United States General Accounting Office

GAO

Staff Study



June 1992

# STRATEGIC INFORMATION PLANNING

## Framework for Designing and Developing System Architectures





GAO/IMTEC-92-51

# Preface





Managing and modernizing the federal government's information resources is an enormously complex task. The purposes of government information systems vary widely from business applications, such as payroll or personnel, to worldwide military command, control, and communications systems. Identifying the best solution to a problem in terms of operational effectiveness, flexibility, maintenance, and cost requires detailed, structured analyses of user needs and identification of technical alternatives and supporting technologies for the system's software, hardware, communications, data management, and security infrastructures.

Over the past several years, we have evaluated numerous attempts to modernize government information systems; among the major problems we found were large cost increases, long development delays, and systems that do not meet users' needs. In many cases, these development shortcomings occurred because of inadequate detailed planning to identify current and future user needs and a premature commitment to a specific design that did not consider alternative solutions or how well each would satisfy users' needs. These problems are not new. Both the Commission on Government Procurement in 1972 and the Blue Ribbon Defense Panel in 1986 produced similar findings.

The complexities of information systems can be vastly different, but the analyses to determine an organization's information needs to carry out its mission are essentially the same regardless of the complexity of the problem. Various public and private organizations have devised methodologies for identifying information needs and planning acquisitions to fill those needs. While the methodologies were slightly different, each espoused a top-down, structured approach to identifying information needs and analyzing how to meet those needs. Our assessment and evaluation of the various methodologies resulted in this generic framework for analyzing, designing, and developing open and flexible information system architectures that can be used to meet any information processing need.

This framework is intended to (1) provide a basis for systematically determining information needs, (2) identify and analyze information and data needs and relationships, (3) identify and analyze alternative ways to satisfy information needs, and (4) provide factors to be considered in arriving at the best way to satisfy information needs. The framework is not meant to be a "cookbook," but a guide for systematically identifying and evaluating an organization's information needs and laying out a disciplined process to satisfy those needs.

This framework is intended for use by General Accounting Office staff when evaluating the process used by federal agencies to develop and acquire information systems. It may also be useful for internal and external auditors, other oversight organizations, and program managers to serve as a guide when planning to acquire or upgrade information systems. It should be used in conjunction with the Information Management and Technology Division's August 1990 publication, Information Technology: A Model to Help Managers Decrease Acquisition Risks (GAO/IMTEC 8.1.6). The model was developed to help managers identify the critical factors needed to manage and control large-scale systems development and acquisition projects that are significant enough to require formal project management.

The framework was prepared under the direction of Samuel W. Bowlin, Director, Defense Security and Information Systems, who can be reached at (202) 512-6221. Other major contributors to this framework are listed in appendix I.

Ralph V. Carlone
Assistant Comptroller General
Information Management and
Technology Division

# Contents



**Abbreviations**

| | |
|---|---|
| CASE | Computer-Aided Software Engineering |
| GAO | General Accounting Office |
| GOSIP | Government Open Systems Interconnection Profile |
| IMTEC | Information Management and Technology Division |
| POSIX | Portable Operating System Interface for Computer Environments |

# What Is Strategic Information Systems Planning?

Strategic information systems planning is a disciplined, systematic approach to determining the most effective and efficient means of satisfying organizational information needs. It is a top-down, structured approach which, to be successful, must employ technical and managerial processes in a systems engineering context. Under this approach, the characteristics of the system's hardware, software, facilities, data, and personnel are identified and defined through detailed design and analysis to achieve the most cost-effective system for satisfying the organization's needs. The process must consider system life cycle management and the organization's policy and budget as important integral factors, and include all organizational participants (e.g., managers, users, maintainers, operators, and designers) throughout the process. It is an iterative process in that changes identified during the process must be evaluated to determine their effect on completed analyses. Strategic information systems planning is not a one-time event—it should be revisited periodically to ensure a system's continued viability in meeting information needs and achieving long-term missions.

## The Importance of Strategic Information Systems Planning

Information systems are important tools for effectively meeting organizational objectives. Readily available, complete, and accurate information is essential for making informed and timely decisions. Being unable to obtain needed data, wading through unneeded data, or inefficiently processing needed data wastes resources. The organization must identify its information needs on the basis of a systematic identification and analysis of its mission and functions to be performed, who is to perform them, the information and supporting data needed to perform the functions, and the processes needed to most usefully structure the information. Successful information system development and acquisition must include a rigorous and disciplined process of data gathering, evaluation, and analysis prior to committing significant financial and human resources to any information system development. While implementing such an approach may not preclude all information system acquisition problems, it should produce detailed knowledge of organizational missions and operations, user information needs and alternatives to address those needs, and an open and flexible architecture that is expandable or that can be upgraded to meet future needs.

## The Impact of Not Systematically Identifying Information Needs

The importance of thoroughly and systematically identifying and analyzing information needs cannot be overemphasized. We have found over the past several years that acquiring information systems costs too much, takes too long, and results in systems that do not do everything the organizations wanted them to do. There are many such examples of failed information system acquisitions; on one system acquisition we evaluated, estimated cost had increased 50 percent and development was expected to take more than twice as long as estimated, yet the system was far from satisfying all the users' original requirements. We have noted other equally shortsighted planning examples, including instances where hardware was chosen early in development and had to be upgraded during development; hardware was chosen that was incompatible with other existing systems; and communications systems could not handle the originally required work load or message formats.

