Preface

The increasingly complex nature of information systems, with corresponding increases in resource expenditures, requires the establishment of guidelines. Guidelines ensure that such systems are developed, acquired, evaluated and operated in an efficient manner, within prescribed budget and schedule constraints, and responsive to mission requirements. Accordingly, the procedures in this Information System Life Cycle (ISLC) Manual have been revised to ensure that adequate management and control mechanisms are established during the life of information systems projects. This manual applies to all information systems developed, maintained, enhanced, and disposed in the Department of the Treasury. The manual is not intended to be applied retroactively to systems already under development.

The manual is designed to be tailored for each project so that an appropriate set of products is produced. While each phase of the ISLC applies to every project, the composition of products and activities within a phase will vary depending on the size and complexity of the project and the system's intended users.

Information systems managers generally agree that systems development projects often are not completed on time, do not meet user requirements, and are not completed within budget. Most failures are the result of not understanding that developing systems require a consistent management approach and user involvement for structuring and controlling the process. This document provides such an approach and is developed to assist information systems and more directly, project managers, in producing quality systems from the onset.

Information systems managers are responsible for managing the development and operation of information systems that improve service delivery to the public, reduce the burden on the public and minimize the cost of Federal program administration. In addition, information systems managers must ensure that information systems comply with regulations concerning accessibility for individuals with disabilities.

Information systems managers must be change agents. They must be able to link systems development projects to the bureau's strategic plan and core business processes. Information systems managers are charged with having a direction for change and exploit windows of opportunity to reinforce change. They ensure that system performance reviews are integrated with planning, that the systems delivered provide value, and that the system outcomes meet internal and external customer needs.

______________________________________ ____________________Deputy Assistant
Secretary for Information Systems Date

Acknowledgement

The Office of Deputy Assistant Secretary for Information Systems, Office of Information Resources Management, would like to thank George Williams, Internal Revenue Service, for his valuable contribution to this extensive revision of the Information System Life Cycle Manual.

INFORMATION SYSTEM LIFE CYCLE MANUAL

TABLE OF CONTENTS

INFORMATION SYSTEM LIFE CYCLE MANUAL

TABLE OF CONTENTS (cont'd)

INFORMATION SYSTEM LIFE CYCLE MANUAL

TABLE OF CONTENTS (cont'd)

INFORMATION SYSTEM LIFE CYCLE MANUAL

TABLE OF CONTENTS (cont'd)

APPENDIX G. REFERENCES

INFORMATION SYSTEM LIFE CYCLE MANUAL

TABLE OF CONTENTS (cont'd)

List of Exhibits

INFORMATION SYSTEM LIFE CYCLE MANUAL

TABLE OF CONTENTS (cont'd)

List of Exhibits

EXECUTIVE SUMMARY

A. INTRODUCTION TO THE INFORMATION SYSTEM LIFE CYCLE

The Information System Life Cycle (ISLC) Manual was developed to address information systems which vary greatly in size, scope of application, complexity of processing, technologies used, and the methodologies and tools used to support the evolution of systems from identification of a requirement through the operation and ultimate disposition of the system. Such variation reflects the diversity of Treasury programs. This manual presents a structured, disciplined approach to solving an information management requirement and for selecting and using the methods, tools, and techniques appropriate for each requirement. The procedures should be modified appropriately to compensate for differences in projects.

Sound life cycle management practices include planning and evaluation in each phase of the information system life cycle. The appropriate level of planning and evaluation is commensurate with the cost of the system as well as with the stability and maturity of the technology under consideration, how well defined the users' requirements are, the level of stability of program and user requirements, and security considerations. Establishing a close relationship with the user organization early in the life cycle is key to successful life cycle management.

This Executive Summary provides an overview of the Department's newly adopted ISLC methodology and briefly describes the overall structure of the life cycle. This Summary also highlights certain topics that are crosscutting in nature.

B. KEY DECISIONS

The key decisions represent significant issues to be addressed by the project team during each life cycle activity. Some decisions will require program management approval. There are three types of decisions: project approach, project execution, and project continuation.

Project Approach Decisions. Project approach decisions address the organization of the project, project scope, methods and tools to be used, and the selection of participants for project activities such as system acceptance testing, reviews, and approvals.

Project Execution Decisions. Project execution decisions address the scope and specific features of the system. These decisions address programmatic, technical, and system support related issues.

Project Continuation Decisions. Project continuation decisions address issues relating to the continued need for the system, the availability of funding, cost versus benefits of the approach, and other needed resources.

Some decisions may be very simple, while others may require a great deal of effort; they must all be addressed explicitly to ensure that they are made in a well-reasoned manner, rather than being overlooked or made by default. How well all three types of decisions are made, and the timeliness of decisions, are crucial to the system's ultimate success in meeting the information management requirements.

C. CROSSCUTTING ACTIVITIES

There are a number of activities which crosscut the phases of the system life cycle. These activities are addressed and summarized below.

Project Plan. The project plan is the crucial document of the information system life cycle. It is first produced in the planning phase and is updated, expanded, and refined continually throughout the life cycle. The project plan covers project scheduling, staffing, resources, adjustments to the life cycle structure, selection of tools and methodologies, identification of applicable reviews and approvals, configuration management methods and other related topics. The project plan is subject to approval by bureau and/or Department program management.

Project Reviews/Quality Assurance. Project reviews and quality assurance (QA) activities are conducted throughout the system life cycle to ensure that the system ultimately established is sound programmatically, technically, economically, and from a system management perspective. Reviews and QA activities help to ensure that key issues are identified and addressed appropriately as early as possible in the life cycle to avoid major, expensive rework during later phases of the project. Reviews and QA activities provide feedback to the project team and also advise program management as a prelude to required system approvals. Specific organizations and individuals who are to participate in project reviews and QA activities are designated early in the life cycle and are selected to reflect the nature of the information management requirement based on their functional, technical, and/or operational expertise.

Project Approvals. Formal approvals occur at designated points in the life cycle to ensure program management support of the project and the resulting system. Conducting reviews and obtaining approvals is not the goal of the life cycle process, but is a key means to the desired end. Program management approvals are obtained in all phases of the life cycle. The specific selection of organizations and individuals to provide approvals for project activity is tailored to meet the specific characteristics of the information management requirement and the proposed solution.

Configuration Management. Continual, consistent documentation of the development and evolution of the system is essential to ensure that at all points in the system life cycle, key analyses and decisions are recorded, that the system is accurately described, and that there is consensus on what is required and what is delivered. Configuration management serves to maintain a controlled library of life cycle products (automated products such as software as well as system and user documentation) and to provide a process to record consideration and disposition of requested modifications to the system. (Documentation will also be used for system maintenance, impact of change analysis, management review and control, system conversion, and audits.)

Data Management. To ensure effective management of Treasury's data, all systems are created and maintained in accordance with the Department's data management policies and practices. Life cycle activities are carried out consistent with the existing and planned data management environment, and data management concerns are addressed during all activities of the life cycle.

Methodologies and Tools. To the extent practical, all systems created and maintained by the Department must utilize state-of-the art development methodologies (or models) and modern systems development (and maintenance) tools. Methodologies and tools must be identified early in the life cycle to ensure consistency across life cycle activities. However, methodologies and tools cannot replace the application of systems life cycle management. For example, prototyping may be used as a method for performing design or development tasks; but the prototype does not substitute for documentation and reviews prescribed by the life cycle guidance. Use of expert system development tools may require that some activities be performed in a different sequence, but again, such tools merely support and do not replace the need for documentation and reviews as prescribed by the life cycle management model.

Cost-Benefit Analysis. Cost-benefit analysis is a vital management tool for linking function and budget. The analysis is used to support investments in information technology. A preliminary assessment of costs and benefits is made during the strategic planning process and is refined and updated as appropriate throughout the remainder of the life cycle. At each phase information is gathered and decisions are made that enable the project team to make increasingly accurate projections of the total costs and benefits of the system over its projected life. Subsequently, the cost-benefit analysis is updated at the end of each phase. The cost-benefit analysis information is used to determine both the establishment and the continuation of a project.

D. KEY PRINCIPLES

This manual refines traditional information system life cycle management approaches to reflect the following principles which the Department views as the foundation for life cycle management.

Life Cycle Management Should Be Used to Ensure a Structured Approach to Information Systems Development and Operation. This ISLC manual describes an overall structured approach to information management. Primary emphasis is placed on the information and systems decisions to be made and the proper timing of decisions. The manual provides a flexible framework for approaching a variety of systems projects. The framework enables information systems projects to combine activities, processes, and products as appropriate, and to select the tools and methodologies best suited to the unique needs of each project.

Each System Project Must Have an Accountable Sponsoring Organization. To help ensure effective planning, management, and commitment to information systems, each project must have a clearly identified sponsor (champion). The sponsor serves in a leadership role, providing guidance to the project team and securing from senior management the required reviews and approvals at specific points in the life cycle.

A Single Project Manager Must be Appointed for Each System Project. The project manager has responsibility for the success of the project and works through a project team and other supporting organization structures, (e.g., working groups, user groups) to accomplish the objectives of the project. Regardless of his/her organizational affiliation, the project manager is accountable and is responsible for ensuring that project activities and decisions consider the needs of all organizations which will be impacted by the system. The project manager develops the project charter to define and clarify lines of authority and responsibilities with the steering committee and project sponsor.

A Comprehensive Project Plan is Required for Each System Project. The project plan is a pivotal element in the successful solution of an information management requirement. The project plan must describe how each life cycle phase will be accomplished to suit the specific characteristics of the project. The plan is used to provide direction to the many activities of the life cycle and must be refined and expanded throughout the life cycle.

Specific Individuals Must be Assigned to Perform Key Roles Throughout the Life Cycle. Certain roles are considered vital to a successful system project and at least one individual must be designated as responsible for each key role. Assignments may be made on a full or part time basis as appropriate. Key roles include program/functional management, quality assurance, security, telecommunications management, data administration, data base administration, and configuration management. For most projects, more than one individual should represent the actual or potential users of the system (i.e., program staff) and should be designated by the program manager of the sponsoring program and organization.

Obtaining the Participation of Skilled Individuals is Vital to the Success of the System Project. The skills of the individuals participating in a system project are the single most significant factor to ensuring the success of the project. The ISLC manual is not intended as a substitute for information management skills or experience. While many of the skills required for a system project are discussed in later chapters, the required skill combination will vary according to the project. All individuals participating in a system development project are encouraged to obtain assistance from experienced information management professionals.

Complete and Accurate Documentation of Activity Results and Decisions for Each Phase of the Life Cycle is Essential. Effective communications and coordination of activities throughout the life cycle depends on the complete and accurate documentation of decisions and the events leading up to decisions. Undocumented or poorly documented events and decisions can cause significant confusion or wasted efforts and can intensify the impact of turnover of project management staff. Activities should not be considered complete, nor decisions made, until there is tangible documentation of the activity or decision.

Data Management Must be Emphasized Throughout the Life Cycle. The Department considers the data processed by systems to be an extremely valuable resource. Accurate data is critical to support organizational missions. The large volumes of data handled by Department systems as well as the increasing trend toward sharing data across systems and programs, underscores the importance of data quality. Systems life cycle activities stress the need for clear definition of data and the design and implementation of automated and manual processes to ensure effective data management.

Each System Project Must Undergo Formal Reviews and Approvals. To ensure that systems effectively address the targeted information requirement, each information systems project is subject to formal review and approval. The reviews must be conducted by skilled professionals, examining tangible products from a programmatic, technical, economic, and project management perspective. Reviews assist the project team as well as those who provide the required project approvals. Approvals are provided by the suitable level of program management, and certify its continued commitment of the project scope, direction, and resource requirements in view of known risks and/or uncertainties.

Consultation with Oversight Organizations Aids the Success of a System Project. A number of Departmental oversight bodies as well as organizations external to the Department have responsibility for ensuring that information systems activities are performed in accordance with federal guidance and standards and use available resources effectively. Each project team should work with these organizations, as appropriate, and encourage their participation and support in the life cycle as early as possible to identify and resolve potential issues or sensitivities to avoid major disruption of the project.

A System Project May Not Proceed Until Resource Availability is Assured. Beginning with the approval of the project, the continuation of a system project is contingent on a clear commitment from the sponsoring program management. This commitment is embodied in the assurance that the necessary resources will be available, not for the next activity only, but as required for the remainder of the life cycle.

E. INFORMATION SYSTEM LIFE CYCLE MODEL

The full life cycle model is divided into seven phases; some projects may require merging some of the seven phases; and not every project will require that the phases be sequentially executed. However, the phases are interdependent. Depending upon the size and complexity of the project, phases may be combined or may overlap. Decisions reached and deliverables completed in the phases are required in subsequent phases to complete successful project development. The life cycle phases are described in the paragraphs below and displayed in Exhibit 1. A small-scale development effort is described in Appendix A for projects not requiring significant resources.

Project Concept Development. Project concept development actually starts the life cycle when a need to develop or significantly change a system is identified. Once a system requirement is identified and documented, it must be reviewed for feasibility and appropriateness. The requirement may involve development of a new system or modification of an existing system. The necessary approvals and funding is needed prior to moving to the planning phase. Some bureaus consider project concept development a distinct phase which may be beneficial to their specific needs.

Planning Phase. The planning phase begins after the project has been approved by an executive level steering committee and resources have been allocated to the project. The project plan identifies the approach to satisfying the requirement and includes the discussion of methods, tools, tasks, resource requirements and project schedules.

Requirements Definition Phase. Functional user requirements are formally defined and delineate the requirements in terms of data, system performance, security, and maintainability requirements for the system. All requirements are defined to a level of detail sufficient for systems design to proceed.

Design Phase. The external physical characteristics of the system are designed during this phase. The operating environment is established, major subsystems and their inputs and outputs are defined, and processes are allocated to resources. Everything requiring user input or approval must be documented and reviewed by the user. The internal physical characteristics of the system are specified and a detailed design is prepared. Subsystems defined during the external design are used to create a detailed structure of the system. Each subsystem is partitioned into one or more design units or modules. Detailed logic specifications are prepared for each module.

Development Phase. The detailed specifications produced during internal design are translated to executable software. Software is unit tested, integrated, and retested in a systematic manner. Hardware is assembled and tested.

Test Phase. The user, with the quality assurance organization, validates that the functional requirements as defined in the functional requirements document are satisfied by the developed or modified system.

Implementation Phase. The system or system modifications are installed and made operational in a production environment. The phase is initiated after the system has been tested and accepted by the user. The phase continues until the system is operating in production in accordance with the defined user requirements.

Operations, Maintenance and Disposition Phase. The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements, and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to an organization's needs. When modifications are necessary, the system reenters the planning phase.

The disposition activities ensure the orderly termination of the system and preserve the vital information about the system so that some or all of the information may be reactivated in the future if necessary. Particular emphasis is given to proper preservation of the data processed by the system, so that the data is effectively migrated to another system or archived in accordance with applicable records management regulations and policies, for potential future access.

CHAPTER 1. INTRODUCTION AND CONCEPTS

A. BACKGROUND

This chapter provides background information and an overview of specific life cycle concepts that are critical in the life cycle management process. The concepts exist throughout the life cycle for every system and are addressed in multiple phases. The concepts serve as the foundation for the life cycle management approach and are discussed in greater detail in the remaining chapters of this guidance document.

The Department of the Treasury spends billions of dollars each year on the acquisition, design, development, implementation and maintenance of information systems vital to mission programs and administrative functions. The need for safe, secure, and reliable system solutions is heightened by the increasing dependence on computer systems and technology to: provide services and develop products; administer daily activities; and perform short and long term management functions. In addition, to ensure privacy and security when developing information systems, establish uniform privacy and protection practices and develop acceptable implementation strategies for these practices.

Management of the resources essential to effective system information is becoming increasingly complex since the rapid advances in information technology in recent years have dramatically increased computer capacity, the range of technologies available, and user accessibility. The elements needed to ensure effective systems are shown in Exhibit 1-1. There has been a proliferation of systems development efforts by a broad range of users with varying levels of sophistication in making development decisions and conducting development efforts. The result has been twofold:

- o A wide range of hardware/software options for implementation of any specific system concept or design and the consequent difficulty of finding resources that can evaluate and apply the diversity of today's technology; and

- o An increasing number of systems development efforts by Treasury employees at many organizational levels, who because of access to their own equipment and other resources, develop their own systems independently of system staffs.

Exhibit 1-1: Elements of an Effective System

The information system life cycle process describes a broad and diverse set of activities for addressing information systems requirements. The activities of the life cycle range from defining the requirements and selecting the solution, to developing, operating, and maintaining the solution. This manual is structured to provide a fair degree of flexibility in addressing those information systems requirements. New project managers, in particular, should consult with their information resources management (IRM) program officials or Senior IRM Officials for guidance in modifying the life cycle to the needs of specific projects.

B. INFORMATION SYSTEM LIFE CYCLE (ISLC) BENEFITS

Information system life cycle management represents the accumulation of many years of experience by information systems professionals, and many lessons learned. This manual was developed by building on those experiences, taking advantage of other organizations' experience and resulting guidance as well as experiences specific to Treasury. The manual is intended to help ensure that each information system project is successful. Specific benefits include:

- Ensuring full consideration of the Treasury program environment, and associated system and data requirements, from project initiation through the entire life of the system;

- Providing early identification of technical and management issues to avoid investments of resources in impractical or infeasible system features;

- Providing an early view of resource needs (including resources needed for continued operations) to ensure that decisions regarding system capabilities consider the full cost of these capabilities, and refining this view throughout the life cycle;

- Fostering realistic expectations by the user community (program managers and their staff) of what the system will and will not accomplish;

- Providing a balanced consideration of the programmatic, technical, management, and cost aspects of proposed system modifications;

- Encouraging periodic evaluations to identify systems which may no longer efficiently and effectively support program needs or consume disproportionate resources;

- Providing clear measures of progress and status to enable effective corrective action if needed; and

- Providing much of the information needed to support information resources management planning and the development of budget requests.

This ISLC methodology provides a structured approach to information systems projects. The methodology addresses the development and the ongoing operation of information systems and it describes the progression of the life cycle through various activities and processes as well as addressing the relationships among the activities and processes.

This ISLC manual has been revised to address use of a wide range of technologies, including computer-aided software engineering (CASE) and other automated tools. Systems that support the Department's programs vary greatly in size, scope, complexity, and technologies. There is wide variance in the methodologies and tools used to support the evolution of the systems from initial problem statement through the operation and ultimate termination of the system. Such variation reflects the diversity of the Department's programs. Therefore, this guidance document does not prescribe a single method applicable without change to every system. Rather, it presents a structured, disciplined approach to systems projects and to selecting and using appropriate methods, tools, and techniques.

C. PURPOSE, SCOPE, AND APPLICABILITY

The purpose of this manual is to explain the importance of life cycle management to all potential participants in the information system life cycle and describe the progression of the life cycle approaches through individual phases. This manual outlines the overall Department of the Treasury guidelines for the ISLC methodology and provides direction on its use and applicability.

This manual applies to all Treasury and contractor personnel who are developing, acquiring or managing new systems or making major enhancements to existing systems. Adherence by system users and all levels of Treasury management across all functional areas is required to implement effective information systems.

D. SIGNIFICANT CHANGES

Much has changed in the information systems area since the last life cycle guidance was published. These changes are reflected by an increasingly program-based approach to justifying and evaluating information systems. As a result, significant changes in this manual are driven by Federal Information Resources Management Regulations (FIRMR)

revision, General Accounting Office (GAO) reports on systems development, Office of Management and Budget (OMB) revised circulars and publications, Total Quality Management (TQM) initiatives, Treasury's revised guidance documents, and the recent GSA's Information
Resources Procurement and Management Review (IRPMR) of Treasury's Information Resources Management (IRM) program. The following are notable changes driving this revision.

- o Proposed investments in information systems should maximize return on investment which include improving delivery of services to the public, reducing information collection burden on the public, and increasing the efficiency of Federal program administration.

- o Proposed information systems must be linked to the achievement of agency/bureau mission.

- o Information system must consider the needs for accommodations of accessibility for individuals with disabilities to the extent that needs for such access exist.

- o The issuance of the Computer Security Act requires computer security plans for information systems processing sensitive but unclassified information.

- o OMB has changed the discount rate for Federal Information Processing (FIP) resources which is used in calculating cost-benefit analyses. The new OMB's discount rate guidelines should be used in performing cost-benefit analyses. The discount rate will be updated annually by OMB at the time of the President's budget submission to Congress.

- o The governmentwide emphasis on reinventing government, performance measures, benefits, quality improvement, business process reengineering, strategic business planning, prototyping, and information engineering has increased.

The following major changes are incorporated into this revision.

- o The revised manual provides descriptions of reviews for each life cycle phase as a means to reassess the continuation of the project when comparing the benefits to be derived to the cost.

- o The revised manual provides "issues checklists" to identify activities or concerns commonly found in an ISLC phase. The issues checklists serve as planning and validation tools. By checking the list of questions at the end of each phase, major issues are addressed as part of the discussion of the activities for each phase.

- The revised manual provides descriptions of deliverables to aid in the development of the deliverables.

- The revised manual includes an information engineering process model to present similarities and adaptability in the standard life cycle methodology.

- The revised manual eliminates the phase-stage-task structure. This ISLC guidance addresses the phase-task approach. If a project is more manageable by breaking it down, the option to include stages is appropriate. Subtasks can also be added if completing the activities can be accomplished with ease.

- The revised manual incorporates guidance for a capacity management and performance measurement program formerly in TD 81-07, "Information Systems Capacity Management and Performance Measurement."

- The acquisition guidelines are removed from this manual. Acquisition procedures are in TD P 83-01, "Guidelines for Acquisition of Federal Information Processing Resources."

E. DOCUMENTATION

The life cycle documentation is an integral part of the information system development process, not as an end in itself. The life cycle methodology specifies which documentation will be generated during each phase. Some analytic methodologies or tools provide documentation or other output corresponding to all or part of a life cycle product. In these cases, these outputs can be used to satisfy the corresponding documentation requirements. Many of the products may serve as the basis for the departmental information collection requirements, e.g., Information Systems Plans, security plan, systems of records notification, reports on excess personal property, equipment inventories, IRM reviews, and cost-benefit analysis. (See the Office of Information Resources Management for more information of specific requirements.)

The ISLC documentation is typically referred to as "deliverables." Deliverables are collected at the end of each phase and stored in the project file. Very few of these deliverables remain unchanged throughout the system life cycle. Some deliverables evolve continuously during the life cycle and others are revised to reflect the results of analyses performed in later phases. Examples of deliverables are displayed in Exhibit 1-2, Information System Life Cycle Deliverables. Deliverables are typically categorized into two major types - process and product. Depending on the size, scope, and complexity of the system development effort, other types of deliverables may be added, e.g., technical.

1. **Process documentation** is necessary to communicate status and direction. By reviewing this documentation, users are able to determine if their needs are being understood and addressed and management is able to verify if appropriate progress is being made during the development process. Examples include:

(1) All project plans;
(2) Project review reports;
(3) Schedules and budgets; and
(4) Supporting materials.

Process documentation addresses the actions required for developing, implementing, and maintaining the system, i.e., the procedures, timelines, and funds required for accomplishing results. Process documentation is not maintained, i.e., kept current subsequent to implementation; however, it should be retained for evaluation and general reference. Refer to each phase for additional process documentation.

2. **Product documentation** will survive the information system development process. That is, it will be retained and maintained in a current condition subsequent to implementation in order to communicate how the system operates to people who may not have been a part of the system development process. These people are users, reviewers, system operators, maintenance operators, and programmer analysts. Some examples of product documentation include, but is not limited to:

(1) User manual;
(2) Operators manual;
(3) Maintenance manual;
(4) Test plan; and
(5) Conversion plan.

Product documentation addresses the system itself - what it is, how it is operated, how to maintain it, etc. The documents describe, in non-technical terms, how the system processes data. These documents are maintained subsequent to installation. Modifications to the system as they occur are appropriately reflected in the documents; new versions of the documents are distributed periodically. Each phase generates a list of deliverables for documentation of that phase.

F. DATA MANAGEMENT

The Department's life cycle management approach emphasizes management of data resources. Because of the large volumes of data handled by Treasury's systems and the increasing trend toward sharing data across systems and programs, a data management approach is required when developing information systems.

The data related activities, products, and decisions which must be addressed during the system life cycle constitute the data management approach. The approach also includes the degree of rigor to be followed when performing these activities and the level of formality to be used when documenting data related life cycle products and decisions.

Implementing a data management approach is key to the project's success. If the approach does not address data dictionary issues as part of a large, high impact project, the risk of time and cost overruns for the project will be increased, as will maintenance costs for the completed system and its data.

One criterion which stands out as a major factor in determining the data management approach is the degree of data sharing. Data sharing includes use of one information system's data by a second system, and using the same data by multiple functions using a single information system. If data sharing is extensive, a rigorous and formal data management approach should be used. Following this type of approach will minimize unexpected, adverse impacts upon the system and the programs it will support.

The benefits of data management include:

o Ensuring that data collected and disseminated meet programmatic requirements fully, including requirements for accuracy and timeliness;

- o Improving management decision-making by providing better access to more accurate and timely data;

- o Increasing productivity in the informaiton collection and processing activities as the understanding and use of available data increases;

- o Ensuring that existing data can be shared to the maximum practicable extent, avoiding the cost of redundant data collection and storage; and

- o Reducing the cost of system maintenance and the time needed to modify implemented systems by designing more stable and flexible databases.

Several important aspects of data management are described below.

1. **A data management plan** is required for each system project. Like the project plan, the data management plan is an important life cycle product. It is introduced in the planning phase, and is updated, expanded, and refined continually throughout the life cycle.

2. **A data dictionary** is mandatory for every system. A data dictionary must be prepared for every system to clearly communicate the attributes of the data processed by the system to system users and other individuals with an interest in the system data.

3. **Data administration concerns cut across multiple systems.** For each project, data administration focuses on the relationship of the project to other projects and systems that process common data. Data administration addresses data definitions, data standards, mechanisms to ensure consistency of data across systems, data quality control procedures, and related issues that frequently cut across project and system boundaries.

G. COST-BENEFIT ANALYSIS

The cost-benefit analysis is a vital management tool for linking function and budget. The cost-benefit analysis must be updated as appropriate throughout the information system life cycle and the level of detail should be commensurate to the size of the investment.

When the information system provides services to the public, bureau managers should quantify the performance of the information system through systematic measurement of outputs. In conducting cost-benefit analyses to support ongoing management oversight, agencies should seek to maximize return on investment over the information system life cycle by establishing and evaluating systematic performance measures. These performance measures should include:

- the effectiveness of program delivery;

- efficiency of program administration; and

- reduction in burden, including information collection, imposed on the public.

The revised cost-benefit analysis at each phase of the information system life cycle provides up-to-date information to ensure the continued viability of an information system prior to and during implementation. Reasons for updating a cost-benefit analysis may include such factors as significant changes in projected costs and benefits, major changes in requirements (including legislative or regulatory changes), or empirical data based on performance measurement gained through prototype or pilot experience.

H. LIFE CYCLE REVIEWS

The life cycle review process consists of a series of reviews conducted in each phase (Exhibit 1-3) to ensure that each phase of the project is completed successfully. This process ensures that all products created during the life cycle meet functional and performance requirements as outlined in all requirements documentation. The requirements for holding specific reviews are determined by the system size and complexity and by management direction.

The completion of a phase represents a logical point at which reviews should occur. The purpose of reviews are:

(l) To ensure that project direction and goals remain consistent with organization strategic plan and goals;

(2) To provide an opportunity to terminate projects which fail to demonstrate an adequate return on investment;

(3) To measure the ongoing progress (i.e., budget, schedule and deliverables) and identify potential problems for corrective actions; and

(4) To approve phase results and authorize further work.

Planning Phase	Requirements Definition Phase	Design Phase	Development Phase	Test Phase	Implementation Phase	Operations, Maintenance and Disposition Phase
Planning Review	Functional Requirements Review	Software Requirements Review Preliminary Design Review Final Design Review	Test Readiness Review	Test Analysis Review	Revalidation Review	Post-Implementation Review Periodic System Review
			Unit/Module Test Subsystem Integration Test	System Qualification Test System Acceptance Test Security Test		

Exhibit 1-3: ISLC Reviews and Tests

Remember each purpose during the review process and achieve each purpose before terminating the review. The appropriate personnel for the review team is defined in the project plan. Review team members can include the system developer, project user, data administrator, database administrator, quality assurance manager, security manager, telecommunications manager, and configuration manager, as appropriate. In general, senior management involvement is greatest in the earlier phases of systems development efforts. This involvement declines in the later, more technical phases or less critical projects involving fewer resources.

The issues checklist in each phase provides useful information to assist the review team. A review checklist for each phase is also provided and may be useful when conducting reviews. These checklists may be modified as appropriate for the systems development effort.

I. TEST ACTIVITIES

The goal of testing is to confirm that both individual system modules and the entire system are executed in accordance with the functional requirements and technical specification. The following types of test activities are identified more specifically in the testing phase of the life cycle and are included in the test plan and test analysis.

1. **Unit/Module Testing** is performed in the development phase by the system development team. It consists of module/program level testing by validating the module's logic, adherence to functional requirements, and adherence to technical specifications.

2. **Subsystem Integration Testing** is conducted in the development phase by the system development team. It tests the system's integrated groupings of software units and modules, otherwise known as subsystems.

3. **System Qualification Testing** is an independent test, often overseen by an organization that did not develop the system, using a hardware/software platform that mirrors the proposed production environment and occurring in a controlled environment. This test is a comprehensive verification and validation process conducted to ensure that all capabilities and requirements of the system are exercised before the system is delivered to the user acceptance test team. System documents and training plans are tested for accuracy, validity, completeness, and useability. During this test, the software performance, response time, and ability to operate under stressed conditions will be tested.

4. **System Acceptance Testing** is conducted by the user, testing every system feature for correctness and conformance to requirements. This test may or may not be conducted using the proposed production platform. System interoperability, all documentation, system reliability, and the level to which the system meets user requirements will be evaluated. Recovery and restart procedures may be evaluated.

5. **Security Testing** is performed in the operational/production environment. Security testing evaluates compliance with security and data integrity guidelines, and it addresses security backup, recovery, and audit trails. It ensures that all security measures have been properly implemented in the operating environment and are effective to satisfy security requirements. It addresses all aspects of security to include internal controls, hardware, software and communications security controls, physical and environmental security, and administrative procedural security requirements.

J. QUALITY ASSURANCE

Quality assurance places emphasis on identifying and recording design and programming discrepancies (and errors) as early as possible, in the development process, recognizing that the later in the process an error is detected, the more costly the error is to correct. Quality assurance embodies the major elements of testing, system evaluation, and configuration management, in addition to specific mechanisms for quality assurance monitoring throughout the system development project.

The specific quality assurance mechanisms used are technical reviews or walkthroughs and formal quality assurance reviews. In the walkthroughs, a team of functional and technical experts is assembled to examine products of the life cycle for correctness and technical quality. The quality assurance review is a management level review to assure that methods and standards have been observed which comply with the Department's quality assurance goals.

Formal inspections throughout the life cycle have proven to be the largest return on investment for any quality assurance technique available with the software industry. It is a proactive approach to error detection and resolution and it has the added benefit of increasing communication and learning.

K. CONFIGURATION MANAGEMENT

Configuration management provides a set of tools and procedures to control changes to a system after life cycle activities have begun. Configuration management also controls the establishment and maintenance of production, testing, training, and development libraries.

Configuration management provides the functional and physical characteristics of hardware and software as specified in the technical documentation and achieved in the end product. The objectives of configuration management are:

- To provide a method to ensure the functional and physical characteristics of the hardware and software are identical as specified in the technical documentation and achieved in the end product;
- To identify and control changes to the functional and physical characteristics of the hardware and software;
- To provide a method for formal evaluation of the impact of proposed changes to the functional and physical characteristics of the hardware and software;
- To provide reports of the status of the functional and physical characteristics of the hardware and software;
- To establish an open path of communication for all parties affected by changes to the system; and
- To identify, establish, and control the configuration baselines.

Configuration management includes four functions:

(1) **Identification**, which is selecting and labeling all functional and physical characteristics of the hardware and software;

(2) **Control**, which is the evaluation, coordination, approval and implementation of all approved changes to the contents of an established configuration baseline, i.e., the initial functional and physical characteristics of the hardware and software;

(3) **Accounting**, which provides the administrative support required for maintaining system baselines and monitoring the status of the system throughout the life cycle as well as recording and reporting the information that is needed to effectively manage the functional and physical characteristics of the hardware and software; and

(4) **Audit**, which is a form of examining the configuration records to verify the

success of the change control and identification processes.

L. PRIVACY ACT CONSIDERATIONS

Privacy is a personal and fundamental right of all individuals. The voluntary participation of citizens in all aspects of government is influenced by the degree of confidence they have that their personal information will be protected. Privacy, therefore, is a critical component of all information systems development efforts. The Privacy Act provisions (Privacy Act of 1974 and the Computer Matching and Privacy Act of 1988) must be considered in the information system life cycle.

In order to ensure that all information systems development effort give appropriate consideration to privacy issues and requirements, it may be necessary to develop a privacy plan. Details of a privacy plan is described in the planning phase. In addition, contacting the bureau and/or Department Privacy Act representative in the planning phase will ensure compliance with the requirements of the Act throughout the life cycle.

M. SECURITY

In order to protect the bureaus' numerous and valuable system resources, effective security policies and procedures must be developed and implemented throughout the Department. The Department has security policies and procedures in place and continues to make additions and updates as the program evolves. The primary purpose of these policies and procedures is to provide a level of security commensurate with the value of the asset being protected. These security policies and procedures include but are not limited to industrial, systems, telecommunications, technical, personnel, and physical security requirements.

The project manager needs to ensure that all security requirements (which include personnel and physical security) are incorporated in the planning phase and are continued during each phase of the life cycle. The project charter, project plan, along with the work breakdown structure (WBS), should identify: how and when the security requirements will be identified, defined, and refined; the security safeguards development efforts required during the project's design phase; the testing criteria, development and implementation efforts required during the design and testing phases; and the accreditation and certification efforts required during the acceptance testing.

The Treasury Security Manual, TD P 71-10, was written and published to consolidate all existing guidance on this subject and to provide supportive details to describe how security requirements may be defined and how safeguards that satisfy those requirements may be developed and implemented. The supportive details contain policies, procedures, and references that provide authority and specific guidance to be used in the preparation of prescribed documents.

Treasury is required by governmentwide regulations to protect its automated information system resources from potential threats by identifying safeguards through vulnerability and risk analyses. The safeguards become the security requirements and considerations that are associated with the development and implementation of new/enhanced software, hardware, equipment, or other information systems technology. The development of security safeguards is an integral part of the life cycle methodology and is inherent in each phase of the life cycle. The Systems Security, Chapter VI, of the Security Manual provides the means and support for Treasury to accomplish its resource protection mission.

The Computer Security Act of 1987 was enacted to improve the security of operations and information, largely computer based, in the civilian agencies. The primary objective of the Act is controlling unauthorized use of the information in Federal computer systems, rather than merely protecting the computer systems themselves. Although computer hardware and software have real value and must be safeguarded, it is the data that is stored, manipulated, displayed, and transmitted by computer systems that represent the greatest vulnerability.

The Act specifies three major requirements for improving the security and privacy of sensitive information in Federal computer systems. To establish minimum acceptable security practices, agencies are required to:

- Determine which automated information applications are sensitive;
- Submit a security plan for their sensitive applications to the National Institute of Standards and Technology (NIST) and the National Security Agency (NSA) for review; and
- Establish security awareness programs and ensure that employees are properly trained to perform their security responsibilities.

These security requirements need to be incorporated into the system development project at the very beginning. The NIST publication on computer security also emphasizes the need to include security at the beginning of the system life cyle and decribes a number of techniques to address security in the life cycle. All Treasury systems maintaining sensitive information are required to have a Computer Security Plan to respond to the Computer Security Act.

N. PROJECT SCHEDULING TECHNIQUES

The need to establish a realistic schedule is essential to the successful management of the implementation of any system project. The schedule is based on the tasks and activities of each phase and incorporated into the project plan. Initially, the schedule is one of the more important tools used to cost the project. During the life of the project, the schedule is used as a basis for measuring progress in the sense of time.

Some Treasury bureaus have already instituted Program Evaluation Review Technique (PERT) and Critical Path Method (CPM) project scheduling methodologies. This resulted in a significant expenditure on equipment, training and resource hours. There may be cost savings involved in consolidating scheduling services within bureaus, possibly even offering, on a cost-reimbursable basis, the services to smaller bureaus.

Schedules developed during a project are important to the management of the project. Resource allocation also uses the schedule as important input. The schedule established in the project plan is the primary measuring stick for progress reporting. Clearly the most accurate possible schedule is important to the successful completion of the project. Two types of project scheduling techniques are described below.

1. **Top-down project scheduling** occurs when pre-determined boundaries or deadlines have been established and the project manager is pressured to meet the proposed schedule. This method produces a highly ambitious schedule that may or may not have a basis in reality. User demands for a system to be completed as soon as possible also encourage creation of an unrealistic schedule. Senior management may provide staunch support to the project manager in exerting enough pressure on the project team to meet an ambitious or unrealistic schedule. But in a situation less severe than a national emergency, the pressure usually cannot be maintained long enough to conclude such an ill-scheduled project or, if the pressure is maintained, it may severely damage or destroy the organization

involved. Therefore the project manager must focus the schedule on the work to be performed. The project manager must support and present the most accurate, not necessarily the quickest or cheapest schedule possible.

2. **Bottom-up project scheduling** breaks a project into its component tasks, determines the time requirements for each task and totals the components to determine an estimated completion date for the project. Since it is easier to make more accurate

estimates of smaller work units, the bottom-up approaches are generally more accurate than the top-down. The most common bottom-up approaches are described below.

a. **Program Evaluation Review Technique (PERT)** analysis illustrates the relationship between tasks and produces PERT charts (Exhibit 1-4) as its output. PERT has several distinguishing characteristics.

- o It forms the basis for all planning and predicting; provides management with the ability to plan for best possible use of resources to achieve a given goal within time and cost limitations.

- o It provides visibility and enables management to control "one-of-a-kind" programs as opposed to repetitive situations.

- o It helps management handle the uncertainties involved in programs by answering such questions as to how time delays in certain elements influence project completion, where slack exists between elements, and what elements are crucial to meet the completion date. This provides management with a means for evaluating alternatives.

- o It provides a basis for obtaining the necessary facts for decision making.

- o It utilizes a so-called time network analysis as the basic method to determine manpower, material, and capital requirements as well as providing a means for checking progress.

- o It provides the basic structure for reporting information.

Exhibit 1-4: Standard PERT Nomenclature

Exhibit 1-4 shows the standard nomenclature for PERT networks. Networks are composed of events and activities. An event is defined as the starting or ending point for a group of activities, and an activity is the work required to proceed from one event or point in time to another. The circles represent events, and arrows represent activities. The numbers in the circles signify the specific events or accomplishments. The number over the arrow specifies the time needed (hours, days, months), to go from event 6 to event 4. The events need not be numbered in any specific order. However, event 6 must take place before event 3 can be completed (or begin).

b. **Critical Path Method** (CPM) analysis permits management to determine the set of all jobs in the project which, if delayed or extended, would affect the overall project completion date. Critical path and slack times are computed. The critical path is that sequence of activities and events whose accomplishment will require the greatest expected time. Exhibit 1-5 shows a typical PERT/CPM network. The bold line in Exhibit 1-5 represents the critical path, which is established by the longest time span through the total system of events. The critical path is composed of events 1-2-3-5-6-7-8-9. The critical path is vital for successful control of the project because it tells management two things:

- Because there is no slack time in any of the events on this path, any slippage will cause a corresponding slippage in the end-date of the program unless this slippage can be recovered during any of the downstream events (on the critical path); and

- Because the events on this path are the most critical for the success of the project, management must take a hard look at these events in order to improve the total program.

Exhibit 1-5: CPM Chart

c. **Network Diagrams** graphically illustrate relationships between tasks within a project and the dependencies which may affect schedule or resource requirements. Within the life cycle, network diagrams (Exhibit 1-6) are used to model the relationship of each task in a project to every other task within that project; model the flow of a project's work from start to finish; and help to determine the shortest time necessary to complete the project. The primary purpose of network planning is to eliminate the need for crisis management by providing a pictorial representation of the total program. The following management information can be obtained from such a representation:

- o Impact of late starts;
- o Impact of early starts;
- o Cost of a crash program; and
- o Slippage in planning.

Exhibit 1-6: Network Diagram

One of the purposes of constructing the PERT chart is to determine how much time is needed to complete the project. PERT, therefore, uses time as a common denominator to analyze those elements that directly influence the success of the project, namely time, cost, and performance. The construction of the network requires two inputs. First, a selection must be made as to whether the events represent the start or the completion of an activity. Event completions are generally preferred. The next step is to define the sequence of events, as shown in Exhibit 1-7, which relates to each event to its immediate predecessor. Network diagrams for large projects can easily be converted into PERT networks once the following questions are answered:

- What job immediately precedes this job?
- What job immediately follows this job?
- What jobs can be run concurrently?

Exhibit 1-7: Sequence of Events.

d. **Bar (Gantt) Chart** is a means of displaying simple activities or events plotted against time or dollars. An activity represents the amount of work required to proceed from one point in time to another. Events are described as either the starting or ending point for either one or several activities.

Bar charts are most commonly used for exhibiting program progress or defining specific work required to accomplish an objective. Bar charts often include such items as listings of activities, activity durations, schedule dates, and progress-to-date. Exhibit 1-7 shows nine activities required to start up a production line for a new product. Each bar in the figure represents a single activity. Exhibit 1-8 is a typical bar chart that would be developed by the program office at program inception.

Exhibit 1-8: Bar (Gantt) Chart for Single Activities

Bar charts provide only a vague description of how the entire program or project reacts as a system. There are three major discrepancies in the use of a bar chart. First, bar charts do not show the interdependencies of the activities, and therefore do not represent a "network" of activities. This relationship between activities is crucial for controlling program costs. Without this relationship, bar charts have little predictive value. For example, does the long-lead procurement activity in Exhibit 1-8 require that the contract be signed before procurement can begin? Can the manufacturing plans be written without the material specifications activity being completed?

The second major discrepancy is that the bar chart cannot show the results of either an early or a late start in activities. How will a slippage of the manufacturing schedules activity in Exhibit 1-8 affect the completion date of the program? Can the manufacturing schedules activity begin two weeks later than shown and still serve as an input to the bill of materials activity? What will be the result of a crash program to complete activities in sixteen weeks after go-ahead instead of the originally planned nineteen weeks? Bar charts do not reflect true project status because elements behind schedule do not mean that the program or project is behind schedule.

The third limitation is that the bar chart does not show the uncertainty involved in performing the activity and, therefore, does not readily admit itself to sensitivity analysis. For instance, what is the shortest time that an activity might take? What is the longest time? What is the average or expected time to activity completion?

Even with these limitations, bar charts do, in fact, serve as a useful tool for program analysis.

e. **Automated Methods** are becoming easier to use and various software packages exist to assist in scheduling very large projects. An automated methodology is used when a large project is undertaken and many of the methodologies use some variation of PERT or CPM. Appendix C has a list of some automated tools. This is not a comprehensive list and the tools listed are not necessarily better than tools not listed. Application of systems analysis, design, and/or development tools will tend to reduce the duration of the activities in which they are used. Certain methodologies and tools, such as those associated with program code generators, will also tend to alter the relative duration of different activities.

Automated project management can provide answers to such questions as:

- How will the project be impacted by limited resources?
- How will the project be impacted by a change in the requirements?
- What is the cash flow for the project (and for each Work Breakdown Structure (WBS) element)?
- What is the impact of overtime and costs?
- What additional resources are needed to meet the constraints of the project?
- How will a change in the priority of a certain WBS element affect the total project?

The more sophisticated packages can provide answers to schedule and cost based upon:

- Adverse weather conditions;
- Weekend activities;
- Unleveled manpower requirements;
- Variable staff size;
- Splitting of activities; and
- Assignment of unused resources.

CHAPTER 2. STRATEGIC PLANNING FOR INFORMATION SYSTEM

A. STRATEGIC PLANNING

Strategic planning provides a framework for analyzing where the bureau is and where the bureau should be in the future. It is conducted by bureau executives for the purpose of setting the direction the bureau is pursuing. Successful strategic planning requires senior management commitment and participation and incorporates the components shown in Exhibit 2-1. Strategic planning offers new opportunities for improving the cost, timeliness, and quality of information services to bureau customers and opportunities to redefine the organization's work force. Each bureau's strategic plan is mission driven and includes a vision statement which describes the work environment to accomplish the mission. The strategic plan identifies goals, objectives and strategies in support of the bureau's mission and vision. Bureau strategic plans are linked to the overall goals and direction the Secretary has set for the Department.

Strategic planning involves change in the business practices of the bureau. These changes will depend, in large part, on the evolution of new technologies that will foster changes in work processes and techniques for management. Strategic planning also involves positioning the bureau to take advantage of new methods and technologies for improved performance currently and in the future. Successful strategic planning efforts include performance measures to monitor and evaluate progress to achieve objectives and strategies. Strategic planning requires rethinking how business is done and reengineering business processes with a focus on quality improvement. The dynamic organizational culture requires an ongoing strategic planning process, since bureau management continually faces the challenge of balancing business effectiveness and efficiency with the responsibilities of operating as a public entity.

In support of strategic planning, the bureau's executive steering committee and Senior Information Resources Management Official (SIRMO) identify strategies and initiatives to achieve the goals and objectives outlined in the strategic plan. For each initiative, they provide profiles of information resources and show how system initiatives link back to their strategic objectives. The profiles are provided in an Information Systems Plan (described later in this chapter). These profiles also provide directions for making budgeting decisions.

Exhibit 2-1: Strategic Planning Components

B. PERFORMANCE MEASURES

The Department's Office of Planning and Management Analysis (OPMA) conducted the study, Criteria for Developing Performance Measurement Systems in the Public Sector, May 1992, which describes approaches used to develop performance measures. The document is available from OPMA upon request.
The Department's efforts to develop performance indicators and measures are expected to exceed recent legislative requirements and improve the Department's ability to evaluate programs and support Treasury's mission. Treasury is seeking to develop a performance measurement system as part of a Departmentwide strategic planning process.
Performance measurement is an element essential in developing effective systems (see Exhibit 2-1). In a strategic planning process, the mission, goals, and objectives of a program are identified. Strategies are then developed to achieve the goals, and a system (including an action plan and milestones) is established to measure progress towards achieving those goals. This measurement system allows managers to determine how well they are achieving their program goals since performance is tied to strategic goals and objectives.

This manual recognizes that bureaus have begun to develop and use performance and quality measures to track improvements to major functional areas (programs). Many of the changes in measured values of performance are attributed to the implementation or significant modification of systems. These measures of system improvements to functional programs originate in the benefit analysis portion of the economic analysis performed on alternatives.

This manual also reflects, for the first time, the need to develop a **baseline** for an existing information system. The system baseline of performance is necessary to demonstrate current status as well as the actual accrual of benefits resulting from implementation of changes. The manual also reflects the need to maintain an audit trail of costs and benefits for systems.

A performance measurement system includes: program goals and objectives; a representative mix of program performance indicators and measures; and benchmarks, standards, trends, and targets, which are used to evaluate program performance. Performance indicators can be program input, output, or outcome data, whereas performance measures show the relationships among performance indicators.

The Administration and Congress are seeking improved methods for reducing the deficit and inefficiency in government. Limited budget resources put an increased emphasis on performance measurement systems. A major step towards measuring results was the passage of the Chief Financial Officer (CFO) Act of 1990, which requires agencies to provide annual audited reports that emphasize financial and program performance measures.

OMB Circular, A-94, Benefit-Cost Analysis of Federal Programs; Guidelines and Discounts requires a program-based approach to justifying and evaluating information systems. A program approach requires examining an information system, typically a project, in the context of the functional programs it supports such as tax examination, passenger or claims processing, or revenue collection. It ties the life cycle cost-benefit analysis of the information system to the functional program through the benefit analysis. It ties benefit analysis of the information system and the functional program to measurable high-level business objectives and requirements.

A program approach, as opposed to a project approach, is used to ensure that in establishing information system requirements:

- High-level objectives and requirements in established business (long-range) plans drive the analysis and assessment of feasible alternatives;
- Technical requirements are understood and identified for the project based on the program requirements; and
- Benefits and costs are identified, analyzed and measured within the confines of an information system, and not solely a computer system, a component of the information system. The information system is the basis through which a Federal Information Processing (FIP) resource derives its full-value.

In accordance with OMB Circular A-94, bureaus and Departmental management must:

- plan for periodic, results-oriented evaluations of program effectiveness. These evaluations will utilize the quantified measures developed in the economic analysis;
- place a high priority on information system projects whose benefits accrue to the general public or to other levels of government;
- understand that a request for funding approval of most information system projects is based on the bureau's willingness-to-pay: this is the maximum amount of resources a bureau is willing to forgo in order to obtain given benefits as a result of

using that FIP resource. Essentially, benefits, such as efficiency and productivity increases, that result from an information system project, may reduce the bureau's operational budget, allocation of staff years, etc. for many years; and

- use OMB's new discount rate guidelines in performing cost-benefit analysis.

C. BUSINESS PROCESS REENGINEERING

Business process reengineering (BPR) is a radical change in the way an organization conducts its business. BPR is the redesign of the organization, culture, and business processes using technology as an enabler to achieve quantum improvements in costs, time, service, and quality. Information technology is not the driver of BPR. Rather, it is the organization's desire to improve its processes and how the use of technology can enable some of the improvements. BPR may not necessarily involve the use of technology. There are circumstances when all BPR will entail is an elimination of steps or the process. For BPR to attain large benefits, then the use of information technology can be justified.

BPR will increase the demand for horizontal rather than stovepipe systems. BPR will cause the organization to shift because fewer resources will be needed to conduct business. New technologies will improve and simplify many management processes which would deliver improved services to internal and external customers.

The primary underpinning of any new system development or initiative should be business process reengineering. When BPR is applied to one or more related business processes, an organization can improve its products and services and reduce resource requirements. The results of a successful BPR program are productivity and quality improvements. BPR is not just about continuous, incremental and evolutionary productivity-enhancements, it also utilizes an approach which suggests scraping a dysfunctional process and start from scratch to obtain larger benefits.

Since BPR has been the focus, the technology infusion has caused organizations to flattened or at least reshape their managerial structure. This is largely due to the fact that dissemination of information has improved and increased through the use of networks, locally and worldwide. In essence, information is passed horizontally instead of top-down. The links among external organizations and customers, electronically, also makes it increasingly important that standard data formats and standards are compatible. Information systems of the future will increasingly use knowledge bases in conjunction with conventional databases. And finally, most of the performance measurements that are required by business components in an organization ensure that business goals are met, the business is streamlined, performance effectiveness is improved, and that BPR truly delivers the benefits that technology can put in place.

