State of Michigan directives.
- Identify physical safeguards required to protect hardware, software, or information from natural hazards and malicious acts.
- Identify communications security requirements.

Work Product: Document the design for the system security controls in accordance with the project design standards. Discuss the design with the system owner and users and submit the completed design for their review and approval. The approved design will be incorporated into the Functional Design Document.

Review Process: Schedule a structured walkthrough to verify that the system security controls are correct, complete and describe the controls in a manner that can be understood by the system owner and users. Include the MDIT Office of Enterprise Security liaison in the walkthrough.

Activity: **5.6 Build Logical Model**

Responsibility: Project Team Analysts

Description: The logical model defines the flow of data through the system and determines a logically consistent structure for the system. Each module that defines a function is identified, interfaces between modules are established, and design constraints and limitations are described. The focus of the logical model is on the real-world problem or need to be solved by the product.

A logical model has the following characteristics:

- Describes the final sources and destinations of data and control flows crossing the system boundary rather than intermediate handlers of the flows.
- Describes the net transfer of data across the system boundary rather than the details of the data transfer.
- Provides for data stores only when required by an externally imposed time delay.

When building a logical model, the organization of the model should follow the natural organization of the product's subject matter. The names given to the components of the model should be specific. The connections among the components of the model should be as simple as possible.

Work Product: The logical model should be documented in user terminology and contain sufficient detail to obtain the system owner's and users' understanding and approval. Use data flow diagrams to show the levels of detail necessary to reach a clear, complete picture of the product processes, data flow, and data stores.

Maintain the logical model and data flow diagrams for incorporation into the Functional Design Document. Keep the logical model and diagrams up-to-date. They will serve as a resource for planning enhancements during maintenance, particularly for enhancements involving new functions.

Review Process: Schedule a structured walkthrough to verify that the logical model is correct, logical, and complete.

Activity: **5.7 Build Data Model**

Responsibility: Project Team Analysts

Description: A data model is a representation of a collection of data objects and the relationships among these objects. The data model is used to provide the following functions:

- Transform the business entities into data entities.
- Transform the business rules into data relationships.
- Resolve the many-to many relationships as intersecting data entities.
- Determine a unique identifier (key) for each data entity.
- Add the attributes (facts) for each data entity.
- Document the integrity rules required in the model.
- Determine the data accesses (navigation) of the model.

Work Product: The data dictionary is developed in this stage. Its purpose it to catalog every known data element used in the user's work and every system-generated data element. Data elements are documented in detail to include attributes, known constraints, input sources, output destinations, and known formats.

The data dictionary can serve as a central repository of information for both developers and end users. The dictionary can include business rules, processing statistics, and cross-referencing information for multiple vendor environments.

To expand the data dictionary, define, analyze, and complete data definitions using the following steps.

- Identify data needs associated with various system features.
- Match (verify) data needs with the data dictionary.
- Match the data dictionary with specific data structures.
- Create data record layouts.
- Ensure that all data can be maintained through add, change, or delete functions.

The data dictionary may be further refined in the System Design Stage to complete the information on data elements, entities, files, physical characteristics, and data conversion requirements.

Sample Attributes: The following is a sample of the type of attributes (information) that should be included for each element in a data dictionary.

Long data name (full name)
Short data name (abbreviation)

Alias
Data definition
Owner(s)
Occurrence(s)/key
Program mode
Input source(s) (e.g., screens, external interfaces, system generated)
Output destination(s) (e.g., screens, reports, external interfaces)
Values/meanings
Protection/security
Default value
Length/precision
Character set (type)
Format
Range
Surface edits
Remarks

Review Process: Schedule a structured walkthrough to verify that the data dictionary is correct and complete. The data model for a software application should be validated against any MDIT or site specific data model.

Activity: **5.8 Develop Functional Design**

Responsibility: Project Team

Description: The functional design describes how the product will be structured to satisfy the requirements identified in the Requirements Specification. It is a description of the structure, components, interfaces, and data necessary before coding can begin.

The functional design is a model or representation of the product that is used primarily for communicating design information to facilitate analysis, planning, and coding decisions. It represents a partitioning of the system into design entities and describes the important properties and relationships among those entities. Design descriptions may be produced as documents, graphic representations, formal design languages, records in a database management system, and Computer Aided Systems Engineering (CASE) tool dictionaries.