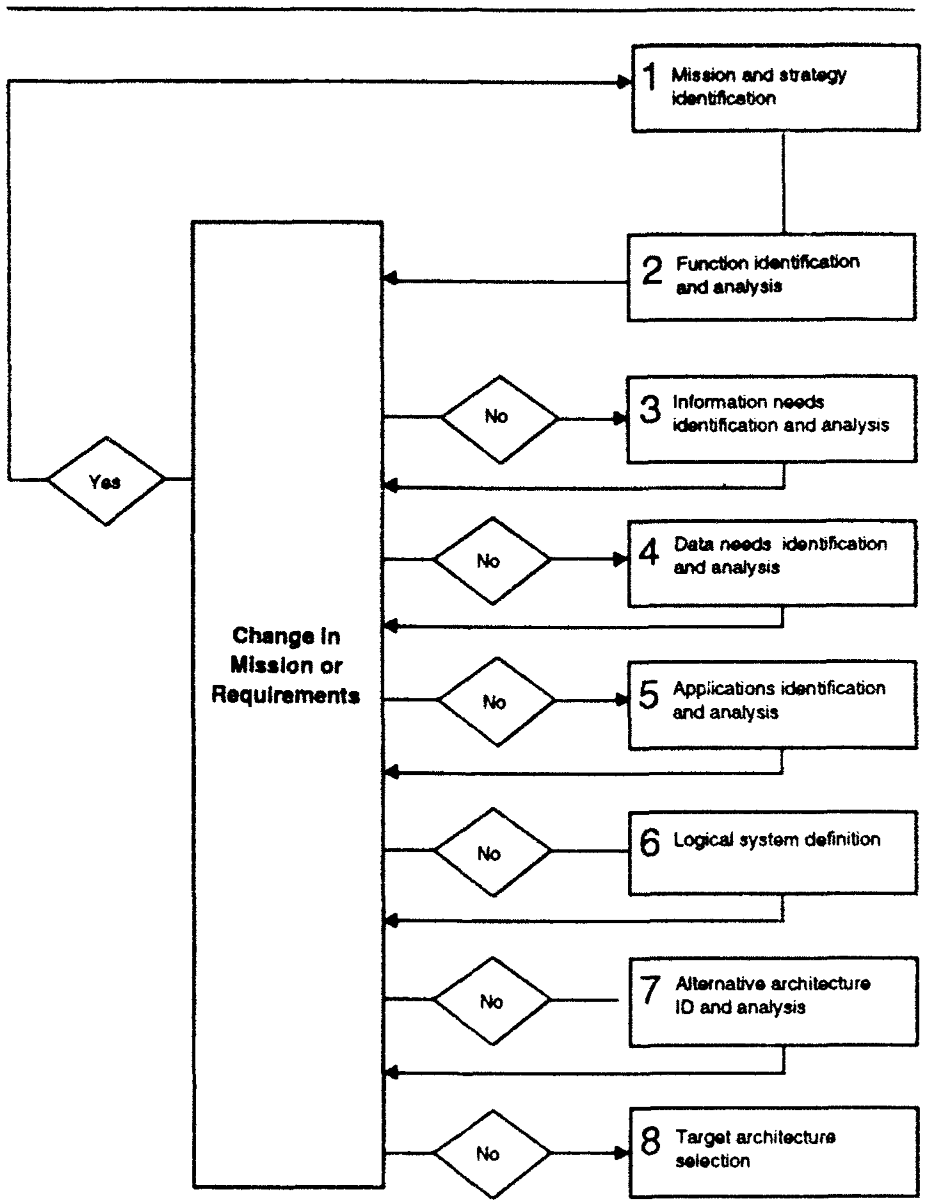
For example, the Department of Defense's acquisition process does not adequately emphasize systematically identifying and analyzing information needs and how to best meet these needs prior to committing resources to develop systems. The limited resources made available to explore alternatives and the practice that military services must defend their system choice before large-scale resources are committed have resulted in focusing prematurely on a single technical approach. Resources are spent proving that the initial system concept choice is right in order to get development go-ahead approval, rather than examining broad alternatives to satisfying the need. This premature commitment to a system concept, technical approach, or design often leads to cost growth, schedule delays, and performance shortfalls. The lesson that is almost always learned through these painful experiences is that had the organization thoroughly identified the information it needed to perform its mission and the most effective and efficient way of providing the information, time and money would have been saved and the acquired system would have better satisfied the organization's needs.

Even when the organization knew its information needs and identified a system to provide this information, it frequently did not consider whether the system could be affordably purchased, maintained, expanded, or upgraded. These factors are as important to deciding how to meet organizational information needs as identifying the information needed. For example, one system acquisition has been halted mid-way through development because the organization could not afford to operate it. We have also seen an organization acquire a new information system, only to see it fail to meet their long-term needs because it could not be effectively and efficiently maintained, expanded, or upgraded.