D. QUALITY IMPROVEMENT PROCESS AND THE LIFE CYCLE

Quality management principles should be integrated into each level of Treasury's planning processes. Effective planning is critical to the Department's ability to develop quality information systems. The ongoing quality improvement effort in the Department embraces understanding work processes better, making quality management a reality. The need for quality planning to deliver improved quality systems and services to our customers more cost-effectively is critical. The declining budget makes it imperative that the Department revamp the way systems are designed and improve quality in services delivered.

This manual uses an approach which integrates people, processes, and products to support continuous improvement of systems, and organization processes to support organizational goals and objectives better. By promoting user involvement, the ISLC approach fosters process understanding and provides a medium for continually improving the way individuals in the organization conduct business. By using the tools in this manual, users and functional specialists can review current processes and identify enhancements to operations. By facilitating this process improvement, the ISLC approach encourages application development projects to improve upon current procedures rather than simply automate them.

This manual will ensure that each information system project is successful and that bureaus avoid learning (and relearning) the pitfalls and lessons of information systems development the hard way. By committing to continuous quality improvement during the systems life cycle, this approach will ensure full consideration of the Department's program environment, and associated

system/data requirements, from project approval through the entire life of the system and produce quality systems.

E. INFORMATION ENGINEERING AND ITS IMPACT ON THE ISLC

Information engineering (IE) is a restructuring of a number of the basic principles and components used in previous and current systems development efforts. IE requires a detailed approach emphasizing corporate business requirements in order to develop integrated software application systems. However, many of the principles or components of the information engineering can be and are adapted in the traditional information system life cycle methodology contained in this manual.

Information engineering is a methodology that tends to help overcome some of the drawbacks of the traditional application-oriented design methodologies. Based upon the philosophy for managing the data systems of an organization, the basic premise of information engineering is that data lay at the center of modern data processing. The IE methodology views the organization as a whole, and begins by addressing the data and information processing needs of the entire organization, regardless of organizational boundaries.

The IE methodology focuses on the development of integrated systems based on user requirements and the enterprise business rules. Data and processes are crafted from the user's perspective. The direct involvement of the user community throughout the IE development process is key to the implementation of successful systems.

This emphasis on information engineering does not invalidate the functions described as part of the information system life cycle. For example, feasibility studies and requirements analyses are necessary to design an appropriate data structure and to select the proper data base management system. Also, applications must still be designed, coded, and tested. It does mean, however, that the interactions of the parties involved in a development effort are different in that the common data structure presents both opportunities and constraints.

a. Information engineering is a three-faceted approach that provides a balance between **data** and **process** requirements, supported by **technology**. This balance is intended to implement the user requirements and business rules. Processes are developed to act upon data to transform the data to achieve the desired user requirements. Projects must maintain a balance of emphasis among data, process, and technology in order to avoid the following common information system problems.

- o Focusing too heavily on **processing** requirements may result in development of stove-pipe or self-contained systems, which, by addressing only limited, function-specific needs, may store data redundantly and are often difficult to maintain or integrate. **Stove-pipe systems are systems developed to solve a specific problem and have little or no interconnection with other systems.** In addition, stove-pipe systems typically provide limited capability to support broad or complex information needs.

- o Focusing too heavily on **data** requirements may result in databases which, having been developed without regard to how data will be used, are difficult to access and typically use extensive computer resources.

- o Focusing too heavily on **technology** may result in a technologically superior system that cannot meet the functional needs of the users.

When data, processing, and technology requirements are addressed equally throughout the life cycle and across systems, the resulting systems environment more effectively addresses functional requirements while ensuring maximum benefits realized from technological resources.

b. The Department currently uses a number of information engineering principles for developing and migrating to an integrated information systems environment. These include:

(1) **Diagramming Techniques**. Diagramming and data modeling are the most effective means to document and understand complex data and processing requirements. Data and process models represent complex requirements in simple, concise formats that can be more easily maintained than textual documentation. Many of the modeling techniques are not new, but have evolved and been applied to numerous system development efforts.

(2) **User Involvement**. User involvement is essential to the identification and development of any system. Intense user involvement at the front end of systems development life cycle, with continued involvement throughout the entire life cycle is critical.

One example of an effective method of user involvement is Joint Application Development (JAD) sessions, which are interactive workshops which bring together application developers and functional specialists or users. The Department successfully used JAD sessions, and through objective facilitation and consensus building, JAD improved communications among technical and functional experts throughout the life cycle.

During the early stages, users are involved in the life cycle front-end planning and analysis of the application. Users participate in developing and validating data and process models that represent their requirements; their interactive involvement ensures that functional requirements are accurately represented.

Later in the design phase, a JAD session can be conducted to verify that the application design and user interface effectively address user needs. User involvement, also a TQM principle, interfaces with the information engineering principle.

JAD sessions also provide a mechanism for managing potential conflict or inconsistencies across functions, by bringing together experts from various functional areas. JAD is an example of a user involvement technique and is proven to be effective in ensuring that the system development activities reflect the total organization. Where traditional, serial interviews often result in inconsistent information and varying requirements, JAD sessions bring users and developers together to resolve issues and to help minimize the extent to which systems staff must interpret information.

(3) **Process Improvement**. Similar to TQM, the information engineering approach supports continuous improvement of processes and results to better support organizational goals and objectives. By promoting user involvement in process modeling, IE fosters process understanding and provides a medium for continually improving the way individuals in the organization do business.

Using diagrams as a tool to concisely represent the organization, functional specialists and managers can review current processes, and identify potential enhancements to operations. By facilitating process improvement, this approach encourages development projects to improve upon current procedures rather than simply automate them.

(4) **Automated Tools**. Automated tools can effectively support the development of information systems. A variety of these automated tools are available and are used in the bureaus. They include:

<u>Computer-aided software engineering (CASE) tools</u>. CASE involves using the computer as a development tool to build models that describe the organization and the systems environment and to document application development from planning to implementation. A CASE tool provides for on-line analysis and documentation; as a result, system documentation, in the form of models and accompanying narratives, can be dynamically

changed to reflect new or changing requirements through the life cycle.

Integrated CASE tools (I-CASE) maintain traceability between life cycle phases by storing all products in a single automated encyclopedia. In addition, CASE tools support quality control by checking for technical consistency, correctness, and balance in and between various products. Some CASE tools support later life cycle phases by developing detailed coding specifications and generating code.

Automated repositories. Automated repositories define, store, manipulate and control the description of an organization's significant information and its information resources. It serves as a centralized mechanism for storing and organizing information related to an information systems environment, including all life cycle phases of each application.

Fourth-generation languages (4GLs). 4GLs are more advanced than traditional high-level programming languages. For example, in dBASE, the command 'LIST' displays all the records in a data file. In second- and third-generation languages, instructions would have to be written to read each record, test for end of file, place each item of data on screen, and go back and repeat the operation until there are no more records to process. As a result, 4GLs are typically more powerful and can result in increased programmer productivity.

Code generators. Code generators are automated tools that translate low-level specifications for a process (pseudo code) to a high-level language program, e.g., COBOL. The program can then be compiled and executed as usual. Code generators currently are used in conjunction with or as part of CASE tools.

(5) **Prototyping**. Prototyping is a way of building a quick, rough version of a desired system or parts of that system for users to evaluate. A prototype illustrates the system to users and designers and allows for an opportunity to review requirements or design and identify potential areas for improving the system before extensive development resources have been expended. A prototype also provides an opportunity to present a sample product to the users in the short term and maintain their interest over its long-term development.

A prototype may range from a static, "throw away" prototype to a more complex, operational prototype. A static prototype simulates screen dialogue and navigation to provide for user review of the external interface and general feel of the system. As a result, a static prototype is typically not reused for system development, but can be very effectively used for early detection of problems in the emerging system. An operational prototype includes logic and core functionality, and can be later used as the first phase or pilot of the application.

These information engineering principles, coupled with this ISLC methodology tailored to the individual requirements of the organization, will help to effect quality improvements across the organization, both in the availability and quality of automated support as well as in ways of doing business.

c. Information engineering incorporates four critical components (Exhibit 2-2) which are coordinated within effective project management and they are described below.

(1) **Methodologies**. IE provides a basis for successful and effective information systems development and integration. The principles of IE must then be applied within the organization's methodology, which is tailored to the unique needs of that organization.

(2) **Techniques**. The IE approach requires the rigorous and structured application of standard techniques throughout the life to ensure effective development of information systems. These techniques may include modeling techniques as well as entity relationship modeling or data flow diagramming, as well as life cycle techniques such as JAD sessions or prototyping.

Exhibit 2-2: Information Engineering Components

(3) **Tools**. The IE approach is effective when appropriate tools are employed to support the methodologies, techniques, and project management. The use of tools help to ensure more productive and efficient systems development. The tools may be automated, such as CASE and 4GLs, or they may be manual, such as presentation tools to support JAD sessions or strategic planning.

(4) **Training**. Since IE introduces some fundamental changes to the overall approach to systems development and the specific life cycle methodology, a rigorous and concerted training and development program should be implemented with IE. The training program, in combination with the inherent benefits of IE principles, techniques and tools, can help to ensure the successful introduction and incorporation of IE into the organization. Training facilitates the smooth introduction and implementation of changes to the operating environment.

Successful implementation of an IE approach within a large and complex environment is dependent on the development of a framework for identifying existing resources and defining new applications to most effectively meet user needs and take advantage of existing resources. This framework is referred to as an information architecture.

F. IRM PLANNING FOR INFORMATION SYSTEMS

IRM planning should be tightly linked to the bureau's long-range strategic business plan via measurable objectives of the programs and the systems. As such, IRM plans should be used as agendas before bureau executive boards to make strategic decisions. The information systems plans should have a set of measurable objectives as a means for measuring their achievement of specific systems in the ISP. Consequently, new initiatives and existing systems should contain performance objectives written in business terms. These objectives may be revised over time as the concept of a new system is refined. They should be based on the expected benefits indicated in the system's cost-benefit analysis.

The Paperwork Reduction Reauthorization Act (PRRA) of 1986 requires Federal agencies to develop and annually revise a 5-year plan for meeting the agencies information technology needs. The legislation is implemented through OMB Circular A-130, which requires agencies to establish a multi-year strategic planning process for acquiring and operating information technology that meets program and mission needs, reflects budget constraints, and forms the basis for their budget requests. These requirements are met through the Department's Information Systems Planning and Information Technology Budgeting processes.

The Department's information systems planning process identifies information needs for the Department and formulates initiatives that apply information resources to meet those needs. The planning process promotes the use of information throughout its life cycle to maximize the usefulness of information, minimize the burden on the public, and preserve the integrity, availability, and confidentiality of information.

The Deputy Assistant Secretary for Information Systems initiates the annual planning cycle by issuing the Departmental Planning Call that provides guidance to the bureaus for preparing their Information Systems Plan (ISP). Each bureau is responsible for coordinating the development of its plan and for incorporating the respective operating components' plans in the overall bureau plan. The planning call encourages each organization to assure that program managers, budget staffs, and Office of Information Resources Management (OIRM) staff communicate and participate in the development of its overall plan. The ISP outlines the organization's strategies and directions and identifies major initiatives which include enterprise-wide planning.

The ISP describes information resources management strategies, major systems and initiatives which the bureau expects to continue or initiate in support of its mission. A major system is any information system, in development or operation, which is critical to the bureau's mission, or is highly visible, or for which the annual cost exceeds one percent of the bureau's budget, or for which the systems life cost exceeds $10 million. An initiative is a proposal for the development/acquisition of a new information system or for an enhancement to an existing system which will result in a major system or for the acquisition of new information technology for which Departmental approval is required.

The ISP establishes an enterprise-wide plan for developing systems to meet bureau information needs; to achieve goals and objectives; to assess the bureau's human resource element; and to evaluate existing systems. The ISP translates the bureau's mission needs into information systems projects. The ISP begins with a recognition of the user's problem to be solved and/or the mission need to be satisfied. During this activity, the need is validated, and a decision is made on the general course of action to be pursued. Bureaus are required to identify all major information systems in the ISP. Bureaus are responsible for developing long-range plans for development or acquisition of major information systems. These plans are integral to the analysis for planning the strategic information architecture for the bureau.

During this process, a functional manager makes a determination of information resource needs that are in direct support of the mission and identifies prospective strategies for meeting those needs. Front-end analysis conducted during this phase includes: a mission analysis to relate the proposed initiative to the bureau's mission; a feasibility study (described later) to determine how it would affect the current situation; and a preliminary cost-benefit analysis (described later) to justify the proposed system in terms of cost and benefits during its operational life.

Once the information system plans are developed, approved, and submitted to the Department, OIRM desk officers coordinate and analyze each plan with the Departmental procurement, telecommunications and budget staff. The plans provide the budget staff information required for reviewing and assessing the information technology (IT) budget initiatives.

G. BUDGET FORMULATION FOR SYSTEMS INITIATIVES

The Department is required by OMB to prepare an annual IT budget. OMB Circular A-11 (Section 43) defines the format for reporting the planned acquisition, operation and use of information technology systems. In A-11, OMB requires agencies to submit a separate report for each major automated information system that has an estimated total life cycle cost exceeding $25 million. In addition, agencies are required to submit a separate report for all financial or mixed (supports both financial and non-financial functions, where financial functions are significant) systems even if they fall below the $25 million threshold.

Circular A-11 also requires agencies to submit an agency acquisition plan that identifies and describes anticipated acquisitions of equipment and services where the cumulative cost exceeds $5 million over a five-year period.

After approved information systems initiatives are forwarded to the Department, OIRM desk officers, telecommunications staff, procurement staff, and budget desk officers analyze the initiatives. The analyses ensure IRM justification supports budget requests and that the Department's most critical IRM priorities are identified and approved in the Department's internal budget hearing process. Budgets are based on approved plans and may not receive consideration if the plan for the system is not in the ISP. This step is part of the budget formulation process.

The initiatives are revised as necessary to reflect the President's budget. This may involve removing items not approved internally or by OMB, or by changing costs associated with initiatives as a result of the internal or OMB hearings. The ISP then becomes an operating plan for budgeting information systems acquisitions and tracking the progress of information systems initiatives.

The IT budget reflects the Department's total current year expenditures, budget year obligations, and planned obligations for the development and acquisition of information technology systems and services. The integration of information systems planning and IT budgeting is fundamental to the Department's information systems planning program. The ISP plan identifies the Department's information systems goals and objectives and identifies major initiatives and strategies to achieve them. The IT budget supports funding requests to implement the initiatives in the ISP.

CHAPTER 3. PROJECT CONCEPT DEVELOPMENT

A. INFORMATION SYSTEM INITIATIVE

Project development begins with the sponsor/user identifying an opportunity or need to develop or enhance an information system. The justification of a functional requirement or an opportunity precedes initiation of the information system life cycle phases. During the justification process a functional manager provides the first critical description of the information management requirement or opportunity and secures the resources needed to further examine the requirement or opportunity and potential solutions. The Department commonly refers to this effort as an information system initiative which supports the mission, function, goals and objectives of the organization.

To begin the justification process, a functional manager typically prepares a concept document (issue paper, decision paper, etc.) which identifies and describes the information management requirement or opportunity. Depending on the sensitivity or criticality of the issue, the functional manager brings the document to the attention of program management and/or executive management. (The justification process can be performed by one or more functional organizations.)

The functional manager must clearly identify and describe the information management requirement or opportunity since it will be critical to the successful development of an appropriate solution. The way the requirement or opportunity is defined will shape the analyses and decisions of the feasibility study (if appropriate) and subsequent phases of the life cycle. There are two important points to be noted here:

- there is a strong emphasis on defining the information management requirement or opportunity and linking it to specific Treasury missions, as required by the Office of Management and Budget (OMB); and

- there is no assumption yet that the solution will necessarily be either a new system or an automated system. A modification to existing manual or automated systems may be the best approach to address the requirement of opportunity; the determination will be recommended in the feasibility study.

Program or executive management decide whether to commit resources to exploring ways to address the requirement or opportunity. Management determines whether staff and/or other resources will be devoted to defining and evaluating alternative ways to respond to the identified requirement or opportunity. At this point, the decision to proceed generally applies to only conducting a feasibility study (Exhibit 3-1) accompanied with a cost-benefit analysis (Exhibit 3-2).

B. FEASIBILITY STUDY

The purpose of a feasibility study is to describe the information management or business requirement or opportunity in clear, technology-independent terms that all affected organizations can agree on. An information management requirement or opportunity can be prompted by factors such as new legislation, changes to regulations, or the growth of a program beyond the support capability of existing systems.

The feasibility study provides an overview of a complex business requirement or opportunity and determines whether feasible solutions exist before full life cycle resources are committed. The requirement or opportunity is assessed in terms of technical, economic, and operational feasibility. The study contains decision criteria, comparisons of general solution possibilities and a proposed program (solution). The study is conducted any time a broad analysis is desired prior to commitment of development resources. Prior to conducting the study, the following key decisions should be addressed.

(1) **What is the specific requirement or opportunity, and for what organization(s)?** Provide an initial recognition of the requirement or opportunity and establish the broad objectives of the remainder of the life cycle. This decision addresses characteristics of the requirement or opportunity such as programmatic or other causes and symptoms of the requirement or opportunity, affected organizations, types of information needed, high level information processing capabilities, an initial perception of the ability of current systems and procedures to address the requirement or opportunity, and the timeframe(s) within which the requirement or opportunity must be resolved.

(2) **What new information needs are associated with the problem?** Provide a context for future life cycle decisions by determining whether a new need exists for information to support a solution. Describe the scope of the need in terms of missions and organizations affected.

(3) **How broad a scope should the solution cover?** Provide an overall context within which potential solutions to the requirement or opportunity are defined, and help ensure that solutions focus on the major priority areas. The scope is determined in terms of the organization(s) (e.g., agency offices, congressional organizations, executive branch agencies), the pertinent portions of the missions or programmatic functions of each organization, and the potential relationship of the current requirement and efforts to formulate its solution to other previously identified requirements and ongoing efforts related to them.

A cost-benefit analysis is prepared as a companion document with the feasibility study. The cost-benefit analysis is the document that provides managers with adequate costs and benefit information to analyze and evaluate alternative approaches. The document provides information for management to make decisions to initiate a proposed program or continue (or discontinue) the development, acquisition, or modification to information systems or resources.

A sample outline of a feasibility study is provided in Exhibit 3-1 and the description follows.

1. **Introduction**

   a. **Origin of Request**. Identify who and describe what precipitated this project request. Provide the objectives of the feasibility study in clear measurable terms.

   b. **Explanation of Requirement**. Describe the information management requirement in programmatic, technology-independent terms. State the specific deviations from the desired situation, the cause, and the cause of the new requirement or opportunity. Describe any new information need(s) associated with the requirement or opportunity. Identify the cause(s) and effect(s) of the requirement or opportunity. Validate the description of the requirement or opportunity with all affected organizations.

1. Introduction

a. Origin of Request

b. Explanation of Requirement

c. Organization Information

d. Glossary

2. Evaluation Criteria

3. Alternative Descriptions

a. Alternative Model

b. Description

4. Alternative Evaluation

5. Recommendation

Exhibit 3-1: Feasibility Study Outline

c. **Organization Information**. Identify the organization(s) mentioned in the Origin of Request section. Identify pertinent current procedures, information, and systems of those organizations. Provide descriptions of the appropriate procedures and systems as appropriate.

Identify all organizational units involved. List the organizational unit(s) at all levels of the bureau, and external organizations, which relate to the requirement or opportunity, and describe the pertinent mission area(s) and programmatic functions of each.

d. **Glossary**. Provide a glossary of all terms and abbreviations used in the feasibility study. If the glossary is several pages in length, place it as an appendix to the study.

2. **Evaluation Criteria**

Give criteria by which the alternatives will be evaluated. Make a distinction between characteristics that must be present in the system for it to be acceptable.

3. **Alternative Descriptions**

Provide a description for each alternative proposed to handle the defined problem. Describe the resources required, associated risk, system architecture, technology utilized, and the manual process flow for each alternative. State at least two alternatives for each feasibility study, one being the alternative of doing nothing if appropriate. Predict the anticipated benefits of each alternative and the likely effects of not taking action on the alternative. State benefits in terms of in technical, operational, and economic feasibility.

a. **Alternative Model**. Present high-level data flow diagram and logical data model, if possible, from current physical processes and data for the proposed system alternative.

b. **Description**. Give a statement of required and desirable features and a concise narrative of the effects of implementing this alternative.

4. **Alternative Evaluation**

Provide a systematic comparison of the alternatives and document potential problems resulting from the implementation of each.

5. **Recommendation**

Provide a narrative that supports the recommended alternative program. Select the most advantageous program to implement the required functional capabilities based on the functional and technical concepts that satisfy the need. The information system should not be obtained at the price of inappropriate development risk or the loss of efficiency, capability, or capacity in the supported function.

C. COST-BENEFIT ANALYSIS

The important decisions of the system life cycle, and the prescribed management approvals often hinge on two key questions: "How much will it cost?" and "Are the benefits worth the costs?" For each systems project, cost-benefit analyses are conducted early in the life cycle and are updated continually as appropriate. Several important aspects of the cost-benefit analysis are described below.

1. **Cost-benefit analysis evolves from rough estimates in the early phases to increasingly detailed and accurate projections as the life cycle progresses.** At each phase, information is gathered and decisions are made that enable the project team to make increasingly accurate projections of the costs and benefits of the system over its life cycle.

2. **Cost-benefit analysis considers costs and benefits throughout the entire life cycle**. Each analysis provides an explicit consideration of onetime costs and benefits as well as those that are realized over the life of the system. The analysis also clearly identifies the incidence of each cost and benefit denoting those organizations that are likely to realize the benefits and those that are likely to incur cost.

3. **Results of the cost-benefit analysis are documented throughout the life cycle**. These results form the basis of the most important decisions of the life cycle, particularly those regarding the scope and direction of the project. Written documentation of the results is critical and includes any significant assumptions made in conducting the analysis.

The purpose of the cost-benefit analysis is to provide managers, users, and designers with adequate cost and benefit information for analyzing and evaluating alternative solutions to a problem. This document is used by management in conjunction with the feasibility study for making decisions about initiating as well as continuing the development, procurement, or modification of software or other FIP related services. Although the feasibility study and the cost-benefit analysis are presented as separate deliverables, due to their inter-relationship, both are needed to assist management with making decisions. Follow the cost-benefit analysis outline in Exhibit 3-2 to conduct a detailed cost-benefit analysis to justify information systems development efforts.

Introduction to the cost-benefit analysis

The development and roll-out of a Federal information processing (FIP) system project can be a large, lengthy, and an expensive endeavor. As a result, when considering alternative courses of action, it is necessary to perform an cost-benefit analysis. A cost-benefit analysis is a systematic quantifiable method for evaluating feasible alternatives prior to developing or acquiring an information system. This analysis is updated at multiple phases during the system's life cycle. OMB Circular A-130 defines an information system as a "discrete set of information resources organized for the collection, processing, maintenance, transmission, and dissemination of information, in accordance with defined procedures, whether automated or manual." The feasible alternatives will:

- meet the requirements and needs of the users and their customers (usually the public or other agencies);

- comply with established technical, business, security, and architectural requirements of the bureau or the Department; and

- meet management objectives of the bureau or Department (measured as part of a quality, performance measurement process or program).

The preferred alternative will be that which offers the most advantageous alternative to the government.

Management's use of cost-benefit analyses

A cost-benefit analysis is an important management tool in:

- deciding which feasible alternative is preferred to meet **program** objectives;
- determining a baseline for measuring whether a system is meeting **performance objectives**; and
- helping plan, budget, and allocate resources for projects competing for scarce resources.

Along with the cost-benefit analysis, many factors are often weighed in management's decision process to proceed with an alternative. These include:

- documented maintenance history of the existing system;
- continued technology support for an existing system;
- ease of use of the existing system; and
- capacity of the present and future systems to meet present and future demands for service.

If the system is a replacement for or a major enhancement to an existing system, management needs to know the costs, benefits and performance levels of the existing information system (termed a baseline) in order to accurately measure changes above or below the baseline. Examples of changes may include:

- improvements to the program such as those mandated by mission change or legislation enacted;
- improvements to productivity;
- improvements to achieve a given performance level for a system or program at a minimum cost; or
- improvements to achieve maximum performance at a given cost.

Management needs to develop and maintain an explanation of: the quantifications, calculations, assumptions, costs and benefits, sources of this data, and other information necessary for future reviews. This is termed an audit trail, and is needed to permit re-analysis of system cost-benefit data throughout the system life cycle and to ensure accountability for results.

The requirements for a benefit analysis provide the basis for developing measures of performance and results for the chosen alternative. These are used to measure the impact of the project in meeting the stated program and mission objectives. These quantified objectives must be tied back to the program and mission objectives described below. These benefit measures are used in management's reviews of prototypes and pilots as well as in post-implementation reviews.

The intended audience and the total cost of the system are the primary determinants of the level of effort expended on the cost-benefit analysis. **AT A MINIMUM, ALL COST-BENEFIT ANALYSES MUST INCLUDE STEPS 1 - 8 AND 11.** In organizing and writing the published cost-benefit analysis, analysts should keep in mind the intended users of the analysis. Display key ideas, charts and analyses in a manner understandable by management.

1. **Objectives and Performance Measures of the Program**

   a. Identify program (functional) performance improvements expressed in terms of bureau mission, goals, and program objectives. Do not lead to a pre-determined conclusion about which alternative is selected. Objectives for the program are reasonable and obtainable. Their definition may require an iterative process.

   b. Identify the quantifiable measures for expected program outcomes. An example of this would be "increase revenues collected by 10%," or "increase response time by 15%."

   c. Identify bureau objectives from bureau's long-range plan which drive these program objectives and their measurable outcomes. Bureau objectives should be detailed, feasible, and attainable.

   d. Explain how the program objectives and performance measures fit into the system architecture of the bureau.

1. Objectives and Performance Measures of the Program
2. Assumptions, Constraints, and Conditions
   a. Assumptions
   b. Constraints
   c. Conditions
3. Feasible Alternatives
4. Costs for each Feasible Alternative
   a. Cost Analysis Components
   b. Cost Projections
   c. Projects for Public Benefits
   d. Negative Costs (Benefits)
5. Benefits for each Feasible Alternative
   a. Tangible Benefits
   b. Intangible Benefits
6. Comparison of Costs and Benefits for each Alternative
   a. Techniques in the Analysis
   b. Discount Rate
   c. Present Value Analysis
   d. Net Present Analysis
   e. Benefit-Cost Ratio
   f. Budgetary Effect
7. Sensitivity Analysis
8. Selection and Recommendation of the Alternative
9. Presentation of Results for Management Use
10. Implementation Process
11. References and Documentation
12. Glossary

Exhibit 3-2: Cost-Benefit Analysis Outline

e. Identify management and technical issues which underlie the choice of alternatives and need to be resolved to select the preferred alternative. Posing these issues as questions involving choices helps to clarify what questions need to be answered in conjunction with the choice of alternatives. The issues are stated in terms of possible answers to a management or technical question, solutions to a problem, or alternative strategies which will resolve the point of dispute. In the analysis report state which choice is recommended and why. Examples include:

(1) Should an agency organize its field offices to deliver products and services to customers by customer, by geographic area, or by product line?

(2) Should a given system be centralized or distributed?

(3) An example of a good objective is "Increase staff productivity required to process records by 1/3 from 6 records per hour per person to 8 records per hour per person." This is a well written objective because it focuses on the output to be expected (the benefit to be achieved in the program), rather than on the technology. Example of an unacceptable objective: "replace a mainframe with distributed processing." This is an unacceptable example because the focus is on the technology, and not on the benefits to be achieved.

2. **Assumptions, Constraints, and Conditions**

All information analyzed in the cost-benefit analysis is either factual, estimated, or assumed. This section discusses assumptions and estimates which represent operating constraints and conditions in the Treasury information system environment. In order to limit the selection of alternatives to those which are practical and feasible, it is necessary to identify assumptions, constraints and conditions which will impact the system. While noted here as the second step in the cost-benefit analysis process, the formulation of assumptions is a part of every phase of the process.

a. **Assumptions** are explicit statements used to describe the present and future environment upon which the cost-benefit analysis is based. Examples of assumptions include the estimated future workload and the estimated useful life of the system. An appropriate assumption would be "transaction volume will average 500,000 per month and is projected to grow at an annual rate of 5 percent." An inappropriate assumption would be "the system must process transactions in an accurate and timely manner." The reason this statement is inappropriate is that it is neither measurable nor sufficiently specific to affect the ranking of alternatives.

b. **Constraints** are external factors to the system which limit alternatives. Constraints may be physical, as with a limitation on the amount of building space; time-related, as with a mandatory deadline; financial, as with a budgetary ceiling; or policy-related, as with applicable public laws, regulations orders and directives.

c. **Conditions** are facts that act on and shape the analysis and are similar to constraints. A condition could be the business nature of the process and how it works. An example could be that the payments can not be made until the statutory date.

All assumptions, constraints, and conditions must be carefully chosen and identified so that responsible officials understand how the costs and benefits of the alternatives identified in the analysis were developed and evaluated.

Assumptions, constraints, and conditions identified at the beginning of the analysis should be global (affecting all alternatives or entire analysis) in nature, and not alternative specific. Assumptions, constraints and conditions which are alternative-specific should be described in appropriate locations of the analysis (e.g., in the cost analysis, benefit analysis).

(1) <u>Use of Assumptions</u>. When assumptions change significantly, the analysis of alternatives will need to be re-examined. Three rules to observe in making assumptions are:

(a) Don't confuse assumptions with facts or risks. Make assumptions only when there is a need to bridge informational gaps.

(b) Be certain the assumptions are realistic.

(c) Ask yourself if your conclusions would be valid if one of the assumptions did not hold. If the answer is yes, then eliminate the assumption because it is not relevant to the analysis.

Three assumptions that should always be included in cost-benefit analysis are:

(a) the estimated future workload, capacity, and utilization of the FIP resource;

(b) the **economic life** (that period of time over which the savings or benefits to be gained from a project may reasonably be expected to accrue) of the project; and

(c) the time period covered by the comparison.

To compare alternatives, the same time period should be used for all resources and all alternatives. The lifetimes of major investments in human resources, applications software, and systems software and hardware are apt to be different, and these differences must be accounted for in the analysis. This is accomplished by ensuring that economic lives are equalized to that of the asset with the longest lifetime.

(2) <u>Estimates</u>. An **estimate** is a calculated numerical value, usually determined through a statistical sample. It is an opinion or judgement. It should be considered more reliable than an outright guess. Keep the following in mind when calculating estimates.

(a) Most cost-benefit analyses contain estimates. The nature and calculation requirements of economic analyses will require estimates that exist on one point, rather than demonstrate estimates which cover a range of numbers. This is called a **point estimate**. The means to get to the point estimate will require a quantitative handling of the potential biases that exist when using estimates.

(b) There is a common-sense way to handle point-estimates which applies the best business or technical judgements available at the time. The results are easy to understand, explain and document.

(c) In order to quantitatively handle uncertainties associated with point-estimates for costs and benefits, developers of the cost-benefit analysis must understand the use of "**expected values**."

(d) Since there is a chance that a number may be one of several values in a range, the probability of those numbers occurring, and their respective values, are used to calculate the expected value. This is the most likely number, given biases, uncertainty, and chance. An "expected value" approach is used in figuring costs, benefits, performance and other measures.

(e) For example: Suppose the cost (or benefit) of an item is believed to be either $1, $10, or $100, and the likelihood that each cost is correct is judged to be 25%, 45% and 30%, respectively. The expected cost (cost times likelihood %) is then $0.25, $4.50, and $30. The individual expected values are added up to get the total expected value, $34.75. More or fewer than three values can be used.

Note that the sum of the chances or likelihoods totals 1.0. This means the value is fully accounted for by the three dollar estimates. The sum can never be more than one, since this represents a "certainty," all the possible outcomes.

3. **Feasible Alternatives**

Only include those alternatives which can achieve the objective - i.e. are feasible and comply with technical and business requirements. One alternative which is probably not feasible, but is often included as a "feasible" alternative is the current system, **as is**, without improvements. Although the cost of the current system is required for consideration in the budget process, in the cost-benefit analysis the current system, **as is**, usually is not a feasible alternative, as it cannot meet the objectives, or the technical and business requirements.

Normally a number of feasible alternatives do exist to meet the managerial objectives and the performance measures stated in section 1 above. **There should always be more than one feasible alternative.** Some alternatives will not be considered feasible, but should be **described** briefly, including their reasons for exclusion. Identification of feasible alternatives is iterative. After identifying feasible alternatives, if the costs outweigh their benefits, the objectives should be reexamined. It may be best to make no changes to the current method of operation.

4. **Costs for each Feasible Alternative**

The purpose of analyzing the cost of each alternative is to determine the differences in costs among the alternatives for each year in the system life cycle. Each feasible alternative should be fully costed -- this includes accounting for the spending for all resources (whether appropriated or non-appropriated) such as Full Time Equivalents, materials, GSA RENT (standard rental charges for office space, utilities, protection, warehouse etc.), equipment, logistical support, maintenance, compensating balances left in commercial banks in exchange for services, etc..

a. **Cost Analysis Components**. The development of the cost analysis must include a separate identification of **Developmental** (non-recurring) and **Operational** (recurring) costs.

A representative listing of costs typically associated with FIP projects is provided below. **It is important to note that line items included in a cost-benefit analysis should not necessarily be limited to those presented below; nor should every cost-benefit analysis include all line items.** The appropriate subset of line items to be included in an benefit analysis is an important judgement which must be made by management.

(1) **Developmental Costs (non-recurring costs)** - costs incurred only once. Acquisition and transition costs fall into this category.

System Design
  Requirements Analysis Package Development
  RFP Development
Current System Measurement (Quality Measures)
Hardware Development/Acquisition
Software Development/Acquisition
Facilities/Non-Labor Costs

Space and Materials
Repair and Alterations to Space
Furniture/Fixtures
Office Supplies
Telecommunications (Data & Voice) Services
System Security
Data Encryption
Labor and Support Costs
Data Processing Services
Systems Integrator
Computer Engineer
Conversion Costs

(2) **Operations and Maintenance Costs (recurring costs)** - costs incurred throughout project or system life.

Hardware Maintenance
Software Maintenance
Facilities/Non Labor Costs
Space and Materials
Office Supplies
Telecommunications (Data & Voice) Services
Data Transmission
System Security
Operation and maintenance of encryption equipment
Labor and Support Costs
Direct Labor
Management Supervision
Benefits
Overhead

All relevant costs must be addressed in the cost-benefit analysis. Anything that involves a cash transaction must be included. This fulfills the objective that analyses identify the **full cost** of the alternatives. (Agency Procurement Requests include only acquisition costs, while the cost-benefit analysis includes both acquisition and non-acquisition costs.)

Sunk costs are not relevant to the cost-benefit analysis because they are costs which have already been incurred or are irrevocable due to a prior commitment. Sunk costs are irrelevant to the cost-benefit analysis because they were incurred at the same level regardless of the alternative chosen. The cost-benefit analysis includes only those cash flows which a decision can affect.

For example, there is an alternative linked to an already completed research effort which incurred expenditures of $300,000. This research cost must be disregarded when estimating the cost of the alternative, as it is a sunk cost, and cannot be affected in any way by the choice among alternatives. Another example is the development of an application which will run on existing hardware. The inclusion of costs for the hardware is not allowed.

Relevant non-information systems costs must be included in the analysis. For example, if workload increases would require future increases to non-information systems staff to perform a bureau mission, goal or operating program, the additional costs must be shown as increased costs for the alternative.

b. **Cost Projections**. Cost estimates must be supported by a reasonably accurate projection of workload and capacity requirements. Specific workload data and associated capacity requirements for each year in the system life must be provided. Appendix D provides guidelines to establish an information systems capacity management and performance management program.

Forecasted changes in the general price level during the planning period (e.g. **inflation**) should not be used. All costs and benefits estimated for each year of the planning period should use the general purchasing power of the dollar at the time of the analysis. This is because inflation is automatically included in the discounting calculation later.

However, a known or expected price increase or decrease in a specific cost item should be included when the magnitude of the price change may affect the decision (for example: an increase in personnel costs projected due to a planned general Federal pay raise or an increase or decrease in the cost of computer).

In general, future inflation is highly uncertain. Analysts should avoid having to make an assumption about the general rate of inflation whenever possible. The treatment of inflation is actually incorporated into the discount rate established by OMB.

c. **<u>Projects for Public Benefits</u>**. Some projects justified solely by benefits to Public. Public investments and projects which only have external benefits (external to the federal government) and are not justified on cost-saving grounds, will include a supplementary analysis with a **25% excess burden**. A good example of such a project is one which has as its sole benefit of reducing paperwork on the public. This supplementary analysis is meant to be an extra hurdle for those projects which are justified by benefits to the public, rather than decreased bureau costs. Thus, in such analyses, costs in the form of public expenditures should be multiplied by a factor of 1.25 and net present value recomputed.

Those projects which show both cost savings and social benefits are **not required** to use excess burden. For further information, consult OMB Circular A-94.

d. **<u>Negative Costs (Benefits)</u>**. Some alternatives may reduce or avoid costs (basically, cost savings and cost avoidances) which other alternatives or the current baseline require. These **negative costs** are "benefits" a project will incur and should be expressed in dollars. They often result when comparing alternatives which have different total life-cycle costs. Examples of factors resulting in negative costs include:

- fewer staff years needed to complete collection or claims cases;
- reduced maintenance costs; and
- avoided storage or communications costs.

Include these "benefits" (**negative costs**) **in** calculations on the cost side, **and not on the benefits side.** This is because **negative costs** are already included in the alternative's total life-cycle cost. Including **negative costs** in the benefits analysis would result in double counting. **Describe the negative costs in the narrative on the benefits side.**

5. Benefits for each Feasible Alternative

Benefits are the positive effects of an alternative, or values of outcomes, expressed in both discounted and non-discounted dollars, in units or in narrative form. Benefits are usually expressed in terms of the bureau's mission, goals, objectives or operating program accomplishments.

It is important that all benefits be expressed in dollars whenever possible. When benefits are not quantified in dollars, they should be placed in unit form or described thoroughly in the written narrative.

Two major categories of benefits (**tangible and intangible improvements**) are listed below, along with some subcategories and examples. A third category, cost savings (negative costs), is identified in Section D above.

a. **Tangible Benefits** are quantifiable benefits which are expressed in dollars or units (such as hours). Tangible benefits generally include:

(1) **Interest Savings**, usually expressed in dollars, are associated with a reduction in interest payments (beyond the statutory tax free period), increased interest earnings, and the loss of interest earnings due to earlier tax refund payments. A good example of interest savings is from the quicker deposit of tax receipts.

(2) **Increased Revenue**, usually expressed in dollars, may come about as a result of a more effective and efficient means of collecting fees or debts, collecting tax revenues, collecting penalties and reducing accounts receivable by increased collections. Benefits of increased revenue may also come about as a result of decrease errors, reduced incorrectly addressed correspondence, improved account access, more investigations and examinations completed, and prosecutions of criminal completed.

(3) **Burden Reduction** is the time and cost (if computable) saved by the customer, or taxpayer as a result, for example, of simplified filing, faster refunds, improved notices (reduction in improper notices and the improvement in the clarity of notices), and improved services. For many information system projects, burden reduction is the most significant benefit. The fact that this benefit may not currently be measured in dollars does not mitigate its significance in decisions about alternatives and about projects' business cases.

b. **Intangibles Benefits** are often deemed "soft" benefits because they are more difficult to quantify. Defining and quantifying intangibles in terms of program improvement increases the value of the overall assessment of the alternatives. This is a subjective process. Uncertainties should also be identified and discussed. This may increase or decrease the benefit level. Intangibles, by their nature, also require a narrative explanation. However, reasonable attempts to quantify intangible benefits should be made. Some examples of intangible benefits include:

- better decision making by management using the system;
- less rework by staff;
- meeting requirements established by a Public Law;
- better compliance with the law; and
- fewer errors.

6. **Comparison of Costs and Benefits for each Alternative**

Well-managed organizations insist that projects flow from a sound strategic plan and should be analyzed on that basis. The essence of a plan is to explicitly consider competitors and to chart a course to build and maintain competitive advantage, highlighting assumptions and sensitivities. This is the Department's aim.

The cost-benefit analysis must include a detailed comparison of alternatives. A major goal of the cost-benefit analysis is to identify the most favorable business investment from among the various alternatives. Evaluating alternative strategies for accomplishing project objectives is the core of the cost-benefit analysis. The purpose is to identify the best choice, or the choice which is the most favorable business investment from among the alternatives. This kind of evaluation is basically an optimization problem.

The procedure described here is used when only a few alternatives are being studied and only a few dozen cost or benefit factors are involved. Bureaus may specify in detail which techniques to use and the formats for them.

a. **Techniques in the Analysis**. The process of evaluating alternatives can involve various techniques. One technique is the use of a matrix to organize information. The rows of the matrix can include, for instance, functional, technical or acquisition details, while the columns represent each alternative. Describe why particular alternatives were selected.

The key idea in the evaluation of alternatives is to find out the differences between the alternatives. The evaluation process for each feasible alternative will involve several steps each of which are described in this section:

- calculate present value life cycle costs;
- calculate present value benefits;
- determine net present value;
- determine benefit/cost ratio;
- weigh the qualitative and intangible factors; and
- determine budgetary effects.

Many different techniques have been developed to handle more complex situations. Linear programming is one example. It can determine which alternative yields the most performance for a given cost, or a given performance for the least cost. Probabilistic tools also are available to model uncertainties.

Rank alternatives by their relative value to the bureau and the Department. The purpose of this evaluation process for each alternative is to consider quantitative and qualitative factors in determining the best alternative.

Total costs and benefits must include, without exception, all significant cash flows (actual costs and benefits). For example, logistic support and the cost of the procurement itself (such as staff, review, travel, paperwork, protests, legal costs) may be significant.

When the least cost alternative is not selected, such as when another alternative provides better benefits, it is imperative that the supporting discussion clearly and specifically identify the criteria or basis for selection.

A special, but common, case is to analyze alternatives which will produce the same results; that is all the alternatives have the same benefits in terms of achieving the objective. In this case since the benefits are the same, the least costly alternative will be the best choice.

b. **Discount Rate**. In October 1992 via OMB Circular A-94, OMB changed the discount rate and modified when the discount rate should be used. **The discount rate is updated annually by OMB at the time of the President's budget submission to Congress. Updated discount rates are available from OMB, Office of Economic Policy.**

c. **Present Value Analysis**. Every cost-benefit analysis must include a **present value analysis**. This means that cost and benefit totals must be discounted so that dollars are expressed in terms of their real value at the time the cost-benefit analysis is prepared. The present value analysis is calculated by multiplying benefits and costs by a factor (discounting) based on the discount rate and time period

Cash flows which happen at very different times have different units of value even though they are nominally all in dollars. This is because "time is money." The purchasing value of money actually varies with time, as a result of factors like inflation, interest, and opportunity costs. It follows that adding together cash flows separated by time will not be sound measures unless they are changed into a common unit of measure. This is an extremely important technique to evaluate the relative worth of projects.

The simplest way to make the change is by discounting all cash flows into today's value, the so-called "present value." Businesses commonly use this conversion to get the best representation of what their financial costs and benefits really are. Discounted cash flows properly focus on cash, not budget or accounting measures. **The interest, inflation, and opportunity costs are all automatically included.**

The technique discounts future cash flows to current dollars, and recognizes the alternative return the Government could get if it had the cash now rather than in the future. Returns on system investments are based on the changing amount actually invested (or not yet recovered by accumulated benefits) at each point in time.

Alternatives which provide a mix of both Federal cost savings and external social benefits (such as burden reduction) should use the annually updated discount rate. Alternatives which provide **only "internal" benefits** which take the form of increased Federal revenue or decreased Federal costs should use a comparable-maturity Treasury interest rate as the discount rate. For example: a five-year project with internal benefits should currently be evaluated with a 3.6% discount rate.

d. **Net Present Value**. One technique used to evaluate alternative allocation of resources is the **net present value**. The **net present value** is used to determine whether or not an alternative will show a return on an investment. The **net present value** is the difference between the total present value benefits and the total present value costs.

e. **Benefit-Cost Ratio (BCR)**. The **benefit-cost ratio** is another technique used to evaluate allocation of resources. The **benefit-cost ratio** involves determining life cycle costs and benefits for each alternative and comparing the BCR ratios as a determinant of efficiency of resource allocation. The BCR is calculated by dividing the present value of benefits by the present value of costs.

Other optimization techniques may be used when deemed appropriate by local management. Consult local reference material on cost-benefit analysis for a full explanation of these and other techniques.

f. **Budgetary Effects**. The study also needs to determine their total costs and benefits and, for budgeting purposes, the budgetary effects as well. Each has different, but useful purposes. Budgets may not need to include all the costs shown in the system's total life cycle costs.

7. Sensitivity Analysis

A sensitivity analysis determines how sensitive the ranking of alternatives is to changes in the system parameters or basic assumptions. After costs and benefits are determined for each alternative and the alternatives are ranked to see which appears best, the results need to be tested to verify if they are reliable.

Knowledge about the future is uncertain, probabilistic, and imperfect. Consequently, analyses depend on estimates and assumptions which are by nature seldom correct. Cost estimates tend to be low and benefits estimates high. It is common to find out that changing estimates or assumptions reorders the ranking, putting a different alternative first. If the change is reasonable or likely to happen, switching to a different recommendation or outcome can avoid serious problems.

Cost-benefit analyses require sensitivity analyses to help deal with uncertainty by determining how easily the ranking of alternatives is changed if estimates and assumptions vary. Sensitivity analysis is used to determine maximum allowable increases and decreases to the values of certain variables needed before the ranking of alternatives changes. The values may change one at a time or in combination. Each variation represents a reasonably possible set of circumstances or environment which might happen. Spreadsheets are extremely useful for testing such "what if" scenarios.

The purpose is to learn whether the alternative which is ranked first is dominantly the best or, on the other hand, would actually not be first, if other likely values of estimates or assumptions happen. **As a result, a sensitivity analysis is always required in a cost-benefit analyses.**

As better information becomes available, adjustments in the analysis may provide management with clearer choices. **It is the analyst's responsibility to provide a narrative explanation of the variables tested and management's responsibility to review them.**

Listed below are some examples of costs or benefits which might be variables for a test:

o Cost estimates (effects of significant increases or decreases in major cost estimates for hardware, software, telecommunications, or maintenance);

o Requirements (legislative mandate, changes in functional or organizational structure);

o Implementation schedule, or systems life (shorter or longer);

o Discount rate;

o Assumptions (effects of alternative assumptions concerning requirements, operations, facilities, software, configuration of equipment or software);

o Workload (effects of variation in the estimated volume, mix or pattern of workload); and

- o Inflation (only if specifically used).

8. **Selection and Recommendation of the Alternative**

Conclusions are made after careful computation of discounted and non-discounted costs and benefits, correct usage of the benefit/cost ratio as well as other comparisons utilizing other techniques given the conditions. Further, the proper use of sensitivity analysis will help in recommending the best alternative. Consider the realities of manpower restrictions, large initial cash outlays, and budgetary constraints for the final recommendation.

9. **Presentation of Results for Management's Use**

For the final cost-benefit analysis, include sufficient narrative, tables and spreadsheets to permit management's easy reading and analysis and understanding of the alternatives and conclusions.

10. **Describe Implementation Process**

Provide a high-level implementation schedule for management consideration. The schedule is further defined after project approval.

11. **References and Documentation**

Provide a list of all references and documentation used for this deliverable. In order to perform reviews of projects during their implementation and after roll-out of the project to all sites, careful documentation of costs and benefits and their sources must be maintained. These data become the basis for reviews and validation of the methods used for resource allocation decisions.

12. **Glossary**

Include a glossary of those terms that are not commonly used outside the project office. This glossary is part of the cost-benefit analysis and ensures that these terms will be understood by the reviewers and will not be misinterpreted.

D. PROJECT MANAGEMENT DECISIONS

After the feasibility study and cost-benefit analysis are conducted and a proposed program is accepted by program and/or executive management, two events occur: (1) the initiative is identified in the information systems plan which forms the basis for a budget request and (2) a system project begins. A number of project continuation and project approach decisions are made by executive management staff of the organizations. Some of those decisions are described below.

a. **Project continuation decisions** include:

(1) **Does a significant information management requirement or opportunity exist?** The feasibility study and cost-benefit analysis should confirm that the defined information management requirement or opportunity exists and is significant enough to warrant an information systems project with life cycle management activities.

(2) **Is developing a new system potentially the best way to address the requirement or opportunity?** The feasibility study should confirm that the information management requirement or opportunity is beyond the capabilities of existing systems and that developing a new system is a promising approach.

(3) **Is the initial estimate of project scale in line with the anticipated benefits?** The cost-benefit analysis should confirm that the projected benefits of the proposed program (solution) exceed the projected resources required. Make funding, personnel, other resources available to proceed with the planning phase.

b. **Project approach decisions** include:

(1) **Who will manage the project?** Identify a lead organization for the project, and appoint an individual with the appropriate skills, experience, credibility and availability to lead the project. Clearly define authority and responsibility of the project manager.

(2) **What reviews are necessary and what organization(s) should participate in the project?** In view of the nature and scope of the proposed program, determine the key organizations and potential individuals who are to participate in the formal reviews of the project. This

decision addresses both programmatic and information management oriented participation, as well as technical interests in the project that may be known at this time.

(3) **What approvals are necessary and what oversight bodies should participate in the project?** In view of the nature and scope of the proposed program, determine the key individuals and steering committee members who will be the approval authorities for the project.

(4) **What special assistance or notification is needed?** Determine whether any particularly unusual programmatic, technical, or information management skills or experience will be needed to conduct the project. Address whether notification of any organizations not participating directly in the project may be appropriate, including external organizations (congressional organizations, OMB, General Accounting Office (GAO), and others). If the requirement or opportunity is widely shared, data administration should play a strong role.

(5) **What is the source of funding?** Determine the organization that will be responsible for providing funding for personnel, contractor support, and other resources needed to undertake the project.

Management approval to commit resources to the proposed program begins the information system life cycle phases. The sponsoring organization identifies planning and budget information for a new activity or update information for any existing activities that will be the focus of this proposed program.

E. ROLES AND RESPONSIBILITIES

a. **Steering Committee**. The concept of a steering committee stems from the need to secure executive or senior management involvement, representing both the user and information systems organizations, in directing and controlling the evolution of an organization's information systems. The steering committee is the "Board of Directors" for information systems and information systems projects. The steering committee is composed of senior level executives and is responsible for:

- o Assuring that the project is consistent with the bureau's strategic long range (mission) plan; overseeing the development of the long range information systems plans; and recommending resource levels based on those plans.

- o Reviewing, monitoring and prioritizing projects; approving all projects; and requesting reviews of these projects.

- o Authorizing project initiation, defining project scope, and selecting the project manager and delegating authority commensurate with the scope, size, and complexity of the project.

- o Requesting independent reviews of the system at any phase of the life cycle.

- o Reviewing the reports of the project manager at each phase and recommending a go/no go decision regarding the next phase of the life cycle. The committee can require the project manager to report progress periodically between phases.