Within the functional design, the design entities can be organized and presented in any number of ways. The goal of this activity is to compile the design entities and their associated attributes in a manner that facilitates the access of design information from various viewpoints (e.g., project management, software configuration management, quality assurance, and testing). Also, the design entities and their attributes must be described in terms that are understandable to the system owner and users.

Work Product: Major work products are the Functional Design and the revised Requirements Traceability Matrix. Each requirement identified in the Requirements Specification must be traceable to one or more design entities. This traceability ensures that the product will satisfy all of the requirements and will not include inappropriate or extraneous functionality. Expand the Requirements Traceability Matrix developed in the Requirements Definition Stage to relate the functional design to the requirements.

Review Process: Conduct a structured walkthrough of the Functional Design and the Requirements Traceability Matrix.

Tasks: The following tasks are involved in developing the functional design.

5.8.1 Develop Functional Design Document
5.8.2 Conduct Functional Design Review

Task: **5.8.1 Develop Functional Design Document**

Description: The Functional Design Document defines the functions of the system in user terminology and provides a firm foundation for the development of the system design. The Functional Design Document should be written from the system owner/users' perspective. This document provides the owner/users with an opportunity to review and provide input to the product design before system design work is completed.

Work Product: Prepare a draft Functional Design Document. Use the designs developed for inputs, outputs, user and system interfaces, and security controls as input to this document. Submit the draft document to the system owner and users for their review and approval. After making the changes needed to resolve problems found during the review, the approved Functional Design Document becomes an official agreement and authorization to use the functional design as the basis for developing the system design. Place a copy of the approved Functional Design Document in the Project File.

Review Process: Conduct structured walkthroughs as needed to assure that the Functional Design Document is accurate, complete, and describes the functional design in a manner that can be understood by the system owner and users.

Resource: A template of the Functional Design Document is available on the MDIT SUITE website.

Task: **5.8.2 Conduct Functional Design Review**

Description: The Functional Design Review is a formal technical review of the basic design approach. The primary goal of the Functional Design Review is to demonstrate the ability of the information system design to satisfy the project requirements. The review should be a series of presentations by the project team to the system owner, users, functional area points-of-contact, and quality assurance representative. Vendors may be invited to participate in the Functional Design Review when an off-the-shelf software product or hardware item is being considered for the system architecture.

Conduct the Functional Design Review to perform the following verifications.

- Evaluate the progress, technical adequacy, and risk mitigation of the selected design approach. Determine whether the approved design approach is being followed by the project team.
- Evaluate the progress, technical adequacy, and risk mitigation of the selected test approach. Review the following items:
    - System Test Requirements from the Requirements Specification document
    - Organization and responsibilities of group conducting tests
    - Planned format, content, and distribution of test reports
    - Planned resolution of problems and errors identified during testing
    - Retest procedures
    - Change control and configuration management of test items
    - Special test tools not required as deliverables
- Evaluate the techniques to be used to meet quality assurance requirements.
- Establish the existence and compatibility of the physical and functional interfaces.
- Determine whether the functional design embodies all of the product requirements.
- Verify that the design represents a system that can meet the functional, data, and interface requirements.
- Review the planned user interfaces to the system. Examples of the types of design information to review:
    - Operating modes for each display station. For each mode, the functions performed, and the displays and controls used.
    - The format and content standards for each screen (e.g., data locations, spaces, abbreviations, the number of digits, all special symbols, alert mechanisms).
    - Control and data entry devices and formats (e.g., keyboards, special function keys, and cursor control).
    - The format of all data inputs and provisions for error detection and correction.
    - The format for all status and error messages and data printouts

(e.g., formats, headings, data units, abbreviations, spacing, and columns).

- Demonstrate any rapid design prototypes used to make design decisions.
- Identify potential high risk areas in the design and any requirements changes that could reduce risk.
- Review to assure that consideration has been given to optimizing the maintainability and maintenance aspects of the product.

Review Items: The following items should be considered for review and evaluation during the Functional Design Review. Be prepared to discuss in technical detail any of these items within the scope of the review.