The framework presented here will introduce structure to the process of deciding how to meet information needs. This framework offers instructions for documenting an organization's mission, determining how this mission is to be carried out and the information and data required to do so, analyzing this evidence and identifying a logical system architecture and configuration, and determining the hardware and software needed to effectively and efficiently meet these needs. We cannot overemphasize the iterative nature of the process; any changes in mission, operations, functions, or information and data needs must be assessed to reveal their impact on analyses already completed, since these changes could have a profound effect on the system to be acquired.

In addition to the framework described, three other factors play a critical role in successful system development. They are systems engineering—the management function that controls the total system development effort; policy and budget—the procedures and funds for system developers to plan, acquire, manage, and operate automated information systems; and system life cycle management—a process that views systems as having stages, such as initiation, definition, design, programming and training, evaluation and acceptance, and installation and operation, each of which has its own planning, development, and management aspects. While not the subject of this paper, each is important to successful system design, development, and implementation.

**Figure 1: Strategic Information Systems Planning Framework**

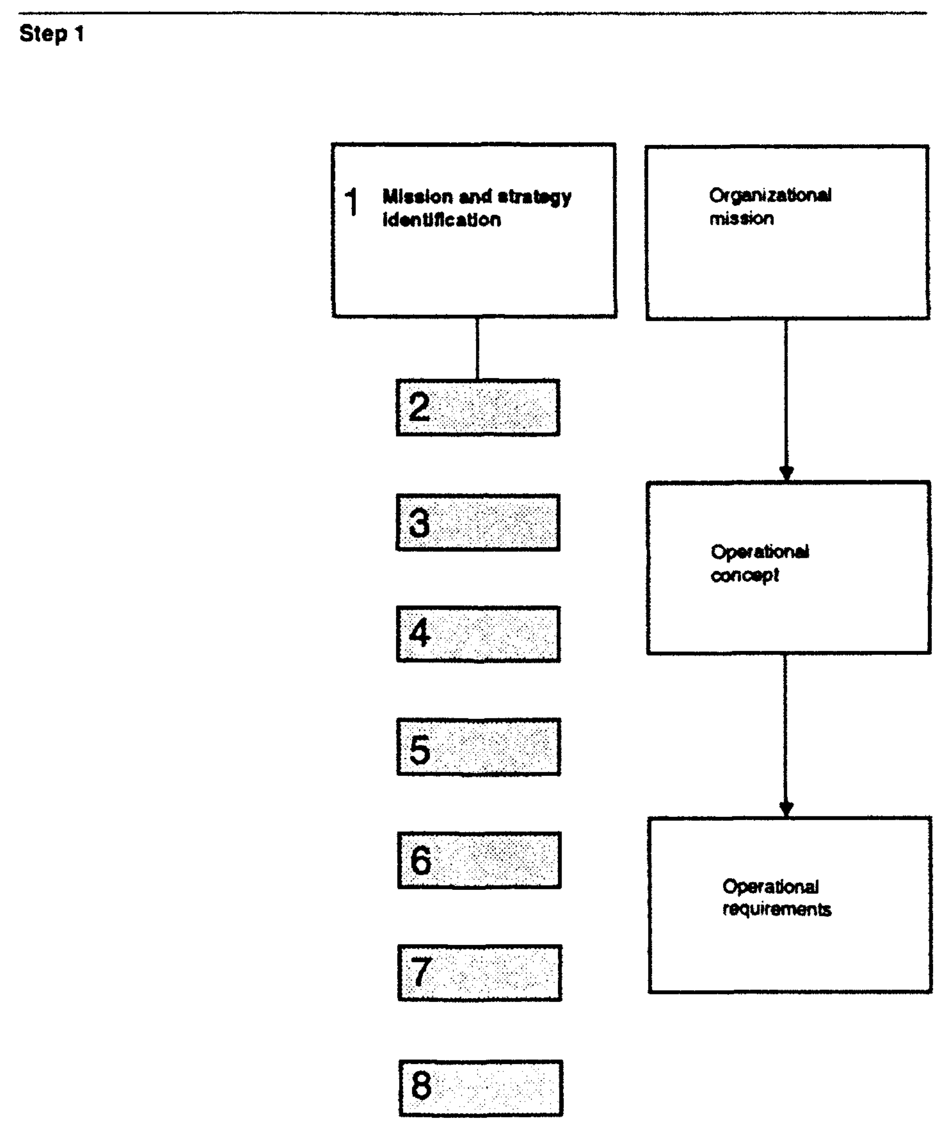


# Elements of the Strategic Information System Planning Framework

The strategic information system planning framework is generically designed so that it can be applied to any kind of information system acquisition. The rigor with which an organization applies the framework and the amount of detail generated when applying the framework depend in large part on the complexity of the system to be acquired. For example, identifying and analyzing a need for a small payroll system would be less rigorous than identifying and analyzing the need for a large-scale military command, control, and communications system incorporating both ground- and space-based platforms. Also, high technical risks and the application of relatively immature technologies require more detailed analyses and risk management, even though the process to be followed and the factors to be considered are the same as for less risky acquisitions.