The steering committee is formed to ensure the systems development efforts are accomplished in a cost-effective manner as possible, using appropriate technology, and ensures the solution meets the needs of the functional user. The committee is the focal point for overall control for ensuring compliance with the life cycle policies and documentation standards. The steering committee may choose to exercise more rigorous control on their largest, most critical projects and may wish to delegate this responsibility to an oversight group of middle managers for the smaller, less critical projects.

b. **Sponsoring Organization**. The sponsoring organization is responsible for:

- o Maintaining active senior level involvement - through the steering committee - throughout the development of the system.

- o Initiating the project request for a new information system or modification to an existing system.

- o Participating in project review activities and reviewing project deliverables.

- o Selecting a functional manager (if not the project manager) as the key participant on the project team with responsibility for defining the functional/user needs.

- o Describing functional requirements to the project manager/team and interfacing with others.

- o Ensuring that the final product meets user requirements.

c. **Project Manager**. The project manager has overall responsibility for coordinating all management and technical aspects of the system development effort. The project manager is responsible for development of the project plan, completing the project within schedule and budget, meeting customer needs, reporting to the steering committee, and developing the project charter. The project manager is also responsible for:

- o Determining the project team organization based on user and information systems organization recommendations.

- o Providing detailed work assignments, making sure written tasks exist for all work. Developing measurement criteria that defines what constitutes acceptable performance of each task.

- o Coordinating and/or performing system planning, design and implementation.

- o Scheduling and directing milestone reviews and participating in reviews conducted by independent staff or review committee.

- o Resolving problems related to all phases.

- o Overseeing preparation of test plans and test reports and managing system tests.

- o Overseeing preparation of required documentation and maintaining a project file. Meeting project standards as outlined in this manual.

Every system needs a clear charter, describing the objectives of the project and certain other key project attributes. The project manager develops the project charter prior to establishment of the project office. The charter is a written understanding between the project manager and the steering committee. The charter defines lines of authority and accountability; defines the reporting channels for accomplishing the objectives outlined in the project management plan; and sets forth the scope, activities, team organization, responsibilities, and the general methods of operation. The larger the project, the more detailed the project charter should be.

d. **Project Team**. The project manager is leader of the project team. Project team members bring technical and/or functional expertise to the project and each member plans and performs tasks involving that individual's area of expertise. Team members may not necessarily serve on the project team for the duration of the project; however, all project team members must be identified in the planning phase of the project. Once the planning is completed, members whose contributions are scheduled for later in the project may not be active participants until shortly before their involvement. Team members who are involved at the beginning of the project and have made their contribution, may return to other duties long before the project is completed. Other members may be involved in the project throughout its life cycle, although not on a full-time basis.

The systems project transcends the boundaries of many organizational units. In order to derive maximum benefit from the life cycle methodology, certain key organizational concepts should be in place. These concepts provide a means for working within the structure of the life cycle. There are several components of organization which are integral to the life cycle and are discussed in the following paragraphs. When appointing team members, the project manager must clarify the member's non-project workload responsibilities. This clarification is required both by the individual and the project manager and will affect the workload estimates rendered by the project team member during the planning process.

Project team members (and number) will vary according to projects. A select number of team members and their responsibilities are described below.

Project User. User involvement is the most critical factor in achieving a high level of acceptance and satisfaction in the life cycle process. A key measure of success for an information system is the degree of user acceptance and satisfaction with the system. The establishment of user participation is an organizational strategy for obtaining input from the user's point of view, which is critical to the success of the systems development effort. The project user is a permanent member of the

project team and interacts with the project manager in a customer-contractor relationship. As such, the project user is responsible for:

- o Monitoring and coordinating, from the user perspective, system projects affecting the user's area.
- o Providing input to all phases of the system life cycle.
- o Defining the functional requirements and participating in development sessions to define data, system, and processing environment.

- o Providing input to scope and implementation costs, screen design, application control flow, and report layouts.

- o Assisting in system testing, participating in reviews, developing user manual and training material, conducting end-user training, collecting acceptance test data and conducting post-implementation reviews.

Data Administrator. The data administrator produces the products associated with the logical data model and its supporting documentation. The data administrator assists the database administrator in transforming the logical data model to the physical database. The data administrator is responsible for:

- o Identifying the system modules for each requested application and constructing the data entity list.

- o Developing the project data model and validating/reconciling the data model with the organization data model.

- o Maintaining information and data architectures.

- o Identifying and formally defining new data elements.

- o Reviewing existing data elements to determine their appropriate use and creating the data elements dictionary input for data elements and records.

- o Participating in system demonstrations for the user organization and management.

- o Collaborating with the database administrator to convert the logical model to a preliminary data structure diagram.

- o Reviewing physical database design to ensure that it accommodates the logical design requirements.

Database Administrator. The database administrator is responsible for on-going management of the database throughout the system life cycle and for:

- o Ensuring that the software and hardware forming the database system meet the needs of the users of the database.

- o Monitoring and tuning performance of the database for optimal use and reducing unnecessary or redundant storage.
- o Designing and implementing the physical database.
- o Determining the record and set physical characteristics for the physical database.
- o Facilitating sharing of data among users.
- o Guaranteeing database security.
- o Defining control procedures for data backup and recovery.
- o Performing regular database backups and performing recovery (software and media) when necessary.

System Developer. The system developer is responsible for:

- o Managing, supervising and controlling the project's technical activities and tasks.
- o Participating in discussions regarding methodology, and tools to be used.
- o Monitoring the technical portion of the project's schedule.
- o Working closely with the project manager to ensure the project progresses according to schedule.
- o Identifying technical tasks for each phase of the project development.
- o Assigning and managing the resources to perform the technical tasks.
- o Ensuring quality control of all products produced.
- o Ensuring that all security aspects are being addressed during development.

Quality Assurance Manager. Every project manager should assure the quality of the information system output to the customer. The quality assurance function should be separated organizationally from the project development staff to avoid the potential for and appearance of a conflict of interest. As such, the quality assurance manager is responsible for:

- o Conducting reviews of the system life cycle products and providing feedback to the project team.
- o Reviewing system for conformance to system requirements and design (post-implementation review).
- o Assuring the quality of the data through the entire life cycle of the project.
- o Participating in the testing of new or modified software.
- o Participating in developing the disposition plan for the system.

Security Manager. The security manager is responsible for:

- o Participating in design reviews to ensure security measures are being appropriately addressed in the design and development of the system.
- o Reviewing security-related documentation (i.e., computer security plans, risk assessment/analysis, contingency plans, security test procedures) to ensure security requirements are being appropriately addressed.
- o Defining and specifying the security of the system.
- o Determining internal control requirements.
- o Identifying security standards applicable to the development of the system including access, encryption, and privacy.
- o Participating in test plans and system testing.
- o Ensuring system certification prior to its operational use.

Procurement Manager. The procurement manager is responsible for:

- o Providing expert advice to the project manager in planning for the acquisition of needed resources other than personnel. These resources include hardware, software, and contractor support services.

- Participating actively in other procurement activities of the system project.

Configuration Manager. The configuration manager establishes and maintains the configuration management records for the system and is accountable for the completeness and integrity of the configuration management records. The configuration manager is responsible for:

- Establishing procedures for configuration and change control.

- Maintaining a complete file and log of all change requests, including requests to modify system characteristics that are not yet approved and baselined, as well as requests to modify approved configuration plans.

- Recording the disposition of all change requests, including approval/disapproval, and completion and implementation of the change.

- Preparing periodic reports of configuration status, as needed, and providing assistance and support to the performance of configuration audits.

- Developing and maintaining the program library of life cycle products.

- Providing physical control over baselined documentation (i.e., retaining at least one official copy of the documentation).

- Managing the movement of software and data among developments, test, training, and production environments.

System Tester. The system tester is responsible for:

- Developing and executing all test plans.

- Overseeing testing of the system and executing the test plan to ensure validity and completeness.

- Generating the test analysis report.

- Ensuring user is satisfied prior to acceptance.

e. **Review Committee**. Depending on the scope and complexity of the system development effort, an independent review committee may be established to conduct the review activities of each phase. The review committee role and responsibilities are outlined in the project charter. The review committee prepares review reports for the steering committee and the project manager. Life cycle phase issues checklists and phase review checklists are available in each phase. Guidance is also provided to conduct the periodic system review and post-implementation review.

F. ESTIMATING STAFF REQUIREMENTS

Estimating project human resource requirements is a four step process described below.

a. **Understand the Scope of the System**. A comprehensive understanding of the scope of the system is required to estimate staff requirements more accurately. Such an understanding will minimize the likelihood of misconstruing the complexity of the system or forgetting tasks. In general, the estimator should be an individual close to the source of the work. This means that the individuals responsible for each task to be performed should develop their own estimates and provide these estimates to the project manager. In this way they will feel a greater sense of responsibility for holding close to their estimates during execution.

b. **Identify the Tasks**. The tasks to be performed must be identified before attempting to estimate. Depending on the phase in the development process for which the estimate is being prepared, identify tasks by extracting and refining larger units of work. A prerequisite to estimating is compiling a definitive list of tasks. Any precedential relationships among the tasks should be well understood before beginning. Tasking for a phase can be done during the preceding phase to better assure valid cost figures are used. Using automated project scheduling system (tools) will help determine the impact of a change to one task upon the entire project's resources.

c. **Estimate Time Required**. Once the tasks for a phase have been identified and documented, estimate the amount of time for each task. The project manager can estimate the time required on a task by task basis, keeping in mind the quantity and experience levels of the staff assigned to a task.

d. **Review Estimates**. Automated project scheduling systems will facilitate production of initial project staffing reports and allow a regular review of the initial staffing projections throughout the life cycle. As the project advances through the life cycle, this method of time estimate reviews is likely to result in increasingly accurate estimates. The project manager should schedule these reviews as tasks are produced.

G. TAILORING THE ISLC STRUCTURE

For some information systems, it may be appropriate to alter the ISLC structure. For very simple systems, (e.g., single-user applications), it may be appropriate to combine parts of activities. For projects using methodologies such as rapid prototyping, some activities may overlap. For very large or complex systems, it may be appropriate to divide the system into major subsystems and manage the evolution of each subsystem through its own life cycle. For these systems, it will be important to develop project team structures and other methods to ensure effective coordination across subsystems. Three considerations described below are important when tailoring the life cycle to suit a particular situation.

- The tailoring of the life cycle methodology is to be clearly documented in the project plan for the system.

- All decisions regarding project approach, execution and continuation are to be explicitly documented (i.e., not made by default or accident). Decisions should be made and recorded no later than the completion of the corresponding activity in the life cycle.

- Plans for system reviews and approvals are to be specifically included regardless of any tailoring of the life cycle. This will ensure appropriate program management participation and oversight. As appropriate, reviews and approvals designated in this guidance may be consolidated to reflect the consolidation of individuals' activities.

Although any systems project may require altering, information systems projects listed and described below will always require tailoring. Organizations may find it necessary to add other types of systems projects to this list, or provide more specific guidance, depending on the systems' functions and requirements.

a. **Systems Development Projects**. Systems development projects vary widely according to such factors as size, complexity and resource requirements. New user requirements or major enhancements to an existing system are usually classified as a major systems development effort. A major system is defined as any information system, in development or operation, which is critical to a bureau's mission, or highly visible, or for which the annual cost exceeds one percent of the bureau's budget, or for which the systems life cost exceeds $10 million. Projects which are very technical, controversial, or sensitive, yet requiring less time, may also be classified as major systems development efforts so that proper management control can be exercised.

The ISLC model describes a full range of tasks and products that may be required for system development. All tasks and products may not be appropriate or cost-effective for every project. The model must be tailored to address the specific requirements of individual development projects. Large, complex projects will require a broader range of activities and products than smaller or less complex systems.

b. **Systems Maintenance**. The ISLC model describes a maintenance phase during which requested changes to existing systems are implemented as required. For significant systems modifications, the activities required to implement changes effectively mirror those of the entire system life cycle. Planning, requirements definition, design, development, test and implementation must take place. Products under configuration control must be updated to reflect changes. For significant system modifications, guidelines provided in the planning through maintenance/operations phase should be followed.

The types of activities categorized as systems maintenance may include the following:

Conversion/Upgrade. The modification of a system to reflect changes in the operating environment of both hardware and software.

Corrective. The removal of one or more defects from the system.

Adaptive. The minor modification of data or function value or content to reflect changes in the business environment or legislation.

Perfective. The alteration, rewriting or restructuring of a system to reduce the maintenance effort or result in more efficient operations.

c. **Emergency Maintenance**. In the event that a system in operational status is suddenly unable to execute in an acceptable manner, maintenance must be performed at once. Effort expended in this regard is classified as "emergency maintenance." Typically such projects are approached in a "do now, report later" mode. Functional managers are responsible for insuring that all necessary documentation/approvals are completed once the system is again operational.

d. **<u>Development or Systems Maintenance Work Performed Under Contract</u>**. When performing systems development or maintenance efforts under contract it is necessary to ensure that adequate documentation is required as deliverables as set forth in this manual. Documentation needed to provide an auditable track of decisions must also be required as deliverables.

e. **<u>Prototype Projects</u>**. A mini-cycle called the prototyping life cycle could be embedded within the requirements definition phase. The mini-cycle is an alternative approach to the generation and validation of written requirements specifications as a means of ensuring that a software product will be responsive to user needs. The four stages of the mini-cycle are described in Appendix A.

H. MANAGING THE LIFE CYCLE

The ISLC methodology ensures that all information systems proceed through a series of phases during their existence. The methodology refers to this as "management of the system life cycle" and every information system normally follows the same pattern of evolution.

The methodology segments the total process into a series of distinct phases in order to facilitate management of the process. Information systems projects are generally viewed in terms of the same standard phases.

Systems generally proceed sequentially through these phases. The end of each phase corresponds to a logical milestone in the system life cycle. Each succeeding phase builds on the activities of the preceding phase, increasing what is known about the intended system.

Just as the project is divided into phases to enhance control, each phase may be divided into smaller work units. This is appropriate whenever a phase becomes difficult to manage as a whole due to its size, complexity, or the number of personnel involved.

In any specific project, this subdivision of work should proceed until units of work result which can be accomplished in a short period of time. On some projects subdivision below the phase level may not be necessary. On very large development efforts, further subdivision of activities may be required. Operational control of the process is facilitated by dividing projects into smaller work units. Advantages include:

(l) **Improved estimates**. More accurate time and cost estimates can be produced by dividing projects into small work units. This is achieved both by reducing the amount of uncertainty contained in any single work unit and by averaging high and low estimates through aggregation. When projects are being contracted out, relatively small work units will facilitate monitoring the contract.

(2) **Improved communications**. Smaller, less complex units of work can be communicated from project manager to team member with less confusion. Progress can be measured frequently and problems quickly identified.

(3) **Enhanced control**. Progress measurement and communication are difficult in large projects. Segmenting a phase into manageable work units enables the project manager to meaningfully track and communicate progress.

I. MANAGING RISK

Since every human endeavor involves some element of risk, it is an appropriate management activity to search for, acknowledge, and minimize where a project may be at risk in such areas as nonconforming products, specification noncompliance, technology failure, cost overrun, schedule slippage, etc. The ISLC methodology can help management minimize risk as described in the following paragraphs.

a. **Optimizing the Investment Decision**. Early phases and tasks of a project account for a relatively small percentage of the total cost. Consequently, if projects are to be abandoned it is during the early phases that this should happen in order to minimize loss. Utilizing the phase and task structure of the ISLC, senior management can most effectively make a go/no-go decision. Thus, the ISLC methodology assists senior management in controlling risk and making good investment decisions without experiencing major losses.

b. **Controlling Risk**. Frequent review points are provided when the project is divided into manageable tasks and activities. By authorizing effort to be expanded only for the next phase and requiring a subsequent approval to proceed beyond that point, senior management can limit their exposure to only the cost of the next phase. Further, since estimates are more accurate for the next phase and diminish beyond this, management may assume they are authorizing work to be performed against an estimate that is reasonably reliable. To facilitate risk management at the end of each phase, the ISLC methodology requires a detailed and accurate time and cost projection for the next phase while permitting a less detailed time and cost estimate for the balance of the project. This is done so that management may operate in a controlled risk environment.

c. **Managing Risk on an Exception Basis**. The role for senior management diminishes through the phases of the ISLC. This is appropriate and represents optimum use of senior management's limited time. In order to address the possibility of subsequent cost or scope changes, however, management may wish to qualify their approval. Thus, approval may be given to proceed as long as cost and benefits remain within specified boundaries and all requirements continue to be satisfied. In the event that a subsequent phase review finds that these conditions no longer exist, the previous approval is rescinded and permission to proceed further must be obtained through a senior management review and approval step.

PROJECT CONCEPT DEVELOPMENT CHECKLIST

Feasibility Study

___ Has a feasibility study document been prepared to provide an analysis of the objectives, requirements, and system concept?

___ Does the document provide an evaluation of alternative approaches?

___ Does it provide an identification of a proposed approach?

___ Does it include a description of the requirements and major performance objectives of the system?

___ Does it include an analysis of existing systems that currently address these requirements and objectives?

___ Does it include a detailed description of the proposed system?

___ Does it include a discussion of alternative systems or approaches?

___ Does it provide the rationale for recommending the proposed system?

___ Are user needs considered in developing the conceptual system design?

___ Is this conceptual design used to determine the technical and operational feasibility of the system?

Cost-Benefit Analysis

___ Has a cost-benefit analysis document been prepared to provide adequate cost and benefit information of analyzing and evaluating alternative approaches?

___ Does the document include a description of alternative systems or approaches?

___ Does it contain the cost of development and operation of each alternative?

Exhibit 3-3: Project Concept Development Checklist (cont'd)

PROJECT CONCEPT DEVELOPMENT ISSUES CHECKLIST

Cost-Benefit Analysis (cont'd)

___ Does it describe the benefits that could be attained through the development of each alternative?

___ Does it include a comparative cost-benefit summary?

Project Development

--- Has a steering committee been established to oversee the information system project?

--- Has the steering committee appointed a project manager?

--- Have the expected benefits and outcomes of the information system project been identified?

--- Has a project charter been developed?

--- Is the role of the functional manager been clearly defined?

--- Has funding been allocated to the information system project?

--- Has a project work site been established for the project team?

--- Are there any physical security issues to be considered for the project team and project work site?

--- Are there any acquisition needs to be considered for the project team?

--- Has the IRM budget submissions and ISP been updated?

--- Has a review committee, if appropriate, been established to conduct reviews of the information system project through the life cycle?

--- Has the steering committee authorized the planning phase to begin?

Exhibit 3-3: Project Concept Development Issues Checklist (cont'd)

CHAPTER 4. PLANNING PHASE

A. GENERAL

The purpose of the **planning phase** is to establish a comprehensive model of the recommended solution to provide a better definition of the requirement to be addressed. This phase provides the basis for acquiring the resources needed to achieve the solution through the remaining phases of the life cycle.

The results of this phase provide a framework which guides the work performed in subsequent phase. That work will refine the understanding of requirements, solution and project management approach. The end of this phase is a critical juncture for the project, the point at which program management makes a firm commitment to obtain the resources required throughout the life cycle, and approves continued work to further define the detailed requirements of the recommended solution.

During the **planning phase** the project team is established and other organizations become involved in the life cycle. This phase establishes active involvement by customer organizations, as well as the initial involvement of organizations or individuals responsible for configuration management, data administration, quality control, change control management, system security and procurement support.

A number of **project approach**, **project execution** and **project continuation decisions** are made in this phase. **Project approach decisions** include: how will the effort be organized, what is the life cycle strategy, what methodologies and tools will be used in the requirements definition phase, what is the procurement approach and should funding be obtained for the full life cycle and by whom?

Project execution decisions include: what is the overall architecture of the system/data, to what extent will the system/data interface with existing system/data, how will technical, programmatic, and other risks be addressed, what are the high-level data requirements, what are the high-level functional requirements, who are the user organizations and can the requirement be addressed with existing system/data resources?

Project continuation decisions include: are sufficient funding and other resources available for the entire system life cycle, have the appropriate approvals been obtained for the system to continue and does the information requirement continue to exist?

In this phase, the prerequisites are the feasibility study and the cost-benefit analysis. The requirement is typically identified in a feasibility study, however, the requirement may have evolved from a legislative mandate, a user problem report, a user proposal, or an identified problem or opportunity.

B. TASKS AND ACTIVITIES

The tasks and activities listed below are performed during the planning phase. These tasks and activities may be expanded or deleted depending on the size of the proposed system.

- Develop the project plan
- Establish the project team
- Assess capabilities of current systems, databases, and procedures
- Define/document the high-level functional and data requirements
- Document connections between the high-level functional and data requirements
- Update the IRM/ISP planning and budget submissions
- Prepare initial data management plan
- Plan procurement activities
- Identify privacy issues
- Conduct a planning phase review
- Obtain approval of the requirements definition phase
- Revise the cost-benefit analysis
- Revise feasibility study

C. ROLES AND RESPONSIBILITIES

The project manager uses the project charter to develop the project plan, establish user contact(s) and establish the project team. The team includes project user(s), data administrator, database administrator, quality assurance manager, security manager, configuration management manager, telecommunications manager, procurement manager, budget and finance manager, Freedom of Information Act/Privacy Act (FOIA/PA) representative, systems operations manager, systems developer, and other representatives as appropriate. The need for participation by personnel outside the immediate development area and the need for additional functional managers in the succeeding phases are also identified.

D. DELIVERABLES, RESPONSIBILITIES, AND ACTION

The deliverables listed below must be available for review at the end of this phase. **In some cases, information may not be available to complete deliverables in this phase.**

In those cases provide a draft of the deliverable and continue to update the deliverable as the information becomes available. The deliverables may be expanded or deleted depending on the size, scope, and complexity of the system. The project plan, data management plan, computer security plan, and configuration management plan are described in the documentation section of this chapter. The remaining deliverables are referenced by Treasury guidance documents.

DELIVERABLES AND PRODUCTS	PRIME RESPONSIBILITY
Project Plan (Exhibit 4-1)	Project Manager
Data Management Plan (Exhibit 4-2)	Data Administrator
Configuration Management Plan (Exhibit 4-3)	Configuration Manager
Privacy Act Federal Register Notice (TD P 25-04)	FOIA/PA Representative
Acquisition Plan (TD P 76-01)	Procurement Manager
Cost-Benefit Analysis (revised)	Project Manager
Feasibility Study (revised)	Project Manager

================================

MANAGEMENT ACTION	PRIME RESPONSIBILITY
Appoints project team	Project Manager
Reviews and approves deliverables	Steering Committee/Project Sponsor
Obtains end-of-phase concurrence from project user	Project Manager
Reviews project at end of phase	Steering Committee
Authorizes Requirements Definition Phase to begin	Steering Committee

The project manager assigns deliverables to team members having expertise in specific areas. This does not necessarily mean that the team member having primary responsibility for a deliverable must prepare the deliverable exclusively. That team member is empowered to request assistance from other team member having expertise in the subject area or from individuals in the user organization.

E. ISSUES FOR CONSIDERATION

During the **planning phase**, there are specific issues that should be addressed. These issues are included in the Planning Issues Checklist in Exhibit 4-4. This checklist is to be used as a reference for the project manager to ensure that objectives of this phase are met. These issues are independent of the project size and effort. The checklist is also for use in conducting the Initial Planning Review as described in Section F of this phase.

F. REVIEW ACTIVITY

The Initial Planning Review is performed at the end of this phase. The review ensures the goals and objectives of the system are identified and the feasibility of the system is established. The project plan is evaluated for completeness, validity, and approach. Other products of the **planning phase** are reviewed. An Initial Planning Review Checklist, at Exhibit 4-5, is provided to help ensure the review goals are met.

The review is conducted by a review organization which is included in the project plan. Participants in the review include the project manager and project user(s) and may include additional project team members. Depending on the size and complexity of the project, the review may be conducted by a review committee at the request of the steering committee.

G1. DOCUMENTATION - PROJECT PLAN

The project plan is prepared for all projects. It incorporates all key project planning documents, using a building block approach to planning. It is a vehicle for documenting project scope, tasks, schedule, allocating resources, and interrelationships with other projects. It also provides details on the involved functional units, required job tasks, and milestone and review scheduling.

Revisions to the project plan occur at the end of each phase and as information becomes available. Software designed for Gantt charts and project descriptions are available to complete the project plan. The size of the project plan should be commensurate with the size and complexity of systems development effort and should generally follow the outline in Exhibit 4-1.

1. Introduction

Provide a description of the project plan's purpose and scope.

a. **Project Description**. Provide a description of the project in as much detail as is required to understand the nature of the project.

b. **Project Background**. Describe why the project is important to the organization, its mission, and the capabilities the project will provide to the organization. Include any background or history that is important to understanding the project.

(1) **Project Development Strategy**. Provide an overview of the development strategy selected for the project. For example, this strategy might include prototyping of the system, use of commercial off-the-shelf software, conversion of an existing system from one hardware and software family to another, and so on.

(2) **Organization of the Project Plan**. Briefly describe the organization of the project plan.

c. **Points of Contact**. Identify the key points of contact for the project plan, including the project sponsor and manager. Identify any additional points of contact.

d. **Project Reference**. List a bibliography of key project references and deliverables produced prior to this point. For example, these references might include cost-benefit analyses, existing bureau documentation describing internal processes, existing documentation of the system if the project is a conversion, and so on.

e. **Glossary**. Provide a glossary of all terms and abbreviations used in the plan. If the glossary is several pages in length, place it as an appendix.

1. Introduction

a. Project Description
b. Project Background
 (1) Project Development Strategy
 (2) Organization of the Project Plan
c. Points of Contact
d. Project References
e. Glossary

2. Organization and Responsibilities

3. Task Description, Schedule, and Resources
- a. Task Name and Description
- b. Task Schedule
  - (1) Target Completion Date
  - (2) Total Staff Months
  - (3) Total Budget
  - (4) Project/Task lnterdependencies
- c. Required Facilities
- d. Tasks Management
- e. Technology Applications
- f. Resource Acquisition Strategy
- g. Phase Related Requirements
  - (1) Detailed Work Estimates
  - (2) Development Activity Tailoring
  - (3) List of Required Deliverables
  - (4) Quality Assurance Activities
  - (5) Privacy Issues
  - (6) Computer Security Activities
- h. Human Resource and System Design Strategy

4. Risk Areas and Control Measures

5. Work Breakdown Structures
- a. Summary Work Breakdown Structure
- b. Project Work Breakdown Structure
- c. Contract Work Breakdown Structure

Exhibit 4-1: Project Plan Outline

2. Organization and Responsibilities

Include the various bureau organizations and staff titles, roles, and responsibilities involved in the development project. Describe team structures, reporting responsibilities and relationships, and guidelines for status reporting internally within the bureau and externally for any contractor support. Identify the following key organization components:

- o Organization sponsor for the project;
- o Manager responsible for the day-to day administration of the project, if different from the sponsor;
- o Quality assurance organization; and
- o Configuration management organization.

3. **Task Description, Schedule, and References**

List all tasks/activities to be completed within the project. If possible, use diagrams and tables (automated tools) to list the tasks and show any possible relationships among them. Repeat any subsection for each known task within the project.

Provide a detailed description of each task and its schedule, budget, and management. Also include an estimate of each software development phase related work effort and deliverables.

a. **Task Name and Description**. Provide the task name and a detailed description of the task. The task description should be as detailed as necessary to define the task clearly and distinguish among similar tasks.

b. **Task Schedule**. Describe the schedule-related information for the task. Address the target completion date, total staff months of effort, total budget, and activity relationships.

(1) **Target Completion Date**. Provide the target or actual completion date for each phase in which the task is active. As a phase is complete, its actual date is substituted for the target date, and the remaining completion dates are adjusted to reflect the current priorities and resources.

(2) **Total Staff Months**. Provide the estimated and actual staff months of effort for each phase. Divide the staff months into general schedule categories and/or contractor labor categories. This section represents the total staff months for the task, and therefore should be equal to the sum of the estimated phase staff months (from Detailed Work Estimate Section).

(3) **Total Budget**. Provide the complete budget required to accomplish the task, which should be equal to the sum of the estimated phases budget (from Detailed Work Estimate Section). If the task is long term (greater than one year) and crosses fiscal years, subdivide by fiscal year, and provide complete task totals.

(4) **Project/Task Interdependencies**. Specify the project/task interdependencies. A chart or diagram may be presented to illustrate the interdependencies.

c. **Required Facilities**. Identify the required computer hardware and software facilities needed to complete the task. Indicate whether contractor support is required. If the computer hardware and software facilities required are not currently available, then the Resource Acquisition Strategy Section is completed.

d. **Task Management**. Supplement the information provided in "Organization and Responsibilities" if the organization structures and responsibilities vary by task.

e. **Technology Applications**. Describe the technologies and tools used to support the task. If problems could arise as a result of the use of new technologies for computer hardware, software, software development methodologies, automated tools, etc., then an assessment of associated risks should be provided in the Risk Areas and Control Measures Section.

f. **Resource Acquisition Strategy**. Briefly describe the acquisition strategy for any specific hardware, software, or consulting support required for completion of the task. Present the schedule, responsibilities, and activities to accomplish the acquisition strategy. This section is further defined in the Acquisition Plan.

g. **Phase Related Requirements**. Describe the detailed work estimates, tailored life cycle activities, and tailored list of deliverables by phase of the life cycle. Some tasks may not have activities in all phases. Describe only those phases that apply to the task.

Note: The actual structure of this subsection may be organized as

best suits the task. For example, suppose the task has activities in the Requirements Definition and Design phases. Sections 1 - 3, described below, could be repeated for each of these phases. Or, Sections 1 - 3 could be listed once, and paragraphs (or subsections) within those sections could address the two phases separately.

(1) **Detailed Work Estimate**. Summarize the task management considerations during each phase of the life cycle. Use diagrams and tables where feasible. The following items may be included as appropriate:

- o Schedule and budget;
- o Activity chart showing activities within task and interrelationships with other tasks within each phase of the life cycle;
- o Staff months of effort;
- o Support tools required by phase; and
- o Contractor support required.

(2) **Development Activity Tailoring**. Provide a complete checklist for each activity or discipline within each phase of the task. Mark the activities to indicate a comprehensive or reduced level of effort.

(3) **List of Required Deliverables**. Identify the documentation anticipated to be developed during each phase of the task. Explain how each document is tailored to the type of development anticipated during the task.

(4) **Quality Assurance Activities**. Identify and describe quality assurance activities, reviews, and milestones for the task by life cycle phase. Provide information regarding contractor and bureau organizational responsibilities for each of the activities, reviews, and milestones. This section is defined further in the Quality Assurance Plan in the requirements definition phase.

(5) **Privacy Issues**. Identify privacy issues to be addressed during the

phases of the information system development effort and define the process to be establish for addressing the privacy issues throughout the life cycle. If a privacy plan is more suitable, cross reference this section with a plan. For more details on a privacy plan refer to Documentation Section G4, Privacy Act Federal Register Notice.

It is important that there be a preliminary analysis of the potential privacy impacts of the proposed information system. The purpose will be to establish for the project team and the review process an awareness of the privacy related issues that will have to be addressed as the system is planned, developed, and implemented.

(6) **Computer Security Activities**. Review and evaluate risk assessment results and owner's evaluation of sensitivity of information and systems to determine if all system vulnerabilities, threats, risks, and privacy issues have been identified and whether an accurate determination has been made of the sensitivity of the system and information. Details of this section can be used in the Computer Security Plan described in Exhibit 5-4.

h. **Human Resource and System Design Strategy**. If the project creates major changes to the existing work force, a plan is developed with the human resources office to address work force planning.

4. Risk Areas and Control Measures

Describe any risk areas for the project. If known, describe risks that might affect the ability to complete the project on schedule and within budget. Identify factors that contribute to these risks. Examples of such factors are the use of new computer hardware or software technology, uncertainty in the ability to define user functional requirements adequately, and a compressed development schedule. This section is further defined in the Risk Assessment Plan in the requirements definition phase.

Address approaches for mitigating the impacts of these factors. Add subsections as necessary to separate different categories of risk or different risk-inducing factors.

5. Work Breakdown Structures

The Work Breakdown Structure (WBS) is a family tree structure that relates to products produced and tasks performed at the various phases of the project life cycle. The typical WBS products may consist of hardware products, services and life cycle products. A WBS displays and defines the product(s) to be developed or produced, and relates the elements of work (tasks) to be accomplished to each other and to the end product(s).

Typically, the three types of WBSs developed during the system development process are: summary, project, and contract.

a. **Summary Work Breakdown Structure**. The Summary WBS is a high-level WBS that covers the first three levels of the Project WBS. The summary WBS is used for management presentations, but is not used for detailed day-to-day project management. The structure of the Summary WBS may vary depending on the nature of the project.

b. **Project Work Breakdown Structure**. The Project WBS is the detailed WBS that is used for the day-to-day management of a project. The Project WBS includes all important products and work elements, or tasks, of the project, regardless of whether these tasks are performed by the bureau or a contractor. The Project WBS may be modified, if necessary, during the life cycle.

For a software system development project, the structure of the Project WBS should also reflect the bureau's life cycle approach. The structure of the Project WBS may vary depending on the nature of the project, and should be customized by the bureau's project manager to reflect the particular project and the particular path through the life cycle. For example, a full scale initial information system development project and a software conversion process would be expected to have somewhat different WBSs. Similarly, the Project WBS for an information system development project with a major hardware complement would differ from a software-only project.

(c) **Contract Work Breakdown Structure**. The Contract WBS (CWBS) is a further breakdown of the contract specific WBS that covers the products and work elements, or tasks, from Project WBS to be performed by a contractor. In addition to items derived from the Project WBS, the CWBS will have contractor specific items which may not reflected in the Project

WBS. Depending on the nature of the project, the contractor may be responsible for a given part of the project development activities (for example, quality assurance), for a specific part of the development life cycle (for example, requirements definition), or for the entire development process.

A preliminary CWBS is usually specified by the bureau in the Acquisition Plan. The contract line items, configuration items, contract work statement tasks, contract specification, and contractor responses will typically be expressed in terms of the preliminary CWBS.

G2. DOCUMENTATION - DATA MANAGEMENT PLAN

The data management plan provides details to major topics that should be addressed for data management during the system life cycle. Data management actually begins in the initiation activities, proceeds as requirements are defined and software is implemented, and continues until the system is replaced or terminated. The purposes of data management are to: (1) focus attention upon ensuring that the data provided by systems meets program requirements; facilitate successful data management for each project; and 3) provide a common approach to data management across organizational lines.

The data management activities performed during the systems life cycle are based on the following basic principles:

- <u>Data is a valuable resource</u>. Data is collected, stored, and used to support major program activities and decisions, making accurate and timely data an important resource.

- <u>Data is defined separately from the technology used to collect and store it</u>. Operating functional area data requirements are recorded clearly prior to designing automated data collection and storage methods, so that program needs are understood and recorded.

- <u>Accurate information about data is essential</u>. Effective management of data collected by functional areas requires that accurate information about data (metadata) be kept.

- <u>Common data management guidelines, methods, and tools are used</u>. A common approach to defining, modeling, designing, and documenting data will improve data quality and make it easier to share data among systems and offices.

As technology evolves, data dictionaries take on an increasingly important role. Many of the most modern relational database management systems require that a production data dictionary be loaded before the DBMS is used. Currently, system and database design activities ar being supported by automated software tools that are also designed around data dictionaries of their own. The data administrator will have to plan and monitor the project's collection, use, and transfer of metadata throughout the system life cycle. Lack of attention to this issue may cause delay of the system, increase the project cost or cause a database to be implemented that does not meet requirements. It is always wise to consider the availability of organization-wide data resource inventory to support the project.

The data management plan provides details of areas that reflect the data management approach. Provide as much detail as possible to ensure that all appropriate information is described. **It is understood that some of the information needed here might not be available since the data management approach and plan are developed in this phase.** But in most instances indicate all areas that will be addressed and describe those areas in the remaining phases. Follow the outline at Exhibit 4-2 to complete the data management plan.

1. **Information Need**
2. **Planning Phase**
    a. Data Administration Responsibilities
    b. High Level Data Requirement
    c. Data Documentation Responsibilities
    d. Life Cycle Methodology and Tools
    e. Plan for Physical Flow of Data
3. **Requirements Definition Phase**
    a. Data Requirements
    b. Data Quality Assurance Plans
    c. Data Security Requirements and Strategy
4. **Design Phase**
    a. Logical Data Model Revision
    b. Physical Database Design
    c. Design Data Dictionary
    d. Data Administrator
    e. Data Conversion Strategy and Plan
5. **Development Phase**
    a. Data Structures for Programming Support

b. Data (Structure) Revision Approach
c. Data Backup, Logging and Recovery Plans
d. Data Documentation and User Training

6. Test Phase

a. Data Testing Strategy and Test Plan
b. Testing Support

7. Implementation Phase

a. Production Data Dictionary
b. Cutover Plans

8. Operations, Maintenance and Disposition Phase

a. Database and Metadata Management
b. Support of Configuration Management
c. Audit and Evaluation Support Plan
d. Response to Evaluation Report
e. Database Data Definition Language and Metadata Disposition
f. Data Disposition
g. Cutover Procedures

Exhibit 4-2: Data Management Plan Outline

1. **Information Need**

Prepare a summary of the information need created by the information management problem discussed in the strategic planning document and feasibility study. If it is necessary to collect new information or access existing information to address a requirement, document it in the data management plan. Provide the scope and the programmatic missions with the information need, and review this information at the start of each phase to be sure it is still valid.

2. Planning Phase

a. **Data Administration Responsibilities**. Provide the data administration responsibilities of the organization sponsoring the project. Also provide the roles and responsibilities of other individuals associated with data administration activities if

appropriate. For example, the data definer may be the person determining meaning of data and prescribing procedures to aggregate and refine the data. Include the primary data users for each data entity identified during the planning and requirements definition phases. In a database, data entity is anything about which information can be stored; for example, a person, concept, physical object or event. Reference the Quality Assurance Plan in the requirements definition phase.

b. **High Level Data Requirements**. Document high level data requirements by addressing the following information.

(1) **Entity List**. List data entities about which data is needed.

(2) **Entity Definitions**. Provide programmatic definitions of each entity.

(3) **Entity Identifiers**. Identify data elements used to uniquely identify each entity.

(4) **Conceptual Data Model**. Develop a graphic representation of entities and their relationships. The conceptual data model shows the kinds of data that should be stored for users, and the relationships among the data. Often a conceptual data model is used to coordinate the data requirements of one project with the requirements of other projects by providing a framework for assigning data elements to specific, high-level data entities.

(5) **Likely Sources of Data**. Identify the likely functions and organizations providing data.

(6) **Information Flow/Data Model Validation**. Document the plan to validate that each entity in the high level information flow is in the conceptual model.

(7) **Data Distribution Plan**. Record the use of each entity in matrices, by function and by distribution/location.

(8) **Information Collection Burden**. Plan for any additional information collection burden hours imposed by the solution to the information need.

c. **Data Documentation Responsibilities**. Provide plans to allocate responsibilities for documenting data requirements, data designs and physical data structure. To save time and effort, research and review plans done for earlier projects of the same scope. Include the data objects to be documented, the attributes to be documented and the functional areas responsible for creating the documentation. Also detail the medium to be used for recording dictionary documentation. Prepare maintenance procedures for dictionary documentation, in coordination with the configuration management plan.

d. **Life Cycle Methodology and Tools**. Describe the methodologies and automated tools to be used during the project to support data management activities. Be sure to include your explicit plan for managing the flow of metadata (information about data) between methods and tools through the life cycle. These tools might include automated data dictionaries, data modeling and data design tools, and database management systems.

e. **Plan for Physical Flow of Data**. Provide plans for the physical data (data sets) flowing through the system. This is important when automated data is added to the system from external sources, or when multiple physical data sets need to be updated from a single source of data. Develop timeliness and error correction procedures later.

3. **Requirements Definition Phase**

The requirements definition phase activities and deliverables are important since they define the system's detailed data requirements. Provide information on the following topics.

a. **Data Requirements**. Include the following information to describe data requirements.

(1) **Interview Plans**. Determine and record interviewees names, functional area subject of the interviews, and relationship between the data requirements interview and the functional requirements interview.

(2) **Data Analysis by Process**. Determine and document the plan to identify and analyze the data elements required by each process during the **functional analysis**. This analysis and the logical data

model will produce the detailed data requirements for the system.

(3) **Entity Normalization**. Determine and record the methodology used to normalize the entities required by each process. Normalization is the process of reducing a data entity to its most basic form, removing repeating data elements, data elements not dependent upon the key of the entity, and data elements dependent on the key of other entities.

(4) **Conceptual Data Model Revision**. Determine and document your plan for updating the conceptual data model as new entities are determined. Review the information flows, or high level data flow diagrams to ensure that data entities representing all the data in the data flows and stores on these flow diagrams are represented in the conceptual data model. Additional data entities are defined, added, and the entity-relationship diagram is updated until is complete.

(5) **Logical Data Model**. Determine and document details to create logical data models for each functional process, and for the project as a whole. A logical data model is a graphic depiction of the logical, or programmatic, data needed to support an organizational mission. The logical data model provides a clear, accurate description of data requirements that physical database designers use to begin design of databases.

Validate that all data entities and elements have been identified by examining the data flow diagrams to ensure that all data elements in the data flows and data stores have been accounted for in the logical data model. Users should then review the model and approve the logical model's accuracy.

Finally, add all the descriptions of the data entities and data elements in the logical model, as well as their definitions, to the requirements data dictionary. The requirements data dictionary and the logical data model diagrams will comprise the detailed data requirements for the project.

(6) **Requirements Data Dictionary**. Determine and document your plan for recording metadata describing data entities and data elements in the requirements data dictionary. Be specific about

responsibilities, activities, and external support needed.

The requirements data dictionary is produced by data modeling activities that take place during the planning and requirements definition phases. Metadata recorded about data entities and data elements include their names, programmatic definitions, purpose in the programs, data administration, data definer, and source.

Many automated software development tools provide good support for the requirements data dictionary. If automated tools have used to perform data modeling, then some of the documentation in the requirements data dictionary can be loaded from the software development tools into the data dictionary. In addition, automated tools used to support physical database design should contain metadata derived from the data modeling tool, or from the requirements data dictionary. If the metadata used in each tools is based upon the metadata produced by the preceding tool, time and effort may be saved while minimizing the risk of inaccurate metadata being used to produce a database design.

(7) **Data Flow/Logical Data Model Validation**. Document the plan to validate the data elements in the logical data model against the data elements in data flows.

b. **Data Quality Assurance Plans**. Document and cross-reference the quality assurance plan in the requirements definition phase if users want to measure the quality of the data in the system periodically. If quantitative measures of quality for specific data elements or combinations of data elements are established, provide a plan for measuring databases' ability to meet these measures. Include the organization responsible for monitoring data quality, who will monitor the data quality, what data will be monitored, how will it be monitored, how often, how will problems be resolved, and who will monitor problem resolution.

c. **Data Security Requirements and Strategy**. Identify data security requirements, record the requirements and implement them. Detail responsibility for identifying sensitive data and protecting the data in accordance with the data administration roles of the system. Data security requirements are addressed in accordance with applicable strategies in the Treasury Security Manual, TD P 71-10, and are provided in greater detail on the systems security and integrity requirements worksheets in the risk assessment plan described in TD P 85-03, Risk Assessment Guideline.

4. **Design Phase**

a. **Logical Data Model Revision**. Describe the approach for revising the overall logical data model.

b. **Physical Database Design**. Document the method to use to prepare a physical database design (data design) for the system. The physical database design is used to create the physical data structures to support the system.

Physical database design activities begin after the logical data model is completed. Information contained in the logical data model and the requirements data dictionary is used as a starting point for the design process. The designer transforms the logical data model into an initial physical database design that can be implemented in a database management system. Data definition language statements are produced during the development phase to provide common data structures for programming and unit testing support.

c. **Design Data Dictionary**. Describe the approach for recording information (metadata) about the physical database design in the design data dictionary. Include an outline of what metadata can be copied from the requirements data dictionary.

The design data dictionary is maintained through the rest of the life cycle. This dictionary contains detailed data documentation of the physical database design. It also contains programmatic definitions of data elements stored in the requirements data dictionary, as the physical design information is created.

Technical information about the physical database design is entered into the design data dictionary. It describes the design of the physical database structures and the manner in which these structures are implemented in the test versions of the databases. In addition to the data elements, the design dictionary contains documentation of databases, physical records, segments, data sets (or files), and keys.

d. **Data Administrator**. Identify and document the data administrator for data in the database. Since distributed databases have multiple administrators, plan carefully. Record this information in the design data dictionary during the design phase.

e. **Data Conversion Strategy and Plan**. Provide details for data conversion plans and data conversion support activities in a Conversion Plan (see Design Phase) if existing data will be used in the new system. Include information such as sources, media, load programs required (automated procedures), and validation plans.

5. **Development Phase**

a. **Data Structures for Programming Support**. Record plans to support software program development and unit testing. Coordinate plans to avoid problems. This section will vary depending upon the database and programming language used.

b. **Data (Structure) Revision Approach**. Document the details of the change control procedures you will use to support changing and adding data structures to the database during this phase. Refer to the system's Configuration Management Plan, described later in this chapter, for guidance concerning the overall framework use. Also document the approach to updating the design data dictionary to keep abreast of all changes.

c. **Data Backup, Logging and Recovery Plans**. Record plans for backup, logging, and recovery of physical data sets stored in the data structures the system creates. Document plans for test databases, and plans for the production versions of data.

d. **Data Documentation and User Training Materials**. Ensure that data management concerns are included in all documentation, not just the data dictionaries, and that user training material includes an emphasis on data creation, collection, validation, and quality assurance issues.

6. **Test Phase**

a. **Data Testing Strategy and Test Plan**. Document general data testing and test plan strategy for testing database performance and functionality to prove a viable database design exists that meets detailed data and system functional requirements. Include this information in the Test Plan in the requirements definition phase.

b. **Testing Support**. Include plans for supporting integration testing and acceptance testing with the databases that have been developed (refer to Test Plan).

7. **Implementation Phase**

a. **Production Data Dictionary**. Document plans to create the production data dictionary to support system integration and acceptance testing.

b. **Cutover Plans**. Record the activities the database administration function must carry out to support cutover to the production system. This cutover usually involves unloading test data, securing the production version of software, and loading data that has been converted from an existing system.

8. **Operations, Maintenance and Disposition Phase**

a. **Database and Metadata Management**. Describe procedures and plans for supporting the production database and keeping the production data dictionary accurate. Describe procedures to keep the design and production data dictionaries in synchronization when enhancements occur.

b. **Support of Configuration Management**. Coordinate data management with the configuration plans of the system. Refer to the Configuration Management Plan for details of the system's approach.

c. **Audit and Evaluation Support Plan**. Document planned activities if the data management activities are required to support the audit and evaluation process.

d. **Response to Evaluation Report**. Document the planned actions if data management related actions are required in response to points or recommendations of an evaluation report.

e. **Database Data Definition Language and Metadata Disposition**. Document plans to archive or pass to another database the data definition language statements that create the data structures in the database, and the metadata in the production data dictionary.

f. **Data Disposition**. Document plans to archive or pass to another database the data sets (data) in the database.

g. **Cutover Procedures**. Document plans for the database administration function to support the cutover from production to archiving or another system.

G3. DOCUMENTATION - CONFIGURATION MANAGEMENT PLAN

The purpose of the configuration management (CM) plan is to establish uniform CM practices to manage the establishment of and changes to the system hardware and software. The plan identifies the physical location of system baselines; the locations in which hard copy documentation is stored; and the automated libraries used to store other documentation and the software components of the system. CM helps maintain the integrity of the system throughout its life cycle and facilitates communication about the system among project team members, users, and other supporting organizations. CM guidelines are applied through the systematic identification, control, and audit of system characteristics. They include the following:

- Configuration identification of functional and physical characteristics of a system through structured documentation baselines;
- Configuration control of changes to the physical and functional characteristics of hardware and software systems and baseline documentation describing them;
- Configuration status accounting about the current configuration and changes to it;
- Configuration audits to verify that system performance and configuration are accurately identified in the baseline documentation; and
- Storage and control of access to the baseline documentation, source code, and executable code.

1. General

2. Configuration Identification

3. Configuration Control

4. Configuration Accounting

5. Configuration Audits

a. Functional Configuration Audit

b. Physical Configuration Audit

6. Problem Reporting and Corrective Action

7. Tools, Techniques, and Methodologies

Exhibit 4-3: Configuration Management Plan Outline

A baseline is a documented technical description that becomes a reference point against which changes can be proposed, evaluated, and incorporated. Follow the outline in Exhibit 4-3 to complete the configuration management plan.

1. **General**

Identify the specific purpose and scope of the CM plan. Name and briefly describe the project covered by the CM plan. Include the name of the configuration manager for the system who will be responsible for establishing and maintaining the configuration management records for the system. In addition, include plans to establish the change control board. The change control board examines requested changes to the system, direct the change request impact analysis and, based on the results, determine the changes that are to be made and those that are not to be made to the system.

2. **Configuration Identification**

Cite the project documents that identify and define the configuration baseline characteristics. Identify the following:

- Documentation that establishes the selected system baseline;

- Documentation and media defining code and documentation that are placed under configuration control, and the corresponding version, release, and change status of each deliverable item; and

- System components comprising the application system, to the module level.

Configuration identification serves to clearly delineate the significant characteristics of the system, providing a common language, or referencing scheme, for describing the system. It is the delineation of specific configuration items that will enable all individuals involved in the evolution of the system to communicate effectively, using the common language. Examples of such characteristics include specific functional and data requirements, specific characteristics of the system design and system documentation.

3. **Configuration Control**

Identify approval/disapproval authority for changes, library controls, and the librarian used for the project. In addition, procedures for controlling the preparation and dissemination of changes to deliverable software and documentation that have been placed under configuration control. Specific change control and software control procedures are covered in detail.

Change control is a process for determining what changes are to be made to the system. Change control requires the documentation of specific requests for modifications, and a review of the requested modifications, and consideration of their impact on the system, before they are made.

Software control is a set of procedures to ensure that the integrity of the system is preserved when approved changes are being made to the system, or in the event of a disaster and restoration of the system is needed. Software control procedures are particularly important during the operation phase of the life cycle. Software control ensures that changes to the computer programs are developed and tested using a copy of the programs and test database, and do not adversely affect system users. However software control is also important during the development and implementation phases to control changes during the planning and installation of the system.

4. Configuration Accounting

State the procedures for generating documentation that will provide traceability of changes to controlled products and communicate the status of configuration items to management. Identify data content and format of accounting records, and their frequency and distribution.

Configuration accounting is an administrative procedure for maintaining system baselines and monitoring the status of the system throughout the life cycle. The configuration accounting procedure may also include documentation for use by the project team throughout the life cycle.

5. Configuration Audits

State the procedures for preparation and execution of audits for establishing the traceability of requirements identified for the project. The procedures identify what will be certified and provided to management for approval.