- Functional flows. Indicate how the system functional flows map the software and interface requirements to the individual high-level components of the product.
- Storage allocation data. Describe the manner in which available storage is allocated to individual components. Timing, sequencing requirements, and relevant equipment constraints used in determining the allocation should be included.
- Control functions. Describe the executive control and start/recovery features of the product.
- Component structure. Describe the high-level structure of the product, the reasons for choosing the components, the development technique that will be used within the constraints of available computer resources, and any support programs that will be required in order to develop and maintain the product and allocated data storage.
- Security. Identify the security requirements and provide a description of the techniques to be used for implementing and maintaining security within the product.
- Information systems engineering facilities. Describe the availability, adequacy, and planned utilization of the information systems engineering facilities including both Government-provided and commercially available facilities.
- Information systems engineering facility versus the operational system. Describe any unique design features that exist in the functional design in order to allow use within the information systems engineering facility that will not exist in the operational product. Provide information on the design of support programs not explicitly required for the operational system that will be generated to assist in the development of the product.
- Development tools. Describe any special tools (e.g., simulation, data reduction, or utility tools) that are not deliverables, but are planned for use during systems development.
- Test tools. Describe any special test systems, test data, data reduction tools, test computer software, or calibration and diagnostic software that are not deliverables, but are planned for use during development.
- Commercial resources. Describe commercially available computer

resources, including any optional capabilities (e.g., special features, interface units, special instructions, controls, formats). Identify any limitations of commercially available equipment (e.g., failure to meet user interface, safety, and maintainability requirements) and identify any deficiencies.

- Existing documentation. Maintain a file and have available for review any existing documentation supporting the use of commercially available computer resources.
- Support resources. Describe the resources necessary to support the product during engineering, installation, and operational state (e.g., operational and support hardware and software personnel, special skills, human factors, configuration management, testing support, documentation, and facilities/space management).
- Standards. Describe any standards or guidelines that must be followed.
- Operation and support documentation. Describe the documentation that will be produced to support the operation and maintenance of the product.

Work Product: The work product is the Functional Design Document. The review of this document will result in one of the following outcomes:

- Approval - indicates that the functional design is satisfactorily completed.
- Contingent Approval - indicates that the functional design is not considered accomplished until the satisfactory completion of identified action items.
- Disapproval - indicates that the functional design is inadequate. Another Functional Design Review is required, once specified changes to the functional design are completed.

After making changes needed to resolve problems found during the review, the now approved Functional Design Document becomes an official agreement and authorization to use the functional design as the basis for developing the system design.

Activity: **5.9 Select System Architecture**

Responsibility: Project Team

Description: Due to the complexities inherent with contracting and procurement processes and establishing hosting environments, the need exists to select the system architecture at this stage of the SEM.

When the system architecture for the product has not been predetermined by the existing IT environment of the system owner and users, evaluate system architecture alternatives to determine which one has the best, cost-effective solution that satisfies the project requirements. By conducting the MDIT Enterprise Architecture (EA) Solution Assessment process, the final system architecture will be determined.

"Cost effective solution" does not imply the least expensive alternative. The best, cost effective solution is the alternative that does the best job of satisfying the project requirements, assures the highest quality product, and provides for an adequate return on investment in a timeframe that is acceptable to the system owner.

Select the specific hardware, software, database management system, and communication facilities based on the following types of considerations.

- MDIT Enterprise Architecture or site-specific information architecture guidelines or standards
- Hardware and software that emphasizes simplicity, flexibility, ease of operation and maintenance
- Cost to procure and maintain potential environment
- Backup and recovery procedures
- Selection of a distributed or centralized processing environment
- Communication requirements
- Data configuration

Obtain support from functional area points-of-contact to aid in the architecture evaluation process. Consultations and input may be helpful from system and database administrators, local area network administrators, operations personnel, system developers, and communication experts.

Tasks: The following tasks are involved in selecting a system architecture.

5.9.1 Evaluate System Architecture Alternatives
5.9.2 Recommend System Architecture

Task: **5.9.1 Evaluate System Architecture Alternatives**

Description: Consider system architecture alternatives within the site's enterprise architecture guidelines that enable the project objectives and requirements to be achieved. The selection of a system architecture depends on many factors such as the experience of the project team with each alternative and the availability of reusable components to facilitate the implementation of an alternative. MDIT Enterprise Architecture (EA) should be consulted during the selection process.