The framework invokes a structured, orderly process of obtaining and analyzing key information before initiating system acquisitions. The framework is made up of eight steps: (1) identifying the mission; (2) identifying the functions to be performed in carrying out that mission; (3) identifying information needed to perform those functions; (4) identifying data needed to perform those functions; (5) identifying specific applications needed to provide that information; (6) specifying a logical system definition; (7) exploring alternative architectures; and (8) finally selecting a target architecture. Since the process is iterative, any major change in mission or requirements during any of the steps may require restarting the process from the top to ensure that the acquired system will meet user needs and be developed within cost and on schedule. Further, each step in the process builds upon the previous step. A detailed discussion of each of these eight steps follows.

**Figure 2: Step 1: Mission and Strategy Identification**



## Mission and Strategy Identification

Automated information system acquisitions begin by identifying a new or confirming an existing mission, determining how the organization will operate to accomplish its mission, and setting down the general operational requirements needed to achieve the mission. This first step outlines what the organization wants to accomplish (its mission), how it wants to accomplish the mission (a concept of operations), and the operational requirements needed to achieve the mission as outlined in the organization's concept of operations (for example, a system operational requirements document).

### Organizational Mission

It is critically important to the success of an information system acquisition that the organization's senior management identify and understand the ultimate use of a prospective system. Senior management's view of the organization's mission and how that mission is achieved greatly influences informational needs; therefore, this view must be obtained early in this stage. Management's identification and understanding of the mission should include the present, as well as a vision of the future and how present and future missions can be synthesized with respect to computer resources. This vision is important because information systems acquired today will likely continue to be used well into the future and must be open, flexible, and capable of being expanded or upgraded. If potential future missions are not articulated at the outset of the organization's information system planning, the resulting system most likely will not effectively and efficiently meet future information needs.

### Operational Concept

The concept of how each mission is (or could be) carried out is a high-level description of the operations that must be performed and who must perform them. This description includes where and how the operations might be carried out and the relationships between the missions and operations. It is at this level that organizational and informational relationships begin to emerge that help identify the proper scope of the operations to be performed. For example, knowing the locations where data is input to a military supply system would be important to identifying the types and amounts of data to be input and the needs of those locations for system output. The more knowledge that is generated about potential system users and their operational needs, the more likely that the resulting system will actually meet the users' needs.