Configuration audits are examinations of the products and related documents submitted for inclusion in a baseline to assure that they are complete, clearly presented, and internally consistent. This examination is oriented to adherence to guidance and standards. These audits support reviews and evaluations of the system by ensuring that required products and documents are complete and provide effective traceability to related products. Audits do not evaluate qualitatively the programmatic and/or technical content of the product. This is done by other reviews and other quality assurance activities. Audits help ensure that the resources used to conduct reviews and evaluations are not applied to products that are not yet ready for the review.

a. **Functional Configuration Audit (FCA)**. Describe the point at which an FCA will be completed and the process to be followed. An FCA is a means of validating that the developer (contractor) has completed the system in accordance with the requirements definition. The FCA ensures that technical documentation and test analysis/reports accurately reflect the functional, operational, and performance characteristics specified.

b. **Physical Configuration Audit (PCA)**. Describe the point at which a PCA will be completed and the process that will be followed. The PCA is a means for validating the system baseline. The PCA ensures that configuration items are developed as specified in the design documents; that the items are tested, verified and/or validated successfully and that any differences are resolved.

6. Problem Reporting and Corrective Action

Describe the procedures for reporting, prioritizing, tracking, and resolving problems that result from reviews, audits, and tests.

7. Tools, Techniques, and Methodologies

Identify the automated tools, techniques, and methodologies to be used to support configuration management.

G4. DOCUMENTATION - PRIVACY ACT FEDERAL REGISTER NOTICE

The Department's Privacy Act Handbook, TD P 25-04, provides guidelines for planning, preparing, and publishing notices of the existence and composition of each Privacy Act system of records maintained by Treasury bureaus. Information about individuals cannot be collected until a notice of that system has been published in the Federal Register.

The project manager establishes initial contact with the bureau's Freedom of Information Act (FOIA) and Privacy Act (PA) representative in this phase. The project manager and FOIA/PA representative work together in the succeeding phases to determine and document system features and requirements pertinent to these Acts. This collaboration is necessary due to the long lead time necessary for inclusion of the Privacy Act Notice in the Federal Register and for the comment period. The notice should be announced in the Federal Register in the design phase. In addition, the FOIA/PA representative works with the project manager and appropriate organizations to prepare a notice to announce the disposal or deletion of a Privacy Act System of Records, if applicable.

A privacy plan may be needed to recognize the importance of privacy and to ensure that all privacy issues are addressed appropriately. The plan will explain how the information system will adhere to privacy laws and regulations, and how it will demonstrate compliance. The plan will identify privacy requirements. The privacy plan will specifically address issues raised during the preliminary privacy analysis, the risk assessment, and those issues identified subsequently.

The succeeding information is recommended to develop a privacy plan and the activities described should be required of every information system project as it proceeds through the system life cycle. These activities will ensure that all privacy issues are addressed and the privacy of all individuals are thoroughly protected.

The project plan will be prepared by the project manager with coordination with the bureau and/or Departmental privacy representative. The project plan will include a preliminary analysis of the potential privacy impacts of the proposed system. The purpose will be to establish for project team members and the review process an awareness of the privacy related issues that have to be addressed as the system is planned, developed, and implemented. In performing this initial analysis, the following questions should be answered:

o What information is needed?

o Why is the information needed?

- o How will the information be used?
- o Who will own the information?
- o Where will the information be obtained?
- o How will the information be protected?
- o Who will have access to the information?
- o How long will the information be stored?
- o How will the information be retrieved?
- o How will the information be disseminated?

The analysis will establish, among other things, whether the proposed system will contain personally identifiable information and/or third party information, and whether the information will be transferred to other agencies. The issues above are examples of issues which would have privacy implications. It is anticipated that as the system development effort progresses and requirements and capabilities are changed, new privacy issues will be raised. The plan should be revised as required.

The privacy plan should be completed in the development phase. The plan should formally detail how the system will meet the privacy objectives of minimizing intrusiveness, maximizing fairness, and satisfying individuals expectations of confidentially.

The plan should specifically address issues raised during the preliminary privacy analysis and the risk assessment. The plan should establish the process by which the project will consider. The plan should explain how the system will adhere to privacy laws and policies and how it will demonstrate compliance.

G5. DOCUMENTATION - ACQUISITION PLAN

The acquisition plan is a document which shows how all hardware, software, telecommunications capabilities, and contractor support services are acquired during the life of the project. The acquisition plan helps to ensure that the needed resources can be obtained and are available at the time they are needed. The plan includes a schedule that lists the activities to be completed as well as deliverables to be produced with appropriate estimated dates of completion. This is necessary to:

- o Provide management with adequate information for making decisions about the procurement of contractor services, including assurance of the availability of funding;
- o Provide technical evaluation personnel with adequate information for analyzing and evaluating vendor proposals;
- o Ensure that vendors will have adequate information for preparing bids; and
- o Provide the source selection official with adequate information on which to base a selection.

The acquisition plan typically has its own mini-life cycle of presolicitation; solicitation and award; and post-award. The life cycle model will vary according to the system development effort. This means that the activities in each area will differ significantly. The model acquisition plan includes the following activities by providing a schedule and estimated completion dates for each. Follow the Guidelines for Acquiring Federal Information Processing Resources, TD P 83-01, and Treasury Acquisition Procurement Regulation (TAPR), TD 76-01, to complete the activities listed below.

Requirements Analysis Package
Statement of Work
Procurement Plan
Acquisition for Contractor Services
Legal Opinion of Statement of Work
Solicitation of Services

Technical Evaluation Report
Source Selection Recommendation
Contract Award
Adjustment of Funds
Contract Performance

The acquisition plan becomes critical after the functional requirements document has been approved. Several acquisitions will probably be needed to procure an entire system and will be a continuous part of the cycle. The acquisition plan is continuously updated and the involvement of users working closely with the project manager and the procurement manager becomes increasingly important. The project manager works directly with the procurement manager and the bureau's procurement and contracting authorities to ensure the timely award of the needed resources.

PLANNING ISSUES CHECKLIST

The project team uses this checklist to ensure that specific issues have been addressed. Consideration is given to the questions below for each system project. Action may not be required for some items.

Planning Phase

___ Have the project scope and purpose been defined?

___ Have the development project organization and responsibilities been defined?

___ Has the work breakdown structure been used as the basis for budget development, allocation and planning?

___ Has planning been based on: contract, schedule, budgets, user requirements, Treasury guidelines, operational and technical environment, and limitations on personnel and resources?

___ Have the background for project plans been defined (i.e., requirements, development environment)?

___ Have management and user approvals to proceed with project development been obtained?

___ Have data administration activities in the data administration manual been addressed?

___ Has the project plan been approved by senior management?

Exhibit 4-4: Planning Issues Checklist
PLANNING ISSUES CHECKLIST

Project Plan

___ Are the project scope and purpose defined?

___ Are the development project organization and responsibilities defined?

___ Are the management and technical controls to be applied to the project addressed?

___ Are the resources to be applied in developing and implementing the software identified?

___ Are the software development schedule and controls described?

___ Are specific procedures for monitoring project progress and reporting defined?

___ Are specific project documentation requirements defined?

___ Is the development approach clearly defined?

___ Are development and test tools described?

___ Are security and data access requirements identified and described?

___ Have software development requirements been included in the work breakdown structure (WBS)?

___ Is the WBS current, and have updates been used to update the software plans?

___ Is work required by the contract related to WBS elements?

___ Are schedules based on WBS, plans, budgets, and technical parameters?

Exhibit 4-4: Planning Issues Checklist (cont'd)

PLANNING ISSUES CHECKLIST

Project Plan (cont'd)

___ Have quality factors that are most important for system development been identified?

___ Have the deliverable documents and sections been identified, based on the selected quality factors?

___ Are schedules consistent with project constraints?

___ Do the schedules identify current user milestones, internal software program schedules, and working schedules?

--- Will the information system create major changes to the existing work force and if yes, are strategies being developed to address work force planning?

Privacy Act Issues

--- Has a process been established to address privacy issues having an impact on the proposed information system?

--- Have project team members been scheduled for training in an awareness of privacy issues relative to the project?

--- Does the magnitude of the information system development effort require a privacy plan?

___ Is the proposed system covered by the Privacy Act? If so, is a Privacy Act Notice being drafted?

___ Have provisions been made for required disclosure accounting?

Exhibit 4-4: Planning Issues Checklist (cont'd)

PLANNING REVIEW CHECKLIST

1. Date/time of meeting _____________.

2. Attendants:
   Project Manager ______________
   Project Sponsor ______________
   Project User ______________
   Budget/Finance Manager ______________
   Data Administrator ______________
   Configuration Manager ______________
   Security Manager ______________
   Procurement Manager ______________
   FOIA/PA Representative ______________

3. Were the following items present for the review or their absence justified?

   Project Plan ____________
   Data Management Plan ____________
   Configuration Management Plan ____________
   Acquisition Plan ____________
   Privacy Act Federal Register Notice ____________
   Updated Cost-Benefit Analysis ____________
   Updated Feasibility Study ____________

4. Were the objectives of the system clearly identified? (Yes/No) If no, explain.

5. Were the objectives of the system found to be feasible? (Yes/No) If no, explain.

6. Was the project plan found to be acceptable? (Yes/No) If no, explain.

7. Was the cost-benefit analysis present? (Yes/No) If yes, was it acceptable? (Yes/No) If no, explain.

8. Other issues raised.

Exhibit 4-5: Planning Review Checklist

CHAPTER 5. REQUIREMENTS DEFINITION PHASE

A. GENERAL

The primary purpose of the **requirements definition phase** is to analyze user needs to develop user requirements. User requirements are defined in a functional requirements document (FRD) at a level of detail sufficient to permit system design to proceed. This phase expands the high level functional and data requirements into specific detailed functional and data requirements. These requirements provide the basis for a more concrete assessment of system benefits and costs. The requirements also form the basis for the detailed design of the system.

During the **requirements definition phase**, the system is defined in more detail, i.e., system inputs, processes, and outputs. This process takes place at the functional level, in that the system is described in terms of the functions to be performed, not in terms of computer programs, files and data streams. The emphasis in this phase is on determining what functions must be performed, not how to perform those functions.

The initial test plan (TP), derived from the FRD, establishes a minimum level of acceptable performance and conformance to stated requirements. The test plan is refined through the **development phase**, as testing conditions and scenarios are identified during design. In addition, the system sponsor serves as the liaison to Freedom of Information Act (FOIA) and Privacy Act (PA) personnel in determining and documenting system requirements and features to these Acts.

The **requirements definition phase** is initiated when a project is approved for development in the Initial Planning Review or by management direction. Documentation on the user requirements from the planning phase are used as the basis for further analysis of user needs and the development of detailed user requirements. The analysis may surface new insights into the overall information systems requirements and in such instances, all deliverables should be revised to reflect this analysis.

A number of **project approach**, **project execution** and **project continuation decisions** are made in this phase. The **project approach decision** includes: what specific methodologies and tools will be used in the design phase? This tool/method portfolio includes CASE tools and prototyping methods, and the linkage of these methods and tools across all future life cycle phases and activities.

Project execution decisions include: what are the detailed functional requirements and their priorities, what are the detailed data requirements and their priorities, what potential requirements will not be included in the new system, and what modifications are needed to be made to previous deliverables?

Project continuation decisions include: Does the information system requirement continue to exist, do the detailed requirements address the requirement sufficiently to permit continuation to the design phase, and is sufficient funding and other required resources available for the remainder of the system life cycle:

B. TASKS AND ACTIVITIES

The following tasks and activities are conducted during this phase. These activities may be expanded depending on the size of the proposed system.

- Conduct detailed analysis
- Survey/document the user operation
- Document the user process work flow
- Identify and document detailed functional requirements
- Identify and document detailed data requirements
- Load requirements data dictionary
- Define the quality assurance/internal control mechanisms
- Develop preliminary test plans and criteria
- Identify security requirements
- Develop risk assessment plan
- Obtain approval of requirements definition phase
- Continue procurement activities
- Update project plan
- Update cost-benefit analysis
- Finalize feasibility study

C. ROLES AND RESPONSIBILITIES

During this phase the project team collects the details of the system functions down to the data element level. The relationships among these functions and the data elements are then analyzed. Data element definitions (DED) and formats are developed and recorded. The existing user process work flow is documented, if any. Throughout this process, the project team members think about how the system architecture might best be configured. This process forces the identification of the modules in the system.

Team members participating in this phase are the project manager, project user, project sponsor, data administrator, database administrator, programmer analyst, security manager, configuration manager, quality assurance manager, procurement manager, FOIA/PA representative, and systems operations manager.

D. DELIVERABLES, RESPONSIBILITIES AND ACTION

The deliverables listed below must be available for review at the end of this phase. The deliverables may be expanded or abbreviated depending on the size, scope and complexity of the systems development effort. The functional requirements document, test plan, quality assurance plan and additional system security requirements are described in the documentation section of this chapter.

DELIVERABLES AND PRODUCTS	**PRIME RESPONSIBILITY**
Functional Requirements Document (Exhibit 5-1)	System Developer/Project Team
Test Plan (Exhibit 5-2)	Security Manager
Quality Assurance Plan (Exhibit 5-3)	Quality Assurance Manager
Computer Security Plan (Exhibit 5-4)	Security Manager
Risk Assessment Plan (TD P 85-03)	Security Manager
Project Plan (revised)	Project Manager
Data Management Plan (revised)	Data Administrator
Acquisition Plan (revised)	Procurement Manager
Cost-Benefit Analysis (revised)	Project Manager
Configuration Management Plan (revised)	Configuration Manager
Privacy Act Federal Register Notice (revised)	FOIA/PA Representative
Feasibility Study (finalize)	Project Manager/Project Sponsor

==================================

MANAGEMENT ACTION	**PRIME RESPONSIBILITY**
Reviews and approves deliverables	Project Manager/Project Sponsor
Obtains end-of-phase concurrence from user	Project Manager
Reviews project at end of phase	Steering Committee
Authorizes Design Phase to begin	Steering Committee

E. ISSUES FOR CONSIDERATION

During the **requirements definition phase** there are specific issues that should be addressed. These issues are included in the Requirements Definition Issues Checklist at Exhibit 5-5. The checklist is used by the project team to ensure objectives of this phase are met. These issues are independent of the project size and scope. The checklist is also used for conducting the Functional Requirements Review as described in Section F in this chapter.

F. REVIEW ACTIVITY

The Functional Requirements Review is performed at the end of this phase. This review ascertains the completeness and attainability of the functional requirements. Project documents are validated for conformance to quality factors as described in the QA plan, and revised if necessary. A Functional Requirements Review Checklist is provided at Exhibit 5-6 to ensure that review goals are met.

Participants in the functional requirements review include the project manager and the project user. Other project team members may be requested to participate in the review. Depending on the size and complexity of the project, the review may be conducted by an oversight board.

G1. DOCUMENTATION - FUNCTIONAL REQUIREMENTS DOCUMENT

The functional requirements document serves as the foundation for system design and development. It captures all user requirements to be implemented in a new or enhanced system to satisfy business needs. It may also contain descriptions of the existing and proposed methods and procedures, performance requirements, inputs and outputs, and other known requirements.

If required for analysis, a general description of the business policies and functions to be supported by the system is provided. This description may be independent of the way that the functions are currently carried out. If the intent is to automate existing functional processes exactly, then the existing processes should be described. Some areas of the outline may not be applicable to the system under development and can be excluded or marked, "Not Applicable." Follow the outline in Exhibit 5-1 to complete the functional requirements document.

1. Introduction

a. Project Description
b. Points of Contact
c. Project References
d. Glossary

2. **Functional Process Requirements**

a. Detailed Functional Descriptions
b. Interfaces Between Functions
c. External Interface Requirements

3. **Detailed Data Requirements**

a. Data Dictionary
b. Logical Data Relationships

4. **User Performance Requirements**

a. Timeliness
b. Testability
c. Traceability

5. **System Performance Requirements**

a. System Sizing and Capacity Requirements
b. System Response Requirements
c. Reliability Requirements
d. System Growth Requirements

6. **Security Requirements**

7. **Quality Assurance**

Exhibit 5-1: Functional Requirements Document Outline

1. **Introduction**

Provide a description of the project and add any additional information as necessary. Do not provide information in this section where the description provided in the following subsections is sufficient.

a. **Project Description**. Provide a brief overview of the project. Use the following subsections to address specific issues.

(1) **Background**. Provide a brief description of the historical background and other background leading up to the systems development project.

(2) **Objectives**. Provide a brief description of the key objectives of the system. Describe any bureau mission-critical aspects of the system.

(3) **Assumptions and Constraints**. Describe the assumptions and constraints involved in the system development. For example, describe cost or other resource constraints, time constraints, equipment constraints, or other constraints.

(4) **Equipment Environment**. Describe the equipment environment that the system is being developed for. For example, describe personal computers, mainframes, local area network or wide area network environments.

(5) **Support Software Environment**. Describe the support software environment. For example, describe operating systems, local area network software, data base management systems, software languages and compilers, computer aided software engineering (CASE) tools, prototyping tools, cross reference listing generators, software profilers, etc.

(6) **Interfaces to External Systems**. Describe the interfaces to other external systems. Reference documentation of those external systems where appropriate.

b. **Points of Contact**. Provide the organization name and title of the key points of contact for the information system development effort. Include the project manager, project user, quality assurance manager, security manager, data administrator, and configuration manager, as appropriate.

c. **<u>Project References</u>**. Provide a bibliography of key project references and deliverables which have been produced prior to this point in the information system development effort. For example, these references will specifically include the project plan, and might include the data management plan, feasibility study, cost-benefit analysis, and acquisition plan.

d. **<u>Glossary</u>**. Provide a list of all terms and abbreviations used in this document. If the list is several pages in length, place it as an appendix.

2. Functional Process Requirements

Provide a description of all automated user functions to be implemented or modified in the system under development.

a. **<u>Detailed Functional Descriptions</u>**. Provide a detailed description of the functions of the system under development. Describe any conventions to be used in the associated subsections. The number, and type, of subsections may be varied as necessary for the particular system being developed.

The term "Function X" is replaced with the name of the function.

(1) **Function X Process Description**. Describe the process associated with the function in sufficient detail to facilitate system development. Include a graphic work flow or process flow, if necessary, to describe the function.

(2) **Function X Inputs**. Describe the inputs to the functions. Include all data elements, files and records, user selected inputs from menus, or any other input to this function. Use graphics or text, as appropriate, to describe these inputs.

(3) **Function X Processing**. Describe any processing, algorithms, calculations, transformation of input data, and so on performed by the function. If appropriate, use graphics or text to describe the processing.

(4) **Function X Outputs**. Describe all outputs of the function.

Include printed reports, display screens of data, transformed data elements, calculated data, records and files, error messages, etc.

b. **Interfaces Between Functions**. Show the relationship between functions provided as a graphical presentation of the relationship. In the graph, cite a name and/or reference number of each function to facilitate discussion of the functions. Provide a textual description of the relationships between the functions if it is necessary for understanding the functional relationships.

c. **External Interface Requirements**. Provide graphical and textual description of the interfaces to external systems. Define record, file, and data base structures that will be interfaced to by the system under development. Define access controls to the external system.

3. **Detailed Data Requirements**

Address the data definitions, data relationships, and data structures in this section. For all new systems or designated systems, all data referenced in this document must be defined. All data elements which are part of the input, processing, and output parts of the functional process requirements in the previous section shall be included.

a. **Data Dictionary**. Provide an alphabetical listing of all data element names, data definitions and explanation of use, formats, and aliases. Address allowable values or edits for correctness.

b. **Logical Data Relationships**. Describe the appropriate relationships among the data by using graphics and text. Group data by record types, file type, schema, subschema, or other groupings if these groupings are known at this point in the system development. Such groupings may be known for systems undergoing conversion and may not be known for newly developed systems. Include relational table structures, entity-relationship diagrams if known at this point.

4. **User Performance Requirements**

Identify all known user performance requirements. The following areas serve as categories for grouping user performance requirements. If the subsections are not applicable to the system under development, indicate "Not Applicable." Add additional subsections as needed.

a. **Timeliness**. Address any requirements for production schedules, user response times, peak user times, and availability.

b. **Testability**. Address any requirements for modularity of the system architecture, system design, and source code. The careful structuring of these components will greatly assist with the testability of the software products.

c. **Traceability**. List all user requirements for reference during future phases. This list may take the form of a matrix to assist in tracing how the requirements are addressed in subsequent phases and documents, including the external design document, internal design document, and test plans. The matrix is created to allow for expansion in future phases. However, in this phase, the user requirement, should be supplied.

5. **System Performance Requirements**

Identify system performance requirements in this section. The subsections serve as categories for grouping performance requirements. If the subsections are not applicable to the system under development, indicate "Not Applicable."

a. **System Sizing and Capacity Requirements**. Include all important or potentially limiting system sizing and capacity requirements. Examples include the computer system memory requirements, and disk drive capacity. Include any telecommunications system throughput capacity requirements, such as the number of transactions per second supported by the LAN, WAN, mainframe, and mainframe front end communications processor.

b. **System Response Requirements**. Define response and timing requirements. For example, the system may be specified to require a response to a user transaction within 2 seconds 95 percent of the time.

c. **Reliability Requirements**. Define system reliability requirements, including allowable down time, required availability, mean time between failure, mean time to repair, allowable error rates, or other factors if appropriate.

d. **System Growth Requirements**. Define any important growth requirements such as the: projected increase in the number of users; projected increase in the number of transactions year-by-year over the system life cycle and file growth; and projected increase in the number of files over the system life cycle.

6. **Security Requirements**

Provide a description of security and privacy requirements. The presence of the following factors may indicate that the system has areas of security concerns:

- Privacy Act considerations (FIPS PUB 41);
- Proprietary information to be handled by the system;
- Computer Security Act considerations;
- Financial systems which authorize funds allocations or funds flows that are based on electronic signatures or passwords;
- Specific risks related to the system and its processing environment; and
- C2 requirements for sensitive, but unclassified systems.

Address risks from theft, disclosure, unauthorized access, eavesdropping, programmed attacks, misrouting, misplacement, erasure and accidental damage considerations. Security requirements are further defined when completing the system security and integrity requirements worksheets in the risk assessment plan and the requirements are also cross-referenced in the contingency plan as appropriate.

7. Quality Assurance

For each quality factor, the system developer should provide a brief explanation of its importance. The system developer works with the user to identify the most appropriate quality factors. In addition, for each quality factor not selected, provide a brief explanation for not selecting that quality factor.

For example, for a particular financial system the correctness and integrity quality factors may be important to ensure that the system is implemented as specified and to prevent theft of data that could lead to financial fraud. For the same financial system, the verifiability quality factor may also be important to verify the system's performance. If the bureau's development staff expect to reuse certain system functions in future systems, then the reusability quality factor would be important. In addition, other quality factors might be applicable, depending on the system to be developed. Therefore the most important quality factors would be selected for developing the system.

All members involved in creating the FRD should consider the following fifteen quality factors when determining and documenting the system requirements. By selecting and referring to these quality factors, FRD developers may begin to document expectations and standards that will facilitate the development of a high quality system. The following quality factors can apply to any system.

- **Correctness**. Extent to which all specified functions are implemented, the design is documented according to standards, and the software's performance is within the criteria set for Performance.

  Requirements in the FRD - what was produced was specified; and what was specified was produced.

- **Efficiency**. The resources needed to provide the required functionality are affordable.

- **Expandability**. Enhancing the software's functionality or performance is easy.

- **Flexibility**. Modifying the software for different environments requires little more than a user interface or change of data, without resorting to expandability-type changes.

- **Integrity**. Assurance that the software and data are not being tampered with or stolen and that the software is performing updates appropriately.

- **Interoperability**. The software can be easily coupled with other systems, and the results will comply with industry standards.

- **Maintainability**. Finding and fixing errors is sufficiently easy to allow production to continue, even with the issuance of new releases.

- **Manageability**. Administrative aspects of software modification are facilitated by tools such as configuration control systems, source code libraries, and media control procedures.

- **Portability**. The ability of software of a system to run on more than one type or size of computer or under more than one operating system.

- **Usability**. System is user-friendly; documentation is thorough and accurate; and few keystrokes are required per command. The implemented system fulfills its functional purposes as determined by its users.

- **Reliability**. Prevention of inaccurate results, slow response, system "freezes" (stops without recovery), and system failures.

- **Reusability**. Portions of the software can be used for other applications, as a result of the availability of a large library for standard software building blocks.

- **Survivability**. Although a system failure may cause a degraded functionality, the user can still execute essential functions of the software (for example, a spreadsheet can be calculated even though the printer is not functioning).

- **Verifiability**. Verification of the software's specified performance is simple.

G2. DOCUMENTATION - TEST PLAN

The purpose of the test plan is to ensure that all aspects of the system are adequately tested and that the system can be successfully implemented. The test plan documents the scope, content, methodology, sequence, management of and responsibilities for test activities. The test plan describes the test activities of: unit/module test, subsystem integration test, system qualification test, system acceptance test, and security test in progressively higher levels of detail as the system is developed.

The test plan provides guidance for the management of test activities, including organization, relationships, and responsibilities. The test case procedures may be included in the test plan or in a separate document, depending on system size. The project user assist in developing the test plan which describes the nature and extent of tests deemed necessary. This provides a basis for verification of unit tests and validation of the system. The validation process ensures that the system conforms to the functional requirements in the functional requirements document and that other applications or subsystems are not adversely impacted. Test analysis reports are developed at each level of testing to record the results of testing and certify readiness for system implementation (see test phase).

Problems, deficiencies, modifications, and refinements identified during testing or implementation should be tracked under configuration control and tested using the same test procedures as those described in the test plan. Specific tests may need to be added to the plan at that time, and other documentation may need updating upon implementation, as well. Notification of implemented changes to the initiator of the change request/problem report and to the users of the system is also handled as part of the configuration control process. A modified system acceptance test, including the users of the system, should be performed before such changes are implemented. The test plan, at Exhibit 5-2, is a dynamic document used for directing the testing of the system throughout the life cycle.

1. Introduction

a. Purpose
b. Background
c. Scope
d. Project References
e. Glossary

2. Limitations and Traceability

a. Limitations
b. Traceability

3. Test Plans

a. Test Levels
   (1) Unit/Module Test
   (2) Subsystem Integration Test
   (3) System Qualification Test
   (4) System Acceptance Test
   (5) Security Test

b. Test Environment and Schedules

c. Test Methods and Evaluation

4. Test Descriptions

a. Test Name (repeat for each test)
   (1) Test Description
   (2) Control
   (3) Inputs
   (4) Outputs
   (5) Procedures

Exhibit 5-2: Test Plan Outline

1. **Introduction**

Provide an overview and background on the application(s) system being tested. Do not provide information at this level of heading, if the description provided in the subsections is sufficient.

a. **Purpose**. Identify the application system being tested by name and describe the purpose of the test plan. Include a summary of the functions of the system and the tests to be performed.

b. **Background**. Provide a brief description of the history and other background leading up to the system development process. Identify user organization and location where the testing will be performed. Describe any prior testing and note results that may affect this testing.

c. **Scope**. Describe the boundaries of the planned tests. Include a summary of any constraints imposed on the testing whether they are due to a lack of specialized test equipment, or constraints on time or resources. Describe constraints in greater detail in the "Limitations" section below.

d. **Project References**. List publications and documents referenced in the test plan. Include any referenced standards the application is expected to meet and also describe the application being tested in greater detail. The referenced documents are listed as project documents, Government documents, and non-Government documents. Add additional headings for clarity if needed.

(1) **Project Documents**. List the project documents that were developed prior to the test plan. Examples may include the project plan, acquisition plan, cost-benefit analysis, functional requirements document, etc. In addition to project documents, documents from related projects may also be referenced.

(2) **Government Documents**. List any applicable Government standards, policies, directives, circulars, and other documents that apply to the test plan and the conduct of testing.

(3) **Non-Government Documents**. List any non-Government documents, such as industry standards or reference documents, that

may apply to the test plan and to the conduct of testing.

e. **Glossary**. Provide a list of all terms and abbreviations used in this document. If the list is several pages in length, it may be placed as an appendix.

2. **Limitations and Traceability**

Elaborate on the limitations of the constraints as summarized in the Scope required above. Cross-reference the functional requirements and detailed specifications to the tests that demonstrate or partially demonstrate that capability.

a. **Limitations**. Describe constraints imposed on the testing, whether they are due to lack of specialized test equipment or constraints on time or resources. Indicate what steps, if any, are being taken to reduce the program risk due to the test limitation(s).

b. **Traceability**. Enhance the traceability matrix created in the FRD to include testing activities which address user requirements. This matrix begins with the user requirements and assists in tracing how the requirements are addressed in subsequent phases and documents, including the external/internal design documents and test plans.

The matrix may be broken up into segments, if appropriate. For example, a separate matrix of the test plan paragraphs that trace to particular paragraphs in the internal design document in the design phase may be provided. The intent is to show that the test design tests all of the functionality, performance, and other requirements associated with each design element (unit, module, subsystem, and system) in the internal design document.

When a test supports a particular requirement, the relationship is noted at their intersection in the matrix. The listed requirements may be explicitly stated, or may be derived or implicit. All explicit requirements must be included. The granularity of the list should be detailed enough that each requirement is simple and testable.

3. **Test Plans**

Describe the levels of tests that take place during development: unit, integration, and system level tests, and the planning that is needed. The test environment is described in terms of milestones, schedules, and resources needed to support testing.

a. **Test Levels**. In the test plan, list the various types of software testing below.

(1) **Unit/Module Test**. This level of testing is generally conducted by the individual who develops the code. The test validates the module's logic, adherence to functional requirements and adherence to technical specifications. The goal of this test is to ensure that all module source statements have been executed and each conditional branch taken. The test and their results are recorded in the software development folder for that module. The folder includes the source code for the module, a clean compiled listing of the code, a list of any outstanding issues or problems in the design or the program discovered during the supervised module desk check review, the test data sets, copies of any output, and the rationale for design decisions. The software development folders containing the test data may be delivered in the final test plan or in an appendix.

(2) **Subsystem Integration Test**. This test examines the subsystems that are made up of integrated groupings of software units and modules. This level of testing is generally performed by the developing organization and is generally conducted in the development environment. It is also generally the first level of testing where problem reports are generated, classified by severity, and their resolution monitored and reported. The subsystem integration test results, including the test data sets and outputs produced from the tests may be delivered as part of the final test plan, with the integration test analysis report, or in an appendix.

(3) **System Qualification Test**. This independent test determines whether the system complies with standards and satisfies functional and technical requirements when executed on target hardware using operational data files and user prepared test data. System documents and training manuals are tested for accuracy, validity, completeness, and useability. During this test, the software performance, response time, and ability to operate under stressed conditions will be tested. External system interfaces are also tested. All findings are recorded in a system qualification test analysis report.

(4) **System Acceptance Test**. This test is performed in an environment which "mirrors" the environment in which the system will be fielded. The environment is a non-production environment. Every feature of the system may be tested for correctness and conformance to functional requirements. System interoperability, all documentation, system reliability, and the level to which the system meets user requirements will be evaluated. Performance tests may be executed to ensure that screen response time, program run time, operator intervention requirements and overall system operations meet user requirements. Recovery and restart procedures may be evaluated. Interfaces to other applications should be tested here as well. The tests results, including the test data sets and outputs produced from the tests may be delivered as part of the final test plan, with the system acceptance test analysis report, or in an appendix.

(5) **Security Test**. This test is performed in the operational (production) environment under the guidance of the security staff. This test evaluates compliance with system security and integrity guidelines. System backup, recovery, security, audit trails, and reconciliation issues are addressed. Include internal controls or application security features mentioned in the context of security testing.

b. **Test Environment and Schedules**. Document key elements of the test environment, including milestones, schedule, and resource requirements in this section.

(1) **Software Description**. Provide a brief description of the inputs, outputs, and functions of the software being tested.

(2) **Milestones**. List the milestone events, and dates for the testing.

(3) **Organizations and Locations**. Identify the participating organizations and the location where the software will be tested.

(4) **Schedule**. Show the detailed schedule of dates and events for the testing by location. Such events may include familiarization, training, test data set generation and collection, as well as the volume and frequency of the input for testing.

(5) **Resource Requirements**. State the resource requirements including the following:

- o **Equipment** - show the expected period of use, types, and quantities of equipment needed.

- o **Software** - list other software that will be needed to support the testing that is not part of the software to be tested. This can include debugging software and programming aid, as well as many current programs to be run in parallel with the new software to ensure accuracy; any drivers or system software to be used in conjunction with the new software to ensure compatibility and integration; and any software required to operate the equipment and record test results.

- o **Personnel -** list the numbers, skill types, and schedules for personnel that are expected during the test from both the user, database, quality assurance, security and development groups. Include any special requirements such as multi-shift operation or key personnel.

(6) **Testing Material**. List the materials needed for the test, such as documentation, software to be tested and its medium, test inputs, sample outputs, test control software, and worksheets.

(7) **Test Training**. Describe or reference the plan for providing training in the use of the software being tested. Specify the types of

training, personnel to be trained, and the training staff.

c. **Test Methods and Evaluation**. Document the test methodologies, conditions, test progression or sequencing, data recording, constraints, criteria, and data reduction in this section.

(1) **Methodology**. Describe the general methodology or strategy of the testing for each type of testing described in this test plan.

(2) **Conditions**. Specify the type of input to be used, such as real-time entered test data, canned data for batch runs, and so on. Describe the volume and frequency of the input, such as the number of transaction per second tested and so on. Sufficient volumes of test transactions should be used to simulate live stress testing at each level, and to incorporate a wide range of valid and invalid conditions. Data values used should simulate live data and also test limited conditions.

(3) **Test Progression**. Describe the manner in which progression is made from one test to another so that the entire cycle is completed.

(4) **Data Recording**. Describe the method to be used for recording the test results and other information about the testing.

(5) **Constraints**. Indicate anticipated limitations on the test due to test conditions, such as interfaces, equipment, personnel, and databases.

(6) **Criteria**. Describe the rules to be used to evaluate test results, such as range of data values used, combinations of input types used, maximum number of allowable interrupts or halts.

(7) **Data Reduction**. Describe the techniques to be used for manipulating the test data into a form suitable for evaluation, such as manual or automated methods, to allow comparison of the results that should be produced to those that are produced.

4. Test Description

Describe each test to be performed in this section. Tests at each level should include verification of access control and system standards, data security, functionality, and error processing. As various levels of testing (unit/module, subsystem integration, system qualification, system acceptance testing, and security) are completed and the test results are documented, revisions or increments of the test plan can be delivered. **The subsections of this section should be repeated for each test within the project**. If there is a large number of tests, they can be placed as an appendix.

a. **Test Name**. Identify the test to be performed for the named unit, module, subsystem or system. Address the following for each test:

   (1) **Test Description**. Describe the test to be performed. Tests at each level of testing should include those designed to verify data security, access control and system standards, system/subsystem/unit functionality, and error processing as required.

   (2) **Control**. Describe the test control, such as manual, semi-automatic, or automatic insertion of inputs, sequencing of operations, and recording of results.

(3) **Inputs**. Describe the data input commands used during the test. Provide listings of input data. At the discretion of the bureau software manager, input data listings may also be requested in computer readable form - for possible future use in regression testing.

(4) **Outputs**. Describe the output data expected as a result of the test and any intermediate messages or display screens that may be produced.

(5) **Procedures**. Specify the step-by-step procedures to accomplish the test. Include test setup, initialization, steps and termination. Also include effectiveness criteria or pass criteria for each test procedure.

G3. DOCUMENTATION - QUALITY ASSURANCE PLAN

The purpose of the quality assurance plan is to indicate the means by which the quality assurance (QA) requirements established for a system development effort will be met. The degree of QA applied and the level of detail contained in this plan should be appropriately tailored to and consistent with the complexity, size, intended use, mission-criticality, and cost of failure of the system development effort. An outline for the QA plan is provided at Exhibit 5-3.

1. General

a. **Purpose**. Identify the specific purpose and scope of the quality assurance (QA) plan. Name and briefly describe the project addressed by the QA plan.

b. **Reference**. List the deliverables that will be used in the QA review.

c. **Objective**. Discuss the system objectives based on the quality requirements established by the project manager and project user. Also discuss the benefits to be realized by meeting the quality requirements.

1. General

- a. Purpose
- b. Reference
- c. Objective
- d. Management
- e. Documentation

2. Reviews and Audits

3. Test and Evaluation

4. Problem Reporting and Corrective Action

5. Tools, Techniques, and Methodologies

Exhibit 5-3: Quality Assurance Plan Outline

d. **Management**. Provide an operational organizational chart from a QA perspective. Describe the tasks covered by the plan with emphasis on QA events selected for the project. Identify units responsible for each project task.

e. **Documentation**. Identify system documentation and the way these documents are to be checked for adequacy. Include the review or audit function involved.

2. **Reviews and Audits**

Describe the procedures for preparation and execution of reviews and audits for establishing the traceability of requirements identified for the project. State what will be certified and provided to management for approval.

3. **Test and Evaluation**

Identify which tasks will be performed; the system to be used; the data to be used; its level of testing; a schedule; and responsibilities for testing and evaluating results.

4. **Problem Reporting and Corrective Action**

Describe procedures for reporting, prioritizing, tracking, and resolving problems that result from reviews, audits, and tests.

5. **Tools, Techniques and Methodologies**

Identify the automated tools, techniques, and methodologies to be used to support QA.

G4. DOCUMENTATION - SYSTEM SECURITY REQUIREMENTS

For each new information system under development containing sensitive information a Computer Security Plan must be prepared. This is the initial document addressing overall security requirements. The Plan is updated through the testing phase and the supporting documentation such as risk assessment plan, contingency planning, recertification, etc., is covered in the proceeding phases. The Plan should be signed by the functional system owner.

The Computer Security Plan, Exhibit 5-4, describes responsibilities, requirements and guidelines for the development, maintenance and submission of inventories, plans, and reports related to Treasury's sensitive but unclassified Federal Information Processing (FIP) systems and communications security program. Address security requirements in each phase of the life cycle and revise them accordingly.

Bureaus are required by governmentwide regulations and Treasury policies to protect their automated information systems resources from potential threats by identifying safeguards through vulnerability and risk assessments. The safeguards become the security requirements and considerations that are associated with the development and implementation of new or enhanced software, hardware, equipment, policy or other FIP technology or products. The development of security safeguards is an integral part of the systems development methodology and is inherent in each phase of the life cycle.

Life cycle security refers to steps taken to ensure that a system is planned, designed, developed, and maintained according to requirements using formalized and rigorous security policies, controls, and standards. However, inadequacies in the planning and development of computer systems/applications are a frequent source of harmful effects, and in most cases, the effort to improve security is concentrated on the application system. System developers often think to improve functionality first, and delay security issues later. However, many opportunities to include effective controls are lost if security is not considered early.

Security decisions must be an integral part of the planning, development, and operation of a computer system/application. Systems must be carefully evaluated and tested during each life cycle phase and reevaluated whenever changes are made that could affect the protection mechanisms. In this way, confidence can be provided that the systems development interpretation of security policies is maintained accurately and without distortion.

The information system life cycle specifies a sequence of reviews, evaluations, walkthroughs and operational tests which guarantee that the requirements definition, design, development, implementation, and operations/maintenance of a system are accomplished under rigorous controls and standards. The sequential nature of the life cycle provides an iterative process in which the security considerations in a previous phase frequently drive the security considerations in the next phase.

DEPARTMENT OF THE TREASURY
BUREAU:
COMPUTER SECURITY PLAN

1. **System Name/Acronym**:

2. **System Description/Purpose/Environment**:

3. **System Components**: For large application systems, list the modules or subsystems that make up the larger system. (Attach additional page if necessary.)

4. **System Status**:
___ Operational
___ Under Development/Acquisition
Operational By:_________

5. **System Category**:
___ Application
___ Hardware Installation/Facility/Processing Center
___ Telecommunications Network - e.g., CDN, WAN, LAN
___ Microcomputer System(s)
___ Other (e.g. specialized system unique to bureau)

6. **Type of Sensitive Data Handled by This System**: (Check ALL that apply)

__ Financial __ Law Enforcement
__ Tax __ Privacy Act
__ Payroll __ Personnel
__ Internal Administrative __ Procurement/Contract
__ Officially Limited Information (e.g., LOU)
__ Other: Specify ________________________________

Exhibit 5-4: Computer Security Plan

7. **System Operator**: For hardware installations and telecommunications networks, indicate whether the system is operated by:

__ Bureau personnel __ Other Treasury agency
__ Other Gov't agency __ Contractor

8. **Network access**: Is this system linked to: (Check ALL that apply)
__ Consolidated Data Network (CDN)
__ Bureau-limited Network
__ Other Government Network(s): Specify __________________
__ Other Commercial Network(s): Specify __________________

9. **Electronic Data Interchange (EDI) capability**:___ Yes
___ No

10. **Schedule of Risk Assessments and Reviews**. To enable information to be obtained from the Management Control Plan (MCP) in your annual FMFIA/Internal Control Report as required under OMB Circular A-123, specify the MCP COMPONENT (OR ASSESSABLE UNIT) NUMBER of this system. ________________

11. **Were any material security or control weaknesses identified during the last audit, review, or risk assessment of this system**?

__ No __ Yes Control number (year of identification and sequential number) of weakness as reported in last annual FMFIA/Internal Control Report
____

12. **Date of last certification to the adequacy of controls and safeguards of this system (in accordance with OMB A-130 and TD P 71-10, Chapter VI, Section 4.B)**.
____

13. **At a minimum, does this system comply with the Treasury <u>Security Manual</u>, TD P 71-10, Chapter IV, Section 4.B.1, requirement for Controlled Access Protection (C2 Level)**?

____ Yes - Date of compliance: _______________
____ No - Date of planned compliance: _______________
and, if applicable, date waiver granted: ________________

Exhibit 5-4: Computer Security Plan (cont'd)

14. **Designated point of contact for further information regarding this security plan and system**:

NAME: ___________________________ PHONE:__________________

OFFICE/DIVISION/BRANCH: __________________________________

15. **The following supporting documentation for this security plan should be available**:

a. Most recent Risk Assessment report that reflects, at a minimum, that Federal and Treasury baseline security requirements have been considered in determining the security and control requirements for this system;

b. Contingency Plan or Disaster Recovery and Continuity of Operations Plan for this system; and

c. Latest signed certification and/or recertification to the adequacy of controls in this system.

NOTE: IF ACCESS TO OR DISTRIBUTION OF THIS SECURITY PLAN AND ACCOMPANYING DOCUMENTATION IS RESTRICTED, THE APPROPRIATE OFFICIALLY LIMITING LEGEND SHOULD BE AFFIXED IN ACCORDANCE WITH TD P 71-10, Chapter VI, Section 2, "LIMITED OFFICIAL USE INFORMATION AND OTHER LEGENDS."

_________________________________ _________

Signature **Date**

____

Title

Exhibit 5-4: Computer Security Plan (cont'd)

Include the following information when reporting the annual Computer Security Plan requirements.

Sensitive System Inventories and Risk Management Plans

Each bureau must submit a sensitive system inventory and report completed and scheduled risk assessments and security reviews as part of the Management Control Plan (MCP) included in the bureau's annual Federal Managers Financial Integrity Act (FMFIA) report to the Treasury. This report is due on November 1 each year in accordance with guidance issued by Treasury's Office of the Comptroller, and will be forwarded in automated form using the Department's Inventory Tracking and Closure (ITC) System. The sensitive system inventory shall list, by system name, each sensitive FIP system in operation or under development, including telecommunication and application systems and facilities.

Material Security and Control Weaknesses

Bureaus must continue to report, in the annual FMFIA report to the Treasury, material security or control weaknesses identified through risk assessments, audits, or security reviews of the sensitive systems identified in their MCP inventories as described above.

Computer Security Plans and Centralized Plan Review and Approval

Each bureau must develop and maintain a Computer Security Plan (to be updated at the conclusion of each risk assessment or other significant system change) that contains all of the elements listed in the format provided for each sensitive system reported in the MCP inventory as described above.

Each bureau must establish a bureau Security Plan Review, Approval, and Coordination function to review and approve each new and updated security plan developed within the bureau, provide independent advice and comment to bureau security plan developers, and coordinate requests from the Department and other Federal regulatory agencies for information on, or submission of, security plans and supporting documentation.

Computer Security Awareness and Training Plans

Each bureau must develop, maintain, and annually update a Treasury Computer Security Awareness and Training Plan in accordance with the Treasury Security Manual, TD P 71-10, Chapter VI, Section 8.A, "Awareness and Training Program for Telecommunications and Automated Information Systems (AIS) Security."

Additional Reporting and Coordination Requirements

A. Report, in annual information systems plans:

1. Summary information regarding on-going computer security awareness training programs, including number and types of government and contractor personnel who received training, types of training, and benefits derived from the training during the past year; and

2. Summary information regarding overall progress in implementing Treasury information systems security policies and development of security plans, and specific information regarding systems that may have been granted temporary exemptions from security policy requirements.

B. Upon request, make security plans and training plans available to the Department or other Federal regulatory agencies for review.

C. Ensure that information reported in inventories, plans, and reports required by this guideline is coordinated and consistent with information systems security-related plans and synopses reported in the annual Triennial IRM Review Plan.

G5. DOCUMENTATION - RISK ASSESSMENT PLAN

TD P 85-03, Risk Assessment Guideline, Volumes 1 and 2, describes the procedures to conduct a risk assessment of information systems and to prepare a risk assessment plan. The system security and integrity requirements worksheets are included in the Guidelines. Risk assessment is synonymous with risk analysis. It refers to the evaluation that is performed of computer systems and applications to determine the degree of risk to computer systems resources posed by threats and vulnerabilities. Risk assessments are performed of systems and applications under development to determine the security measures that need to be designed into the system or application. Risk assessments are performed of implemented systems and applications as part of the certification and accreditation processes to verify that there is an acceptable level of risk for the data processed.

A risk assessment plan is prepared to determine the level of sensitivity and risk associated with the system and its data. The basic goal is to provide required security commensurate with the perceived risks of the system as related to its data and processing environment. The required level of controls and security become an integral part of the design specification, and provide integrity assurance to system management. The security and controls implemented will therefore be practical, cost-effective, and based on the specific needs of the application, rather than on some arbitrary edict or the judgement of the application designer. In the event that there are any privacy issues that are separate from security issues, those issues can be addressed in the risk assessment.

REQUIREMENTS DEFINITION ISSUES CHECKLIST

The project team uses this checklist to ensure specific issues have been addressed. Consideration is given to the questions below for each systems project. Action may not be required for some items.

Functional Requirements Document

___ Has a Functional Requirements Document been prepared to provide basic understanding between users and designers of the system?

Does the document include:

___ A statement of objectives to be met by the new system?

___ A description of existing methods and procedures?

___ A description of proposed methods and procedures?

___ The performance (functional) requirements of the new system?

___ A description of inputs and outputs?

___ A description of data elements, dictionaries, tables, and reference files?

___ A description of the equipment needed to process the system?

___ A description of system software needed to support the system?

___ A description of interfaces with other systems?

___ A description of the system sizing and capacity requirements (mainframe, LAN, WAN)?

Exhibit 5-5: Requirements Definition Issues Checklist

REQUIREMENTS DEFINITION ISSUES CHECKLIST

Functional Requirements Document (cont'd)

___ The system response, reliability and growth requirements?

___ A requirements traceability matrix?

Test Plan

___ Has a plan been developed to test the computer-based system?

___ Does it include the detailed specifications, descriptions and procedures for all tests?

___ Does it include test data reduction and evaluation criteria?

___ Are all computer programs desk checked by the programmer and supervisor before program assembly or compilation?

___ Are all computer programs reviewed after assembly or compilation to insure that errors disclosed by these translator routines are corrected?

___ Is dummy test data, as opposed to live data, used to test computer programs?

___ Is each program, subsystem, and the entire system, tested?

___ Is test data treated just like live data?

___ Are sufficient volumes of test transactions entered which have a wide range of valid and invalid conditions?

___ Is sufficient time allocated for thorough testing?

Exhibit 5-5: Requirements Definition Issues Checklist (cont'd)

REQUIREMENTS DEFINITION ISSUES CHECKLIST

Test Plan (cont'd)

___ Have resources been allocated for testing purposes?

___ Are programming aid software packages used to improve computer programs efficiency and effectiveness?

___ Are new programs run parallel to old ones to help ensure their accuracy?

___ Does the system acceptance process evaluate whether the entire system is performing in accordance with system specifications and processing standards?

___ Is system acceptance performed by individuals independent of the analysis, design, and development of the system?

___ Once system acceptance has been completed, is a written certification that the system performs in accordance with all functional and performance specifications required?

___ Are system acceptance transactions treated just like live transactions?

___ Are sufficient resources allowed for acceptance testing?

___ Are program modifications thoroughly tested to make sure the modifications functions properly?

___ Are program modifications subject to system acceptance before being placed in operations?

___ Are offices that initiate changes in master files or program instructions furnished with a notice or other documentation showing changes actually made?

___ Do users make the final decision on whether the modification meets their needs?

Exhibit 5-5: Requirements Definition Issues Checklist (cont'd)
REQUIREMENTS DEFINITION ISSUES CHECKLIST

Test Plan (cont'd)

___ Is documentation changed to reflect the modifications?

___ Does the system acceptance process evaluate both manual and automated procedures?

Risk Assessment Plan

__ Has a system security and integrity requirements worksheet been prepared? If so, does it include:

___ A description of contingency steps to be taken in the event of hardware/software failures?

___ A description of internal controls over and within the system?

___ The system's security and privacy requirements?

___ A description of system integrity and quality assurance requirements?

Exhibit 5-5: Requirements Definition Issues Checklist (cont'd)

FUNCTIONAL REQUIREMENTS REVIEW CHECKLIST

1. Date/time of meeting _____________.

2. Attendants:

Project Manager	_______________
Project Sponsor	_______________
Project User	_______________
Quality Assurance Manager	_______________
Configuration Manager	_______________
Data Administrator	_______________
Security Manager	_______________
Procurement Manager	_______________
FOIA/PA Representative	_______________

3. Were the following present for the review or their absence justified?

Functional Requirements Document	____________
Test Plan	____________
Quality Assurance Plan	____________
Risk Assessment Plan	____________
Computer Security Plan	____________
Updated Project Plan	____________
Updated Data Management Plan	____________
Updated Configuration Management Plan	____________
Updated Cost-Benefit Analysis	____________
Updated Privacy Act Federal Register Notice	____________
Updated Acquisition Plan	____________
Final Feasibility Study	____________

4. Were the system's functions completely defined? (Yes/No) If no, explain.

Exhibit 5-6: Functional Requirements Review Checklist
FUNCTIONAL REQUIREMENTS REVIEW CHECKLIST

5. Were the functions sufficiently isolated in the requirements document(s) to make their traceability to the design document feasible? (Yes/No) If no, explain.

6. Were the plans as presented found to be:

Complete ____	Consistent ____	Unambiguous ____
Accurate ____	Feasible ____	Testable ____

If not, explain.

7. Was the cost-benefit analysis present? (Yes/No) If yes, was the analysis acceptable (Yes/No)? If no, explain.

Exhibit 5-6: Functional Requirements Review Checklist (cont'd)

CHAPTER 6. DESIGN PHASE

A. GENERAL

The purpose of the **design phase** is to transform the detailed requirements definition into a complete, detailed specification for the system. The analyses of this phase are performed within the framework of the system initiative, converting the functional and data requirements into a complete system design which will guide the work of the development phase. The decisions made in this phase address in detail how the system will meet the previously defined functional and data requirements. The **design phase** activities may be conducted in an iterative fashion, producing first an external (general) design that emphasizes the functional features of the system, and then an internal (detailed) design that expands the design by providing all the technical detail.