When investigating alternatives with EA, consider the following issues.

- Those functions or portions of functions that are to be automated and the functions that will be manual. Conduct an examination of *what* the automated portion of the project will encompass.
- The technical solution for the objectives. The determinations of *how* the product is to be designed; (e.g., online vs. batch, client-server vs. mainframe, Oracle vs. SQL Server).
- The system owner's and users' IT environment and the needs created by the technical solution. Consider any hardware and software that must be acquired, including system access software, operating system software, database management system, and communications facilities.

The following procedure provides one approach for evaluating the architecture alternatives.

- Conduct a Business Case Analysis to determine the most cost-effective alternative. On the benefits side, include the improvements over the current process being used to support the business application. On the costs side, include any degradation from current capabilities along with the rationale for allowing the degradation.
- Create and evaluate a data flow diagram for each alternative.
- Identify how users would interact with the features associated with each alternative (such as the generation of queries and reports).
- Create a list of the risks associated with each alternative and develop a plan for mitigating each risk.
- Compare the performance capabilities of each alternative. How fast will each alternative be able to process the user's work given a particular hardware resource. Performance is usually expressed in terms of throughput, run time, or response time. Five factors that frequently affect performance include:
    - Number of intermediate files in a system (parked data between programs)
    - Number of times a given file is passed
    - Number of seeks against a disk file
    - Time spent in calling programs and other system overhead

    - Time taken to execute actual program
- Compare the security and access control features of each alternative. To what extent does the alternative provide security against human errors, machine malfunction, or deliberate mischief. Some common controls include:
    - Check digits on predetermined numbers
    - Batch control totals
    - Creation of journals and audit trails
    - Limited access to files
- Compare the ease with which each alternative allows the system to be modified to meet changing requirements, such as:
    - Fixing errors
    - Changing user needs
    - Mandatory/statutory modifications
    - Enhancements

Work Product: Maintain records on each alternative that is evaluated. Use this information to develop a summary of the system architecture alternatives. The summary will be integrated into the materials presented to the system owner when a system architecture recommendation is made. Place a copy of the records for each alternative and the summary in the Project File.

If a Business Case Analysis is conducted, prepare a report that describes the process used for the analysis, a summary of the alternatives considered and the results obtained, and place a copy in the Project File. The report will be integrated into the materials presented to the system owner when a system architecture recommendation is made.

Review Process: Conduct structured walkthroughs on records of each alternative that is evaluated.

Task: **5.9.2 Recommend System Architecture**

Description: Based on the results of the architecture alternatives evaluation, develop a recommendation for a system architecture that is cost-effective and will facilitate the achievement of the project requirements. This recommendation is contained in the MDIT Hosting Solution document. Prepare a presentation for the system owner and users that provides the following types of information to support the recommendation. MDIT EA and Technical and Data Center Services should be consulted during the selection process.

- Review the limitations or problems with any current manual or IT system that will be resolved by the product.
- Present the logical model for the product. Highlight new functions that would be incorporated.
- For each architecture alternative that was evaluated, present the following information.
  - A description of the alternative.
  - An overall data flow diagram showing how the alternative would be implemented.
  - The way the system would look to the users in terms of hardware, user interface, reports, and query facilities.
  - The estimated benefits of the alternative.
  - The estimated cost and time to implement the alternative.
  - A statement of the element of risk associated with the alternative.
- Present the recommended alternative and explain why it was selected.

Formal acceptance of the project team's recommendation by the system owner is required before the project can move forward. Any delay in making this decision could result in a slippage of the project schedule.

Work Product: Document the project team's recommendation for the most cost-effective and viable architecture alternative. Provide a summary of each alternative that was evaluated. Describe the rationale for proposing the recommended architecture. Describe the impact of this alternative on the system owner and users organization(s) and other systems. Include any background information that was relevant to the decision process. Provide access and/or copies of DIT-015B and other related templates to appropriate users/DIT staff for preparation.

Review Process: Conduct a structured walkthrough to assure that the most cost-effective and viable architecture alternative is being recommended.

Approval: Present the project team's recommendation for the system architecture to the system owner and users. The recommendation should be delivered as both a document and as a presentation. Place a copy of the MDIT Hosting Solution Document in the Project File.