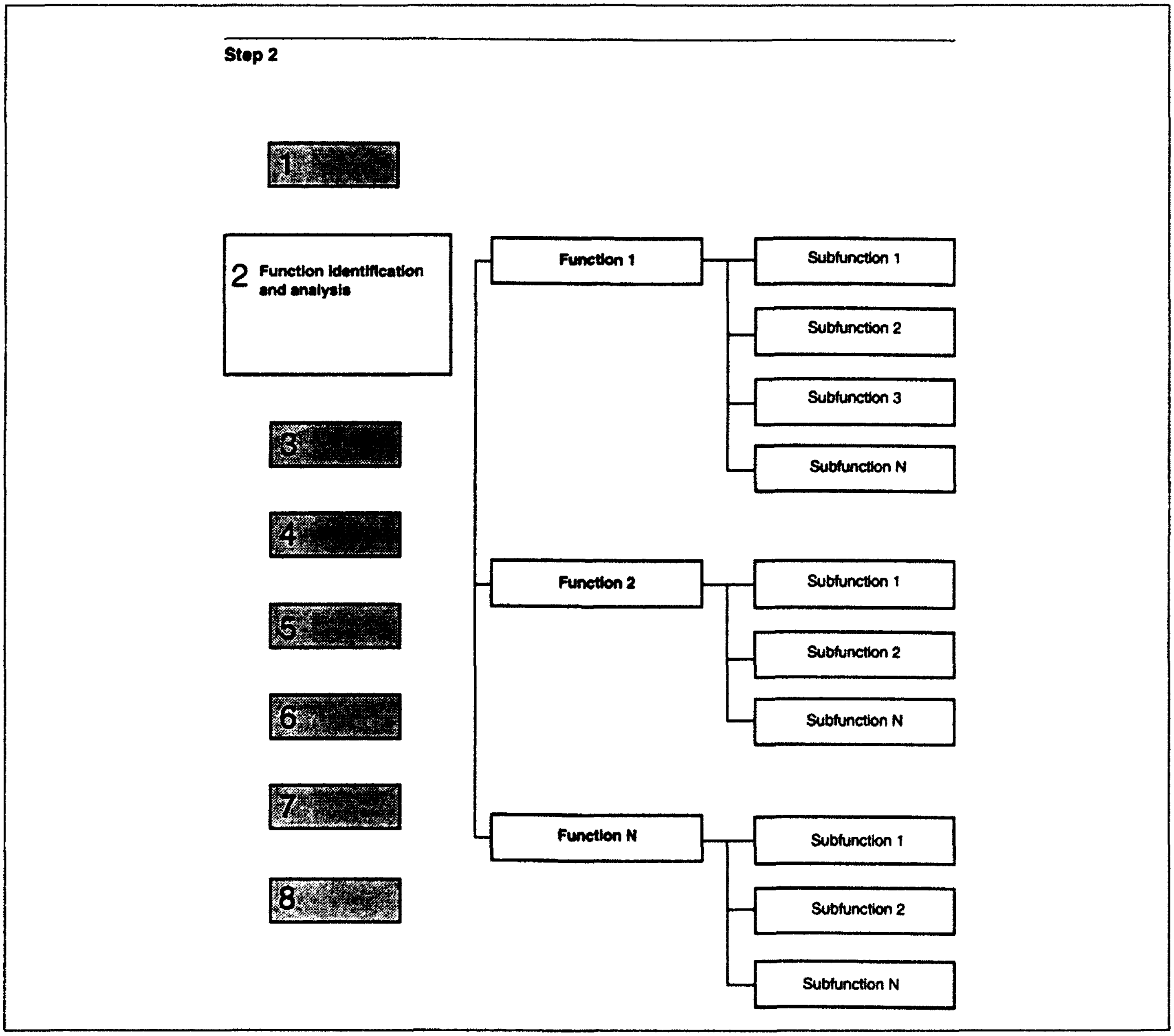
## Operational Requirements

Once the need has been identified, documented, validated, and justified, a difficult and critical task begins—identifying operational requirements. Requirements are the blueprint that system developers will use to design and develop the system. Ill-defined or incomplete requirements and requirements growth have been identified by many system developers and program managers as one of the root causes for system failure. Without adequately defined, organizationally approved requirements, a system will need extensive and costly reengineering before it can become fully operational.

Operational requirements identify the level of performance needed to accomplish the program's mission and provide the information that will drive later decisions on the logical system definition, alternative system architectures, and hardware and software designs. Examples of requirements include the amount, type, source, frequency, and speed at which data and information must be gathered, edited, correlated, fused, updated, displayed, printed, and transmitted in support of specific organizational needs. In addition, security, reliability, availability, and maintainability must be realistically defined because these system requirements drive subsequent system design choices, such as hardware and software, and have a significant impact on system development cost, schedule, and performance.

Requirements cannot be developed in a vacuum. A cadre of personnel intimately knowledgeable in all areas essential to carrying out an agency's mission must be represented during the requirements-setting process. There is no one methodology that ensures success in the requirements-setting process; however, various methods should be investigated. These include structured interviews, surveys, sampling techniques, brainstorming, and prototyping. Complex needs may require extended, intense studies, the use of more rigorous methodologies, and possibly the application of commercially available automated tools.

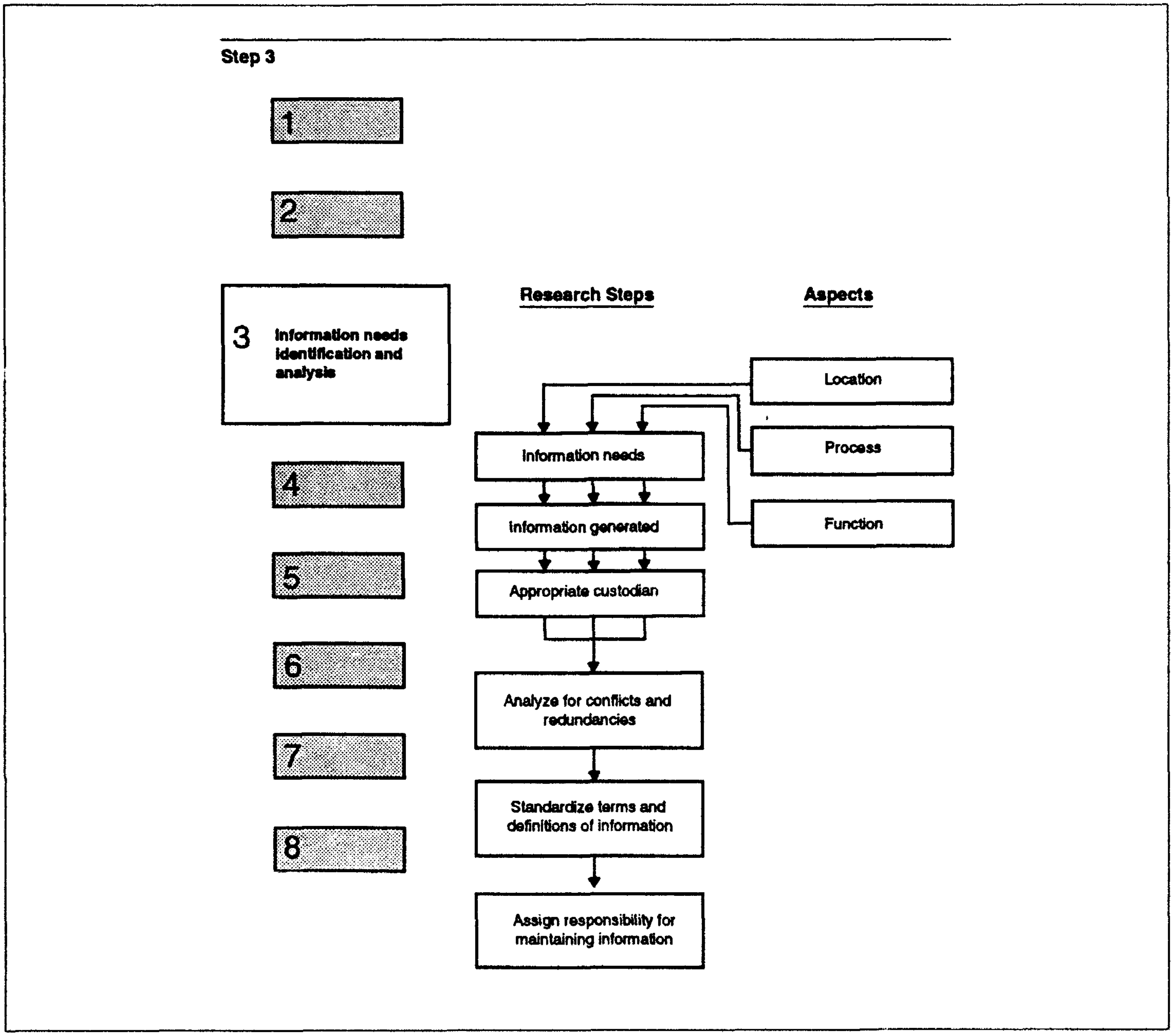
Figure 3: Step 2: Function Identification and Analysis