In the **external design of the system**, the external characteristics of the system are defined. The operating system is established and the automated system is packaged into major design subsystems. The inputs and outputs of each subsystem are defined, the interfaces to external systems are designed, and the administrative activities are established. Security and auditing needs are addressed.

In the **internal design of the system**, the detailed structure of the system from the subsystems identified in the external design document is created. Each subsystem is partitioned into one or more design units or modules. For each design unit or module, the process is described in a structure chart, flowchart, pseudocode, or some other acceptable format. Detailed logic specifications are then written for each module described and data usage is physically defined to the elemental level. Everything requiring user input and approval is completed in this activity.

The decisions of this phase reexamine in greater detail many of the parameters of the system concept addressed in the previous phases. The decisions made in this phase also address in detail how the system will meet the previously defined functional and data requirements. The system design prepared in this phase will be the basis for the activities of the development phase. The overall objective is to establish a complete design for the system. The prerequisites for this phase are the project plan, the functional requirements document, and the test plan.

A number of **project approach, project execution and project continuation decisions** are made in this phase. **Project approach decisions** include: what specific methodologies and tools will be used in the development and implementation phases, how will user support be provided, how will development and implementation be phased, and how will newly identified risks and issues be handled?

Project execution decisions include: what modifications must be made to the initial information system requirement, what modifications will be made to current procedures, what modifications will be made to current systems/databases or to other systems/databases under development, and how will conversion of existing data take place?

Project continuation decisions include: does the information system requirement continue to exist, does the system design address the requirement sufficiently to continue the development activities, and are sufficient funding and other resources available for the remainder of the life cycle.

B. TASKS AND ACTIVITIES

The following activities are performed in this phase. These activities may be expanded or deleted depending on the size of the proposed system.

- o Design and document technical environment and system architecture
- o Design and document system interfaces
- o Document communications requirements
- o Design and document inputs, input processing
- o Design and document physical database(s)
- o Select software packages
- o Load physical database design into design data dictionary
- o Define and document manual procedures
- o Design and document conversion activities
- o Define and document implementation activities
- o Prepare external design document
- o Design and document report and internal processing
- o Design and document report and other data access capabilities
- o Prepare internal design document
- o Perform design reviews
- o Perform design audits
- o Prepare contingency plan
- o Continue configuration accounting and change control
- o Continue procurement activities
- o Expand system acceptance test documents
- o Update data management plan
- o Update the project plan

- o Update cost-benefit analysis
- o Finalize functional requirements document
- o Obtain approval of the design phase

C. ROLES AND RESPONSIBILITIES

Project members participating in this phase in addition the project manager and project user may include the programmer analyst, data administrator, data base administrator, quality assurance manager, security manager, configuration manager/configuration control board, procurement manager, system developer, system maintenance manager, telecommunications manager, capacity planning manager, procurement manager and systems operations manager.

D. DELIVERABLES, RESPONSIBILITIES, AND ACTION

The deliverables listed below must be available for review at the end of this phase. The deliverables may be expanded or abbreviated depending on the size, scope and complexity of the systems development effort. The external/internal design documents; operators, user, and maintenance manuals; and training, conversion, implementation, and contingency plans are described in the documentation section in this chapter. Revise previous documents as appropriate.

DELIVERABLES AND PRODUCTS	PRIME RESPONSIBILITY
External Design Document (Exhibit 6-1)	Project Manager
Internal Design Document (Exhibit 6-2)	Programmer Analyst
Operators Manual (Exhibit 6-3)	Systems Operations Manager
User Manual (Exhibit 6-4)	Project User
Training Plan (Exhibit 6-5)	Project User
Maintenance Manual (Exhibit 6-6)	Systems Operations Manager
Conversion Plan (Exhibit 6-7)	Project Manager
Implementation Plan (Exhibit 6-8)	Project Manager
Contingency Plan	Security Manager/Project User
Project Plan (revised)	Project Manager
Data Management Plan (revised)	Data Administrator
Cost-Benefit Analysis (revised)	Project Manager
Configuration Management Plan (revised)	Configuration Manager
Quality Assurance Plan (revised)	Quality Assurance Manager

Risk Assessment Plan (revised)	Security Manager
Computer Security Plan (revised)	Security Manager
Test Plan (revised)	Security Manager
Acquisition Plan (revised)	Procurement Manager
Privacy Act Federal Register Notice (final)	FOIA/PA Representative
Functional Requirements Document (final)	System Developer/Project Team

=================================

MANAGEMENT ACTION	**PRIME RESPONSIBILITY**
Reviews and approves deliverables	Project Manager/Project Sponsor
Reviews project at end of phase	Steering Committee
Conducts mid-project review	Steering Committee
Obtains end-of-phase concurrence from user	Project Manager
Authorizes Development Phase to begin	Steering Committee

E. ISSUES FOR CONSIDERATIONS

During the **design phase** a number of specific external and internal design issues are addressed. These issues are provided in the Design Issues Checklist in Exhibit 6-9. The checklist is used as a reference for the project manager and system developer to ensure objectives of this phase are met. These issues are independent of the project size and effort. The checklist is also for use in conducting the reviews in Section F of this phase.

F. REVIEW ACTIVITY

Three reviews are conducted during the **design phase**. The Software Requirements Review validates that the functional requirements are technically feasible. The Software Requirements Review Checklist at Exhibit 6-10 is provided to ensure review goals are met. Technical approaches to satisfy the functional requirements will be discussed. Detailed objective system functions, performance requirements, security requirements, and system platform characteristics will be presented by the project team.

The Preliminary Design Review is an ongoing review of the system design process as it evolves through the **design phase**. The Final Design Review confirms that modifications prompted by earlier reviews are incorporated. The Final Design Review Checklist at Exhibit 6-11 is provided to ensure review goals are met.

Participants in the preliminary design and final design reviews include the project manager, project user, and the system developer. Other project team members may be requested to participate in the reviews. Depending on the size and complexity of the project, the reviews may be conducted by a review committee.

G1. DOCUMENTATION - EXTERNAL DESIGN DOCUMENT

The external design document describes the requirements, operating environment and design characteristics for an information system. The external design document is used in conjunction with the functional requirements document to provide a complete system specification of all user requirements for the system. The external design document includes all information required for the review and approval of the project development during the external design as outlined in Exhibit 6-1.

1. **Introduction**

   a. **Purpose and Scope**. Provide a description of the external design document purpose and scope in this section.

   b. **Project Executive Summary**. Provide a description of the project from a management perspective. Provide an overview of the framework within which the high-level system design took place. Include the following information in the summary if appropriate.

      (1) **System Overview**. Describe the system in narrative form using non-technical terms. Provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should show interfaces to external systems, if applicable. Provide a high-level data flow diagram for the system and subsystems, if applicable.

      (2) **Design Constraints**. Describe any constraints in the system design. Describe any assumptions made by the project team in developing the system design.

1. **Introduction**

    a. Purpose and Scope
    b. Project Executive Summary
        (1) System Overview
        (2) Design Constraints
        (3) Future Contingencies
    c. Organization of this Document
    d. Points of Contact
    e. Project References
    f. Glossary

2. **System and Subsystem Specification**

    a. System/Subsystem Overview
    b. Specification Model
        (1) Process Model
            (a) Manual Components
            (b) Automated Components
        (2) Logical Data Model
        (3) Physical Data Model
    c. External System Interfaces
    d. Data Communications Interfaces
    e. Data Control and Audits

3. **Procedural Requirements**

4. **Traceability**

Figure 6-1: External Design Document Outline

(3) **Future Contingencies**. Describe any contingencies that might arise in the design of the system.

c. **<u>Organization of this Document</u>**. Describe the organization of the external design document.

d. **<u>Points of Contact</u>**. Provide the organization code and title of the key points of contact (and alternates if appropriate) for the information system development effort. These points of contact should include the project manager, project sponsor and/or functional manager, project user, quality assurance manager, security manager, and configuration manager, as appropriate.

e. **<u>Project References</u>**. Provide a bibliography of key project references and deliverables which have been produced prior to this point. For example, these references might include the project plan, feasibility study, cost-benefit analysis, acquisition plan, quality assurance plan, configuration management plan, data management plan, and functional requirements document.

f. **<u>Glossary</u>**. Provide a glossary of all terms and abbreviations used in the document. If the glossary is several pages in length, it may be placed as an appendix.

2. **System and Subsystem Specifications**

Describe the system and subsystem specifications for the project in this section. Provide a summary of the intended capability of system and subsystems in terms of major components (specification model, manual components, automated components, interfaces, networks, and audits). Add as much detail as necessary to fully define all external specifications that will satisfy the system design objectives.

a. **<u>System and Subsystem Overview</u>**. This overview supplements the project executive summary with the following additional technical details.

o A narrative description of the preliminary design of the system identified in the project executive summary with a detailed description of each subsystem.

- o High-level diagrams of each subsystem with more detail than the diagrams in the project executive summary, if more detail is known at this point.

- o A matrix of requirements versus design components.

b. **Specification Model**. Describe the specification model in terms of a work flow model, logical data model, and physical data model, if appropriate. Include the following data.

(1) **Process Model**. Provide work flow or process flow diagrams that were used to define the sequence of work activities within the processes being automated. Provide the parts of the work flow or process flow that will remain as manual processes graphically, if possible. Provide the lowest level (greatest detail) of processes developed at this point. The manual and automated parts of the work flow or process flow are discussed in more details below.

If the work flow or process flows have been developed in detail in the functional requirements document, those sections may be referenced here. It may be appropriate to repeat the diagrams here.

(a) Manual Components. Describe the processes that will be accomplished manually. Include a discussion of the analyses performed to identify alternatives for manual components; a matrix of the manual components and the requirements addressed by them; and references to other documents which will offer greater detail on manual activities, such as procedures manual.

(b) Automated Components. Describe the automated support offered to the user, i.e., the parts of the work flows or process flows that are being automated. Include a discussion of the analyses performed to identify alternatives for software components; and a matrix of software components and the requirements addressed by the components.

(2) **Logical Data Model**. Provide detailed data flow diagrams, data element definitions for items on the data flow diagrams, input and output data flow diagrams, and control flow diagrams, if appropriate. Where appropriate, provide the following:

- o Data flow diagrams, which are at a greater level of detail (for the lower level, more detailed processes) than provided in the project executive summary. Provide data dictionary definitions for the data elements within the data flows. This may replicate some of the information contained in the functional requirements document.

- o Input and output designs - the data flow summaries and layouts for each automated system input and output declared on the subsystem level data flow diagrams.

- o Provide control flow diagrams if the system is an on-line transaction processing data base system or other system that allows system data flows and controls flows to be specified, structured, and manipulated separately.

- o Develop a logical data model for subsystems in accordance with the data administration guidelines.

- o For data base management systems, include additional logical diagrams; i.e., entity-relationship diagrams, table structures, etc.

(3) **Physical Data Model**. Describe the physical record and file structures. For conversion applications and maintenance of existing systems, a physical data model would exist. Identify the change to be made for implementation in the Internal Design Phase. If available, provide:

- o For DBMSs, a physical description of the DBMS schemas, subschemas, records, sets, tables, files, etc., for DBMSs. Provide physical size and storage requirements for both online and offline.

- o For non-DBMSs, the file organization (e.g., sequential, index sequential, and random access file structures). Provide the record layout, and projected file volume of data. Define the transfer medium (e.g., tape or disk).

c. **External System Interfaces**. Address the design of interfaces to external systems. Reference the external interface requirements sections of the functional requirements document. Also provide diagrams of the interfaces at the subsystem level; file, record, table, and data base schema structures involved in the interface of both the external system and the system under development; definitions for data elements involved in external interfaces; and access control features of the external system and the system under development.

d. **Data Communications Interfaces**. Provide a logical and physical specification of any interface to data communications networks. Provide diagrams of the interfaces to the network(s) when feasible. Describe or reference the hardware and software of the communications network itself.

e. **Data Control and Audits**. Describe the audit trails which will enable the user to track any one transaction from its original source to final storage in some given data base or file, and from output back to its original source. Describe the user data interfaces as they relate to the preservation of control.

3. **Procedural Requirements**

Address all system procedural requirements from a user's perspective. Describe how the data is entered into the automated system; describe how file recovery or reconstruction will take place; and address full backup procedures in case of a total or partial system failure or central shutdown which should reference the contingency plan.

4. **Traceability**

Enhance the traceability matrix created in the FRD to include features from the external design which address user requirements. This matrix begins with the user requirements and assists in tracing how the requirements are addressed in subsequent phases and documents, such as the external design document, internal design document and test plans.

G2. DOCUMENTATION - INTERNAL DESIGN DOCUMENT

The internal design document describes the detailed system and subsystem designs that will be used in developing the information system. It contains the database design, file structures, input formats, output layouts, and module processing logic to be used by the project team during system development. The sections and subsections of the internal design document may be organized, rearranged, or repeated as necessary to reflect the best organization for a particular project. The internal design document is outlined in Exhibit 6-2.

1. Introduction

a. **Purpose and Scope**. Provide a description of the internal design document purpose and scope in this section.

b. **Organization of this Document**. Describe the organization of the internal design document.

c. **Points of Contact**. Provide the organization and title of the key points of contact (and alternates if appropriate) for the information system development effort. These points of contact should include the project manager, project sponsor, system developer, programmer analyst, quality assurance manager, security manager, configuration manager and other points of contact as appropriate.

d. **Project References**. Provide a bibliography of key project references and deliverables which have been produced prior to this point. For example, these references might include the project plan, feasibility study, cost benefit analysis, acquisition plan, quality assurance plan, configuration management plan, functional requirements document, data management plan, computer security plan, and external design document.

1. **Introduction**

    a. Purpose and Scope
    b. Organization of this Document
    c. Points of Contact
    d. Project References
    e. Glossary

2. **System Design Overview**

    a. System Architecture
    b. Subsystem Architecture

3. **Unit Design Organization**

4. **File and Data Base Design**

    a. Data Base Management System Files
    b. Non-Data Base Management System Files

5. **Input and Output Design**

    a. System Input Design
    b. System Output Design

6. **Detailed Module Design**

7. **Traceability**

Exhibit 6-2: Internal Design Document Outline

e. **Glossary**. Provide a glossary of all terms and abbreviations used in the document. If the glossary is several pages in length, it may be placed as an appendix.

2. **System Design Overview**

Describe the system and subsystem architectures and their design specifications in the this section.

a. **System Architecture**. Describe the system architecture and structure of the automated system. The overview information in this section may partially repeat some of the content of the external design document (e.g., the system description, the flow diagrams). Include the following in this section:

- o A narrative description of the system. Describe all input and output of the system.

- o A high-level system schematic to illustrate the overall architecture. Provide the breakout of subsystems and the interfaces between subsystems within the system in this diagram or another diagram, as appropriate. Show interfaces to external systems and telecommunications systems in the system architecture.

- o A high-level process flow, data flow and control flow diagrams of flows that occur within the system architecture. The flow diagrams should trace the system operations from initial data input through final output.

- o A system data dictionary that is complete up to this point in the system development process.

b. **Subsystem Architecture**. Define the next lower-level of design specification for each subsystem within the system. Include the following in this section:

- o A diagram of each subsystem. Provide a diagram showing the breakout of design units within each subsystem, either on the subsystem diagram or on another diagram.

- A narrative description of each subsystem. Describe all input and output of the subsystem.

- The process flow, data flow, and control flow diagrams for flows that occur within the subsystem. The diagrams should trace the system operations from initial data input through final output.

3. **Unit Design Organization**

Describe the segmentation of the system into subsystems (this section may reference the external design document); segmentation of the subsystems into design units (a subsystem may map to one or more than one design unit per subsystem); and the segmentation of design units into design modules. Show the relationship between design modules and the projected actual computer program compilation units graphically or in tables. There may be one or more than one design module per program compilation unit depending on the software design approach and the computer languages used. The degree and type of modularity above may be modified as necessary for the project under development.

Use structured organization diagrams which show the various segmentation levels down to the lowest level. All features on the diagrams should have reference numbers and names. Include a narrative which expands upon and enhances the understanding of the functional breakdown.

4. **File and Data Base Design**

Define the final design of all DBMS files and the non-DBMS files associated with the system under development in this section. Additional information may added as required for the particular project.

a. **Data Base Management System Files**. Define the final design of the DBMS files and include the following information:

   - The final logical design. Depending on the DBMS, third or fourth normal form table layouts, entity-relationship diagrams of the data base structure, or other logical design information should be provided.

- o A physical description of the DBMS schemas, subschemas, records, sets, tables, storage page sizes, etc.

- o Access methods (e.g., indexed, via set, sequential, random access, sorted pointer array, etc.).

- o Estimate of the DBMS file size or volume of data within the file, and data pages, including overhead resulting from access methods and free space.

- o Definition of the update frequency of the data base tables, views, files, areas, records, sets, data pages, etc. Estimate the number of transactions if the data base is an on-line transaction based systems.

b. **Non-Data Base Management System Files**. Provide the detailed description of all non-DBMS files and include the following information:

- o A narrative description of the usage of each file; including whether the file is used for input, output, or both; whether this file is a temporary file; an indication of which modules read and write the file, etc.; and

- o File structures. Include information that will:

  - identify record structures;

  - identify record keys or indexes;

  - identify or reference data elements within the records;

  - define record length (fixed or maximum variable length);

  - define blocking factors;

  - define file access method. For example, index sequential, virtual sequential, random access, etc.;

- estimate the file size or volume of data within the file, including overhead resulting from file access methods; and

- define the update frequency of the file. If the file is part of an on-line transaction based system, provide the estimated number of transactions per unit time, and the statistical mean, mode and distribution of those transactions.

5. **Input and Output Design**

Provide the detailed design of the system and subsystem inputs and outputs. Any additional information may be added to this section and may be organized according to whatever structure best presents the system input and designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels. Additional information may be added to the subsections, if the suggested lists are inadequate to describe the project inputs and outputs.

a. **System Input Design**. Describe the system input design by providing the following:

o The input media used for external data transfers. For example, electronic data interchange, magnetic tape, scanned paper, etc. If appropriate, the input record types, file structures, and data base structures provided in the File and Data Base Design section may be referenced. Define data element definitions.

o The layout of all input data screens. Provide a graphic representation of the screen. Define or reference all data elements associated with the screen.

o Edit criteria for the data elements. The following types of edits could be performed:

- specific values;
- range of values;
- mandatory/optional;
- numeric, alphabetic values; and
- length.

o Miscellaneous messages associated with screen inputs:

- copies of the form if the input data is to be keyed or scanned for data entry from printed forms;

- description of any access restrictions or security considerations;

- each transaction name, code, and definition if the system is a transaction based processing system; and

- the Privacy Act Warning incorporated into the screen flow if the system is covered by the Privacy Act.

b. **System Output Design**. Provide a description of the system output design. System outputs include reports, data display screens, and files. The output files are defined in Section 4, File and Data Base Design, and may be repeated or referenced. The following should be provided, if appropriate:

o Identification of codes and names for reports and data display screens.

o Description of report and screen contents. Provide a graphic representation of the report or screen layout. Define or reference all data elements associated with the report or screen.

o Identification and description of output files.

o Description of the purpose of the output, including identification of the primary users.

o Report distribution requirements, if any.

o Description of any access restrictions or security considerations.

o Description of any form requirements.

6. Detailed Module Design

A module is the lowest level of design granularity in the system. Depending on the software development approach, there may be one or more modules per program. Provide all of the detailed information logic,
and data necessary to correctly write source code for all modules in the system in this section. At the point at which this document is written, development of the detailed design has been completed for the modules, and that design is documented in this section.

If there is a large number of modules, or if the module documentation is bulky, place it in an appendix or in a separate document. Add additional diagrams and information if necessary to describe the module, its functions, and structure adequately. For state-of-the-art software development areas, such as expert system development and object-oriented design, industry-standard module specification practices should be followed. Include the following information in the detailed module designs:

- The layout of all input data screens.

- A narrative description of each module, its function(s), the conditions under which it is used (called or scheduled for execution), its overall processing, logic, interfaces to other modules, interfaces to external systems, security requirements, etc. Explain any algorithms used by the module in detail.

- Data elements, record structures, and file structures associated with module input and output.

- Graphical representation of the module processing, logic, flow of control, and algorithms, using an accepted diagramming approach (for example, structure charts, action diagrams, flowcharts, etc.).

- Data entry and data output graphics. Define or reference associated data elements. If the project is large and complex, or if the detailed module designs will be incorporated into a separate document, then it may be appropriate to repeat the screen information here.

- Report layout.

7. **Traceability**

Enhance the traceability matrix created in the FRD to include features from the internal design which address user requirements. This matrix begins with the user requirements and assists in tracing how the requirements are addressed in subsequent phases and documents.

The requirements matrix may be broken up into segments. For example, a separate matrix of the internal design document paragraphs may trace to particular paragraphs in the external design document and the functional requirements document may be provided.

G3. DOCUMENTATION - OPERATORS MANUAL

The operators manual provides computer control personnel and computer operators with a detailed operational description of the information system and its associated environments. The contents of an operators manual is provided at Exhibit 6-3.

1. **General**

   a. **Introduction and Purpose**. Describe the introduction and purpose of the operators manual, the name of the system it applies to, and the type of computer operation.

   b. **Project References**. List, as a minimum, the user manual, maintenance manual, and other pertinent documentation.

   c. **Glossary**. List any definitions or terms unique to this document or computer operation and subject to interpretation by the user of this document.

2. **System Overview**

   a. **System Application**. Provide a brief description of the system including its purpose and uses.

   b. **System Organization**. Describe the operation of the system by the use of a chart depicting operations, and interrelationships.

1. **General**

   a. Introduction and Purpose
   b. Project References
   c. Glossary

2. **System Overview**

   a. System Application
   b. System Organization
   c. Software Inventory
   d. Information Inventory
      (1) Resource Inventory
      (2) Report Inventory
   e. Processing Overview
   f. Communications Overview
   g. Security
   h. Privacy Act Warning

3. **Description of Runs**

   a. Run Inventory
   b. Run Sequence
   c. Diagnostic Procedures
   d. Error Messages
   e. Run Descriptions
      (1) Control Inputs
      (2) Primary User Contact
      (3) Data Inputs
      (4) Output Reports
      (5) Restart/Recovery Procedures
      (6) Backup Procedures

Exhibit 6-3: Operators Manual Outline

c. **Software Inventory**. List the software units, to include name, ID, and security considerations. Identify software necessary to resume operation of the system in case of emergency.

d. **Information Inventory**. Provide information about data files, and data bases that are produced or referenced by the system.

(1) **Resource Inventory**. List all permanent files and data bases that are referenced, created, or updated by the system.

(2) **Report Inventory**. List all reports produced by the system. Include report name and the software that generates it.

e. **Processing Overview**. Provide information which is applicable to the processing of the system. Include system restrictions, waivers of operational standards, and interfaces with other systems.

f. **Communications Overview**. Describe the communications functions and process of the system.

g. **Security**. Describe the security considerations associated with the system.

h. **Privacy Act Warning**. Include a Privacy Act warning if the system is covered by the Privacy Act.

3. **Description of Runs**

a. **Run Inventory**. List the runs showing the software components, the job control batch file names, run jobs, and purpose of each run if any portion of the system is run in batch mode. For on-line transaction-based processing, provide an inventory of all software components that must be loaded for the software system to be operational.

b. **Run Sequence**. Provide a schedule of acceptable phasing of the software system into a logical series of operations. If the system is a batch system, provide the execution schedule, which shows the following at a minimum:

- o Job dependencies;
- o Day of week/month/date for execution;
- o Time of day or night (if significant); and
- o Expected run time in computer units.

c. **<u>Diagnostic Procedures</u>**. Describe the diagnostic or error detection features of the system, the purpose of the diagnostic feature and the setup and execution procedures for any software diagnostic procedures.

d. **<u>Error Messages</u>**. List all error codes and messages with operator responses, as appropriate.

e. **<u>Run Descriptions</u>**. Provide detailed information needed to execute system runs. For each run include:

(1) **Control inputs**. Describe all operator job control inputs, for example, starting the run, selecting run execution options, causing an on-line or transaction-based system to become active, and running the system through remote devices, if appropriate.

(2) **Primary User Contact**. Identify the user contact (and alternate if appropriate) for the system, including the person's name, organization, address, and telephone number.

(3) **Data Inputs**. Describe the following if data input is required at production time:

- o Who is responsible for the source data;
- o Format of the data;
- o Data validation requirements; and
- o Disposition of input source and created data.

(4) **Output Reports**. Identify the report names, distribution requirements, and any identifying numbers expected to be

output from the run. Describe reports to be produced from the system run by other than standard means.

(5) **Restart/Recovery Procedures**. Provide instructions by which the operator can initiate restart or recovery procedures for the run.

(6) **Backup Procedures**. Provide instructions by which operator can initiate backup procedures. Cross reference applicable instructions with procedures in the contingency plan.

G4. DOCUMENTATION - USER MANUAL

The user manual contains all information that is essential for the user to make full use of the information system. This manual includes a description of the system functions and capabilities, contingencies and alternate modes of operation, and step-by-step procedures for system access and use. Use graphics where possible in this manual. The manual format may be altered if another format is more suitable for the particular project. At a minimum, the manual should follow the outline at Exhibit 6-4.

1. **Introduction**

a. **Purpose and Scope**. Provide a description of the purpose and scope of the user manual.

b. **Organization of the User Manual**. Describe the organization of the users manual.

c. **Points of Contact**. Identify the organization codes and staff (and alternate if appropriate) who may assist the system user. If a help desk facility or telephone assistance organization exists, describe it in this section.

d. **Project References**. Provide a bibliography of key project references and deliverables which have been produced prior to this point in the system development process. For example, these references might include the quality assurance plan, configuration management plan, data management plan, functional requirements document, external design document, internal design document, etc.

1. **Introduction**

   a. Purpose and Scope
   b. Organization of the User Manual
   c. Points of Contact
   d. Project References
   e. Primary Business Functions
   f. Glossary

2. **System Capabilities**

   a. Purpose
   b. General Description
   c. Privacy Act Considerations

3. **Description of System Functions**

   a. Function X Title
   b. Detailed Description of Function
   c. Preparation of Function Inputs
   d. Results of Function Operations

4. **Operating Instructions**

   a. Initiate Operation
   b. Maintain Operation
   c. Terminate and Restart Operation

5. **Error Handling**

6. **Help Facilities**

Exhibit 6-4: User Manual Outline

e. **Primary Business Functions**. Provide a business perspective to the user's primary responsibilities and tasks as they are supported by the system. Introduce the business functions so that in later sections the focus may rest on the systematic steps to support the business functions.

f. **Glossary**. Provide a glossary of all terms and abbreviations used in the manual. If the glossary is several pages or more in length, it may be placed as an appendix.

2. System Capabilities

Provide a brief overview of the system and its capabilities.

a. **Purpose**. Describe the purpose of the application system.

b. **General Description**. Provide an overview of the system's capabilities, its functions, and operation. Identify and define the specific high-level functions performed by the system. Use graphics and tables, if appropriate.

c. **Privacy Act Considerations**. If the system is protected by the Privacy Act, include a warning of the Privacy Act's civil and criminal penalties concerning the unauthorized use and disclosure of system data.

3. Description of System Functions

Describe each specific function of the system. Describe any conventions to be used in the associated subsections in this high-level section.

Each of the following subsections is repeated as often as necessary to describe each function within the system. The term "Function X" in the subsection title is replaced with the name of the function.

a. **Function X Title**. Provide a descriptive title of each specific function of the system.

b. **Detailed Description of Function**. Provide a summary description of each function. Include the following, if appropriate:

- Purpose and uses of the function;
- Initialization of the function, if applicable;
- Execution options associated with this function;
- Description of function inputs;
- Description of expected outputs and results;
- Relationship to other functions; and
- Summary of function operation.

c. **Preparation of Function Inputs**. Define the inputs required. These inputs should include the basic data required to operate the system. The definition of the inputs include the following:

- Title of each input;
- Description of the inputs, including graphic depictions of display screens;
- Purpose and use of the inputs;
- Input medium;
- Limitations and restrictions;
- Format and content on inputs, and a descriptive table of all allowable values for the inputs;
- Sequencing of inputs;
- Special instructions;
- Relationship of inputs to outputs; and

- o Examples.

d. **Results of Function Operations**. Describe all expected results after completion of the system operation. Include the following in the description if applicable:

- o Description of results, using graphics, text, and tables;
- o Form in which the results will appear;
- o Output form and content;
- o Report generation;
- o Instructions on the use of outputs;
- o Restrictions of the use of outputs, such as those mandated by Privacy Act and Computer Security Act restrictions;
- o Relationship of outputs to inputs;
- o Function specific error messages;
- o Function specific or context sensitive help messages associated with this function; and
- o Examples.

4. Operating Instructions

Provide a detailed set of step-by-step instructions on how to operate the system.

a. **Initiate Operation**. Include procedures to load the software into the computer, if the software is distributed on diskette or should be downloaded. Describe how to sign on to a mainframe or LAN based software system. Describe how to establish the required mode of operation. Describe how to set any initial parameters that are required for operation. Describe how to operate the computer program and use the various system functions.

b. **<u>Maintain Operation</u>**. Define procedures to maintain operation of the software where user intervention is required.

c. **<u>Terminate and Restart Operation</u>**. Define procedures for normal and unscheduled termination of the system operations. Define how to restart the system.

5. **Error Handling**

Address error message and help facilities. Additional information and subsections may be added as necessary. Provide a list of all possible error messages including the following:

- Any numeric error codes associated with the error message;
- A description of the meaning of the error message; and
- A discussion of how to resolve the error.

6. **Help Facilities**

Describe the help facilities built into the software. If the help facilities are context sensitive, they respond differently depending on what the user is doing in the software, then the help facilities should be described in the "Results of Function Operations."

If there is a bureau or contractor help desk of information facility that the user can contact for resolution of errors, these organizations are identified in this section. Include telephone numbers.

G5. DOCUMENTATION - TRAINING PLAN

The training plan outlines the objectives, needs, strategy, and curriculum to be addressed in training users on the new or enhanced information system. The training plan presents the activities needed to support the development of training materials, coordination of training schedules, reservation of personnel and facilities, planning for training needs, and other training-related tasks.

Develop the training activities to teach user personnel the use of the system as specified in the training criteria. To develop a training plan, refer to outline in Exhibit 6-5.

1. Introduction

a. Background and Scope
b. Points of Contact
c. Organization of this Document
d. Project References
e. Security and the Privacy Act
f. Glossary

2. Requirements Traceability

3. Instructional Analysis

a. Development Approach
b. Issues and Recommendations
c. Needs and Skills Analysis

4. Instructional Methods

a. Training Methodology
b. Training Database
c. Testing and Evaluation

5. Training Resources

a. Course Administration
b. Resources and Facilities
c. Schedules
d. Future Training

6. Training Curriculum

Exhibit 6-5: Training Plan Outline

Include the target audiences and the topics on which training must be conducted on the list of training needs. In the training strategy, include how the topics will be addressed. This information will include the format of the training program, the list of topics to be covered, materials, time, space requirements, and proposed schedules. Discuss quality assurance in terms of testing, course evaluation, feedback, and course modification/enhancement.

1. **Introduction**

   Provide a management summary of the entire plan. Do not provide information at this level if the description provided in the following subsections is sufficient.

   a. **Background and Scope**. Provides a brief description of the project from a management perspective. Identify the users and purpose of the system. Also provide a high-level summary of the training plan and its scope.

   b. **Points of Contact**. Provide the organization name (code) and title of key points of contact for the system development. Include points of contact such as the project manager, the program manager, the quality assurance manager, security manager, the training coordinator, and the training person representative, as appropriate.

   c. **Organization of This Document**. Briefly describe the organization of the training plan.

   d. **Project Reference**. Provide a bibliography of key project references and deliverables that have been produced prior to this point. For example, these references might include the project plan, functional requirements document, test plan, implementation plan, data management plan, conversion plan, and external and internal design documents.

   e. **Security and the Privacy Act**. As appropriate, provide a brief overview of the system's security controls and the need for security and protection of sensitive bureau data. If the system handles sensitive or Privacy Act information, include information regarding labeling system outputs as sensitive, or Privacy Act-related. In addition, if the system is protected by the Privacy Act, include a warning of the Privacy Act's civil and criminal penalties concerning the unauthorized use and disclosure of system data.

f. **Glossary**. Provide a glossary of all terms and abbreviations used in the plan. If it is several pages in length, it may be placed as an appendix.

2. Requirements Traceability

Present the traceability matrix which lists user requirements, as documented in the functional requirements document, and trace how they are addressed in documents such as the external and internal design documents, test plans, and training plans. Cross-reference the user requirements and training needs in appropriate sections of the training plan.

The requirements matrix may be broken up into segments, if appropriate. For example, provide a separate matrix of the training plan paragraphs that trace to particular paragraphs in the internal design document, external design document, functional requirements document, and the statement of work.

3. Instructional Analysis

a. **Development Approach**. Describe the approach used to develop the course curriculum and ensure quality training products. In this description include the methodology used to analyze training requirements in terms of performance objectives and to develop course objectives that ensure appropriate instruction for each target group. List or identify the topics or subjects on which the training must be conducted overall objectives of this training.

b. **Issues and Recommendations**. List any current and foreseeable issues surrounding training. Provide recommendations for resolving each issue; constraints and limitations may also be listed.

c. **Needs and Skills Analysis**. Describe the target audiences for courses to be developed. Target audiences could include, for example, technical professionals, user professionals, data entry clerks, clerical staff members, ADP and non-ADP managers and executives, and so on. List the tasks that must be taught in order to meet objectives successfully, and the skills that must be learned to accomplish those tasks. A matrix may be used to provide this information. Indicate the training needs for each target audience. If appropriate, discuss needs and courses in terms of staff location groupings as well (i.e., headquarters and field personnel).

4. **Instructional Methods**

a. **Training Methodology**. Describe the training methods to be used in the proposed courses in relation to the needs and skills identified above, taking into account such factors as the course objectives, the target audience for a particular course, media characteristics, training setting criteria, and costs. List the training approach to be used (e.g., lecture outlines, audiovisual aids, instructor and student guides, student workbooks, examinations, and/or reference manuals). Sample formats of approach can be included in an appendix, if desired.

b. **Training Database**. Define the training database and how it will be used during computer systems training. Usually, if required, this database contains simulated production data related to various training scenarios and cases developed for instructional purposes. Include an explanation of how the training database is to be developed. If this subsection is not applicable to the system involved, indicate "Not Applicable."

c. **Testing and Evaluation**. Describe methods used to establish and maintain quality assurance over the curriculum development process. This description should include methods to be used to pilot test and evaluate training effectiveness, evaluate student progress and performance, and apply feedback to modify or enhance the course materials and structure. One source of feedback for this latter process could be a course or module specific course/instructor evaluation form. This form should gather trainee reactions on the following topics: scope and relevance of course/module; appropriateness of objectives; usefulness of assignments and materials; effectiveness of course training materials; stronger and weaker features of the course; adequacy of the facilities; timing/length of the course/module; effectiveness of the instructor(s); and participant suggestions/comments.

5. **Training Resources**

a. **Course Administration**. Describe the methods to be used to administer the training program, including procedures for class enrollment, student release, reporting of academic progress and course completion/certification, monitoring of the training program, training records management, and security, as required.

b. **Resources and Facilities**. Provide a description of the resources required by both instructors and students for the training, including facilities (e.g., classroom, training, laboratory, extra tables); equipment (e.g., overhead projector, projection screen, flipchart and/or whiteboard with markers, computer and printer workstations); and materials (e.g., memo pads and pencils, diskettes, viewgraphs, slides). Information in this section can be generic in nature and apply to all courses. Specific course information and special needs may be itemized here as well or, if many differing courses are involved, in Section 6, Training Curriculum.

c. **Schedules**. Present a schedule for implementing the training strategy. Include key tasks to be completed, such as when to set up training facilities and schedule participants; other activities essential to training; and dates on which those tasks and activities must be finished. Indicate responsible parties. Provide an overview of tasks, deliverables (e.g., approach, evaluation forms), scheduled versus actual milestones, and estimated efforts as work plan. In the final version of the training plan, actual course schedules, by location, should also be included.

d. **Future Training**. Describe scheduled training modifications and improvements. This information can include periodic updating of course contents, planned modifications to training environments, retraining of employees, and other predicted changes. Indicate procedures for requesting and developing additional training.

6. Training Curriculum

Provide descriptions of the components that make up each course. If there is a large number of courses or modules being described, place these descriptions in an appendix. Repeat the subsections of this section, if any, for each course.

Each course may be composed of one or more modules. A course description should be developed for each module. At a minimum, each course description should include the course/module name; the length of time the course/module will take; the expected class size (minimum, maximum, optimal); the target audience; course objectives; module content/syllabus; specific training resources required (i.e., devices, aids, equipment, materials, and media to be used); and any special student prerequisites. The course description could also include information on instructor-to-student ratio, total number of students to be trained, estimated number of classes, location of classes, and testing methods.

G6. DOCUMENTATION - MAINTENANCE MANUAL

The maintenance manual provides maintenance personnel with the information necessary to maintain the system effectively. The manual provides the definition of the software support environment, the roles and responsibilities of maintenance personnel, and the regular activities essential to the support and maintenance of program modules, job streams, and database structures.

In addition to the items identified for inclusion in the maintenance manual, any additional information may be provided to facilitate the maintenance and modification of the system. Appendices to document various maintenance procedures, standards, or other essential information may be added to this document as needed. The maintenance manual is outlined in Exhibit 6-6.

1. **Introduction**

Provide general reference information regarding the maintenance manual. Whenever appropriate, additional information may be added to that described here.

a. **Purpose of the Manual**. Describe the purpose of the manual and reference the system name and identifying information about the system and its programs.

d. **Points of Contact**. Identify the organization(s) responsible for system development, maintenance, and use, as well as points of contact (and alternate if appropriate) for the system within each organization.

1. **Introduction**

a. Purpose of this Manual

b. Points of Contact
c. Project References
d. Glossary

2. **System Description**

a. System Application
b. System Organization
c. Security and the Privacy Act
d. System Requirements Cross-Reference

3. **Support Environment**

a. Equipment Environment
b. Support Software
c. Data Base Characteristics
d. Personnel

4. **System Maintenance Procedures**

a. Conventions
b. Verification Procedures
c. Error Conditions
d. Maintenance Software
e. Maintenance Procedures

5. **Software Unit Maintenance Procedures**

Exhibit 6-6: Maintenance Manual Outline

c. **Project Reference**. Provide a bibliography of key project references and deliverables produced during the information system development life cycle. If appropriate, reference the functional requirements document, external design document, internal design document, data management plan, test plan, test analysis report(s), operators manual, user manual, load module description, source code description, and job control language description.

d. **<u>Glossary</u>**. Provide a glossary with definitions of all terms, abbreviations, and acronyms used in the manual. If the glossary is several pages in length, place it as an appendix.

2. **System Description**

Provide a brief overview of the system to be maintained.

a. **<u>System Application</u>**. Provide a brief description of the purpose of the system, the functions it performs, and the business processes that the system is intended to support. If the system is a database or an information system, include a general description of the type of data maintained, and the operational sources and uses of that data.

b. **<u>System Organization</u>**. Provide a brief description of the system structure, major system components, and the functions of each major system component. Include charts, diagrams, and graphics as necessary.

c. **<u>Security and the Privacy Act</u>**. Provide a brief overview of the system's security controls and the need for security and protection of sensitive data. For example, include information regarding procedures to log on/off, provisions for use of special passwords, access verification, and access statuses as appropriate. If the system handles sensitive or Privacy Act information, include information regarding labeling system outputs as sensitive, or Privacy Act-related. In addition, if the system is covered by the Privacy
Act, include a warning of the Privacy Act's civil and criminal penalties concerning the unauthorized use and disclosure of system data.

d. **<u>System Requirements Cross-Reference</u>**. Include a document that cross-references the detailed system requirements with the external design document, internal design document, and test plans. This document, also referred to as a traceability matrix in other documents, assists maintenance personnel by tracing how the user requirements developed in the functional requirements document are met in other products of the life cycle. Since this information is provided in the internal design document, Exhibit 6-5, it may be appropriate to repeat or enhance that information here.

3. **Support Environment**

Describe the operating and support environment for the system and program(s). Include a discussion of the equipment, support software, database characteristics, and personnel requirements for supporting maintenance of the system and its programs.

a. **Equipment Environment**. Describe the equipment support environment, including the development, maintenance, and target host computer environments. Describe telecommunications and facility requirements, if any.

(1) **Computer Hardware**. Discuss the computer configuration on which the software is hosted and its general characteristics. Identify the specific computer equipment required to support software maintenance, if that equipment differs from the host computer. For example, if software development and maintenance are performed on a platform that differs from the target host environment, describe both environments. Describe any miscellaneous computer equipment required, such as hardware probe boards that perform hardware-based monitoring and debugging of software.

(2) **Facilities**. Describe the special facility requirements, if any, for the system and program maintenance and include any telecommunications facilities required to test the software.

b. **Support Software**. List all support software, such as operating systems, transactions processing systems, and database management systems, as well as software used for maintenance and testing of the

system. With the support software lists include the appropriate version or release numbers, along with their documentation references.

c. **Database Characteristics**. Include an overview of the nature and content of each database used by the system. Reference other documents for a detailed description, including the external design and internal design documents as appropriate.

d. **Personnel**. Describe the special skills required for the maintenance personnel. These skills may include knowledge of specific versions of operating systems, transactions processing systems, high-level languages, screen and display generators, database management systems, testing tools, and computer-aided software engineering tools.

4. **System Maintenance Procedures**

Include information on the procedures necessary for programmers to maintain the software.

a. **Conventions**. Describe all rules, schemes, and conventions used within the system. Examples of this type of information include:

- System-wide labeling, tagging, and naming conventions for programs, units, modules, procedures, routines, records, files, and data element fields;

- Procedures and standards for charts and listings;

- Standards for including comments in programs to annotate maintenance modifications and changes; and

- Abbreviations and symbols used in charts, listings, and comments sections of programs.

If the conventions follow a standard programming practices and standards document, that document may be referenced, provided that it is available to the maintenance team.

b. **Verification Procedures**. Include requirements and procedures necessary to check the performance of the system following modification or maintenance of the system's software components. Address the verification of the system-wide correctness and performance.

Present in detail system-wide testing procedures. Reference the original development test plan if the testing replicates development testing.

Describe the types and source(s) of test data in detail.

c. <u>**Error Conditions**</u>. Describe all system-wide error conditions that may be encountered within the system, including an explanation of the source(s) of each error and recommended methods to correct each error.

d. <u>**Maintenance Software**</u>. Reference any special maintenance software used to maintain the system, and its supporting documentation.

e. <u>**Maintenance Procedure**</u>. Describe known step-by-step, system-wide maintenance procedures, such as procedures for setting up and sequencing inputs for testing. In addition, present standards for documenting modifications to the system.

5. Software Unit Maintenance Procedures

For each software unit within the system, provide the information requested. If the information would be identical for each of the software units, it is not necessary to repeat for each software unit. If the information in any of the following areas is identical to information provided in Section 4 above for the system maintenance procedures, then reference that area.

<u>**Unit Name And Identification**</u>. Provide the name or identification of each software unit that is a component of the system. Repeat the following information for each unit name.

o **Description**. Provide a brief narrative description of the software unit. Reference other sections within the life cycle which contain more detailed descriptive material.

o **Requirements Cross-Reference**. Include the detailed user requirements satisfied by this particular software unit. It may be a matrix which traces the system requirements from the functional requirements document, through the external design document, internal design document, and test plans for the specific software units. Other life cycle documentation may be referenced as appropriate.

o **Conventions**. Describe all rules, schemes, and conventions used within the program. If this information is program-specific, provide that information here. If the conventions are all system-wide, discuss them in the Conventions area in Section 4. If the conventions follow a standard programming practices and standards document, that document may be referenced here.

- **Verification Procedures**. Include the requirements and procedures necessary to check the performance of the program following modification or maintenance, and addresses the verification of program correctness, performance, and detailed testing procedures. If the testing replicates development testing, it may be appropriate to reference the original development test plan.

- **Error Conditions**. Describe all program-specific error conditions that may be encountered, an explanation of the source(s) of each error, and recommend methods to correct each error. If these error conditions are the same as the system-wide error conditions described in Section 4 above, that section may be referenced here.

- **Listings**. Provide a reference to the location of the program listings.

G7. DOCUMENTATION - CONVERSION PLAN

The conversion plan describes the strategies involved for converting data from an existing system to another hardware or software environment. It is appropriate to re-examine the original system's functional requirements for the condition of the system prior to conversion in order to determine if the original requirements are still valid. A description of the conversion plan is included at Exhibit 6-7.

1. Introduction

Provide a brief description of introductory material in this section.

a. **<u>Purpose and Scope</u>**. Describe the purpose and scope of the conversion plan. Reference the information system name and provide identifying information about the system undergoing conversion.

b. **<u>Points of Contact</u>**. Identify the sponsoring organization. Provide the name of the responsible organization and staff (and alternate if appropriate) who serve as points of contact for the system conversion. Include telephone numbers of key staff and organizations.

c. **Project References**. Provide a bibliography of key project references and deliverables that have been produced prior to this point in the project development. These documents may have been produced in a previous development life cycle that resulted in the initial version of the system to undergo conversion, or may have been produced in the current conversion effort, as appropriate.

d. **Glossary**. Provide a glossary of all terms and abbreviations used in the plan. If it is several pages in length it may be placed in an appendix.

2. **Conversion Overview**

Provide an overview of the following aspects of the conversion effort.

a. **System Overview**. Provide an overview of the system to undergo conversion. Describe the general nature or type of the system. Include a brief overview of the processes the system is intended to support. If the system is a database or an information system, also include a general discussion of the type of data maintained, the operational sources, and uses of that data.

b. **System Conversion Overview**. Provide an overview of the planned conversion effort.

1. **Introduction**

a. Purpose and Scope
b. Points of Contact
c. Project References
d. Glossary

2. **Conversion Overview**

   a. System Overview
   b. System Conversion Overview
      (1) Conversion Description
      (2) Type of Conversion
      (3) Conversion Strategy
      (4) Conversion Risk Factors
   c. Conversion Tasks
      (1) Conversion Planning
      (2) Pre-conversion Tasks
      (3) Major Tasks and Procedures
   d. Conversion Schedule
   e. Security

3. **Conversion Support**

   a. Hardware
   b. Software
   c. Facilities
   d. Materials
   e. Personnel
      (1) Personnel Requirements and Staffing
      (2) Training of Conversion Staff

Exhibit 6-7: Conversion Plan Outline

(1) **Conversion Description**. Provide a description of the system structure and major components. If only selected parts of the system are to undergo conversion, identify which components will and will not be converted.

If the conversion process is to be organized into discrete phases, identify which components will undergo conversion in each phase. Include both hardware, software, and data, as appropriate. Charts, diagrams, and graphics may be included as necessary. Develop and continuously update a milestone chart for the conversion process.

(2) **Type of Conversion**. Describe the type of conversion effort. The software part of the conversion effort usually falls into one of the following three categories:

- o **Intralanguage conversion** is a conversion between different versions of the same computer language, or different versions of a software system, such as a DBMS, operating system, or LAN management system.

- o **Interlanguage conversion** is the conversion from one computer language to another, or from one software system to another.

- o **Same compiler conversions** use the same language and compiler versions. Typically, these conversions are performed to make programs conform to standards, improve program performance, convert to a new system concept, and so on. These conversions may require some program redesign, and generally require some reprogramming.

In addition to the three categories of conversions described above, other types of conversions may be defined as necessary.

(3) **Conversion Strategy**. Describe the strategies for conversion of system hardware, software, and data.

- o **Hardware Conversion Strategy**. Describe the strategy to be used for the conversion of system hardware, if any. Describe the new (target) hardware environment, if appropriate.

- o **Software Conversion Strategy**. Describe the conversion strategy to be used for the software.

- o **Data Conversion Strategy**. Describe the data conversion strategy, data quality assurance, and the data conversion controls.

- **Data Conversion Approach**. Describe the specific data preparation requirements and the data that must be available for the system conversion. If data is to be transported from the original existing system, provide a detailed description of the data handling, conversion, and loading procedures. If the data is to be transported using machine-readable media, describe the characteristics of those media.

- **Data Quality Assurance and Control**. Describe the strategy to be used to ensure data quality before and after all data conversions. Describe the approach to data scrubbing and quality assessment of the data prior before it is moved to the new or converted system. The strategy and approach may be described in a formal transition plan or document if more appropriate.

(4) **Conversion Risk Factors**. Describe the major risk factors in the conversion effort and strategies for their control or reduction. Describe the risk factors that could impact the conversion feasibility, the technical performance of the converted system, the conversion schedule, or costs. In addition, a review should be made to ensure the current backup and recovery procedures are adequate as well as operational.

c. **Conversion Tasks**. Describe all of the major tasks associated with the conversion, including planning and pre-conversion tasks.

(1) **Conversion Planning**. Describe planning for the conversion effort. If planning and related issues have been addressed in another life cycle document, reference those documents here. The following list provides some examples of conversion planning issues that could be addressed:

- Analysis of the workload projected for the target conversion environment to ensure that the projected environment can adequately handle that workload, and meet performance and capacity requirements.

- o Projection of the growth rate of the data processing needs in the target environment to ensure that the system can handle the projected near-term growth, and that it has the expansion capacity for longer-term future needs.

- o Analysis to identify missing features in the new (target) hardware and software environment that were supported in the original hardware and software, and were used in the original system.

- o Development of a strategy for recoding, reprogramming, or redesigning the components of the system that used hardware and software features not supported in the new (target) hardware and software environment, but used in the original system.

(2) **Pre-conversion Tasks**. Describe all tasks that are logically separate from the conversion effort itself, but that must be completed prior to the initiation, development, or completion of the conversion effort. Examples of such preconversion tasks include the following:

- o Finalize decisions regarding the type of conversion to be pursued.

- o Install changes to the system hardware, such as a new computer or communications hardware, if necessary.

- o Implement changes to the computer operating system or operating system components, such as the installation of a new LAN operating system, or a new windowing system.

- o Acquire and install other software for the new environment, such as a new DBMS or document imaging system.

(3) **Major Tasks and Procedures**. Address the major tasks associated with the conversion and the procedures associated with those tasks.

- o **Major Task Name**. Provide a name for each major task. Provide a brief description of each major task required for the conversion of the system. Include the tasks required to perform the conversion, prepare data, and test the system. If some of these tasks are described in other life cycle documents, reference those documents here.

- o **Procedures**. For each major task describe the approach to that task. Provide as much detail as necessary to describe these procedures.

d. **<u>Conversion Schedule</u>**. Provide a schedule of activities to be accomplished during the conversion. Describe the pre-conversion tasks and major tasks for all hardware, software, and data conversions described in the Conversion Tasks Section, showing the beginning and ending dates of each task. Charts may be used as appropriate.

e. **<u>Security</u>**. If appropriate for the system to be implemented, include an overview of the system security features and the security during conversion.