## Function Identification and Analysis

After the organization's mission has been defined and the operations required to carry out the mission have been identified, all the specific functions (tasks) needed to meet established requirements must be determined. Functions are the tasks or actions that must be performed to meet the operational requirements. A complete analysis of the functions and their interrelationships is conducted to identify functional inputs and outputs, overlaps and redundancies, and the most basic component processes of each function. The organization's functions are depicted with lower-level flow diagrams until they can no longer be broken down into smaller units, or subfunctions. In our supply system example, a function could be inventory and its subfunctions could be (1) counting, (2) verification, and (3) reorder. In principle, each functional procedure should be self-contained and easily described in a single, unambiguous purpose and process specification. If more than one purpose or process statement is needed to describe the function, probably more than one functional component is being described. This effort can be time-consuming; however, this level of detail is necessary to ensure that all functions are identified as early as possible to avoid difficult and costly changes later. This definition of all functions and their relationships is called the functional architecture.

Figure 4: Step 3: Information Needs Identification and Analysis



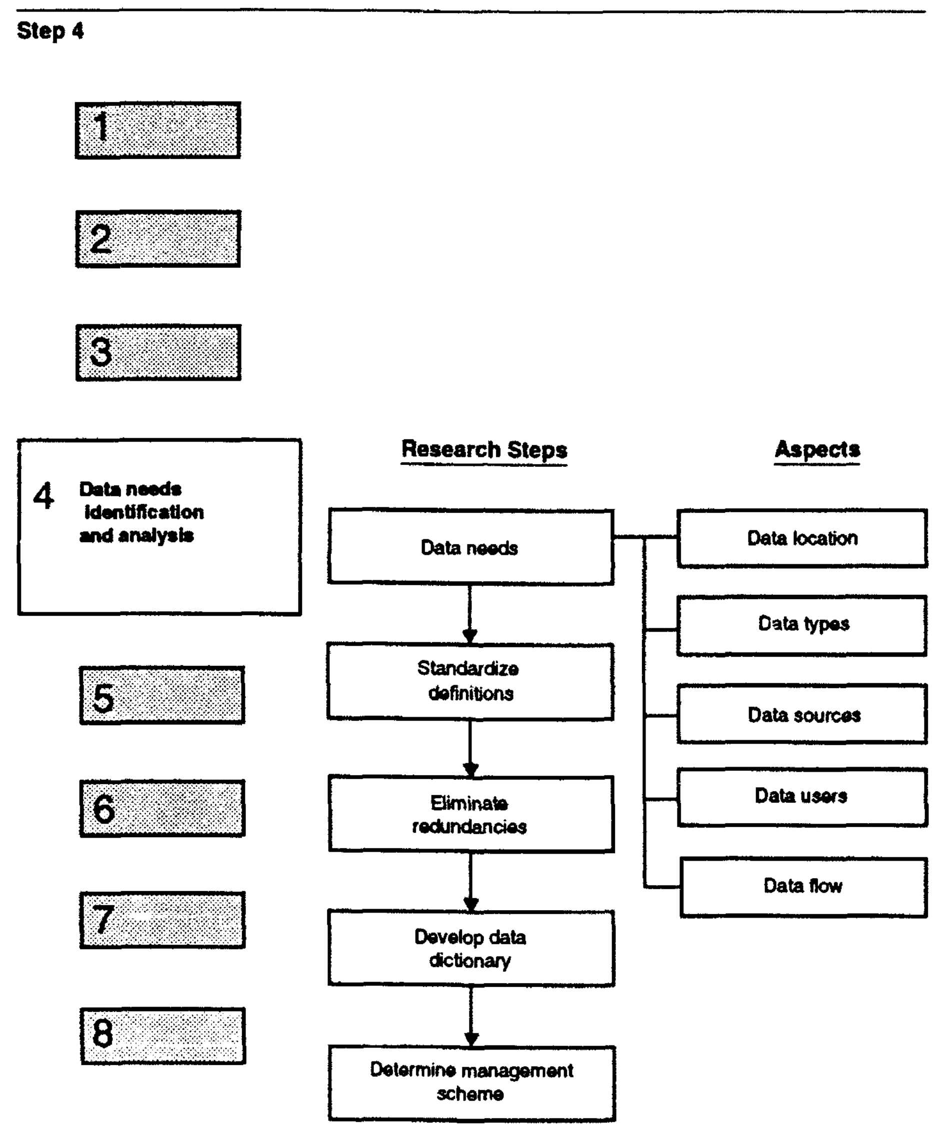
## Information Needs Identification and Analysis

After the various functions have been analyzed, all information needed to effectively perform and manage the functions to accomplish the organization's mission should be identified. It is important to identify the types and uses made of information throughout the organization, including who in the organization creates each specific item of information and who needs and uses that information. Many functions or locations use specific information to carry out their responsibilities. However, it is important that only one functional element or organization be identified as responsible for maintaining the information generated to ensure that information integrity is maintained.

Further, the specific information needed for each function (what is to be done), process (how it is to be done), and location (where it is to be done) should be described or identified. Commonalities and differences in the way information is described should be noted. Different descriptions for seemingly similar information should be researched to determine if a common term or definition can be used. For example, if all supply system locations generate information about the number of each part used in their operations but some locations have included broken and returned parts as used and other locations have not, a standard definition should be agreed upon by all locations. This description of the information needed by functions to perform processes at the various locations is called the information architecture.

Further, government contract services are available to help in the requirements-setting process. For example, the General Services Administration has developed a number of support services contracts to assist agencies. Other government agencies also can provide guidance and consulting services and may be used as independent validation agents; these include the National Institute of Standards and Technology and the General Services Administration's Office of Technology Assistance.

Figure 5: Step 4: Data Needs Identification and Analysis

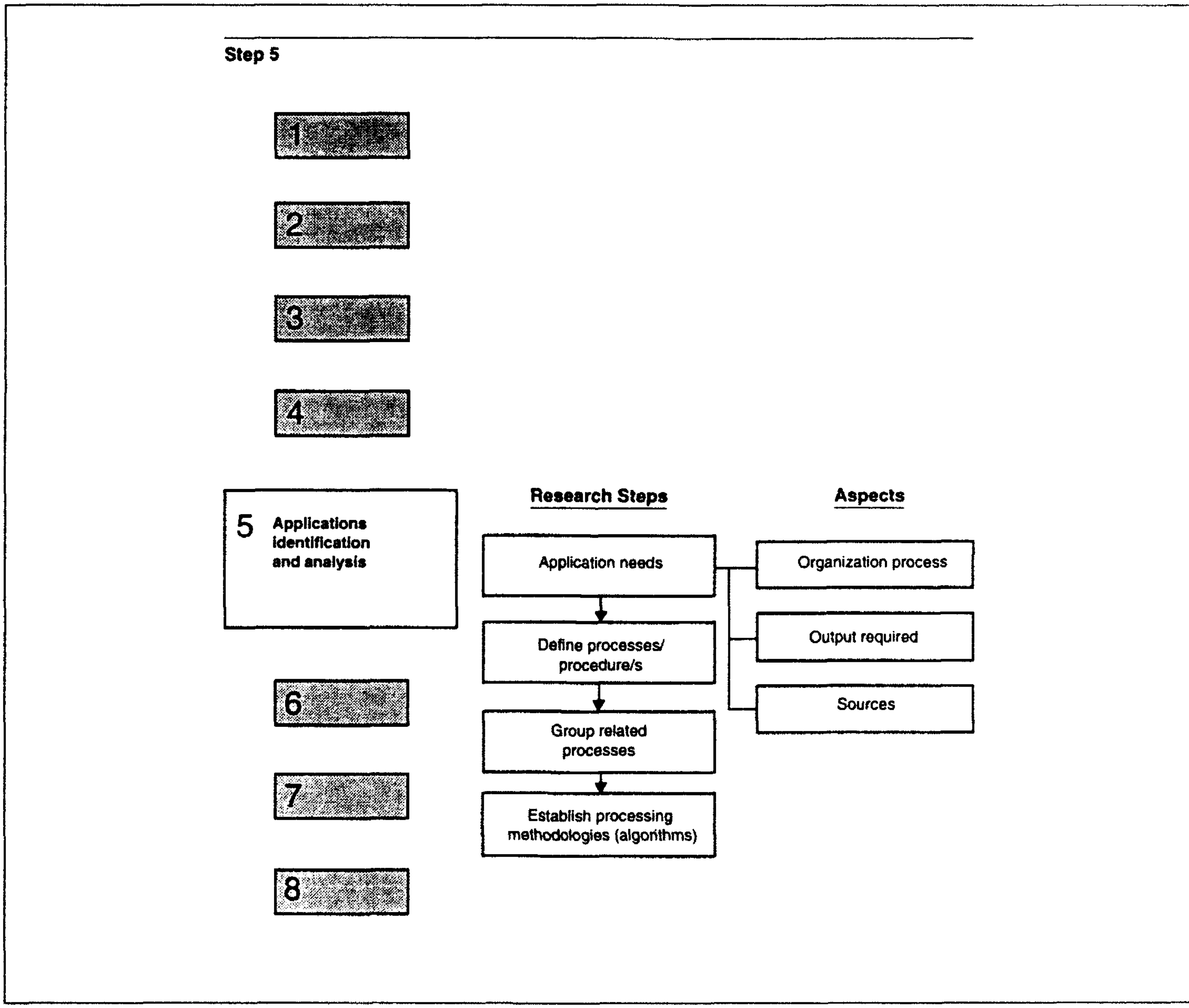


## Data Needs Identification and Analysis

Information is made up of data. It is important to identify all data, who creates and uses it, and how it flows through the organization. Different organizational entities use different subsets of data, all of which relate to the same piece of information. In our supply system example, data that relate to a particular part could include size, weight, length, number of units on hand or on order, average number of parts used in a month, etc.; many locations may use some or all of the data. Because of this data sharing, standard data names and definitions should be contained in a standard data dictionary accessible by all organizational entities.