- o **System Security Feature**. The description of the system security features, if provided, should contain a brief overview and discussion of the security features that will be associated with the system when it is converted. Reference other life cycle documents as appropriate. Describe the changes in the security features or performance of the system that would result from the conversion.

- o **Security During Conversion**. Address all security issues specifically related to the conversion effort itself.

3. Conversion Support

Describe the support necessary to implement the system. If there are additional support requirements not covered by the categories shown here, add other subsections as needed.

a. **<u>Hardware</u>**. Provide a list of support equipment and include all hardware to be used for the conversion.

b. **<u>Software</u>**. Provide a list of software and databases required to support the conversion. Describe all software tools used to support the conversion effort, including the following types of software tools, if used:

- o Automated conversion tools, such as software translation tools for translating between different computer languages or translating within software families (e.g., between release versions of compilers and DBMSs);
- o Automated data conversion tools for translating between data storage formats associated with the different implementations, such as different DBMSs or operating systems;
- o Quality assurance and validation software for the data conversion, i.e., automated testing tools;
- o Computer-aided software engineering (CASE) tools for reverse engineering of the existing application;
- o CASE tools for capturing system design information and presenting it graphically;
- o Documentation tools, such as cross-reference lists and data attribute generators; and
- o Commercial off-the-shelf software, as well as software written specifically for the conversion effort or for Treasury.

c. **Facilities**. Identify the physical facilities and accommodations required during the conversion period.

d. **Materials**. Provide a list of support materials.

e. **Personnel**. Describe personnel requirements and any known or proposed staffing, if appropriate. Also describe the training, if any, to be provided for the conversion staff.

- o **Personnel Requirements and Staffing.** Describe the number of personnel, length of time needed, types of skills, and skill levels for the staff required during the conversion period.
- o **Training of Conversion Staff**. Address the training, if any,

necessary to prepare the staff for converting the system. It should provide a training curriculum listing the courses to be provided, a course sequence, and a proposed schedule. If appropriate, it should identify which courses should be attended by particular types of staff, by job position description. Training for users in the operation of the system is not included in this section, but is normally included in the a Training Plan.

G8. DOCUMENTATION - IMPLEMENTATION PLAN

The implementation plan describes how the information system is to be installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort (such as hardware, software, facilities, materials, and personnel), and any site-specific implementation requirements. The plan is developed during the design phase, and is updated during the development phase; the final version is provided in the test phase, and is used for guidance during the implementation phase. The implementation plan is outlined at Exhibit 6-8.

1. **Introduction**. Provide an overview of the information system and other include any additional information that may be appropriate in this section.

1. **Introduction**

    a. Purpose
    b. System Overview
    c. Project References
    d. Glossary

2. **Management Overview**

    a. Description of Implementation
    b. Points of Contact

c. Major Tasks
d. Implementation Schedule
e. Security

3. **Implementation Support**

a. Hardware, Software, Facilities, and Materials
b. Personnel
(1) Personnel Requirements and Staffing
(2) Training of Implementation Staff

4. **Implementation Requirements by Site**

a. Site Name or Identification for Site X (for each site)
(1) Site Requirements
(2) Site Implementation Details

Exhibit 6-8: Implementation Plan Outline

a. **Purpose**. Describe the purpose of the implementation plan. Reference the system name and identify information about the system to be implemented.

b. **System Overview**. Provide a brief overview of the system to be implemented, including a description of the system and the system organization.

(1) **System Description**. Describe the general nature or type of system. Include a brief overview of the processes the system is intended to support. If the system is a database or an information system, include a general discussion of the description of the type of data maintained, and the operational sources and uses of that data.

(2) **System Organization**. Provide a brief description of the system structure and major system components essential to the implementation of the system. Include a description of both hardware and software, as appropriate. Charts, diagrams, and graphics may be included as necessary.

c. **Project References**. Provide a bibliography of key project references and deliverables that have been produced prior to this point in the project development. For example, these references might include the project plan, data management plan, acquisition plan, functional requirements document, test plan, conversion plan, and external and internal design documents.

d. **Glossary**. Provide a glossary of all terms and abbreviations used in the manual. If it is several pages in length, it may be placed in an appendix.

2. Management Overview

Provide a brief description of the implementation and major tasks involved in this section.

a. **Description of Implementation**. Provide a brief overview description of the system and the planned implementation approach.

b. **Points of Contact**. Identify the sponsoring organization and the name of the responsible organization(s) and titles of the staff that serves as points of contact for the system implementation along with their telephone numbers. These points of contact could include the project manager, program manager, security manager, database administrator, configuration management manager, or other managers with responsibilities relating to the system implementation. The site implementation representative for each field installation or implementation site should also be included, if appropriate. List all managers and staff with whom the implementation must be coordinated.

c. **Major Tasks**. Provide a brief description of each major task required for the implementation of the system. Add as many subsections as necessary to this section to describe all of the major tasks adequately. The tasks described in this section are not site-specific, but generic, or overall project tasks, which are required to install hardware and software, prepare data, and verify the system. Include the following information for the description of each major task, if appropriate:

- o What the task is to accomplish;
- o Resources required to accomplish the task;
- o Key person(s) responsible for the task; and
- o Criteria for successful completion of the task.

Examples of major tasks are:

- o Providing overall planning and coordination for the implementation;
- o Providing appropriate training for personnel;
- o Ensuring that all manuals applicable to the implementation effort are available when needed;
- o Providing all needed technical assistance;
- o Scheduling any special computer processing required for the implementation;
- o Performing site surveys prior to implementation;
- o Ensuring that all prerequisites have been fulfilled prior to the implementation date;
- o Providing personnel for the implementation team;
- o Acquiring special hardware or software;
- o Performing data conversion prior to loading data into the system; and
- o Preparing site facilities for implementation.

d. **Implementation Schedule**. Provide a schedule of activities to be accomplished during implementation. Show the required tasks described in the Major Tasks section above in chronological order, with the beginning and ending dates of each task.

e. **Security**. If appropriate for the system to be implemented, include an overview of the system security features and requirements during the implementation. If the system is covered by the Privacy Act, include Privacy Act concerns.

(1) **System Security Features**. Provide a brief overview and discussion of the security features that will be associated with the system when it is implemented. Include the primary security features associated with the system hardware and software. Security and protection of sensitive bureau data and information should be discussed, if applicable. Reference the sections of previous deliverables that address system security issues, if appropriate.

(2) **Security During Implementation**. Address security issues specifically related to the implementation effort, if any. For example, if LAN servers or workstations are to be installed at a site with sensitive bureau data preloaded on nonremovable hard disk drives, address how security would be provided for the data on these devices during shipping, transporting, and installation, since theft of the devices could result in the compromise of sensitive bureau data.

3. **Implementation Support**

Describe the support software, materials, equipment, and facilities required for the implementation, as well as the personnel requirements and training necessary for the implementation. The information provided in this section is not site-specific. If there are additional support requirements not covered by the subsections listed below, others may be added as needed.

a. **Hardware, Software, Facilities, and Materials**. List support software, materials, equipment, and facilities required for the implementation, if any.

(1) **Hardware**. Provide a list of support equipment and include all

hardware to be used for testing the implementation. For example, if a client-server database is implemented on a LAN, a network monitor or "sniffer" might be used, along with test programs, to determine the performance of the database and LAN at high utilization rates. If the equipment is site-specific, list it in Section 4 below.

(2) **Software**. Provide a list of software and databases required to support the implementation. Identify the software by name, code, or acronym. Identify which software is commercial off-the-shelf and which is bureau specific. Identify any software used to facilitate the implementation process. If the software is site-specific, list it in Section 4. below.

(3) **Facilities**. Identify the physical facilities and accommodations required during implementation. Examples include physical work space for assembling and testing hardware components, physical desk space for software installers, and classroom space for training the implementation staff. Specify the hours per day needed, number of days, and anticipated dates. If the facilities needed are site-specific, provide this information in Section 4 below.

(4) **Material**. Provide a list of support materials, for example, magnetic tapes and disk packs.

b. **Personnel**. Describe personnel requirements and any known or proposed staffing, if appropriate. Also describe the training, if any, to be provided for the implementation staff.

(1) **Personnel Requirements and Staffing**. Describe number of personnel, length of time needed, types of skills, and skill levels for the staff required during the implementation period. If particular staff members have been selected or proposed for the implementation, identify them and their role in the implementation.

(2) **Training of Implementation Staff**. Address the training, if any, necessary to prepare staff for implementing and maintaining the system; it does not address user training, which is the subject of the Training Plan. Describe the type and amount of training required for each of the following areas, if appropriate for the system:

- o System hardware/software installation;
- o System support; and
- o System maintenance and modification.

Present a training curriculum listing the courses to be provided, a course sequence, and a proposed schedule. If appropriate, identify which courses should be attended by particular types of staff/by job position description.

If training is to be provided by one or more commercial vendors, identify them, the course name(s) and a brief description of the course content.

If the training is to be provided by bureau staff, provide the course name(s), and an outline of the content of each course. Identify the resources and support materials required to teach the course(s) as well as proposed instructors.

4. **Implementation Requirements by Site**

Describe the specifics implementation requirements and procedures. If these requirements and procedures differ by site, repeat these subsections for each such site; if they are the same for each site, or if there is only one implementation site, use these subsections only once.

The "X" in the subsection number is replaced with a sequenced number beginning with " 1." For each subsection with the same value of "X" is associated with the same implementation site. If a complete set of subsections is to be associated with each implementation site, then "X" is assigned a new value for each site.

a. **Name or Identification for Site X**. Provide the name of the specific site or sites to be discussed in the subsections of this section.

(1) **Site Requirements**. Define the requirements that must be met for the orderly implementation of the system. Describe the hardware, software, and facilities site-specific requirements for this area.

Any site requirements that do not fall into these three categories and were not described in Section 3, Implementation Support, may be described here, or other subsections may be added following Facilities Requirements below.

- **Hardware Requirements**. Describe the site-specific hardware requirements necessary to support the implementation, e.g., LAN hardware for a client-server database designed to run on a LAN.

- **Software Requirements**. Describe any software required to implement the system e.g., software specifically designed for automating the installation process.

- **Data Requirements**. Describe specific data preparation requirements and data that must be available for the system implementation. An example would be the assignment of individual IDs associated with data preparation.

- **Facilities Requirements**. Describe the site-specific physical facilities and accommodations required during the system implementation period. Some examples of this type of information are provided in Section 3 above.

(2) **Site Implementation Detail**. Address the specifics of the implementation for this site. Include a description of the implementation team, schedule, procedures, and database and data updates.

- **Implementation Team**. If an implementation team is required, describe its composition and the tasks to be performed at this site by each team member.

- **Schedule**. Provide a schedule of activities, including planning and preparation, to be accomplished during implementation at this site. Describe the required tasks, in chronological order with the beginning and ending dates of each task. If appropriate, charts and graphics may be used to present the schedule.

- **Detailed Implementation Procedures**. Provide a sequence of detailed procedures required to accomplish the specific hardware and software implementation at this site. If necessary, other documents may be referenced.

  If appropriate, include a step-by-step sequence of the detailed procedures. A checklist of the installation events may be provided to record the results of the process.

  If the site operations startup is an important factor in the implementation, then address startup procedures in some detail. If the system is to replace an already operating system, then address the startup and cut-over processes in detail. If there is a period of parallel operations with an existing system, address the startup procedures which includes technical and operations support during the parallel cycle, and the consistency of data within databases of the two systems.

- **Database Procedures**. Describe the database environment where the software system and the database(s), if any, are to be installed. Include a description of the different types of database and library environment (e.g., production, test, and training databases).

  Include the host computer database operating procedures, database file and library naming conventions, database system generation parameters, and any other information needed to effectively establish the system database environment. Include database administration procedures for testing changes, if any, to the DBMS prior to the system implementation.

- **Data Update Procedures**. If the data update procedures are described in another document, such as the Operators Manual or Conversion Plan, that document may be referenced here. The following are examples of information to be included:

  Control inputs;
  Operating instructions;
  Database data sources and inputs;
  Output reports; and
  Restart and recovery procedures.

G9. DOCUMENTATION - CONTINGENCY PLAN

Each system component must develop and test contingency plans which will enable them to continue to operate during and after a catastrophic event that significantly interrupts their normal processing. Each contingency plan must be tested at a frequency commensurate with the risk and magnitude of loss or harm that could result from the disruption of a systems operation.

A contingency plan must be developed for each mission-critical system, that is, those systems that significantly affect bureau programs, property, finances and other resources. Contingency plans should be prepared systematically to cover as many disruptions as possible.

A contingency plan is an action plan for ensuring processing continuity for all mission-critical systems. The following three main parts are required: emergency response plan, backup operation plan and recovery plan. Identify as much information as possible to initiate a draft of the contingency plan.

1. **Emergency Response Plan**

An emergency response plan defines procedures to be used to respond to a disruption in computing resources for an extended period of time, due to natural causes, sabotage or other reasons. It is a documented plan of action for the first 36 hours after an emergency and must include the following: telephone call roster, disaster assessment team, offsite storage activation, hot-site activation (if appropriate) and emergency response checklist validation.

2. **Backup Operation Plan**

A backup operation plan is developed to ensure that essential tasks can be completed and that work in process can be recovered subsequent to disruption of an operation. These essential tasks include backing up systems software, hardware, data, communications, documentation and contingency plans at an offsite storage location. The backup operation plan must provide for appropriate testing of the effectiveness of the backup operation for mission-critical systems.

3. **Recovery Plan**

A recovery plan defines preplanned recovery procedures to permit a smooth, rapid restoration of a system operation from a disaster. The plan should provide for the following: a survey of damages, the implementation of short and long-term plans, a strategy for repairing or rebuilding the site,

ordering instructions necessary for a hardware equipment and implementing direction for security measures.

Depending on the size and type of system, different scenarios can be designed for different levels of degradation. An action plan for a mainframe computer will be more comprehensive and indepth than one for a personal computer. The responsible systems personnel should consult with their computer security officer for guidance developing contingency plans.

DESIGN ISSUES CHECKLIST

The project team uses this checklist to ensure specific external and internal design issues have been addressed. Consideration is given to the questions below for each system project. Action may not be required for some items.

External Design

___ Were the users needs used in the conceptual system design and were those needs reflected in the physical design?

___ Are the performance requirements of the system adequately defined?

___ Is the equipment configuration needed to process the system defined?

___ Is all the necessary system software defined?

___ Have all security and Privacy Act requirements of the system been met?

--- Has the final Privacy Act notification been prepared and approved and forwarded to be announced in the Federal Register?

___ Is the operating environment defined?

___ Are hardware, software, and services acquisitions planned?

___ Are I/O designs consistent with approved I/O requirements definitions?

___ Has the logical data base been approved?

___ Are data management considerations defined?

___ Are data communications incorporated into the design and acquisition considerations?

___ Have system acceptance criteria been documented?

--- Have operating backup facilities been addressed

Exhibit 6-9: Design Issues Checklist

DESIGN ISSUES CHECKLIST

External Design Document

___ Have detailed system/subsystem specifications been developed for the system?

___ Does the system development process include a detailed system analysis and design?

___ Does the system development process include a conceptual system design?

___ Do the detailed system/subsystem specifications include an overall narrative description of the system?

___ Do the detailed system/subsystem specifications include the design characteristics of the system and provide a system flowchart?

___ Were all processing procedures, both manual and automated, prepared before implementation and reviewed to ensure that the detailed design specifications were followed?

___ Do the detailed system/subsystem specifications include interfaces to other systems?

___ Have manual procedures and support measures been documented?

___ Does data repository contain all data element names, attributes validation rules and definitions?

___ Have all data elements been defined and their usage identified?

___ Have master files/databases been defined?

___ Are definitions of inputs from and outputs to interfacing system referenced?

Exhibit 6-9: Design Issues Checklist (cont'd)

DESIGN ISSUES CHECKLIST

External Design Document (cont'd)

___ Have edit criteria and transaction conditions been established and defined for online transactions?

___ Have backup and reconstruction of I/O files been defined?

--- Have telecommunications requirements been documented fully?

Internal Design

___ Does the system structure clearly show boundaries of each design unit or job stream?

___ Have any reusable segments from existing systems been analyzed?

___ Are hardware, software, and services procurements underway?

___ Is there clear separation of mainframe files from work files, tables and output files?

___ Has the physical database/file design been documented?

___ Have teleprocessing specifications been completed?

___ Have data control and auditing requirements been defined?

___ Does data repository contain all data flows and data stores used throughout the system?

___ Is a conversion effort planned and documented?

--- Has a contingency plan been drafted?

Exhibit 6-9: Design Issues Checklist (cont'd)
DESIGN ISSUES CHECKLIST

Internal Design Document

___ Does the Internal Design Document include detailed system analysis and design?

___ Is system software needed to support the program?

___ Have the input record formats and descriptions been provided?

___ Have detailed program specifications been developed for all programs of the system?

___ Do these specifications include a narrative description of the program and its functions?

___ Do these specifications describe the program performance requirements?

___ Have the controls over and within the program been described?

___ Have lists of constants, codes, and tables been provided?

___ Has a description of the program's logic, including flowcharts and decision tables, been supplemented by narrative explanations?

___ Have the logical and physical characteristics of all databases used by the program, including file layouts and data element definitions, been provided?

___ Have the database specifications been included?

___ Has the I/O design been fully documented?

___ Have checkpoint/restart provisions been defined?

___ Have all interactive dialogues been fully documented?

Exhibit 6-9: Design Issues Checklist (cont'd)
DESIGN ISSUES CHECKLIST

Operators Manual

Does the manual include:

___ A diagram showing the inputs, outputs, data files, and sequence of operations of the computer based system?

___ An inventory of all programs included in the system?

___ An inventory of each permanent file that is referenced, created, or updated by the system?

___ A list of the various runs possible and a summary of each run's purpose?

___ A description of the manner in which progressive advances from one run to another is made to complete the entire run cycle?

___ Job control statements needed for each run?

___ Emergency Procedures?

___ Operator instructions for each run?

___ A description of procedures for running the system through remote devices?

___ The input and output files for each run?

___ The output reports produced for each run?

___ The output reports that need to be produced by other means?

___ The restart/recovery procedures?

Exhibit 6-9: Design Issues Checklist (cont'd)

INTERNAL DESIGN ISSUES CHECKLIST

Operators Manual (cont'd)

___ A warning about the applicability of the Privacy Act and caution about the civil and criminal penalties for unauthorized disclosure of system data?

___ A reference for the user to contact FOIA/PA branch with questions on the Privacy Act?

Does the manual exclude:

___ Program logic charts or decision tables?

___ Copies of program listings?

User Manual

Does the manual include:

___ A narrative description of the system?

___ A description or diagram of the system?

___ The structure and role of each system component?

___ A description of inputs, the flow of data through the processing cycle, and the outputs?

___ The performance capabilities of the system?

___ The requirements for preparing and entering input data?

___ A description of the equipment needed to process the system?

Exhibit 6-9: Design Issues Checklist (cont'd)

DESIGN ISSUES CHECKLIST

User Manual (cont'd)

___ The requirements relevant to each output, such as format and frequency?

___ Step-by-step procedures required to initiate processing?

___ A list of error codes or conditions generated by the system and the corrective actions to be taken by the user?

___ A warning about the applicability of the Privacy Act and caution about the civil and criminal penalties for unauthorized disclosure of system data?

___ A reference for the user to contact FOIA/PA branch with questions on the Privacy Act?

Maintenance Manual

Does the manual include:

___ A detailed description of each program in the system?

___ A description of the equipment needed to process the system?

___ A description of the system software needed to support the application program?

___ A description of the database being used by the application programs?

___ A description of the programming conventions used to develop the application programs?

___ A description of all error conditions, their sources, and procedures for their correction?

___ A description of the program listings and flowcharts of decision tables?

Exhibit 6-9: Design Issues Checklist (cont'd)

DESIGN ISSUES CHECKLIST

Maintenance Manual (cont'd)

___ A warning about the applicability of the Privacy Act and caution about the civil and criminal penalties for unauthorized disclosure of system data?

___ A reference for the user to contact FOIA/PA branch with questions on the Privacy Act?

___ A detailed description of security aspects or considerations?

Training Plan

___ Have target audiences for training been identified and a training needs analysis performed?

___ Have performance criteria based on required skills for each target group and enabling course objectives been developed?

___ Are choices of instructional methods and media for effective training justified by the subject/course objectives, audience, and constraints imposed?

___ Have sufficient materials, resources, and time been allocated to specified courses to ensure appropriate instruction for each target audience group?

___ Are specific courses described in enough detail to permit evaluation of the planned training schedule, objectives, training materials, required equipment and resources, and administrative procedures to be used?

___ Have methods been defined for maintaining quality control over course development and effectiveness?

___ Do documented procedures exist that explain the use and development of a training database if required?

___ Have plans been specified for training follow-on personnel and updating courses as required?

Exhibit 6-9: Design Issues Checklist (cont'd)

DESIGN ISSUES CHECKLIST

Conversion Plan

___ Are automated data retrieval and analysis packages or specially written computer programs used for evaluating data records?

___ Are there effective procedures to ensure that no data is lost or erroneously changed during the conversion to the newly designed system?

___ Has sufficient computer time been allocated for the conversion process?

___ Was the newly designed system tested in parallel with the old system?

___ Was sufficient time allocated for parallel processing to allow for adequate comparison of results from both systems?

___ Has a risk analysis be conducted to determine any significant changes to the physical facility, hardware, or operating system?

___ Do documented procedures exist that explain the methods for data conversion and entry?

___ Does the information systems organization have a control group responsible for data conversion and entry of all source documents received from user organizations?

___ Have conversion operations been established as close to the origination of the source documents as possible?

___ Do conversion operations record document information directly onto machine-readable media, as opposed to intermediate media, such as coding documents?

___ Does the information systems organization have a schedule that shows, by application, when data requiring conversion will be received and needs to be converted?

Exhibit 6-9: Design Issues Checklist (cont'd)
DESIGN ISSUES CHECKLIST

Conversion Plan (cont'd)

___ Are all record counts developed during conversion balanced with those of the information systems control group, and are all discrepancies reconciled?

___ Are all predetermined control totals developed during the conversion balanced with those of the information systems control group, and are all discrepancies reconciled?

___ Are rejected transactions caused by data conversion or entry errors corrected by the information systems organization control group?

___ Have the system components to undergo conversion been identified?

___ Has the type of conversion been defined and described?

___ Has the conversion strategy been defined in detail for all hardware, software, and data conversion?

___ Have quality assurance controls and data scrubbing procedures been developed for the data to be converted?

___ Have factors been identified which could result in risks to the technical performance, schedule, and cost of the conversion effort?

___ Have strategies for controlling or reducing the project risks been developed?

___ Have the workload requirements been matched with the projected performance and capacity of the new or converted system?

___ Does the new or converted system have adequate performance, capacity, and expandability for the projected short-term and long-term growth in processing requirements?

Exhibit 6-9: Design Issues Checklist (cont'd)
DESIGN ISSUES CHECKLIST

Conversion Plan (cont'd)

___ Has an analysis of the new (target) hardware and software environment been performed to identify missing features that are being used in the current and software environment?

___ Have strategies been developed for working around any missing features in the new (target) hardware and software environment?

___ Have all major pre-conversion tasks been identified, and schedules developed for performing those tasks?

___ Have all major tasks been identified?

___ Have the resources, key personnel, and criteria for successful completion been identified for all major tasks?

___ Has a conversion schedule been developed for each of the major tasks?

___ Have all security issues relating to the conversion been identified?

___ Has conversion support been planned for, including hardware, software, facilities, materials, and personnel?

___ Have personnel requirements, staffing, and training been identified for the conversion effort?

Implementation Plan

___ When the system is ready for initial operation, will its implementation be coordinated with all personnel involved and other systems affected?

___ After the system is in operation for several months, will a post-implementation review of the entire system, both manual and automated, be performed by a review (audit) staff?

Exhibit 6-9: Design Issues Checklist (cont'd)
DESIGN ISSUES CHECKLIST

Implementation Plan (cont'd)

___ With the advent of the database management system, has a database administrator position been established?

___ Does the administrator make sure that adequate testing is performed before changes to the database management system are implemented?

___ Have all of the major tasks been identified?

___ Have the required resources, key persons, and criteria for successful completion been identified for each tasks?

___ Have the security and privacy requirements during the implementation been defined?

___ Have all support requirements for the implementation effort been identified, including hardware, software, facilities, and materials?

___ Have staffing requirements been completely defined for the implementation effort?

___ Have all training requirement for the implementation staff been identified?

___ Have the implementation requirements, including hardware, software, data, and facilities, been identified for all potential implementation sites?

___ Do the identified site implementation hardware requirements include not only the hardware to be installed, but also any hardware to be used in the implementation process? Have all site differences in hardware requirements been taken into account?

Exhibit 6-9: Design Issues Checklist (cont'd)
DESIGN ISSUES CHECKLIST

Implementation Plan (cont'd)

___ Do the identified site implementation software requirements include not only the software to be installed, but also any software to be used in the implementation process, such as database load programs?

___ Have all of the site-specific data requirements been identified, including site-specific user identification codes and passwords, geographic region-specific data, and so on?

___ Have all site-specific facilities requirements been identified?

___ Has the implementation team been identified?

___ Has the implementation schedule been broken out by site, including planning activities for each site?

___ Have the detailed implementation procedures to be followed by the implementation team been developed?

___ Have all database implementation procedures been developed for each site?

___ Have all data update procedures been developed for each site?

Exhibit 6-9: Design Issues Checklist (cont'd)

SOFTWARE REQUIREMENTS REVIEW CHECKLIST

1. Date/time of meeting ____________.

2. Attendants:

Project Manager	________	Configuration Manager	_______
Data Administrator	________	Project User	_______
Security Manager	________	Project Sponsor	_______
System Developer	________	System Tester	_______
Quality Assurance Manager	_______	Procurement Manager	_______
FOIA/PA Representative	_______	Database Administrator	_______
Telecommunications Manager	_____	Programmer Analyst	_______

3. Were the following presented for review or their absence justified?

Internal Design Document	____________
External Design Document	____________
Operators Manual	____________
User Manual	____________
Training Plan	____________
Maintenance Manual	____________
Conversion Plan	____________
Contingency Plan	____________
Implementation Plan	____________
Updated Configuration Management Plan	____________
Updated Test Plan	____________
Updated Project Plan	____________
Updated Acquisition Plan	____________
Updated Cost-Benefit Analysis	____________
Updated Data Management Plan	____________
Updated Computer Security Plan	____________
Updated Risk Assessment Plan	____________
Updated Quality Assurance Plan	____________
Final Functional Requirements Document	____________
Final Privacy Act Federal Register Notice	____________

Exhibit 6-10: Software Requirements Review Checklist

SOFTWARE REQUIREMENTS REVIEW CHECKLIST

4. Were technical approaches to satisfying the functional requirements discussed/adopted? (Yes/No) If no, explain.

5. Were the following technical areas adequately discussed? (Yes/No) If no, explain.

System Communications _____ Security Procedures _____
Software Interfaces _____ System Requirements _____
System Data Requirements ____

6. Were detailed presentations of the following made? (Yes/No) If no, explain.

Objective System Functions ____ Performance Requirements ____
Security Requirements _____ System Platform _____

7. Other issues raised.

Exhibit 6-10: Software Requirements Review Checklist (cont'd)

FINAL DESIGN REVIEW CHECKLIST

1. Date/time of meeting ____________

2. Attendants:
   Project Manager _______________
   Project User _______________
   System Developer _______________
   System Tester _______________
   Quality Assurance Manager _______________
   Configuration Manager _______________
   Project Sponsor _______________
   System Maintenance Operator _______________
   Telecommunications Manager _______________
   Security Manager _______________
   Programmer Analyst _______________
   Data Administrator _______________
   Database Administrator _______________

3. Has the traceability of the functional requirements been maintained in the detailed design of the system? (Yes/No) If no, explain.

4. Is there evidence available to indicate the software development group is prepared to produce the system? (Yes/No) If no, explain.

5. Has the test documentation been presented and found acceptable? (Yes/No) If no, explain.

6. Other issues raised.

Exhibit 6-11: Final Design Review Checklist

CHAPTER 7. DEVELOPMENT PHASE

A. GENERAL

The purpose of the **development phase** is to convert the deliverables of the design phase into a complete information system. Although much of the activity in the **development phase** addresses the computer programs that make up the system, this phase also puts in place the hardware, software and communications environment for the system, and other important elements of the overall system.

The activities of this phase translate the system design produced in the design phase into a working information system capable of addressing the information system requirements. The elements of the system are developed (or acquired), tested, and integrated; hardware, system software, communications, applications, procedures, and associated documentation. At the end of this phase, the system is ready for the activities of the testing phase.

A number of **project approach, project execution and project continuation decisions** are made in this phase. **Project approach decisions** include: what individuals will participate in the acceptance testing? Be ready to select specific individuals from the sponsoring program organizations to represent their organizations during acceptance testing. This includes confirmation of testing procedures and logistics.

Project execution decisions include: how should new requirements be handled, what modifications must be made to the system design, have security and backup issues been identified and resolved, and have all uncertainties been adequately addressed?

Project continuation decisions include: does the information system requirement continue to exist, does the developed system address the requirements sufficiently to be implemented, and are sufficient funding and other resources available for the remainder of the life cycle?

In this phase, the prerequisites are the:

- o project plan and schedule indicating target date for completion of each module and target date for completion of system testing;

- o internal design document containing program logic flow, and identification of any existing code to be used;
- o external design document containing the subsystems, and the inputs and outputs of each subsystem; and
- o unit/module and integration test plans containing testing requirements schedule and test case specifications for unit and integration testing.

B. TASKS AND ACTIVITIES

The following activities are performed during this phase. These activities may be expanded or deleted upon the size of the proposed system.

- o Acquire and install system environment
- o Create and test database(s)
- o Prepare test case procedures
- o Prepare test files
- o Create program code and compile and refine the program modules
- o Continue configuration accounting and change control
- o Perform test readiness review
- o Continue procurement activities
- o Update training, conversion and implementation plans
- o Update project plan
- o Update cost-benefit analysis
- o Finalize the test plan
- o Obtain approval of the development phase

C. ROLES AND RESPONSIBILITIES

Team members involved in this phase are the project manager, project user, data administrator, data base administrator, system developer, quality assurance manager, security manager, configuration manager, telecommunications manager, programmer analyst, capacity planning manager, and systems operations manager.

D. DELIVERABLES, RESPONSIBILITIES AND ACTION

The deliverables listed below must be available for review at the end of this phase. The deliverables may be expanded or abbreviated depending of the size, scope, and complexity of the systems development effort. The methodology and format for generating source and object codes, load module, and job control language will be dependent on the language and available development tools. The project team will coordinate the guidelines for these documentation.

DELIVERABLES AND PRODUCTS	PRIME RESPONSIBILITY
Source code	Programmer Analyst
Object code	Programmer Analyst
Load module	Programmer Analyst
Job Control Language	Programmer Analyst
Source language data definitions members (create/finalized)	Programmer Analyst
Test files/data	Quality Assurance Manager
Software Development Folder	System Developer
Test Plan (revised)/test case procedures	Quality Assurance Manager
Computer Security Plan (revised)	Security Manager
Risk Assessment Plan (revised)	Security Manager
Quality Assurance Plan (revised)	Quality Assurance Manager
Operators Manual (revised)	Systems Operations Manager
User Manual (revised)	Project User
Maintenance Manual (revised)	Systems Operation Manager
Training Plan (revised)	Project User
Data Management Plan (revised)	Data Administrator
Project Plan (revised)	Project Manager
Cost-Benefit Analysis (revised)	Project Manager
Conversion Plan (revised)	Project Manager
Implementation Plan (revised)	Project Manager
Contingency Plan (revised)	Security Manager/Project User
External Design Document (finalized)	Project Manager
Internal Design Document (finalized)	Program Analyst
Acquisition Plan (finalized)	Procurement Manager

================================

MANAGEMENT ACTION	PRIME RESPONSIBILITY
Reviews and approves deliverables	Project Manager/Team Members
Obtains end-of-phase concurrence from user	Project Manager
Authorizes Test Phase to begin	Steering Committee

E. ISSUES FOR CONSIDERATION

During the **development phase** there are specific issues that should be addressed. These issues are included in a Development Issues Checklist at Exhibit 7-1. The checklist is used as a reference for the system developer to ensure objectives of this phase are met. These issues are independent of the project size and effort. The checklist is part of the review package. The checklist is also for use for conducting the Test Readiness Review as described in Section F of this phase.

F. REVIEW ACTIVITY

The Test Readiness Review is performed during this phase. It ensures that the unit/module and integration testing has been successfully completed. A Test Readiness Review Checklist is provided at Exhibit 7-2 to ensure review goals are met. Participants in the test readiness review include the project manager, programmer analyst, system developer, data administrator, database administrator, and the project user. Other project team members may be requested to participate in the review. Depending on the size and complexity of the project, the review may be conducted by an oversight committee.

G1. DOCUMENTATION - SOURCE CODE, OBJECT CODE, LOAD MODULE, JOB CONTROL LANGUAGE DESCRIPTIONS, AND SOURCE DATA MEMBERS

The methodology and format for generating source code is dependent on the language and available development tools. Guidelines for documentation will be coordinated within the project team. Source language data definition members are translated identifying and describing all data elements and their relationships that make up the database structure.

G2. DOCUMENTATION - SOFTWARE DEVELOPMENT FOLDER

The software development folder contains documentation pertaining to the development of each unit/module, including the test cases, software, test results, approvals, and any other items that will help to explain the functionality of the software. The folder is a dynamic document that is maintained by the system development team and should be constantly updated as system development progresses. The methodology and format for generating the content of the folder depend on the programming language, system, and available development tools. Documentation standards will be coordinated within the project team. The software development folder should include the following information for each unit:

- Requirements in the function requirements document allocated to this unit/module;
- Description of the unit's functionality, in narrative format;
- Source Code;
- Controlled libraries/directories/tables;
- Development methodologies;
- All data necessary to conduct unit testing;
- Test analysis results;
- Project manager signoff for design walkthrough, approval of code, and completion of each unit;
- Completed traceability matrix displaying the unit's functional requirements;
- Completed traceability matrix displaying the unit's test case satisfying the functional requirements in the test plan; and
- Completed software development folder check-off.

SOFTWARE DEVELOPMENT FOLDER CHECK-OFF

REQUIREMENTS

Has each requirement in the functional requirements document allocated to this unit been identified using the traceability matrix?

Have derived requirements found during the development of this unit been identified, justified, and put in the functional requirements document?

FUNCTIONALITY

Is the functionality of this unit fully described?

Is the description in narrative form?

Was a design walkthrough conducted?

Was permission granted to begin programming?

SOURCE CODE

Is the source code listing of the unit included in this folder?

LIBRARIES, DIRECTORIES AND TABLES

Are all coded entities included in the folder?

DEVELOPMENT METHODOLOGIES

Are all development methodologies for the development effort in the folder?

TEST DATA

Is all data necessary to conduct testing referenced in this folder?

TEST ANALYSIS

Was the unit thoroughly tested and all logical paths verified?

SYSTEM DEVELOPER

I certify that this software development folder is complete, the unit ________ defined in this folder has successfully completed development and unit testing, and the unit is ready to be baselined and integrated into the system.

Date: _______ System Developer: ____________________

DEVELOPMENT ISSUES CHECKLIST

The project team uses this checklist to ensure that specific issues have been addressed. Consideration is given to the questions below for each system project. Action may not be required for some items.

___ Has the software unit development folder been completed?

___ Have walkthroughs been conducted prior to coding?

___ Does the detail design satisfy bureau programming standards?

___ Does the code carry out the design intent and standard practices?

___ Have unit tests been conducted to verify compliance with design intent and programming standards?

___ Have job control commands for integration test, acceptance test, and the production environment been prepared and validated?

___ Are commercial software products to be concurrently tested?

___ Have security functions been performed according to project specifications?

___ When errors are found in a particular module, are there efforts to find other, perhaps related or subsequent errors in the same module or segment?

___ Is the test document complete and well organized?

___ Is there confirmation that all program data definitions are derived from the repository?

Exhibit 7-1: Development Issues Checklist

TEST READINESS REVIEW CHECKLIST

1. Date/time of meeting ____________.

2. Attendants:
   Project Manager _______________
   System Developer _______________
   System Tester _______________
   Quality Assurance Manager _______________
   Configuration Manager _______________
   Programmer Analyst _______________
   Systems Operations Manager _______________
   Database Administration _______________
   Security Manager _______________
   Project User _______________
   Data Administrator _______________

3. Were the following presented for review or their absence justified:

   Test Files/Test Data/Database ____________
   Test Results ____________
   Object Code ____________
   Source Code _____________
   Load Modules ____________
   Software Development Folder ____________
   Developer's Certification Statement ____________
   Updated User Manual ____________
   Updated Operators Manual _____________
   Updated Conversion Plan ____________
   Updated Project Plan ____________
   Updated Data Management Plan ____________
   Updated Configuration Plan ____________
   Updated Contingency Plan ____________
   Updated Risk Assessment Plan ____________
   Updated Computer Security Plan ____________
   Updated Quality Assurance Plan ____________
   Updated Test Plan ____________

Exhibit 7-2: Test Readiness Review Checklist

TEST READINESS REVIEW CHECKLIST

Updated Training Plan	____________
Updated Implementation Plan	____________
Updated Cost-Benefit Analysis	____________
Final Source Language Data Definitions Members	____________
Final External Design Document	____________
Final Internal Design Document	____________
Final Acquisition Plan	____________

4. Is there evidence of successfully completing the software development test? (Yes/No) If no, explain.

5. Was the document as presented found to be acceptable? (Yes/No) If no, explain.

6. Is the configured system ready for acceptance test? (Yes/No) If no, explain.

7. Is there a system developer certification statement to show evidence that integration testing was successfully completed, including a list of all know deficiencies? (Yes/No) If no, explain.

8. Other issues raised.

Exhibit 7-2: Test Readiness Review Checklist (cont'd)

CHAPTER 8. TEST PHASE

A. GENERAL

The primary purpose of the **test phase** is to prove that the developed system satisfies the requirements defined in the functional requirements document. The secondary purpose is to perform an integrated system function as specified by the design parameters. This function will be the responsibility of the quality assurance staff and will be heavily supported by the user participants.

The prerequisites are the functional requirements document, project plan and schedule, system baseline software and documents, and a test plan containing all test requirements and schedules.

B. TASKS AND ACTIVITIES

The following activities are performed during this phase. These activities may be expanded or deleted depending on the size of the proposed system.

- Conduct unit/module, subsystem integration, system qualification, system acceptance, and security tests
- Prepare test analysis report to document the results of each formal test
- Initiate user training
- Finalize program modules
- Finalize manuals and plans

C. ROLES AND RESPONSIBILITIES

Personnel involved in this phase are the project manager, project user(s), quality assurance manager, security manager, configuration manager, systems operations manager, programmer analyst, data administrator, capacity planning manager, system developer, system tester, telecommunications manager, and data base administrator.

D. DELIVERABLES, RESPONSIBILITIES, AND ACTION

The deliverables listed below must be available for review at the end of this phase. Sample descriptions of the test analysis report, test problem report and test analysis approval determination are provided in the documentation section of this chapter.

DELIVERABLES AND PRODUCTS	PRIME RESPONSIBILITY
Test Analysis Report (Exhibit 8-1)	Security Manager
Test Problem Report (Exhibit 8-2)	System Developer
Test Analysis Approval Determination (Exhibit 8-3)	Project Manager
Project Plan (revised)	Project Manager
Cost-Benefit Analysis (revised)	Project Manager
Computer Security Plan (revised)	Security Manager
Software Development Folder (revised)	Program Analyst
Configuration Management Plan (revised	Configuration Manager
Contingency Plan (revised)	Security Manager/Project User
Program Modules (finalized)	Programmer Analyst
Operators Manual (finalized)	Systems Operations Manager
User Manual (finalized)	Project User
Training Plan (finalized)	Project User
Maintenance Manual (finalized)	Systems Operations Manager
Conversion Plan (finalized)	Project Manager
Implementation Plan (finalized)	Project Manager
Data Management Plan (finalized)	Data Administration
Risk Assessment Plan (finalized)	Security Manager
Quality Assurance Plan (finalized)	Quality Assurance Manager
Test Plan (finalized)	Security Manager
Test Files (finalized)	Quality Assurance Manager
Developer's Certification Statement (finalized)	System Developer

===============================

MANAGEMENT ACTION	PRIME RESPONSIBILITY
Obtains end-of-phase concurrence from user	Project Manager
Certifies system operating according to specifications	Quality Assurance Manager/ Project User/Project Manager/ System Developer
Authorizes Implementation Phase to begin	Steering Committee

E. ISSUES FOR CONSIDERATION

During the **test phase** there are specific issues to be addressed. These issues are included in the Test Issues Checklist in Exhibit 8-4. The checklist is used as a reference for the project manager to ensure objectives of this phase are met. These issues are independent of the project size and effort. The checklist is used for conducting the Test Analysis Review as described in Section F of this phase.

F. REVIEW ACTIVITY

The Test Analysis Review is performed during the **test phase**. Deficiencies are recorded in the report. A Test Analysis Review Checklist is provided at Exhibit 8-5. A number of project team members participate in the test analysis review, at a minimum, the project manager and project user. The test analysis review checklist indicates the other project team members who may be requested to participate in the review. Depending on the size and complexity of the project, the review may be conducted by a review committee.

G. DOCUMENTATION - TEST ANALYSIS REPORT

The test analysis report documents each type of software testing: unit/module, subsystem integration, system qualification, system acceptance, and security, as defined in the test plan. The report records the results of the tests, presents the capabilities and deficiencies for review, and provides a means of assessing whether the software can progress to the next stage of development or testing. The report for each type of test is added to the software development folder for the module or system being tested. Reports are created as required in the remaining phases. The set of test analysis reports provides a basis for assigning responsibility for deficiency correction and follow-up, and for preparation of a statement of project completion.

Test problem report forms are generated as required and attached to the test analysis reports during testing at the integration level and higher. The disposition of problems found, starting with integration testing, will be tracked and reported under configuration control.

A test analysis approval determination form is first generated in the Test Readiness Review meeting at the end of the **development phase**, after unit/module and subsystem integration testing has been completed. This form is

1. Introduction

a. Purpose and Scope
b. Project References
c. Glossary
d. Security

2. Test Analysis

a. Test (repeat for each test)

(1) System Functions
(2) Functional Capability
(3) Performance Capability

3. Software and Hardware Requirements Findings

a. Requirement Number and Name (repeat)

(1) Findings
(2) Limitations

4. Summary and Conclusions

a. Demonstrated Capabilities
b. System Deficiencies
c. System Refinements
d. Recommendations and Estimates
e. Test Problem Report
f. Test Analysis Approval Determination

Exhibit 8-1: Test Analysis Report Outline

also generated with the Test Analysis Reports created for each type of testing at the level of system qualification test and higher. The test analysis report is outlined at Exhibit 8-1.

1. **Introduction**

   a. **Purpose and Scope**. Identify the test and the software application system being tested, and describe the purpose and scope of this test analysis report. The report summarizes the results of tests already conducted and identifies testing that remains to be conducted. List the specific objectives of the test, if appropriate. Provide a brief summary of the project objectives, and identify the project sponsor and users.

   b. **Project References**. Provide a bibliography of key project references (and alternate if appropriate) and deliverables applicable to software testing of the system. These might include the functional requirements document, user manual, operators manual, maintenance manual, test plan, and prior test analysis reports.

   c. **Glossary**. Define all terms and provide a list of abbreviations used in the test analysis report that the reader may be unfamiliar. If the list is several pages in length, it may be placed as an appendix.

   d. **Security**. Describe any security considerations associated with the system/module being tested, the test analysis, and the data being handled, such as confidentiality requirements, audit trails, access control, and recoverability. If this test analysis report is not documenting the formal security test, also summarize the security capabilities included in the system/module test and itemize the specific security deficiencies detected during the conduct of the test. The results of specific tests, system findings, and a thorough analysis of deficiencies, along with recommendations, will be covered in the forthcoming sections. Reference those portions of this document which specifically address system security issues. If no deficiencies were detected during the system/module test, state this fact.

2. **Test Analysis**

   Describe the results of each test performed. If there is a large number of tests, place them in an appendix. Tests at each level should include verification of access control and system standards, functionality, and error process. Repeat the subsections of this section for each test performed.

a. **Test Name**. Describe the test performed for the specified unit, module, subsystem, or system. For each test, provide the subsections below.

(1) **System Functions**. Provide a brief high-level description of the functions tested and describe the system capabilities designed to satisfy these functions. Describe each system function separately.

(2) **Functional Capability**. Describe the capability to perform each function as it has been demonstrated in the test. It assesses the manner in which the test environment may be different from the operational environment and the effect of this difference on the capability.

(3) **Performance**. Compare quantitatively the performance characteristics of the software with the criteria provided in the test plan, where applicable. Identify the deficiencies, limitations, and constraints detected in the system during the testing with respect to each function. A test history or log can be included as an exhibit, if appropriate.

3. **Software and Hardware Requirements Findings**

Summarize the test results, organized according to the detailed numbered requirements listed in the test plan under **Traceability**. Each numbered requirement is described in a separate paragraph(s) consisting of the subsections listed below. Repeat the subsections of this section for each numbered requirement within the project.

a. **Requirement Number and Name**. The requirement number provided in the title to this section is the number from the requirements traceability matrix in the test plan and the name provided is the requirement's short name.

(1) **Findings**. Briefly describe the function of the requirement, including the software and hardware capabilities designed to satisfy this requirement. It states the findings as to the demonstrated capabilities from one or more tests.

(2) **Limitations**. Describe the range of data values tested, including both dynamic and static data. Identify the deficiencies, limitations, and constraints detected in the software and hardware during the testing with respect to this requirement.

4. Summary and Conclusions

Provide an overview and summary analysis of the testing program for each level. Describe the overall capabilities and deficiencies of the testing software module or system.

a. **Demonstrated Capabilities**. Describe the capabilities of the software as demonstrated by the tests. Where tests were intended to demonstrate fulfillment of one or more specific performance requirements, present findings showing a comparison of the results with these requirements. Include an assessment of the effects of any differences in the test environment versus the operational environment that may have had an effect on this demonstration of capabilities. In addition, provide a statement, based on the results of the system/module test concerning the adequacy of the system/module to meet overall security requirements.

b. **System Deficiencies**. Describe the deficiencies of the software as demonstrated by the tests. Generate test problem reports for each deficiency as required. If the test problem reports are tracked in an automated database, include reports extracted from the database as an appendix. Identify all problems by name and number when placed under configuration control. Describe the cumulative or overall impact on performance of all detected deficiencies.

c. **System Refinements**. Itemize any improvements that can be realized in system design or operation, as determined during the test period. Accompanying each improvement or enhancement suggested should be a discussion of the added capability it provides and the impact on the system design. Identify these improvements by name and number when placed under configuration control.

d. **Recommendations and Estimates**. Provide a summary statement describing the overall readiness for system implementation. For each deficiency, address the impact on system performance and design of correcting or not correcting that item. Include any estimates of time and effort required for correction of each deficiency, and any recommendations on the following:

- o The urgency of each correction
- o Parties responsible for corrections
- o Recommended solution or approach to making the corrections should be made

If the test problem reports are tracked in an automated database, include this information in that database, and include a report extracted from the database as an appendix.

e. **Problem Report**. Generate multiple copies of test problem reports (sample in Exhibit 8-2) related to the deficiencies found in the test results and track problem(s) until resolved. The test problem report will vary according to the information system development project, its scope and complexity, and by bureau.

f. **Test Analysis Approval Determination**. Generate one copy of this determination (sample in Exhibit 8-3) following the Text Problem Report as a final result of test reviews and testing levels above integration test. In addition, use this form as a stand-alone document to indicate awareness of the results of a specific test and, eventually, the user's acceptance. This form briefly summarizes the perceived readiness for implementation of the software. In the case of system acceptance test, it serves as the Government's recommendation.

TEST PROBLEM REPORT

TO: ____________________________

FROM: ____________________________

PREPARER/CONTACT:_________________ **PHONE:** ___________

PROGRAM BEING TESTED:_____________________________________

DESCRIPTION OF PROBLEM

DISPOSITION OF PROBLEM

Action Taken and Date Corrected

Risk Impact if Problem Not Corrected

Changes Required for Existing Documentation

SIGNATURES: _____________ _____________

Project Manager System Developer

_____________ _____________

Date Date

Exhibit 8-2: (Sample) Test Problem Report

TEST ANALYSIS APPROVAL DETERMINATION

DATE: ____________________

FROM: ____________________

TO: ____________________

We have reviewed the test material for the following project request:

TITLE:

We recommend:

( ) a. Full acceptance. No problems encountered.

( ) b. Full acceptance. The attached Test Analysis Report describes the problems encountered which are now corrected.

( ) c. Full implementation with later correction. The attached Test Analysis Report describes the impact and expected results of this alternative.

( ) d. Partial implementation. The attached Test Analysis Report describes the impact and expected results of this alternative.

( ) e. Rejection. The attached Test Analysis Report describes the reasons rejected.

SIGNATURE: ________________ ________
Project Manager Date

SIGNATURE: ________________ ________
Project Sponsor Date

Exhibit 8-3: (Sample) Test Analysis Approval Determination

TEST ISSUES CHECKLIST

The project team uses this checklist to ensure specific issues have been addressed. Consideration is given to the questions below for each system project. Action may not be required for some items.

Test Phase

___ Is a test plan present?

___ Is the test plan completed as described for validation in the test plan description?

___ Has the capability for entry of the transaction or master files been provided?

___ Have transaction files been provided?

___ Does the validation process accept the input required in production?

___ Have the proper number and type of records been selected for the database process?

___ Does output reviewed agree with output requested?

___ Have all findings, problems encountered, and recommendations been documented?

___ Is the test analysis report attached?

___ Does such a report exist for each level of testing as required, i.e., unit/module test, system integration test, system qualification test, system acceptance test, security test?

___ Are all validated programs released to operation?

Exhibit 8-4: Test Issues Checklist
TEST ISSUES CHECKLIST

Test Analysis Report

___ Has a test analysis report been developed which documents the test analysis results and findings?

___ Does this report present the demonstrated capabilities and deficiencies of the system?

___ Has the report been used to prepare a statement of the system's readiness for implementation?

___ Have unit/module tests been conducted and a Test Analysis Approval Determination prepared?

___ Was the System Qualification Test conducted and Test Analysis Approval Determination prepared?

___ Does the system acceptance test evaluate both manual and automated procedures?

___ Once system acceptance test has been completed, is a written certification that the system performs in accordance with all functional and performance specifications required?

___ Has the system acceptance test been conducted and Test Analysis Approval Determination prepared?

Exhibit 8-4: Test Issues Checklist (cont'd)

TEST ANALYSIS REVIEW CHECKLIST

1. Date/time of meeting ____________.

2. Attendants:
Project Manager ________________
Project Sponsor ________________
System Developer ________________
Project User ________________
System Tester ________________
Quality Assurance Manager ________________

Configuration Manager ______________
Telecommunications Manager ______________
Security Manager ______________
Data Administrator ______________
Database Administrator ______________

3. Were the following presented for review or their absence justified:

Test Problem Reports ____________
Test Analysis Report ____________
Test Analysis Approval
 Determination ____________
Updated Project Plan ____________
Updated Cost-Benefit Analysis ____________
Updated Configuration Plan ____________
Updated Contingency Plan ____________
Updated Computer Security Plan ____________
Updated Software
 Development Folder ____________
Final Quality Assurance Plan ____________
Final Operators Manual ____________
Final User Manual ____________
Final Maintenance Manual ____________
Final Training Plan ____________
Final Implementation Plan ____________
Final Conversion Plan ____________

Exhibit 8-5: Test Analysis Review Checklist

TEST ANALYSIS REVIEW CHECKLIST

Final Data Management Plan ____________
Final Program Modules ____________
Final Developer's Certification
 Statement ____________
Final Test Plan ____________
Final Test Files ____________
Final Risk Assessment ____________

4. Was the test analysis report presented and found to be acceptable? (Yes/No) If no, explain.

5. Was the decision made to implement the system? (Yes/No) If no, explain.

Exhibit 8-5: Test Analysis Review Checklist (cont'd)

CHAPTER 9. IMPLEMENTATION PHASE

A. GENERAL

The **implementation phase** builds on the results of all prior phases. The **implementation phase** includes efforts required to prepare for implementation of the new system, implementation of the system in a production environment, and resolution of any problems identified during the implementation process. During the **implementation phase**, the system or system modifications are installed and made operational in a production environment; user training is conducted; data conversion is completed; and the system is turned over to the user. A successful system implementation is critical to the operation phase.

This phase implements the components of the system developed and/or acquired during the development phase. This phase also uses approaches for data conversion, and training. Any deficiencies in implementing the system will handicap the initial use of the system. Therefore, it is important that deficiencies are addressed prior to full implementation as it will be more costly to resolve deficiencies once the system is turned over to the user. Modifications to the system should be completely documented in order to provide accurate documentation to system users, operators, and other affected personnel. At the end of this phase, the production baseline is established and it consists of the production system, database(s) and data dictionary.

A number of **project approach decisions** are made in this phase. **Project approach decisions** include: what specific post-implementation methodologies and tools will be used, which changes or enhancements should be made now, does the information system requirement continue to exist, does the implemented system address the requirement sufficiently to be put in operation, and are sufficient funding and other resources available for the remainder of the life cycle?

The prerequisites are the ISLC baseline documentation and baseline software.

B. TASKS AND ACTIVITIES

The following activities are performed during this phase. These activities may be expanded or deleted depending upon the requirements of the proposed system.

- Conduct system tuning tests
- Complete user and operator training (include other affected personnel)

- o Load design data dictionary to production data dictionary
- o Install system in production environment
- o Convert the database(s) and data to the production environment
- o Confirm that the system is ready for operation
- o Continue configuration accounting and change control
- o Work with data administrator to enter information about the system into the data repository
- o Certify system security and readiness features (certification and accreditation)
- o Complete procurements
- o Perform operational reviews
- o Obtain approval of implementation phase

C. ROLES AND RESPONSIBILITIES

Team members involved during this phase include the project manager, project sponsor, system developer, project user, programmer analyst, data administrator, data base administrator, quality assurance manager, security manager, configuration manager, capacity planning manager, telecommunications manager, and systems operations manager. The quality assurance manager reviews the system to ensure that the application software has been properly tested and is ready to move into the production environment. The security manager reviews the system to ensure that adequate security measures have been built into the system, commensurate with the sensitivity of the system.

D. DELIVERABLES, RESPONSIBILITIES, AND ACTION

The deliverables listed below must be available for review at the end of this phase. The quality assurance and security managers are responsible for many of the deliverables.

DELIVERABLES AND PRODUCTS	PRIME RESPONSIBILITY
Delivered System	Project Manager
Security Certification Statement (Exhibit 9-1)	Security Manager
Security Accreditation Statement (Exhibit 9-2)	Security Manager
Production version of data repository	Data Administrator
Project Plan (revised)	Project Manager
Cost-Benefit Analysis (revised)	Project Manager
Configuration Management Plan (revised)	Configuration Manager
Software Development Folder (finalized)	Programmer Analyst

Computer Security Plan (finalized)	Security Manager
Test Analysis Report (finalized)	Security Manager
Test Problem Report (finalized)	System Developer
Test Analysis Approval Determination (finalized)	Project Manager
Contingency Plan (finalized)	Security Manager/Project User

================================

MANAGEMENT ACTION	PRIME RESPONSIBILITY
Formally accepts system	Project User

E. ISSUES FOR CONSIDERATION

During the **implementation phase** there are specific issues that should be addressed. These issues are included in an Implementation Issues Checklist at Exhibit 9-3. The checklist is used as a reference for the project manager to ensure objectives of this phase are met. These issues are independent of the project size and effort. The checklist is also for use in conducting the Revalidation Review as described in Section F of this phase.

F. REVIEW ACTIVITY

The Revalidation Review is performed during the **implementation phase**. At the end of this phase, a revalidation report may be prepared to characterize the user acceptance of the system. This report may make recommendations for system modifications to correct errors in implementation, improve user satisfaction, or improve system performance. For contractor development, analysis must be performed to determine if additional activity is within the scope of the original contract.

User acceptance testing typically takes one of the following four forms:

Parallel Operation. The old and new systems are operated in parallel until the new system successfully performs all user requirements.

Staged Conversion. The system is installed in parts, e.g., by organizational unit or by system function.

Pilot Operation. An immediate conversion is made to the new system; the old system is maintained as backup. Where multiple sites exist, one or two are usually selected for the pilot operation while the others continue to use the old system until the new system has been proven.

Immediate Conversion. The old system is completely discontinued following installation of the new system.

G1. DOCUMENTATION - SECURITY CERTIFICATION STATEMENT

A security certification refers to the technical evaluation of a system or an application to verify that the installed security safeguards are adequate and work effectively for the system or application. To verify that the installed security safeguards are adequate and work effectively requires that these safeguards be tested. Certification takes place after system tests have been completed and the results of the tests indicate that the system or application meets all applicable Federal and Treasury policies, regulations, and standards. The security certification statement is signed by the security manager. Exhibit 9-1 is an example of a security certification statement.

G2. DOCUMENTATION - SECURITY ACCREDITATION STATEMENT

Accreditation refers to the confirmation by a responsible official that a system that is processing sensitive information has an acceptable level of risk and the information being processed is adequately protected. The completion of a risk assessment and certification of the system is consequently performed as part of the accreditation of the system. Accreditation can also be referred to as the certification of an application. The security accreditation statement is signed by the quality assurance manager and the security manager. Exhibit 9-2 is an example of an accreditation statement.

SECURITY CERTIFICATION STATEMENT

I have carefully considered the security and integrity requirements and vulnerabilities of the ___________ system. Base on the review of the security requirements, vulnerabilities, and potential threats against the security and integrity measures implemented or planned, I have determined that the security capabilities are adequate for known risks, and that the statement of systems security and integrity requirements is a fair representation of the level of security and integrity of the _____________ system, except for the weaknesses noted in the certification report.

Based upon the report and my judgment, I hereby certify, subject to the corrections recommended in the certification report, that the ____________ system meets the documented and approved security requirements.

Weighing the remaining residual risks against operational requirements, I recommend that the ______________ system be accredited for continued operation, and that the recommendations included in the certification report be implemented.

Signed: ________________ Date: ______
Security Manager

Exhibit 9-1: Security Certification Statement

SECURITY ACCREDITATION STATEMENT

We have carefully considered the security and integrity requirements and the vulnerabilities of the ____________ system. Based on review of the requirements, vulnerabilities, and potential threats against the system, and security integrity measures implemented or planned, we have determined that the security capabilities are adequate for known risks, and that the statement of systems security and integrity requirements worksheet is a fair representation of the level of security and integrity of the _____________ system as of ____________.

We have determined that the requirements in the Systems Security chapter of the Treasury <u>Security Manual</u>, TD P 71-10, have been satisfied, and the ____________ system is in the best interest of the Department of the Treasury. Based on our authority and judgment, and weighing the remaining residual risks against operational requirements, we authorized operation (or continued operation) of the _____________ system. We further authorized initiation of the following corrective actions to enhance security and integrity.

(CORRECTIVE ACTIONS)

Signed: __________________ Date: ______
Quality Assurance Manager

Signed: __________________ Date: ______
Security Manager

Exhibit 9-2: Security Accreditation Statement

IMPLEMENTATION ISSUES CHECKLIST

The project team uses this checklist to ensure specific issues have been addressed. Consideration is given to the questions below for each system project. Action may not be required for some items.

___ Has all software been validated and documented?

___ Has the validated version of software and documentation been provided for implementation?

___ Has a production schedule been provided?

___ Has an operators manual been provided?

___ Has a user manual been provided?

___ Has training been completed for initial startup personnel?

___ Have data bases/files been initiated for production?

___ Is all teleprocessing equipment in place?

___ Are points of contact with phone numbers listed for the specific types of problems that may occur?

___ Are all user service agreements in place?

___ Has the project team validated the system is ready for production?

___ Has a maintenance log been established?

___ Do the programs released to production match the programs validated?

Exhibit 9-3: Implementation Issues Checklist

IMPLEMENTATION ISSUES CHECKLIST

___ Have all security documents been completed?

___ Have preparations been made to migrate to the production environment?

___ Has repository application data been frozen?

___ Have approvals been received on test analysis approval determination document prior to implementation?

___ Has a contingency plan been developed or refined to cover the new application or enhancements?

___ Has the new application or enhancement been included in the contingency plan testing procedures?

--- Has an analysis of potential versus estimated benefits and costs been conducted for a fully operational system?

--- Has an analysis of costs and benefits for the fully implemented system been conducted versus system and process changes that have occurred during the development?

Exhibit 9-3: Implementation Issues Checklist (cont'd)

CHAPTER 10. OPERATIONS, MAINTENANCE AND DISPOSITION PHASE

A. GENERAL

In this phase the system is fully operational. Some systems are operated completely by users, particularly systems that operate on microcomputers. It is essential that users are fully aware of and carry out their responsibilities for all facets of operations, especially performing system backups. Providing user support is usually an ongoing activity. New users will require training to effectively use the system, and other users may require assistance as well. Emphasis of this phase is to ensure the user needs have been met and the system continues to perform as specified in the operational environment. This phase continues until the system is retired from use, which is the end of the system life cycle.

Maintenance activities for the system occur to ensure that any previously undetected errors are fixed. Maintenance activities take advantage of hardware upgrades or new releases of system software and application software packages used to operate the system (e.g., upgrades and releases). Identifying potential modifications needed to ensure that the system continues to operate as intended and produces quality data. Determine whether modifications to the system and database(s) are needed to resolve errors or performance problems, or to provide new capabilities. New capabilities may take the form of routine maintenance, or may constitute enhancements to the system or database(s) which respond to user request for new/improved capabilities. The maintenance manual is used and updated accordingly.

Disposition activities occur at the end of the system life cycle. The activities ensure the orderly termination of the system and preserve vital information about the system so that some or all of it may be reactivated in the future if necessary. A disposition plan is prepared to address all facets of archiving, transferring and disposing of the system and the data. Particular emphasis is given to proper preservation of the data processed by the system so that the data is effectively migrated to another system or archived in accordance with applicable records management regulations and policies, for potential future access. The disposition activities preserves information not only about the current production system, but also about the evolution of the system through its life cycle.

A number of **project approach, project execution and project continuation decisions** are made in this phase. **Project approach decisions** include: what evaluations of the system/data should be conducted, what new or additional user support activities are needed, what improvements in system/data functionality, quality, and/or performance are needed and what adjustments to the system/data management approach are needed?

Project execution decisions include: what changes/enhancements to the system/database(s) are needed, should a particular enhancement be implemented during this phase or given its own life cycle? Decisions at the end of the system life include: what will be the system termination date, which software components should be preserved, which data should be preserved, what should be done with remaining equipment, and how should life cycle products be archived?

Project continuation decisions include: does the information system requirement continue to exist, does the production system address the requirement sufficiently to be continued in operation and are sufficient funding and other resources available for the remainder of the life cycle?

Review activities occur several times throughout this phase. Each time the system is reviewed, one of two decisions is made: either the system is continued in operation (with or without modifications) or the system is to be terminated, and its functions and data transferred to other systems. A Post-Implementation Review is conducted to ensure that the system functions as planned and expected; to verify that the system cost is within the estimated amount; and to verify that the intended benefits are derived as projected. Normally, this is a one-time review and it occurs after major implementation, but also after a major enhancement to the system.

The Periodic System Review is also conducted in this phase and it is a continuous system review. The Periodic System Review is performed to evaluate system performance, user satisfaction with the system, adaptability to changing business needs, and new technologies that might improve the system. The periodic system review is diagnostic in nature and can lead to development or maintenance activities. Any major system modifications needed after the system has been implemented will follow the life cycle process from planning through implementation. A project plan, including feasibility study, will identify modification to existing system documentation (change pages) rather than new system documentation (e.g., functional requirements document, internal design document, etc.) The appropriate reviews and testing will be conducted, based on the scope of the modification.

The following are prerequisites for this phase:

- ISLC baseline documentation;
- production software;
- system hardware;
- operations schedule;
- user service agreements;

- o maintenance log;
- o manuals; and
- o periodic re-certification and accreditation.

B. TASKS AND ACTIVITIES

The following activities are performed for operations and maintenance of the system. These activities may be expanded or deleted, depending upon the size of the system.

- o Closeout the system development project
- o Identify system problems
- o Identify potential modifications needed to the system
- o Develop and implement maintenance changes
- o Conduct configuration accounting and change control
- o Monitor system/database performance
- o Conduct post-implementation review
- o Conduct periodic system review
- o Conduct user satisfaction review (if appropriate)

The following activities are performed at the end of the system life cycle. The disposition activities ensure the orderly termination of the system, and preserve vital information about the system so that some or all of it may be reactivated in the future if necessary. Particular emphasis is given to proper preservation of the data processed by the system, so that the data is effectively migrated to another system or disposed of in accordance with applicable records management regulations and policies, for potential future access. These activities may be expanded or deleted, depending on the size of the system.

- o Prepare system disposition plan
- o Notify users of disposition date
- o Archive or transfer data
- o Archive or transfer software components
- o Archive life cycle deliverables
- o End the system in a planned, orderly manner
- o Dispose of equipment

C. ROLES AND RESPONSIBILITIES

The levels of support from the project team are dependent upon the project type. The project sponsor takes the lead role for activities in this phase. Users are responsible for identifying system problems. Records management staff is involved when disposition of the system is proposed.

D. DELIVERABLES, RESPONSIBILITIES, AND ACTION

Descriptions of a post-implementation review, a periodic system review, a disposition plan, and a user satisfaction review are provided in the documentation section of this chapter.

DELIVERABLES AND PRODUCTS	PRIME RESPONSIBILITY
Post-Implementation Review Report (Exhibit 10-1)	Project Sponsor
Periodic System Review Report (Exhibit 10-2)	Project Sponsor/User
User Satisfaction Review (Exhibit 10-3)	Project Sponsor
Disposition Plan	Project Manager/Quality Assurance Manager
Problem Reports	Project User
Change Control Reports	Configuration Manager
Project Plan (finalized)	Project Manager
Cost-Benefit Analysis (finalized)	Project Manager
Configuration Management Plan (finalized)	Configuration Manager

==============================

MANAGEMENT ACTION	PRIME RESPONSIBILITY
Closeout of the system development project	Project Manager

E. ISSUES FOR CONSIDERATION

During this phase there are specific issues to be addressed. These issues are included in the Operations, Maintenance and Disposition Issues Checklist in Exhibit 10-4. The checklist is used as a reference for the system developer to ensure objectives of this phase are met. The checklist is also used in conducting the Post-Implementation Review and the Periodic System Review as described in Section F of this phase.

F. REVIEW ACTIVITY

The Post-Implementation Review and Periodic System Review are performed during this final phase. The Post-Implementation Review is conducted within six months after implementation of the system and ensures that the system functions as planned, that benefits are derived and that the system is within the estimated costs. The Periodic System Review, however, is conducted as long as the system is in operation (continuously) to ensure that system performance is optimized and that users are satisfied with the system. Conduct a user satisfaction review if appropriate.

Participants for both reviews may include the project manager, project sponsor, and project user(s). Depending on the size and complexity of the system, the reviews may be conducted by a review team or an external independent team.

G1. DOCUMENTATION - POST-IMPLEMENTATION REVIEW REPORT

The post-implementation review is used to evaluate the effectiveness of the system development after the system has been in production for a period of time. The objectives are to: determine if the system does what it is suppose to do - does it support the user as required in an effective and efficient manner; assess how successful the system is in terms of functionality, performance, and cost versus benefits; and assess the effectiveness of the life cycle development activities that produced the system. The review results can also be used to strengthen the system as well as system development procedures.

The review is scheduled to follow the release of a system or system revision by an appropriate amount of time to allow determination of the effectiveness of the system. A representative from the functional development group or other member of the major user organization participates in the review. The project sponsor ensures that all documentation needed for the review and that all personnel needed to participate in the review are accessible.

The reviewer and an assigned team collect the information needed for the post-implementation review by interviewing end users and their managers, system administrators, and computer operations personnel. The report is then prepared and provided to the user organization who requested it and the information systems organization, who may jointly use the findings to initiate other actions.

The post-implementation review is a free-form report and not all sections will be relevant or necessary to the final product. A description of the post-implementation review is outlined at Exhibit 10-1.

1. **Introduction**

   a. **Project Identification**. Provide the identifying information associated with the project including the applicable project control code, system acronym, and system title.

   b. **Requesting Organization**. Provide the name of the requesting organization.

   c. **History of the System**. Describe the system's history and predecessor, if any. State the mission needs and information requirements including how the system is expected to help users.

   d. **Functional System Description and Data Usage**. Describe what the system does functionally and how the data is used by the system.

2. **Evaluation Summary**

   The purpose of this section is to provide a summary of the overall adequacy and acceptance of the system.

   a. **General Satisfaction With the System**. Describe the users' experience with the implemented system. Comments should address:

      o The level of user satisfaction;

1. Introduction
- a. Project Identification
- b. Requesting Organization
- c. History of the System
- d. Functional System Description and Data Usage

2. Evaluation Summary
- a. General Satisfaction with the System
- b. Current Cost-Benefit Justification
- c. Needed Changes or Enhancements
- d. Projected Cost-Benefit Justification

3. Analysis and Implementation
- a. Purpose and Objectives

   b. Scope
   c. Benefits
   d. Development Cost
   e. Operating Cost
   f. Training

4. Outputs

   a. Usefulness
   b. Timeliness
   c. Data Quality

5. Security

   a. DataBase Protection
   b. Disaster Recovery
   c. Controls
   d. Audit Trails
   e. Allowed Access

6. Computer Operations

   a. Control of Work Flow
   b. Scheduling
   c. User Interface
   d. Computer Processing
   e. Peak Loads

7. Maintenance Activities

   a. Activity Summary
   b. Maintenance Review
   c. System Maintenance

Exhibit 10-1: Post-Implementation Review Outline

- The strengths of the system including specific areas of success;
- Any problems;
- Frequently used features;
- Infrequently used features;
- Features not used; and
- Suggested improvements.

b. **Current Cost-Benefit Justification**. Assess whether the system is paying for itself. Base the assessment on the anticipated benefits and costs projected during the project concept development and revised during the previous phases of the ISLC. This section is intended merely to recap the costs and benefits; provide details of costs and benefits in other sections. Comments should address:

(1) The extent of the benefits and whether they are reported to be less or greater than those projected in the concept analysis and functional requirements report;

(2) Whether any difference is permanent or will change over time; and

(3) Whether the system is or will be cost-justifiable.

c. **Needed Changes or Enhancements**. Gauge the magnitude of effort needed to change or improve the system. Describe the nature and priority of the suggested changes; more detail is to be provided in other sections. Comments should address:

(1) The suggested changes;

(2) The scope of the changes; and

(3) The resource requirements to effect the changes.

d. **Projected Cost-Benefit Justification**. Project whether future use of the system, after any needed or desired changes, will continue to be economical. Focus on ongoing costs experienced since the system was placed into operation. Comments should address:

(1) The projected benefits and operating costs; and

(2) The extent of economic feasibility.

3. Analysis and Implementation

The purpose of this section is to gauge the completeness of the concept analysis/functional specifications and of implementation according to the study.

a. **Purpose And Objectives**. Evaluate the adequacy of the original definition of purpose and objectives presented in the functional requirements reports and whether the objectives were achieved during implementation. Include an evaluation of whether any objectives have changed or should have changed. Comments should address:

(1) Extent to which goals were met;

(2) The level of the objective definition;

(3) Extent to which objectives were met; and

(4) Possible changes to the objectives.

b. **Scope**. Analyze whether proper limits were established in the feasibility study and whether they were maintained during implementation. Comments should address:

(1) Variations from the scope definition as agreed to in the concept analysis;

(2) The extent to which the scope was followed; and

(3) Any possible future changes to the scope.

c. **Benefits**. Analyze whether the benefits anticipated in the concept and functional analyses were realized. Detail all benefits, quantifiable or non-quantifiable, and any quantifiable resources associated with each. Comments should address:

(1) The adequacy of the benefit definition;

(2) The level of benefits realized;

(3) The anticipated benefits that can be realized; and

(4) The reason for the variance between planned and realized benefits, if any.

d. **Development Cost**. Determine the adequacy of the development cost estimated and any deviation between the estimated and actual development costs. Comments should address:

(1) The adequacy of the original and subsequent cost estimates;

(2) The actual costs, by type; and

(3) The reasons for any difference between estimated and actual costs.

e. **Operating Cost**. Analyze the adequacy of the operating cost estimates and any deviation between the estimate and the actual operating costs. Summarize the resources required to operate the system. Comments should address:

(1) The adequacy of the operating estimates;

(2) The actual operating costs; and

(3) The difference.

f. **Training**. Evaluate whether all levels of user training were adequate and timely. Comments should address:

(1) The timeliness of the training provided;

(2) The adequacy of the training;

(3) The appropriateness of the training;

(4) Identification of additional training needs by job category; and

(5) The ability of the personnel to use the training provided.

4. Outputs

The purpose of this section is to evaluate the adequacy and usefulness of the outputs from the system. Care must be taken to ensure that all reports are evaluated.

a. **Usefulness**. Measure the extent to which the users need the output of the system. Comments should address:

(1) Identification of the level of need as, for example:

(a) Absolutely essential;

(b) Important and highly desirable;

(c) Interesting; proves what is already known;

(d) Incomplete - does not provide all necessary information; and

(e) Unnecessary.

(2) Identification of information/reports needed but not currently generated by the system or unable to be obtained;

(3) Demonstration of the ability to do without the reports; and

(4) Alternatives for obtaining the information where improvements can be achieved.

b. **Timeliness**. Determine whether output production performance meets user needs. Comments should address:

(1) The frequency with which output arrives on time, early, and late; and

(2) The amount of follow-up needed to obtain the output.

c. **Data Quality**. Assess the need to provide for effective use of shareable data to enhance performance and system interoperability. Comments should address:

(1) Data accuracy; and

(2) Data reliability.

5. **Security**

The purpose of this section is to determine whether the system provides adequate security of data and programs. In addition to access security, procedures for backup, recovery, and restart should be reviewed.

a. **Data Protection**. Determine whether the security, backup, recovery, and restart capabilities adequately safeguard data including master, transaction and source. Online systems naturally require special techniques (e.g., logging). Comments should address:

(1) The adequacy of the security, backup, recovery, and restart procedures;

(2) The suggested changes; and

(3) The effort required to make the changes.

b. **Disaster Recovery**. Determine whether appropriate files, programs, and procedures are established to enable recovery from a disaster resulting in loss of data. Comments should address:

(1) The adequacy and currency of off-site storage procedures;

(2) The extent that procedures cover:

(a) Master data;

(b) Transaction data;

(c) Source programs;

(d) Object programs; and

(e) Documentation (e.g., systems, operations, user manuals).

(3) The results of any adequacy-of-recovery test.

c. **Controls**. Evaluate the adequacy of the controls on the data base, source documents, transactions, and outputs of the system. Review each area thoroughly for financial controls and file control counts. Comments should address:

(1) The level of controls present in the entire system and on each component (e.g., transaction batch, file);

(2) The adequacy of the controls; the strengths and possible areas for improvement; and

(3) The amount of resources required, if any, to obtain improvements.

d. **Audit trails**. Review the ability to trace transactions through the system and the tie-in of the system to itself. Comments should address:

(1) The thoroughness of the audit trails;

(2) The level of improvements necessary, if any; and

(3) The requirements of audit trails as outlined in the trusted criteria, e.g., C2 requirements, if any.

e. **Allowed Access**. Evaluate the adherence to restriction of access to data. State desired privacy criteria for the system and then evaluate how they have been followed up to this point. Comments should address:

(1) Established privacy criteria;

(2) Recommended privacy criteria;

(3) Adherence to and violations of privacy;

(4) The cost of providing this level of privacy; and

(5) The potential effect on individuals if the privacy criteria are not followed.

6. **Computer Operations**

The purpose of this section is to ascertain the current level of operational activities. Although the user point of view is primary to the post-implementation review, the computer operations view is also important to investigate.

a. **Control of Work Flow**. Evaluate the user interface with the data processing organization. The submittal of source material, the receipt of outputs, and any problems getting work in, through, and out of computer operations should be investigated. Comments should address:

(1) Any problems in getting the work accomplished;

(2) The frequency and extent of the problems;

(3) Suggested changes; and

(4) The effort required to make the changes.

b. **Scheduling**. Determine the ability of computer operations to schedule according to user needs and to complete scheduled tasks. Comments should address:

(1) Any problems in getting the work accomplished;

(2) The frequency and extent of the problems;

(3) Suggested changes; and

(4) The effort required to make changes.

c. **User Interface**. Analyze the useability of the system. The transaction throughput and error rate is included in this analysis. Comments should address:

(1) The volume of data processed (number of transactions);

(2) The number of errors made;

(3) The frequency of problems with the interface;

(4) The suggested changes; and

(5) The effort required to make the changes.

d. **Computer Processing**. Analyze computer processing issues and problems. Some areas to review are:

(1) The correct or incorrect use of forms and off-line files;

(2) The adequacy of instructions (e.g., forms lineup and proper responses on the console); and

(3) The extent of reruns, if any.

e. **Peak Loads**. Assess the ability of computer operations to handle peak loads and to resolve backlogs when they occur. Any off-loading that could be helpful should be investigated. Comments should address:

(1) The level of user satisfaction;

(2) The adequacy of the response time (for online systems);

(3) The effect of delays on online and/or batch systems;

(4) Suggested changes; and

(5) The effort required to make the changes.

7. Maintenance Activities

The purpose of this section is to evaluate maintenance activity involving the system.

a. **Activity Summary**. Provide a summary of maintenance activity to date. Include type, number of actions, and scope of changes required. Estimate a projected maintenance workload based on

findings of the review. Discuss the adequacy of maintenance efforts or whether major enhancement/revision is required.

b. **Maintenance Review**. Review completed and pending changes to the system. Provide conclusions regarding the benefits to be achieved by completing recommended changes. Provide conclusions about the amount of maintenance required based on activity which has occurred to date.

c. **System Maintenance**. Discuss the system maintenance based on the design, types of changes required, documentation, and knowledge about the system (both user and technical personnel).

G2. DOCUMENTATION - PERIODIC SYSTEM REVIEW REPORT

The purpose of the periodic system review is to assess the system's performance and user satisfaction. This review process occurs repeatedly to ensure that the system is performing cost-effectively and that it continues to meet the functional needs of the user. The report provides a description of the review process, its focus, and results. The report also may be used to document management approvals regarding further enhancements or development of the system under review. Depending on the timing and focus of the review, it may involve investigation of system response time, database capacity, newer technologies available, business functions, and continued user satisfaction with the system. The periodic system review report is outlined at Exhibit 10-2.

1. Introduction

a. Purpose

b. Scope

c. Project References

d. Points of Contact

e. Glossary

2. Review Process

a. System Overview

b. Functional System Description and Data Usage

c. Performance Review

(1) System Response
(2) System Capacity
(3) System Correctness
(4) Other

d. User Satisfaction Review

3. Findings

4. Recommendations

5. Approvals and Appendices

a. Approvals

b. Appendices

Exhibit 10-2: Periodic System Review Report Outline

1. **Introduction**

Provide a brief description of introductory material in this section. Whenever appropriate, other information may be added.

a. **Purpose**. Describe the purpose of the periodic system review report. Provide the name and identifying information about the system reviewed. Provide the timing of the review to differentiate the periodic system review reports created in the life of a system.

b. **Scope**. Delineate the boundaries of the system review. Since this review may address initial production performance and/or continued user satisfaction with the system, describe the specific aspects of the review conducted.

c. **Project References**. Provide a bibliography of key project references produced for this system.

d. **Points of Contact**. Identify the sponsoring bureau organization. Provide the name of the responsible organization(s) and titles of the staff that conducted the system review.

e. **Glossary**. Provide a glossary of all terms and abbreviations used in the report that may be unfamiliar to the reader. If it is several pages in length, it may be placed as an appendix.

2. **Review Process**

Provide an overview of the review process and its approach. This information may differ, depending on whether the system review focused on performance, user satisfaction, or both.

a. **System Overview**. Provide a general overview of the system reviewed. Examples of information that would be relevant to this section include:

- System name;
- Date of initial implementation;
- Date of latest modification
- Type of system (e.g., administrative, financial);

- Type of processing (batch, on-line, transaction processing);
- System diagram and narrative description;
- Number of computer programs within the system;
- Programming language(s) and database management systems used;
- Processing frequency;
- Total monthly processing hours;
- System origination (commercial off-the-shelf or bureau-developed);
- Testing methodology (test data, live data) for initial system tests;
- Testing methodology (test data, live data) for latest modification;
- Availability of test results;
- Date of last system review, if any;
- List of users; and
- List of issues identified in last system review.

Expand this list as necessary to include all important aspects of the system relevant to the system review. It is not necessary to provide information on all of the items in the list above if they are not relevant to the review.

b. **Functional System Description and Data Usage**. Describe what the system does functionally and how the data is used by the system.

c. **Performance Review**. Depending on the purpose of the review, address system response, capacity, correctness, and other pertinent performance factors.

(1) **System Response**. To evaluate the responsiveness of the system, it may be appropriate to turn on a system monitor on mainframe-based systems. For example, for a transaction processing system, data on the number of times each of the system's programs have been executed during a work day, week, or month should be collected as appropriate. The monitor may also provide data on the average and worst case delay experienced by the programs, and the average and worst case queue lengths. To evaluate the responsiveness of the system for LAN-based systems, it may be appropriate to place a monitor or protocol analyzer on the LAN.

(2) **System Capacity**. Examine the capacity of the system reviewed to determine whether any performance limitations result from operating the system near the limits of its capacity. For example, for mainframe computer applications using a DBMS, lack of main memory or selection of inappropriate buffer sizing during system generation could result in excessive disk reads and writes that would slow the applications' response. Similarly, a lack of adequate excess hard disk storage could result in large queues at disk controllers, substantially slowing the actual, observed average disk access time. On LAN based systems, hosting all applications on a server with only one large disk drive and controller could lead to bottlenecks in performance for LAN-based applications. In addition, there may be simple system capacity considerations, such as in an application hosted on a system that has only enough hard disk space available for a limited number of data records.

(3) **System Correctness**. Depending on the purpose of the review, it may be appropriate to examine the correctness of the system calculations, output, and reports. Presumably, this was done during unit testing and system testing. The intent of examining correctness during the periodic system review is to determine that the system is operating correctly with actual operational data inputs - since the

operational data may differ somewhat from the test data. Examples of items to be evaluated include:

- Values used for case codes;
- Correctness of field definitions;
- Values within data fields;
- Combinations of data fields;
- Calculations;
- Missing data;
- Extraneous data;
- Amounts;
- Units;
- Logic paths and decisions;
- Limits or reasonableness checks;
- Signs;
- Cross-footing of quantitative data; and
- Control totals.

If the system maintains an audit trail log of hardware and software failures, examine this log to determine the failure modes of the system.

(4) **Other**. Discuss the approach to any performance issues that are not easily categorized under the topics listed above.

d. **User Satisfaction Review**. A user satisfaction review (Exhibit 10-3) determines the effectiveness, correctness, and ease of use of the system from the users' perspective. If appropriate, this review can be used at any point during the information system life cycle. Summarize the results of the review.

3. Findings

Describe the major findings, results, or conclusions of the review. The intent is to provide management information for decision making about the system under review.

Rank or prioritize the findings by importance, if applicable. Otherwise, group them logically, as appropriate. The ranking, prioritizing, or grouping facilitates making a logical linkage to Section 4, which provides recommendations regarding the findings.

Provide as much detail as necessary to describe the findings clearly and support the recommendations. The following list provides some examples of information that might be included in this section:

- o What and where short-term problem areas exist (e.g., missing tapes, misrouted material);
- o What and where long-term problem areas exist (e.g., machine capacity problems);
- o References to meetings, interviews, and surveys conducted, with a description of their results or outcomes; and
- o References to supporting statistics or reports.

4. Recommendations

Present the recommendations derived from the findings of the system review. These recommendations should be phrased as proposals for management consideration and approval.

Depending on the purpose and scope of the specific system review as defined by bureau management, it may be appropriate to provide multiple, alternative recommendations for the findings. If alternative recommendations are provided, then describe the advantages, disadvantages, costs, trade-offs, etc., associated with each alternative.

Rank, prioritize, or group the recommendations logically, as appropriate. Relate the ranking, prioritization, or grouping of the recommendations to that of the findings in Section 3 above.

5. Approvals and Appendices

Reference any management approvals and include any appendices needed to support the periodic system review report.

a. **Approval**. Reference or describe the final approval of the periodic system review report, which may come from different levels of authority within the bureau, depending on the size and importance of the items being reviewed.

Thus, complete this section after the initial periodic system review report has been presented to management. After management approval of the report, update this section. Also update this section to provide an annotation of the recommendations or course of action selected by management, if appropriate.

b. **Appendices**. Reference any additional items necessary to support the system review from other documents, or add to the appendices, as appropriate.

G3. DOCUMENTATION - DISPOSITION PLAN

The disposition plan is the most significant deliverable in the disposition of the information system and the plan will vary according to system and bureau requirements. The objectives of the plan are to end the operation or the system in a planned, orderly manner and to ensure that system components and data are properly archived or incorporated into other systems. At the end of this task, the system will no longer exist as an independent entity. The completion of the system life cycle is carefully planned and documented to avoid disruption of the organizations using the system, or of other systems that will use the data and/or software of the present system.

The software, hardware, and data of the current system are disposed of in accordance with organization needs and pertinent laws and regulations. Software or data of the system may be transferred to other existing systems, migrated to an entirely new system, or archived for future use. Hardware is made available for future use, added to surplus, or discarded.

In conducting the disposition task, several items are to be considered:

o All known users should be informed of the decision to terminate operation of the system before the actual termination date.

o Although the current system may be terminated, in many cases the data will continue to be used through other systems. The specific processing logic used to transfer the data to another system is developed as part of the data conversion planning for that system.

o In some instances, software may be transferred to a replacement system. For example, a component of the current system may become a component of the replacement system without significant rewriting of programs.

- o Effective reactivation of the system in the future will depend heavily on having complete documentation. It is generally advisable to archive all documentation, including the life cycle products generated during the earliest tasks of the life cycle as well as the documentation for users and for operation and maintenance personnel.

The disposition plan addresses how the various components of the system are handled at the completion of operations, including software, data, hardware, communications, and documentation. The plan also notes any future access to the system. The plan is lead/performed by the project manager, supported by the records management staff, the project team and the functional staff, and reviewed by the quality assurance manager. Other tasks include the following.

- o Notify users of termination date. Notify all known users of the system date of the planned date after which the system will no longer be available. Work with the FOIA/PA representative process any Federal Register regarding system of records notification.

- o Store or transfer data. Copy data to be archived onto permanent storage media, and store media in location designated by the disposition plan. Work with the project management team for other systems to effect a smooth transfer of data from current system to these systems.

- o Store or transfer software components. Copy software onto permanent storage media, and store media in location designated in disposition plan. (Software to be stored may include communications and systems software as well as application software.) Work with the project team for other systems to ensure effective migration of current system software to be used by these systems.

- o Archive life cycle products. Store other life cycle products, including system documentation, in archive locations designated by the disposition plan.

- o Dispose of remaining equipment. Dispose of equipment used exclusively by this system in accordance with the disposition plan (refer to excess procedures).

- o Complete Disposition Plan. Update the disposition plan to reflect actual disposition of data, software and hardware.

USER SATISFACTION REVIEW

This review is designed to obtain user feedback on information systems. Feedback gathered in this review can help to determine whether information systems are accurate and reliable.

System Identification

1. Name of system ____________________

2. Data processing identification number, if any ____________________

3. Type of system ____________________

4. Part of system to be evaluated ____________________

____

User Identification

6. Name ____________________

7. Date ____________________

8. Title ____________________

9. Organization ____________________

10. Phone number/address ____________________

11. What is the extent of your knowledge about the system?

____

____

____

Exhibit 10-3: User Satisfaction Review

12. For what purpose do you use the system? **YES** **NO**

- Initiate transactions __ _
- Authorize changes to the system __ _
- Operate computer terminal __ _
- Maintain data controls __ _
- Design/program applications __ _
- Other (explain) ____________

____

13. In relation to the work of your office environment, estimate the importance of the system on a scale from 1 (not important) to 10 (very Important). ____

14. State the ease of understanding the system on a scale from 1 (difficult) to 10 (very easy to understand). ____

15. Can the system be used as is, without correction, further identification, or analysis? **YES** **NO**

16. In your judgement, is the system: **YES** **NO**

- Accurate and reliable? __ _
- Available when needed? __ _
- Current and up-to-date? __ _
- Useful? __ _

For each "no" answer, please explain below, and provide examples.

____

____

____

____

Exhibit 10-3: User Satisfaction Review (cont'd)

17. In your opinion, should the system: **YES** **NO**

- Provide more data? __ -
- Provide less data? __ -
- Be combined with other output products? __ -
- Be considered obsolete? __ -
- Be improved to make your job easier? __

For each "yes" answer, please explain below.

____

____

____

____

18. If you maintain manual records to supplement computer processed information, briefly explain why.

____

____

____

19. Does the system duplicate any other information you receive? _**YES** _**NO**

If "yes," briefly explain. ________________________________

____

____

Exhibit 10-3: User Satisfaction Review (cont'd)

20. Can you readily obtain, from other sources, the information in the system? **_YES _NO**

    If "yes," list the sources. ______________________________

    ____

    ____

21. Do you supply the input data for this system? **_YES _NO**

22. When you receive output, do you check it for quality? **_YES _NO**

    If "no," please identify the person or group performing this function. _______
    ____

23. Is the system ever rerun by the data processing department **_YES _NO**

    If "yes":

    - How frequently? ___________
    - Why were the reruns necessary? ___________
    - How do you make sure that the rerun material is correct? ___________

24. If you have had/were to have problems with this system, with whom did/would you discuss them? ___________________________________

    ____

25. Do you maintain correspondence with the data processing department or other user departments concerning the system? **_YES _NO**

    If yes, attach copies of recent correspondence.

26. Did anyone in your department help design the system? **_YES _NO**

Exhibit 10-3: User Satisfaction Review (cont'd)

27. Could you effectively perform your duties: **YES** **NO**

- Without this system? __ _
- If the system output were produced less often? __ _

28. Does the system save any clerical effort? __ _

Explain. ____

____

29. Can this system and its outputs be improved to make your job easier? _
Explain. ____

____

30. How often do you use this system? **YES** **NO**

- Daily? __ _
- Weekly? __ _
- Monthly? __ _
- Annually? __ _
- Never? __ _
- Other? (Explain) ____________________________________

For each "yes" answer, please explain below.

Exhibit 10-3: User Satisfaction Review (cont'd)

OPERATIONS, MAINTENANCE AND DISPOSITION ISSUES CHECKLIST

The project team uses this checklist to ensure that specific issues have been addressed. Consideration is given to the questions below for each information system project. Action may not be required for some items.

Operations and Maintenance

___ Is the system working as specified?

___ Are problems and recommendations for enhancements being identified?

___ Have discovered problems been acted upon?

____ Has a post-implementation review been scheduled?

___ Have periodic system reviews been scheduled?

___ For each review, has the proper system documentation been identified for use?

___ Have review results been documented?

--- Has a review of planned versus actual project schedule been completed? (This review will help the bureau learn about adequate project planning and estimating techniques, and reasons delays occurred and how these delays can be eliminated or reduced.)

Disposition

--- Has a disposition plan been drafted?

--- Has the system termination date been identified?

--- Have plans been made to identify which software components should be preserved?

--- Have plans been made to identify which data should be preserved?

Exhibit 10-4: Operations, Maintenance and Disposition Issues Checklist

OPERATIONS, MAINTENANCE AND DISPOSITION ISSUES CHECKLIST

Disposition (cont'd)

--- Has a determination been made regarding what to do with the remaining equipment?

--- What plans have been made to identify how life cycle products should be archived?

Post-Implementation Review Report

--- Has a determination been made who will perform the post-implementation review and the participants involved?

--- Who will ensure all required information system deliverables be available for review by the post-implementation review leader or team?

--- What, if any, signature authority be required after the post-implementation review is performed?

--- Who will receive copies of the post-implementation review report?

--- What role will the steering committee have in the post-implementation review, e.g., will the committee be expected to approve and accept the report?

--- Who will be responsible for addressing any findings and recommendations made in the report?

--- Is the post-implementation review scheduled in the IRM Review Plan?

Periodic System Review Report

___ Has the system been clearly identified in describing the purpose of the system under review?

___ Has the scope been clearly delineated in describing the scope of the review?

Exhibit 10-4: Operations, Maintenance and Disposition Issues Checklist

OPERATIONS, MAINTENANCE AND DISPOSITION ISSUES CHECKLIST

Periodic System Review Report (cont'd)

___ Has all important, relevant data been provided in the system overview? Has extraneous and unimportant information been excluded from the system overview?

___ Will the periodic system review involve a system response evaluation? If so, has the approach to evaluating the system response been adequately described?

___ Will the periodic system review involve a system capacity evaluation? If so, has the approach to evaluating the system capacity been adequately described?

___ Will the periodic system review involve a system correctness evaluation? If so, has the approach to evaluating the system correctness been adequately described?

___ Has the approach for reviewing user satisfaction been adequately described?

___ Have criteria been developed for determining an appropriate ranking, prioritization, or grouping of major findings?

___ Have unimportant or extraneous findings been removed from the description of the major findings?

___ Does the organization of the **Findings** provide clarity, conciseness, and understandability to the major findings?

___ Has the organization of **Recommendations** been developed to allow the recommendations to be linked clearly to the findings presented in **Findings**?

___ Have all appropriate appendices and management approval statements been included?

Exhibit 10-4: Operations, Maintenance and Disposition Issues Checklist (cont'd)

OPERATIONS, MAINTENANCE AND DISPOSITION ISSUES CHECKLIST

Disposition Plan

The disposition plan describes the rationale for terminating system operations, documents the plan for terminating operations and effectively archiving the various components of the system, and provides information about the location of the archived materials. The disposition plan is vital to ensure that information about the system can be accessed to support reactivation of the system, or future reuse of portions of the current system by other systems.

--- Will termination of the system require announcements of system of records notification in the Federal Register?

--- Does the disposition plan include an introduction to cover the purpose of the plan and reference to related documents?

--- Is there a description of the system which includes system objectives, users and structure?

--- Does the disposition plan cover the rationale for terminating system operation including the events leading to the termination, effective date of termination and any successor system?

--- Does the disposition plan cover future access to the software, data, and life cycle documentation?

--- Does the disposition plan cover data, software, documentation and hardware disposition and a schedule for each area?

--- What are the archive results for the data, software and hardware?

Exhibit 10-4: Operations, Maintenance and Disposition Issues Checklist (cont'd)

APPENDIX A. SMALL-SCALE DEVELOPMENT EFFORTS

A project manager may use these guidelines for the development of small-scale systems or for enhancements to existing systems that require less than 6 person months of work effort to develop. The life cycle methodology has been tailored by compressing seven phases into three, and requiring fewer products than the traditional life cycle methodology.

I. INTRODUCTION

A. GUIDELINES FOR SMALL-SCALE DEVELOPMENT EFFORTS

This guideline may be used for system development or enhancement efforts which require minimal enterprise-wide resources and work-power. Local microcomputer applications typically fit this category.

The project manager may use this guideline in situations where the development effort will take less than a total of 6 person months and/or costs less than $100,000.

The project manager documents the decision to use this guideline. In practice, procedures to determine project size are not very precise. The project manager may choose to consult with the quality assurance unit to determine if the guidelines should be used.

B. LIFE CYCLE PHASES AND PRODUCTS

Even though this guideline applies to systems with relatively short development timeframes, the small-scale project goes through development phases similar to those for development of a large-scale system. In addition, formal user involvement continues to be a mandatory part of the Requirements and User Acceptance Reviews.

There are some differences, however. The phases of the small-scale life cycle methodology (Exhibit A-1) overlap and are of shorter duration than in the full-scale methodology. In addition, life cycle products are combined in the small-scale project, and its life cycle reviews are fewer, may be more informal, and have fewer participants.

PHASES, DELIVERABLES, AND REVIEWS

Life Cycle Phases

Planning/Requirements Definition	Design/Development	Test/Implementation

Life Cycle Deliverables

Requirements Document	Design Document	Test Analysis Statement
- Project Plan	Test Strategy	Delivered System
- Feasibility Study	User Manual	
- Cost-Benefit Analysis		
- Functional Requirements Document		

Life Cycle Reviews

Requirements	Test Readiness	User Acceptance

Exhibit A-1: Methodology for Small-Scale Development Efforts

II. PLANNING/REQUIREMENTS DEFINITION PHASE

During this phase, these activities are performed:

- project plan is developed;
- user requirements are identified;
- alternative solutions are considered;
- a brief cost/benefit analysis is prepared;
- life cycle activities are defined; and
- project schedules are established.

A. PLANNING/REQUIREMENTS DEFINITION PHASE DELIVERABLES

1. Requirements Document

The requirements document is the only deliverable required for the planning/requirements definition phase. **It contains a section for the project plan, the feasibility study, and the cost benefit analysis; a section for project planning activities; and a section documenting user requirements.**

a. Project Plan Section

Automated project management tools may be used to set up the project activities and schedules.

The project plan section:

- documents project scope and objectives;
- provides justification for using the guidelines for smallscale development efforts;

- o discusses assumptions and constraints;
- o describes equipment and support software environment;
- o describes interfaces to external systems;
- o identifies resources needed to develop the project;
- o identifies level-of-effort by system developers, users, technical support staff, quality assurance staff, and security staff; and
- o identifies the life cycle products, reviews and testing activities with their planned start and end dates.

b. Feasibility Study and Cost Benefit Analysis Section

This section:

- o describes each automated solution being considered to solve the business problem;
- o provides a list or description of acceptable system characteristics (differentiating between essential and "nice-to-have" characteristics);
- o describes how each option* meets the requested characteristics;
- o shows the cost and benefits of the present process and each option under consideration;
- o expresses tangible and intangible benefits in terms of dollar value, if possible;
- o describes any extenuating circumstances which may affect the selection; and
- o identifies and justifies the selected option.

* At a minimum, there should be two options: one implementing the automated solution and another continuing the present process.

c. Functional Requirements Section

This section:

o describes the desired functions of the system;

o includes a data dictionary and logical data relationships;

o shows available relational table structures and entity relationship diagrams;

o identifies requirements for;

- performance
- system sizing and capacity
- system responsiveness
- reliability
- system growth
- user response time
- peak user times
- system availability

o describes security requirements if the following factors are present;

- Privacy Act considerations (FIPS PUB 41)

- Computer Security Act considerations, e.g., sensitive but unclassified data is being processed by the system

- Proprietary information to be handled within the system

- Financial system authorizing funds allocations or using fund flows which are based on electronic signatures or passwords

- Specific risks related to the system and its processing environment

2. Prototyping Techniques and Joint Application Development (JAD) Sessions

JAD sessions are alternate ways to develop requirements, but all requirements should be documented. It is an approach which emphasize teamwork between the users and the system developer. The team determines the system objectives, requirements and the business functions and processes to be supported. Within the planning/requirements Phase, there can be imbedded a mini-cycle called the prototyping life cycle (see Appendix B for more details). The mini-cycle steps are:

- o Specify Requirements - Analyze user requirements.

- o Build Model - Use an application generator to develop a prototype of the main portion of the user's required capability.

- o Demonstrate Model - Demonstrate the prototype to the user and identify improvements needed. Repeat this cycle a specified number of times until the user is satisfied and the functional requirements are clearly defined.

- o Evaluate Model - If the user is satisfied with the performance of the prototype system, it may be established as the user's system. All documentation must be developed including the Users Manual and Design Document. The final prototype cycle can be considered the system acceptance test if an adequate test plan is prepared prior to the final cycle and a test analysis statement is completed to document user acceptance is completed.

B. REQUIREMENTS REVIEW

A requirements review is conducted to reach agreement on the identification of user requirements, project activities and schedules, and the acceptability of the proposed automated solution. The program manager, as the user representative, and the project manager acknowledge agreement on work schedules, system benefits being commensurate with estimated costs, and the expected functionality of the system. Any organization

providing technical, quality assurance, or security support should review and concur with the Requirements Document. The project manager is responsible
for resolving any issues raised by these organizations. The requirements review may be the final JAD or prototyping session.

III. DESIGN/DEVELOPMENT PHASE

During this phase:

- proposed system structure and interfaces are developed;
- system is partitioned into design units using;

  - structure charts
  - hierarchial input/output charts
  - flow charts
  - pseudo-code

- design modules are coded;
- Test Plan is developed;
- unit and system tests are conducted;
- acceptance test readiness is determined; and
- User Manual is prepared.

As the system development effort progresses through the design/development phase, additional requirements may be identified or existing requirements changed. The project manager is responsible for incorporating approved changes into the User Manual.

A. DESIGN/DEVELOPMENT DELIVERABLES

1. <u>Design Document</u>

This document combines both the internal and external design characteristics of the system. This document should contain enough information to ensure that the system is maintainable.

The Design Document:

- o includes flow diagrams addressing the design of interfaces to external systems;
- o addresses the system and subsystem architecture
- o includes;

  - system architecture diagrams
  - high-level process flow diagrams
  - data flow diagrams
  - control flow diagrams
  - system input and output design
  - security requirements

- o defines data elements associated with each screen.

If the system is covered by the Privacy Act, the Privacy Act Warning should be incorporated into the screen flow.

2. <u>User Manual</u>

This manual:

- o documents all information necessary for the user to perform system functions;
- o describes the purpose of the software system;
- o describes types of hardware and software used;

- o indicates whether the system is covered by the Privacy Act;
- o identifies procedures to initiate, maintain, terminate, and restart operations; and
- o describes error handling.

If on-line help is available, an abbreviated User Manual may be used.

3. <u>Test Strategy</u>

The test strategy:

- o describes the test type (unit, system, system acceptance) and purpose;
- o provides information on needed test resources, environment, and schedules;
- o documents test scenarios to verify the system requirements;
- o includes descriptions of input test data to exercise program routines, error handling, security, and performance;
- o describes procedures to accomplish the test; and
- o describes the expected output.

B. TEST READINESS REVIEW

The system developer may conduct unit tests during development but should conduct at least one system test before the system is given to the user for a System Acceptance Test. After the unit and system tests are conducted, the project manager conducts a Test Readiness Review with the program manager to ensure that the system and associated User Manual including related software support documentation are ready for the System Acceptance Test. Quality Assurance staff may participate in this review to verify that the system documentation is in place and that there is evidence of appropriate unit and system testing.

IV. TEST/IMPLEMENTATION PHASE

During this phase:

- o user and developer test the system to ensure that it meets requirements;
- o user and developer test the User Manual;
- o developer updates appropriate life cycle documentation based on changes identified as a result of testing;
- o testers prepare a Test Analysis Statement;
- o program manager conducts a User Acceptance Review;
- o program manager directs system implementation;
- o developer distributes the user support documents distributed to system users; and
- o developer provides any needed training.

A. TEST/IMPLEMENTATION DELIVERABLES

1. Test Analysis Statement

This statement documents each type of software testing: unit/module, system, and system acceptance. It is a record of the results of the tests and describes the capabilities and deficiencies to be discussed at the Test Analysis Review.

2. Delivered System

After the system and its associated documentation has been fully tested, updated and approved, it is released to the production environment. Any mainframe or local area network release procedures should be followed unless the system will be implemented on a stand-alone computer.

B. USER ACCEPTANCE REVIEW

In this review, the test results recorded as part of the Test Analysis Statement are discussed. Participants include the project manager and users and may include representatives from the quality assurance unit and security unit. At this time, the users will indicate their acceptance determination on the Test Analysis Approval Determination. Quality assurance staff verifies that testing guidelines were adequately addressed, and security staff verifies that adequate security requirements were included.

APPENDIX B. PROTOTYPING

A. INTRODUCTION AND OVERVIEW

Since the beginning of systems development, experience has shown that frequently after a system has been designed, developed, tested, and implemented, the users find that it is not what they really wanted, or the system does not work the way they intended it to. As a means of preventing this, the practice of prototyping a system is becoming more and more widespread to define a system's functional requirements.

A prototype of a new system is simply a working "model" of the system. Such a model might range from reproducing or simulating only selected functions or processes described in the functional requirements, up to a fully functional model. Even fully functional, a prototype should not be a developed system containing actual code that could be validated and put into production. If it were, it would be defeating the purpose of prototyping, which is **rapid** development of an easily modified model.

Typically prototyping is used today for on-line systems development. This is because the user is intimately involved in the operation of an inter-active on-line system with its data entry and retrieval through screen displays.

One way to build prototypes is to use specialized tools that aid in developing them. It should be noted that, increasingly, these tools have the capability of automatically generating code once the design is finalized. Another popular approach is to use a fourth generation language that facilitates the building of prototypes; and then use third generation language, such as COBOL, for greater efficiency in the production system's code.

Prototyping can be useful in all of the initial phases of the life cycle. In the planning phase, prototyping might be used in finalizing the feasibility study and updating the cost benefit analysis. However, probably the most important benefits are found in using prototyping during the requirements definition and design phases.

At the Department, prototyping is almost an entirely new dimension in system design and development. At this writing, it is not possible to predict every case where prototyping can be useful or provide illustrations of every case. If designers are aware of prototyping, greater use of prototyping should follow as they see new and expanded ways to build prototypes.

B. OBJECTIVES AND BENEFITS

The objectives of this discussion of prototyping are to provide an awareness of the benefits of prototyping, to give a description of prototyping levels, and to state some of the functions to be sought in tools that facilitate prototyping.

Some of the benefits of prototyping are:

- Users and programmer analysts can see early in the life cycle what the new system will do and look like so that they verify requirements through direct experience with the planned system.
- A prototype can be developed in far less time than developing the full system.
- Modifications and changes in the design are far less costly and much easier to make here than in the development phase.
- The prototype is a dynamic model to allow for changes in requirements during their development.
- Alternative approaches can be physically seen and experimented with.

C. LEVELS OF PROTOTYPING

a. **Screens Only**. The most common form of prototyping is in the development of screens for on-line systems. A simple prototype of a screen for users and programmer analysts might allow them only to view it on a terminal and see if it contains the necessary fields, in their correct position, and in color - if available. Sophisticated prototypes would not only let users and programmer analysts view the screen, but would permit interactive modification of the screen as they look at it.

b. **Screens and Data Editing**. The most sophisticated screen prototype would do all of the above, and, in addition, would allow programmer analysts to include editing rules and screen messages so that users could actually input the kind of data they would to the real system, and the prototype would respond the way the real system would.

c. **Basic Working Model**. At the next level, the prototype would be capable of performing input/output functions. A model of the data base would be accessible and one could simulate adding, updating, and deleting records. If the system is to have query or reporting functions, these would be modeled at this level.

d. **Full Function Model**. The full function model would be able to duplicate all of the desired processes of the system, but would do so in "miniature" or as a facsimile only.

In general, users and programmer analysts should progress upward through these levels one at a time. There should be full agreement on screen layout before moving to the basic working model and this level should be fully completed before attempting a full functional model. As obvious as this may seem, the mistake is often made to leap directly into developing a full scale model before such fundamental items as input format have been completely defined.

D. PROTOTYPING AND THE SYSTEM LIFE CYCLE METHODOLOGY

Oftentimes, in the traditional life cycle methodology, after a need has been perceived by users and a request for a new system has passed through strategic planning and the planning phase, a project manager and user prepare functional requirements in the requirements definition phase. While the requirements are drawn up with input from the users in this phase, sometimes users are excluded from the actual writing of the requirements.

The next user involvement is approval of the requirements in the functional requirements document. The users are presented with a completed document and asked to say whether the system described is what they originally wanted. Most often they do not have the information systems background needed to understand the technical language of the written functional requirements, and they are not sure they understand the non-technical areas either. Faced with this, users typically sign-off on the requirements hoping the system is going to be what they want, but far from sure that it will be.

When this approach is followed, it comes as no surprise that users frequently find that the system is not what they expected; or that while it is what they originally wanted, their needs changed during the time the design was out of their sight.

Prototyping is one way to avoid this. It does not require a change in the life cycle phases - just a modification of what happens in the phases. Paramount to the concept of prototyping is that user involvement becomes the primary consideration. The goal is to develop system functional requirements that absolutely meet the users' needs and expectations.

Within the requirements definition phase there could be an imbedded a mini-cycle that could be called the prototyping development life cycle. This mini-cycle has phases of its own: (1) specify requirements; (2) build model; (3) demonstrate model; and (4) evaluate model. These are all described in more detail in Appendix A. This cycle is repeated as often as necessary until the functional requirements have been refined to ensure they are accurate and complete.

E. PITFALLS OF PROTOTYPING

There are no disadvantages of prototyping that would rule out its use - its value in defining precise and accurate functional requirements is enormous - yet there are two things that could be pitfalls if not handled correctly.

a. The user must understand that this is a prototype only. What the user sees, the user could mistake for the real system, and not realize that development of the actual system will take some additional time - perhaps considerable additional time. Care must be exercised to avoid the "I can see it, why can't I have it **now**?" problem.

b. In developing a prototype, it is necessary to build in realistic response time. If, in the prototype, the user sees almost instantaneous response; the user is sure to be upset later when the actual system has a response time of many seconds, especially when the system is being heavily used.

APPENDIX C. AUTOMATED TOOLS

Computer Aided Software Engineering (CASE) technology is an automated engineering discipline for project management, software development, and maintenance. It includes automated structured methodologies and automated tools.

Automated tools vary in cost, complexity of use and suitability to the task, and the market changes rapidly. It is suggested that an information systems group which wishes to purchase an automated tool first contact others who have used this tool. Another alternative is to purchase or reference a publication which reviews software and hardware. These publications may be in the form of monthly periodicals with a low subscription rate and a narrow band of subject items or in the form of costly and extensive binders containing a wealth of information on hardware, software and technology updates.

When selecting a package, evaluate its capability regarding:

Number of tasks allowed - will it accommodate the number of tasks in your project?

Number of resources allowed - will it accommodate the number of personnel and material items?

Subproject capability - can the software divide large projects into smaller ones?

PERT/CPM - does the software use PERT or CPM?

Import/export capability - can access to other software packages be made from within this package?

Network diagram scaling capability - can the diagrams be reduced to computer screen size?

Presentation graphics capability - are the graphics easy to use and flexible?

The following list is representative of some available tools and is not a recommendation of any one product over another. Some software tools are multi-functional and may be listed more than once.

1. **Project Management Tools**

Expert Choice, Expert Choice Inc.
Project for Windows, Microsoft
Project Manager Workbench, Applied Business Technology Corp.
Windows, Computer Associates
Timeline, Symantec Inc.
Harvard Total Project
Qwiknet
Pmis

2. **Software Development Products**

Aims Plus
Analyst/Designer Toolkit, Yourdon, Inc.
Application Factory
CorVision, CORTEX Corp.
Design Aid, Certified Software Specialists, Ltd.
Design Machine
Developer Workstation
Excelerator, Intersolv, Inc.
Information Engineering Workbench, Knowledgeware, Inc.
Life-cycle Productivity System, American Management Systems, Inc.
Life-cycle Manager
Predict Gateway, Software AG

3. **Strategic Planning and Business Modeling Tools**

IEW-Planning Workstation (Information Planner) - KnowledgeWare
4Front strategy (SSP), 4Front planner - Holland Systems
Information Strategy Planning - Texas Instruments
PC Prism - Index Technology
Tip Plan.It - T&IP
Janus - D. Appleton
Super-Mate - LBMS
Methodmanager - MSP

4. **CASE Workstations for Analysis and Design**

Excelerator - Index Technology
IEW - Analysis, Design Workstations - KnowledgeWare
Information Engineering Facility - Texas Instrument
Managerview - Manager Software Products (MSP)
Design/Aid - Nastec
Data Analyst, Database Administrator - Bachman Information Systems
ER-Designer - Chen & Associates
USER:Expert Systems - Information Engineering Systems
Data-Mate - LBMS
Depictor - Applied Data Research
Architect - Cullinet
Requirements Analysis Workbench - Computers & Engineering Consultants
Rule Tool - Cadware
Precise*PC-IAST - CDC
Design/1 - Arthur Andersen
Analyst Workbench - Visible Systems
Diagraphics (DFDP) - ADPAC
Analyst Toolkit - Yourdon
Pro-kit/Analyst - IST (MCAUTO)

5. **Logical Database Design**

Relational Generators (View Synthesis & Automated Normalizers):

Data Designer - KnowledgeWare
Facets - T&IP
Designmanager - MSP
Design4data (LDD) - Holland Systems
Normal - Cincom Systems

(2) Others:

Janus - D. Appleton
ADL/IRMA - Arthur D. Little
NIAM - CDC

6. Software Maintenance Products

Pathvu, Compuware Corp.
Recorder, Advanced Systems Concepts, Inc.

7. System Development Methodologies

Data Structured Systems Development (DSSD), Ken Orr and Associates
Information Engineering, James Martin Assoc.
Information Engineering Methodology, ISCE
Methodware, D. Appleton Co.
Method/1, Arthur Anderson
PRIDE-ASDM, M. Bryce & Associates
RAP (Requirements Analysis Planning) and PDM (Prototype Development Methodology, from DACOM
Systems Development Methodology/70 (SDM/70) and Systems Development Methodology/Structural (SDM/Structural), AGS Management
Spectrum/Productivity, Spectrum International
Structural Analyst, Design and Implementation of Information Systems (STRADIS), McDonnell Douglas Information Systems
System Development Standards, Cara Corp.

APPENDIX D. CAPACITY MANAGEMENT AND PERFORMANCE MEASUREMENT PROGRAM

These guidelines establish procedures to measure and manage the capacity and performance of information systems equipment and software. All bureaus must establish and maintain an information systems capacity management/performance measurement program and utilize this program in the information systems planning and acquisition process.

The objectives of a capacity management/performance measurement program include:

a. information systems which are responsive to the mission, goals and security requirements of the organizations they support;

b. improved productivity within the Department of the Treasury through the measured and controlled application of information systems resources to the processing of present and forecast workload;

c. optimal performance of equipment and software as a result of continuously monitoring and analyzing information system resource utilization;

d. maintenance of agreed upon levels of service in terms of response time, transaction volume, and cost between users and managers of information systems;

e. acquisition plans based on workload forecasts;

f. cost and productivity data on which to base cost-benefit analysis of current and projected processing;

g. timely corrective action in the event of declining reliability of systems or components of systems; and

h. closer working relationships and improved exchange of information between users and managers of information systems resources.

GUIDELINES FOR A CAPACITY MANAGEMENT AND PERFORMANCE MEASUREMENT PROGRAM

I. INTRODUCTION

An information systems organization has a finite amount of capacity to process a given workload. This capacity results from the configuration of equipment and resident software which can be adjusted or incremented to increase the amount of workload that can be processed. Performance is the rate at which workload is processed by an information system and includes measurements of volume and of time. A capacity management program must incorporate the measurement and evaluation of performance as well as the projection of future workload and performance requirements for planning purposes. Information systems are designed to provide functions required by an organization in the performance of its mission. These functions and related requirements, such as internal controls and security, should not be compromised by capacity management.

A. **PURPOSE**. These guidelines address both the technical and the managerial aspects of capacity management/performance measurement and are intended only to outline the major elements of a complete program. Detailed implementation plans must be developed applying these guidelines to each information systems environment. While capacity management/performance measurement requires the use of technical tools and the understanding of ADP concepts, it should not be treated as simply a technical program. In addition to guiding the system manager in the day-to-day management of resources, it will also provide important input to information systems plans and budgets. Measurement data must be appropriately presented to managers to enable them to make information systems decisions from a business perspective.

B. **PROGRAM REQUIREMENTS**. A capacity management/performance measurement program must include:

1. establishment of performance objectives;

2. measurement of actual or projected performance;

3. analysis and presentation of performance measurement results;

4. analysis of costs and value associated with performance levels; and

5. adjustment of capacity when appropriate to meet performance and/or cost-benefit objectives.

C. USE OF DATA. Stacks of system performance printouts which are never examined are pointless. In addition, a program is ineffective if utilization and performance trends are not routinely presented to management for decisions regarding the allocation of resources.

II. MANAGEMENT OF THE PROGRAM

A. PLANNING. A plan for capacity management/performance measurement should address objectives, the current environment, and changes required to meet the objectives. The objectives for any such program should include those outlined in the previous section, as well as others that address a bureaus particular goals and vulnerabilities. The plan should outline activities and responsibilities which will provide continuous management of information resources to meet the business goals of the bureau. The emphasis in the plan should be on activities to avoid crises, as well as on appropriate reaction to unanticipated events. Planning should address each of the following areas described below.

1. **Performance Measurement**. A program for capacity management must begin with an established baseline of performance which states both the amount of workload that is being processed and the time required to process it. Workload can usually be described in terms of volume of input and its source. Time factors vary with the type of workload. Batched transactions are processed at a certain rate per hour while interactive transactions are processed with a certain response time to the user as well as at a certain rate.

2. **Performance Evaluation**. When the performance baseline has been established, it can be evaluated in terms of reasonableness in relation to expected performance for the system and its components. Adequacy of performance to support a bureaus mission objectives should be assessed.

3. **Cost-Benefit Analysis**. Information system capacity must be managed in accordance with value to be obtained from costs incurred. The measurement of system performance when applied to operating costs becomes the basis for work units costs. The benefit/cost ratio should be considered in making any decision to increase capacity or improve performance.

4. **System Service Objectives**. After existing performance has been measured and evaluated, objectives should be set for delivery of service to the system users. These objectives should be stated in terms of workload and time and must be negotiated with system users and documented in service level agreements. Performance priorities for systems should be established with reference to business plan priorities and critical success factors. Whether or not users are charged for service, they should be made to understand the costs associated with alternative service levels and should be prepared to justify, based on value to be obtained, requirements for more costly service levels. For systems already in existence, negotiating service level agreements may consist simply of documenting actual performance. If that performance is not considered to be acceptable, an analysis of the cost and benefit of improving it is required before a decision is made and a new service level agreement is agreed upon.

5. **Monitoring Performance**. Performance of each equipment and software installation must be continually monitored to identify any failure to achieve agreed upon service levels as well as to identify system components which are becoming bottlenecks, declining in reliability, or otherwise affecting performance. Trends in workload volume and type should also be identified by the monitoring function. The information required and the systems to capture and report it must be identified and the appropriate tools acquired as described in Section III below.

6. **Projecting Capacity Requirements**. In addition to performance monitoring, consideration should be given to modeling and simulation techniques to estimate future capacity requirements when significant changes in the nature of the workload are anticipated.

7. **Reporting to Management**. Periodic reports showing, for example, actual performance versus performance objectives, problems, potential problems, and proposed solutions should be made available to management. Exception reporting is desirable. These reports should present capacity and performance in terms of workload and cost so that the manager can assess the impact of capacity management decisions on the accomplishment of the mission and goals of the bureau.

8. **Response to New Initiatives**. Whenever a new application is planned, a service level agreement for that application should be negotiated with the

user. The service level agreement is one requirement to be addressed in system design and acquisition planning.

9. **Acquisition Planning**. All acquisitions of equipment and software should be planned based on measurements and projections of required capacity and performance. Acquisition planning to enhance or replace equipment approaching saturation should allow sufficient time for following procurement regulations and installing equipment. A capacity management plan should establish capacity measures which, when attained, will cause acquisition plans and actions to be initiated.

    a. Cost. The cost of an acquisition must be justified in terms of the value to be realized from increased information system capacity and/or improved performance. While other factors, such as physical obsolescence or new functionality, may dictate new acquisitions, present and projected workload and performance requirements should be the basis on which a given processing capacity is specified. Planning must address raw volumes as well as the type of processing to be provided, the system design features desired, the software to be employed, etc., to ensure that adequate capacity is specified.

    b. Requirements. The requirements for equipment and for software should always include mechanisms for monitoring capacity and performance in the operational environment to allow evaluation of the product's ability to perform according to requirements, as well as its adequacy as the operating environment changes.

10. **Staffing**. An adequate capacity management/performance validation program requires a trained and experienced staff to employ the technical tools, analyze data, and inform management of the issues.

B. **SERVICE LEVEL AGREEMENTS**. Negotiation of service level agreements and decisions regarding configuration of information systems resources to deliver specific performance levels require the participation of both system and user management.

C. **CAPACITY MANAGEMENT IN CONTRACTOR-OWNED FACILITIES**. A capacity management/performance validation program is applicable to information systems support performed through contractor owned facilities and should be provided for in the contract.

D. **INFORMATION SHARING**. Managers of capacity management and performance validation programs are encouraged to keep in contact with one another, both within the Department and with other public and private organizations with such programs, for the purpose of sharing experience and techniques.

III. TECHNICAL APPROACHES TO CAPACITY MANAGEMENT AND PERFORMANCE MEASUREMENT

A. **INTRODUCTION**. A variety of types of tools and techniques is available for capacity management and performance measurement. Some of them will combine two or more functions and certain functions will be performed by components whose primary purpose is not capacity management/performance measurement. While not all tools and techniques are applicable in every situation, the functionality of each should be considered in developing a program.

B. **JOB ACCOUNTING SYSTEM**. This tool is used to collect and present information about the operation and costs of jobs that are processed by a configuration of equipment and software.

C. **CHARGEBACK SYSTEM**. This tool is used to distribute the costs of data processing to those who use it in support of the performance of the organizational mission.

D. **HARDWARE/SOFTWARE MONITOR**. These tools (consisting of equipment attached to the computer system and internally loaded software) provide data which can be used to optimize the configuration, improve throughput, determine resource utilization, determine character of workload, improve response time or turnaround, and develop a system model.

E. **PROGRAM ANALYZER**. This tool is used to identify and analyze those programs and procedures which are the major resource consumers so that optimization techniques can be applied.

F. **OPERATIONS CONTROL**. This tool is used to inform system operators of capacity usage during routine processing so that adjustments can be made immediately if warranted by declining performance.

G. **SIMULATORS/MODELERS**. These tools are used to design a system model which can be altered to reflect differing information systems scenarios in terms of workload to be processed, type of processing to be performed, and system configuration. For each model developed, they can produce projections from which information system resource needs can be derived.

H. **BENCHMARKING**. This is a technique for measuring the performance of a system to create a point of reference from which follow-on measurements can be analyzed. Most commonly, benchmarking is used to measure the capacity and performance of a newly acquired system to determine that it meets the stated requirements. A benchmark can also be revised, run, and the results compared to the original to measure the impact of changes to workload over the life of a system, or it can be rerun with a changed configuration to determine the effectiveness of the change.

IV. **CONCLUSION**. A capacity management/performance measurement program for information systems does not involve new resource management concepts. The program manager must emphasize the management aspects of the program while providing adequate technical facilities. As with any information system project, the technology must not be allowed to stand in the way of sound business practices.

APPENDIX E. GLOSSARY

-A-

Acceptance Testing - Testing performed by program staff during implementation to verify that the system solves the information systems problem, performs satisfactorily, and is ready for release to users.

Accreditation Statement - A policy statement verifying that a computer system and the processing area have met all requirements necessary to process sensitive data in a secure environment. The accreditation is performed during the implementation phase.

Acquisition Plan - A plan that sets out the overall strategy for managing an acquisition.

Adaptation - The addition to or subtraction from an existing life cycle. Through adaptation, a unique life cycle is developed for a particular project.

Assumptions - Judgments made by the project manager and the members of the project team to cover gaps in available information when the plan is being developed. Assumptions are necessary to produce the project plan in the absence of information concerning a specific issue. The plan can be met if the assumptions prove true. The plan will not be met if the assumptions prove false. All assumptions represent risks.

Attribute - A property or characteristic of an entity. For example, the entity "employee" may have the following attributes: employee name, social security number, date of birth, grade level, and salary.

Audit Trail - A collection of documentation used to trace changes to the project objectives and their impact on cost/schedule performance. The audit trail can consist of paper and/or machine readable files.

-B-

Baseline - A formal departure point for control of future changes in performance and design of a system or configuration item. There can be functional, allocated and product baselines based upon approved functional, allocated, and product configurations.

Baseline Schedule - The planned schedule of all start and finish dates of all sub-tasks, tasks, phases, and milestones in the Critical Path Network Diagram; this schedule is approved at the end of the planning phase.

Business Process Reengineering - The need to capture some of recent thinking about ensuring that organizations do not merely automate old processes. Prior to automation, organizations need to reassess work processes to assure alignment with organization mission, and identify opportunities for operational streamlining. Other terms used to described this process is work process redesign and business process restructuring.

-C-

Capacity Management - As applied to information systems, capacity management refers to activities of planning, monitoring, and adjusting a configuration of equipment and software to process data at levels needed by users.

Capacity Management and Performance Measurement Plan - Performance measurement measures the ability of a system to meet performance requirements for a specified workload in a specified time frame. Capability validation measures only a system's ability to meet certain mandatory and optional functional requirements. This plan will provide the details for conducting tests to validate performance and capability.

Certification Statement - A formal statement by the Senior Information Resources Management Official stating that a specific application or facility has met all requirements necessary to provide data integrity, availability and confidentiality. The certification is made prior to implementation.

Computer-Aided Software Engineering (CASE) - Software that is used in any and all phases of developing an information system, including analysis, design, and programming. For example, data dictionaries and diagramming tools aid in the analysis and design phases, while application generators speed up the programming phase.

Configuration - The functional and/or physical characteristics of hardware and software as specified in the technical documentation and achieved in the product.

Configuration Accounting - A process for maintaining system baselines, including adding products to a baseline, denoting the components of each product, and monitoring and recording the disposition of requested modifications to the system.

Configuration Management - A function which serves to systematically identify the items that make up a system, and formally control any modifications to those items, in order to help maintain the integrity of the system, and facilitate communication about the system throughout its life cycle.

Configuration Management Plan - A plan for identifying, documenting, and controlling changes to system components.

Contingency Plan - A plan for ensuring the continued operation of computer systems in the event of an unexpected interruption or loss of computer capabilities.

Cost-benefit Analysis - Sometimes referred to as the economic analysis, a study that projects the costs and benefits of an information system. Costs include all resources required for development as well as operating the system. Benefits are tangible and intangible.

Critical Path - The longest path, in estimated total elapsed time, through a project network, from the start of the first task, through the completion of the last task. The critical path consists of a series of tasks which must be completed on their scheduled completion dates, if the project is to be completed on schedule. There is a critical path in both Critical Path Method and Program Evaluation and Review Technique networks.

Critical Path Method (CPM) - The project schedule technique used to identify early start, early finish, late start, and late finish dates for individual jobs and the critical path to determine the duration of the entire project.

-D-

Data - Representation of facts, concepts, or instructions in symbols suitable for communication, interpretation or processing by human or automated means.

Data Administration - The management function responsible for the planning, definition, organization, protection, and efficiency of data on databases within an organization. It is the analysis, classification, and maintenance of an organization's data and data relationships. It includes the development of data models and data dictionaries, which, combined with transaction volume, are the raw materials for database design.

Data Administrator - The person who coordinates activities within the data administration organization. In some small organizations, the data administrator and the database administrator are one in the same; however, when the two responsibilities are managed separately, the database administrator's function is more technical.

Data Collection - The recording and capturing of data on behalf of an organization.

Data Definer - The person or organization who determines the essential qualities or meaning of data, and who prescribes and defines procedures which aggregate and refine data.

Data Dictionary - A centralized component of a database management system containing information about data, including its meaning, relationship to other data, origin, usage, and format.

Data Element - The smallest unit of data that has meaning in describing information. A piece of data which would not be meaningful if decomposed.

Data Entity - A person, place, thing, concept, or event that is of interest to the organization. A data entity is something about which the organization store data. An entity is typically described as a noun and graphically represented in entity relationship diagrams.

Data Flow Diagram - A graphic and textual representation of processing requirements for a function or system. Used to describe inputs to a process and their transformation by that process into outputs.

Data Integrity - The quality of data that exists as long as accidental or malicious destruction, modification, or loss of data are prevented.

Data Management - A function of data administration which is responsible for data-related activities of the system life cycle, such as logical data modeling during requirements definition, database design, database management, and the documentation of data-related decisions and products.

Data Management Plan - A plan to document the data management strategy which includes a description of data integration/sharing supported by automated software tools for system and database development, and an organizationwide inventory or directory of the data collected.

Data Modeling - Identification of the design principles for a data model.

Data Security - The protection of data against unauthorized disclosure, transfer, modification, or destruction, whether accidental or intentional.

Database - A collection of interrelated data stored together with controlled redundancy to serve one or more systems or applications.

Database Administration - The technical design and management of the database and often falls within the jurisdiction of data administration.

Database Administrator - The person responsible for the physical design and management of the database and for the evaluation, selection and implementation of the database management system.

Database Management System (DBMS) - Software that controls the organization, storage, retrieval, security and integrity of data in a database. It accepts requests from the application and instructs the operating system to transfer the appropriate data.

Design Data Dictionary - Data dictionary created during design to support design and development of the information system. It represents an expansion of the requirements data dictionary, and contains all the metadata stored in dictionary. In addition, it contains descriptions of the physical data base structures and the manner in which they are implemented in the test versions of the data base(s).

Design Phase - The third phase of the life cycle. In this phase the comprehensive analysis of user requirements is used to define the external characteristics of the system.
The detailed structure of the system from the subsystems is also created.

Development Phase - The fourth phase of the life cycle. This phase includes those activities directly related to producing the project's end-product.

Disposition Plan - A plan to document the termination of a system and the archiving and disposing of the system hardware, software, data and life cycle deliverables.

-E-

Economic Analysis - Sometimes referred to as the cost-benefit analysis, a systematic approach to quantifying and evaluating the relative value of proposed investments . Management uses cost and benefit information to evaluate alternatives.

Entity Relationship Diagram - A graphic representation of the data and data relationships required to support a function or system. The diagram is developed to show relationships between entities (objects about which data is stored) used in the processes encompassed by a function or system.

-F-

Functional Specification - A part of the project objectives that describes the end-of-work item from the perspective of the customer. Blueprint for the design of an information system. It provides documentation for the database, human and machine procedures, and all the output, processing and output detail for each data entry, query, update and report program in the system.

-G-

Gantt Chart - A graphic representation of a project schedule that shows each job as a bar whose length is proportional to its duration. The bars appear in rows with the placement of the bars indicating the job start and end times. Color and graphics indicate critical jobs, float, delay, job completion, and the baseline schedule.

-H-

High-Level Milestone - An important event in life cycle, such as the start of a new phase, a decision point, etc. A high-level milestone is concrete and easily verifiable.

High-Level Milestone Schedule - The assignment of target start and finish dates for high-level milestones in the schedule plan.

-I-

Implementation Phase - The sixth phase of the life cycle. In this phase the system is fully implemented. Implementation may take the form of the installation of new hardware and system software; the installation of a new database and application programs or the adoption of new manual procedures.

Information Engineering - The application of an interlocking set of formal techniques for the planning, analysis, design, and construction of information systems on an organization-wide basis or across a major component of an organization.

Information Life Cycle - The stages through which information passes, typically characterized as creation or collection, processing, dissemination, use storage, and disposition.

Information Management - The planning, budgeting, manipulating, and controlling of information throughout its life cycle.

Information Resources - Includes both government information and information technology.

Information System - A discrete set of information resources organized for the collection, processing, maintenance, transmission, and dissemination of information in accordance with defined procedures, whether automated or manual. (Information systems planning in the Department of Treasury addresses automated systems only.)

Information System Life Cycle - The phases through which an information system passes from beginning to end.

Information Technology - The hardware and software operated by a Federal agency or by a contractor of a Federal agency or other organization that processes information on behalf of the Federal government to accomplish a Federal function, regardless of technology involved, whether computers, telecommunications, micrographics, or others.

Initiative - A proposal for the development/acquisition of a new information system or for an enhancement to an existing system which will result in a major system or for the acquisition of new information technology for which Departmental approval is required.

-J-

Joint Application Development (JAD) - A working meeting between functional users and application developers. The objective of JAD varies with the project phase. JAD may be thought of as a controlled and structured, dynamic, interactive group session. An impartial mediator or facilitator leads the group during its activities.

-L-

Level of Detail - A set of entries in a work breakdown structure which divides the project into logical component parts. The lowest level contains the tasks. Intermediate levels are logical units of work larger than tasks, but smaller than the project.

Line Item of Work - The lowest level of work appearing in the project plan. The line item of work could be a task or sub-task.

Logical Data Model - A depiction of the logical, or programmatic, data needed to support an organization. The components of a logical data model include data entities and relations, data elements and attributes, keys, secondary keys, and relationships between entities.

-M-

Maintenance, Disposition and Operations Phase - See Operations, Maintenance and Disposition phase.

Major Information System - Any information system, in development or operation, which requires special management attention because of its importance to the bureau's mission; its high development, operating or maintenance costs; or its significant role in the administration of bureau programs, finances, property, or other resources. (Treasury's major information systems includes systems for which the annual cost exceeds one percent of the bureau's budget, or for which the systems life cycle cost exceeds $10 million or a highly visible project.)

Matrices - Graphic way to represent relationships between processes, data, organizations, and technologies. Useful in verifying completeness of requirements, areas of responsibility or involvement, etc.

Metadata - Information about an organization's information and data activities. Data about data, such as its definition or its physical characteristics.

Mid-Project Review - A review by the project manager and the user of the project conducted at the approximate development mid-point (during the external and internal design phases). Its purpose is to determine the status and condition of the project, the degree of user support, and whether changes are necessary.

Milestone - A significant point in a project. A milestone has no duration, and represents the start of a portion of work, or the completion of a portion of work. One milestone may represent both the start and completion of significant portions of work. At the most detailed level, each task commences and ends with milestones.

-N-

Network Diagram - The diagram of all known project phases, tasks, sub-tasks, and product milestones, showing predecessor relationships, durations, and all paths from the project's start to termination.

-O-

Operations, Maintenance and Disposition Phase - The seventh and final phase of the life cycle. During operations, the system is put into operation and turned over to the user for operation and maintenance. The newly-implemented software is used on an ongoing basis and maintained in operation. The system is terminated at the end of the life cycle.

-P-

Performance - One of the technical objectives of a project. This objective concerns how the product will perform once implemented. The performance criteria of the system are spelled out as part of this element of the technical objectives in the requirements definition phase.

Performance Measurement - As applied to information systems performance measurement refers to the validation of the responsiveness of the configuration of equipment and software to the needs of users and the availability and reliability of individual components and the system as a whole.

PERT (Program Evaluation and Review Technique) - A scheduling technique in which three estimates are developed for each task: optimistic estimate, most likely estimate, and pessimistic estimate.

Phase - A group of related tasks on a project, grouped at an intermediate level of detail in the work breakdown structure. The division of a project into phases is governed by the ISLC model.

Planning Phase - The first phase of the life cycle; during the planning phase, objectives of the project are defined, general requirements are documented, and alternatives are identified and analyzed.

Post-Implementation Review - A comprehensive assessment of a newly-installed system after a period of full operation conducted by the system development team and users. It evaluates the overall productivity gains to the organization. It compares the newly operational system's performance to criteria established in the functional requirements document and to the benefits and costs in the cost-benefit analysis document. It includes an assessment of the quality and effectiveness of human resources support.

Process Documentation - Administrative, planning, and technical documents produced during the project.

Product Documentation - Documentation associated with project products, such as training manuals, computer programs, etc.

Project - An undertaking with prescribed objectives, magnitude, duration and a specific outcome.

Project Charter - The written understanding between the project manager and approving officials as to the project's purpose, scope, objectives, major activities, organization, lines of authority and accountability, responsibilities of various participants, and general method of operation.

Project Files - The central office files in which all information about the project is stored for used by the project manager and the members of the project team. The project files are archived upon completion of the project for future use in the planning and execution of projects of a similar nature.

Project History File - File containing important project documentation that records the methodology and accomplishments of the project, as well as an audit trail of significant incidents during the life of the project.

Project Manager - A managerial position as the leader of an effort to achieve a set of project objectives. The project manager is the person who has accepted responsibility and been provided commensurate authority for meeting the project objectives.

Project Objectives - The project objectives are the quality measures, technical objectives (specifications, standards, and assumptions), time frame, and budget upon which an original project plan or a current approved plan has been developed.

Project Plan - The set of project documents which unify schedule and cost through a common work breakdown structure. The project plan includes the WBS, the schedule, the cost, quality measures, and all plans defined in the life cycle. The project plan may include other documents, at the request of senior management, such as those which address technical, legal, and regulatory issues.

Project Team - The staff of the project consisting of the project manager, the task leaders, and the personnel assigned to perform the tasks.

Project Tracking - The process of monitoring to ensure that the project meets the cost, time, and quality objectives that are identified in the planning phases.

Prototyping - A strategy for determining user requirements by extracting, developing, and presenting user needs in a working model (prototype).

-Q-

Quality - The extent to which a product or service meet customer requirements and is fit for use.

Quality Assurance (QA) - A function that ensures that all products of the life cycle are substantively accurate and address the information systems problem.

Quality Assurance Plan - A plan that includes all inspection and acceptance procedures, quality requirements, and contract administration requirements that ensure quality.

-R-

Reliability - One of the technical objectives of a project. This objective concerns how the product will perform. Reliability measures the extent to which the system will continue to operate rather than failing or shutting down.

Repository - Database of information about applications software that includes author, data elements, inputs, processes, outputs and interrelationships.

Requirements Analysis Package (RAP) - A document that defines the problem with the existing system or operations and proposes the most efficient solution to the problem. It is presented to management as the justification to fund the project. Refer to TD P 83-01, "Guidelines for Acquiring Federal Information Processing Resources."

Requirements Definition Phase - The second phase of the life cycle. In this phase a comprehensive analysis of user requirements is conducted.

Risk Assessment - Sometimes referred to as a risk analysis, an evaluation of a system for security by identifying events that could threaten the system and vulnerabilities not adequately protected against by existing safeguards.

-S-

Scheduling - The creation of start and finish dates for every activity in the project based upon a given project start date and activity duration.

Security Plan - Plan developed during the requirements definition phase that ensures development and implementation of security measures for the system under development.

Sensitive but Unclassified Information - Any information in the loss, misuse, or unauthorized access to or modification of which could adversely affect the national interest or the conduct of

Federal programs, or the privacy to which individuals are entitled under Section 552a of Title 5, United States Code (Privacy Act), but which has not been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept secret in the interest of national defense or foreign policy.

Specifications - The key, unique components of the technical objectives of a project. The specifications consist of such items as: quantity, quality, performance, maintainability, reliability, survivability, operability, the ability to be manufactured, flexibility, ease of transportation, ease of installation, consistency with the corporate image, modularity, etc.

Standards - The Government and institutional guidelines with which the product of the project must comply. Included are the Federal Information Processing Standards (FIPS), International Telegraph and Telephone Consultative Committee (CCITT) and Federal Communications Commission (FCC) standards, and even local building codes in areas where turnkey projects are being executed.

Statement of Work (SOW) - A technical document that clearly and accurately describes the product or service that the contractor will develop, perform, or deliver.

Status Report - A periodic (monthly) comparison of planned versus actual performance.
Subtask - A sub-division of a task, also called a step.

-T-

Task - A well defined piece of work which appears at the lowest level of detail of the project's work breakdown structure. The task begins at the start of the project, or upon completion of its predecessors, and has an estimated duration as well as a resource allocation. The task culminates with a milestone, an output which signifies the completion of the effort.

Technical Specifications - The underlying technical requirements which make it possible to meet the customer's functional specifications.

Test Plan - A plan containing specifications, descriptions, and procedures for testing the project's end-product, i.e., the system. This plan is normally developed during the requirements definition phase and revised in later phases.

Test Phase - The fifth phase of the life cycle. In this phase the programs completed during the development phase are tested under a system acceptability test (SAT) to determine if the system accomplishes its objectives.

Traceability - To maintain a recorded path between life cycle phases.

-U-

User - Individual or office for whom the system is being developed. Also an organizational or programmatic entity that receives information processing services from an information processing services organization.

-V-

Variance - The difference, in time, dollars, person hours, and quality between the master plan baseline and the actual performance of the project team, as of a specific reporting date.

-W-

Walkthrough - A highly-structured meeting to review the completeness and quality of selected module(s) of the system, or of the entire system.

Work Breakdown Structure (WBS) - A key ingredient in the integrated project plan. It is a logical, hierarchical list of tasks for the project, consisting of various levels of detail which may include phases and steps as intermediate levels. Work breakdown structures are usually drawn with the top block representing the project objectives and the lowest level representing the tasks.

APPENDIX F. ACRONYMS

ADP	Automated Data Processing
CASE	Computer-Aided Software Engineering
CBA	Cost-Benefit Analysis
COBOL	Common Business Oriented Language
CCITT	International Telegraph and Telephone Consultative Committee
CM	Configuration Management
CPM	Critical Path Method
CWBS	Contract Work Breakdown Structure
DA	Data Administration (Administrator)
DASIS	Deputy Assistant Secretary for Information Systems
DBA	Database Administration (Administrator)
DBMS	Database Management System
DM	Data Management
FCC	Federal Communications Commission
FEDSIM	Federal Systems Integration and Management Center
FIP	Federal Information Processing
FIPS	Federal Information Processing Standards
FIRMR	Federal Information Resources Management Regulations
FMFIA	Federal Managers Financial Integrity Act

4GLs	Fourth Generation Languages
FOIA	Freedom of Information Act
FRD	Functional Requirements Document
GAO	General Accounting Office
GOSIP	Government Open Systems Interconnection Profile
GSA	General Services Administration
I-CASE	Integrated Computer Aided Software Engineering
IE	Information Engineering
IRD	Information Resources Dictionary
IRDS	Information Resources Dictionary System
IRM	Information Resources Management
ISLC	Information System Life Cycle
ISP	Information Systems Plan
IT	Information Technology
JAD	Joint Application Development
LAN	Local Area Network
MCP	Management Control Plan
NIST	National Institute of Standards and Technology
OIRM	Office of Information Resources Management
OMB	Office of Management and Budget

OTM	Office of Telecommunications Management
PA	Privacy Act
PERT	Program Evaluation Review Technique
PWBS	Project Work Breakdown Structure
QIP	Quality Improvement Process
RAD	Rapid Application Development
RAP	Requirements Analysis Package
SDM	Space-Division Multiplexing
SIRMO	Senior Information Resources Management Official
SOW	Statement of Work
SWBS	Summary Work Breakdown Structure
TD	Treasury Directive
TD P	Treasury Department Publication
TQM	Total Quality Management
WAN	Wide Area Network
WBS	Work Breakdown Structure

APPENDIX G. REFERENCES

1. Public Laws (PL) and Regulations

PL 89-306	Brooks Act, as amended, Oct 1965.
PL 93-579	Privacy Act of 1974, as amended.
PL 96-511	Paperwork Reduction Act of 1980, Dec 1980.
PL 99-500	Paperwork Reduction Reauthorization Act of 1986.
PL 100-235	Computer Security Act of 1987, Jan 1988.
PL 100-503	Computer Matching and Privacy Act of 1988.
PL 101-576	Chief Financial Officer Act, Nov 1990.
PL 103-62	Government Performance Measurement and Results Act of 1990, Jul 1993.

2. Office of Management and Budget (OMB) Circulars and Bulletins

OMB CIR A-11	Preparation and Submission of Budget Estimates.
OMB CIR A-94	Benefit-Cost Analysis of Federal Programs; Guidelines and Discounts.
OMB CIR A-123	Internal Control Systems.
OMB CIR A-127	Financial Management Systems.
OMB CIR A-130	Management of Federal Information Resources.
OMB CIR A-132	Federal Productivity and Quality Improvement in Services Delivery.
OMB BUL 90-08	Guidelines for Preparation of Security Plans With Sensitive Information, Jul 1990.

OMB Report Model Framework for Management Control Over Automated Information Systems, Jan 1988.

OMB Report Budget Examining Techniques for Evaluation Information Technology Investments, Office of Information and Regulatory Affairs, Office of Management and Budget, Mar 1992.

3. **General Accounting Office**

GAO/OCG-93-24TR Internal Revenue Service Issues (Transition Series), Dec 1992.

GAO/OCG-93-5TR Information Management and Technology Issues (Transition Series), Dec 1992.

GAO/IMTEC 8.1.6 Information Technology: A Model to Help Managers Decrease Acquisition Risks, Aug 1990.

GAO/IMTEC 8.1.4 Information Technology: An Audit Guide for Assessing Acquisition Risks, Dec 1992.

GAO/IMTEC-92-51 Strategic Information Planning Framework for Designing and Developing Systems Architecture, Jun 1992.

GAO/GGD-89-1 Managing IRS, Actions Needed to Assure Quality Service in the Future, Oct 1988.

GAO Standards for Audits of Governmental Organizations, Programs, Activities and Functions, 1981 (rev).

4. **Federal Information Processing Standards (FIPS)**

FIPS 11-3 Guidelines-American National Dictionary for Information Processing Systems, Feb 1991.

FIPS 31 Automated Data Processing Physical Security and Risk Management, Jun 1974.

FIPS 38 Documentation of Computer Programs and Automated Data Processing Systems, Feb 1976.

FIPS 39 Glossary for Computer Systems Security, Feb 1976.

FIPS 41 Computer Security Guidelines for Implementing the Privacy Act of 1974, May 1975.

FIPS 64 Program and Automated Data Systems for Initiation Phase, Aug 1979.

FIPS 65 Automated Data Processing Risk Analysis, Aug 1979.

FIPS 73 Security of Computer Applications, Jun 1980.

FIPS 75 Validation, Verification, and Testing of Computer software, Sept 1980.

FIPS 76 Planning and Using a Data Dictionary System, Aug 1980.

FIPS 77 Planning and Management of Database Administration, Sept 1980.

FIPS 87 Guidelines - ADP Contingency Planning, Mar 1981.

FIPS 88 Integrity Assurance and Control in Database Administration, Aug 1981.

FIPS 99 Framework for Evaluation and Comparison of Software Tools, Mar 1983.

FIPS 101 Life Cycle Validation, Verification and Testing of Computer Software, Jun 1983.

FIPS 102 Computer Security Certification and Accreditation, Sept 1983.

FIPS 105 Guideline for Software Documentation Management, Jun 1984.

FIPS 106 Guidelines for Software Maintenance, Jun 1984.

FIPS 123 Data Descriptive File for Information Interchange, Sept 1986.

FIPS 124 Functional Specifications for Database Management Systems, Sept 1986.

FIPS 132 Software Verification and Validation Plans, Nov 1987.

5. National Institute of Standards and Technology (NIST)

NIST 500-56 Validation, Verification, and Testing for the Individual Programmer, Feb 1980.

NIST 500-75 Validation, Verification, and Testing of Computer Software, Feb 1981.

NIST 500-98 Planning for Software Validation, Verification, & Testing, 1982.

NIST 500-105 Software Conversion Management, 1983.

NIST 500-116 Improved FIPS Cost-Benefit Methodology - Phase II, 1984.

NIST 500-121 Planning and Implementing Computer System Reliability, 1985.

NIST 500-136 Computer Software Acceptance Testing, 1986.

NIST 500-139 Data Base Directions: Information Resource Management, 1986.

NIST 500-147 Requirements Analysis for Office Automation Systems, 1987.

NIST 500-148 Application Software Prototyping and Fourth Generation Languages, 1987.

NIST 500-149 Guide on Data Entity Naming Conventions, Oct 1987.

NIST 500-153 Guide to Auditing for Controls and Security: A System Development Life Cycle Approach, 1988.

NIST 500-161 Software Configuration Management: An Overview, Mar 1989.

NIST 500-165 Software Verification and Validation: Its Role in Computer Assurance and its Relationship with Software Project Management Standards, Sept 1989.

NIST 500-169 Executive Guide to the Protection of Information Resources, 1989.

NIST 500-170 Management Guide to the Protection of Information Resources, 1989.

NIST 500-171 Computer User's Guide--Protection of Information Resources, 1989.

NIST 500-172 Computer Security Training Guidelines, 1989.

NIST 500-180 Guide to Software Acceptance, Apr 1990.

NIST An Introduction to Computer Security: The NIST Handbook (Draft) May 31, 1994.

6. **General Services Administration (GSA)**

Federal Acquisition Regulations (FAR)

Federal Information Resources Management Regulations (FIRMR)

Federal Systems Integration and Management Center (FEDSIM), Information Resources Management Strategic Planning Guide, December 1993.

GSA Acquisition Series Guides

GSA Handbook - Life Cycle Management of Telecommunications Systems (IRMS), 1987.

GSA Phased Approach to Life Cycle Management (FEDSIM), 1988.

GSA Survey of Life Cycle Management Methodologies (FEDSIM), 1988.

GSA Information Resources Security, 1989.

GSA Go For 12 Program, 1987.

Lessons Learned to Date From Business Process Reengineering in the Federal Government, Information Resources Management Service, KAP-94-2-I, February 1994.

7. **Treasury Directives and Publications**

TD 25-04 Implementation of the Privacy Act of 1974, as amended

TD P 25-04 Privacy Act Handbook

TD 71-02 Safeguarding Officially Limited Information

TD 71-04	Administration of Security Violations
TD P 71-10	Security Manual
TD P 76-01	Treasury Acquisition/Procurement Regulation
TD 81-01	Management of Departmental Information Systems
TD 81-02	Treasury Information Resources Management Review Program
TD 81-06	Information Systems Planning
TD P 83-01	Acquisition of Federal Information Processing Resources
TD P 85-03	Risk Assessment Guideline

8. Federal Agency Life Cycle Publications

Department of Agriculture. A Project Manager's Guide to Application Systems Life Cycle Management and Application Systems Life Cycle Management Manual.

Department of Commerce. Life Cycle Management Guideline.

Department of Defense, Defense Systems Management College. Systems Engineering Management Guide.

Department of Education. Software Life Cycle Management and Documentation Manual.

Environmental Protection Agency. System Life Cycle Management Guidance.

General Services Administration. Systems Management.

Department of Health and Human Services (DHHS). DHHS Information Systems Life Cycle Management Guidance.

Department of Health and Human Services, Social Security Administration. Software Engineering Technology Manual.

Department of Housing and Urban Development (HUD). HUD's System Development

Methodology and Managing HUD's System Development Process.

Department of the Interior. A Project Manager's Guide to Application Systems Life Cycle Management and Application Systems Life Cycle Management Handbook.

Department of Labor. Systems Engineering Concepts and Procedures Manual and Systems Engineering Standards Manual.

National Aeronautics and Space Administration. Automated Information Management Program Guidance: Lifecycle Management and Information System Life-Cycle and Documentation Standards.

Resolution Trust Corporation, System Development Life Cycle Manual.

Department of Transportation. FIP Hardware Life Cycle Management Policy and FIP Software Life Cycle Management Policy.

Department of the Treasury, Internal Revenue Service, Information Engineering Life Cycle (IELC), Version 1.0, Document 9038, Catalog Number 14930H.

9. **Supplemental References**

Department of Health and Human Services. Information Resources Management Long Range Plan FY 1994-1998.

Department of Veterans Affairs, Planning and Acquisitions Handbook, IRM Planning, Acquisitions and Security Service.

Davenport, T.H. and J.E. Short. "The New Industrial Engineering: Information Technology and Business Process Redesign." Sloan Management Review. Summer 1990. Pages 11-27.

Freedman, Alan. Electronic Computer Glossary, Version 6.12. Landmark Research International Corporation. The Computer Language Co., Inc., 1981-1993. Point Pleasant, PA.

Government Operations Committee of the President's Council on Management Improvement, Managing the Risk and Uncertainty of Technological Change (Draft), "A Review of Federal Practices for Acquiring Major Information Systems," October 1990.

Hammer, Michael and James Champy. Reengineering the Corporation: A Manifesto for Business Revolution. Harvard Business School Press, 1993.

Internal Revenue Service, Department of the Treasury, Office of Management and Budget, Study of Tax Systems Modernization Benefits, February 1992.

Kerzner, Harold, Project Management A Systems Approach to Planning, Scheduling and Controlling, 3rd Ed., Van Norstrand Reinhold, New York, 1989.

Office of Planning and Management Analysis (OPMA), Financial Services Directorate (FSD), Deputy Assistant Secretary (Departmental Finance and Management), Assistant Secretary (Management), Department of the Treasury, Federal Planning Anthology, OPMA Working Paper Series, December 1992.

Office of Planning and Management Analysis (OPMA), Financial Services Directorate (FSD), Deputy Assistant Secretary (Departmental Finance and Management), Assistant Secretary (Management), Department of the Treasury, Criteria for Developing Performance Measures Systems in the Public Sector, OPMA Working Paper Series, May 1992.