The compilation of data, including who creates and uses it and how, presents a stable basis for the processes and information used by the organization to accomplish its mission. This compilation is called the data architecture. Some methodologies combine the information and data architectures; we separated them because we believe it is important to distinguish between information used to perform and manage functions and the detailed data types and subsets that make up the information. The data architecture will be a key ingredient to determining the type of system needed to manage the data within the prospective system, and when used with the information architecture, will determine the applications needed to process the data.

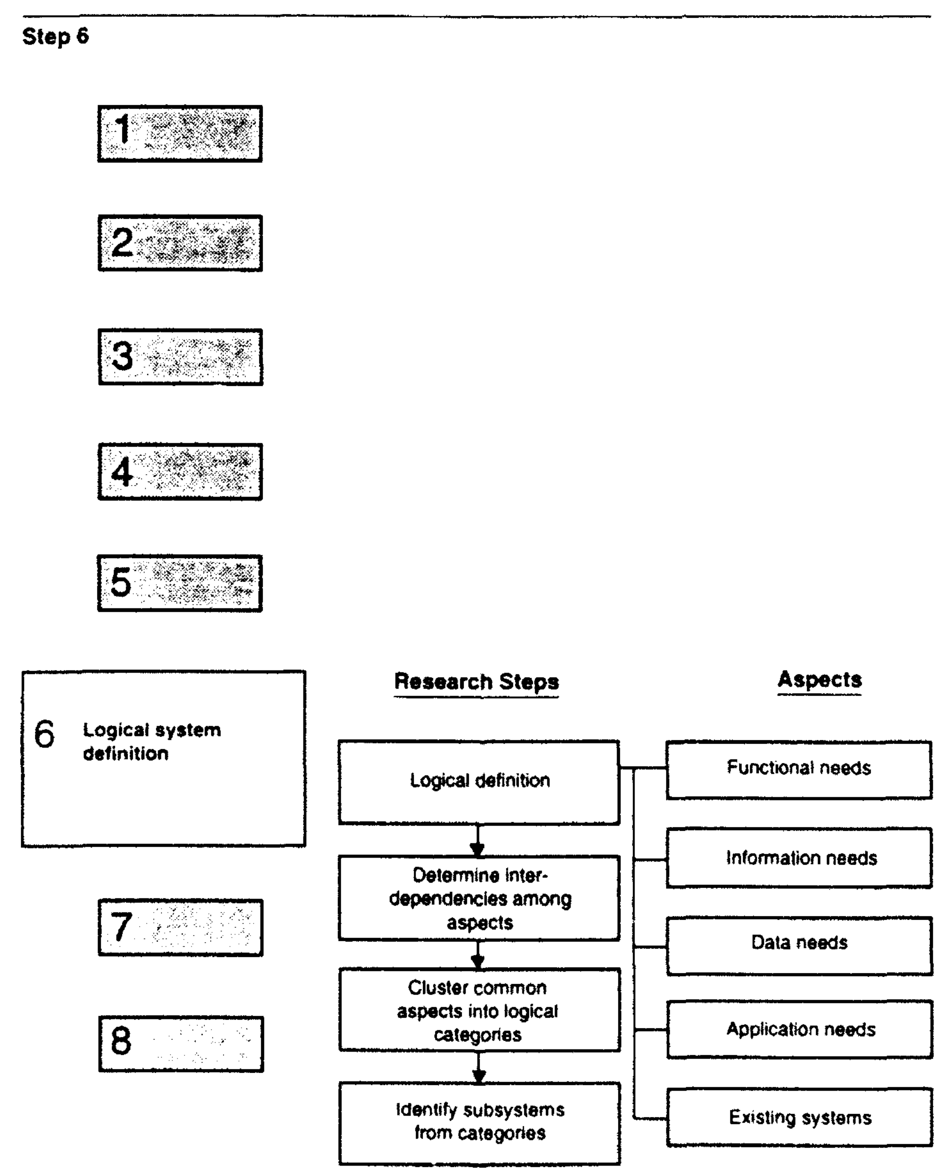
Figure 6: Step 5: Applications Identification and Analysis



## Applications Identification and Analysis

Once the information needed to perform or manage functions has been identified and the basic data and its structure defined, algorithms that manipulate (for example, combine or compare) the data to produce the information must be designed. For example, the Federal Aviation Administration receives data such as altitude and azimuth for aircraft. Algorithms are designed that use these data to compute the aircraft's position and display it on a screen for use by an air traffic controller. These algorithms are embodied in computer software known as applications programs. A description of how the applications manipulate data to create information, and how this relates to the functions and the mission, is called the applications architecture.

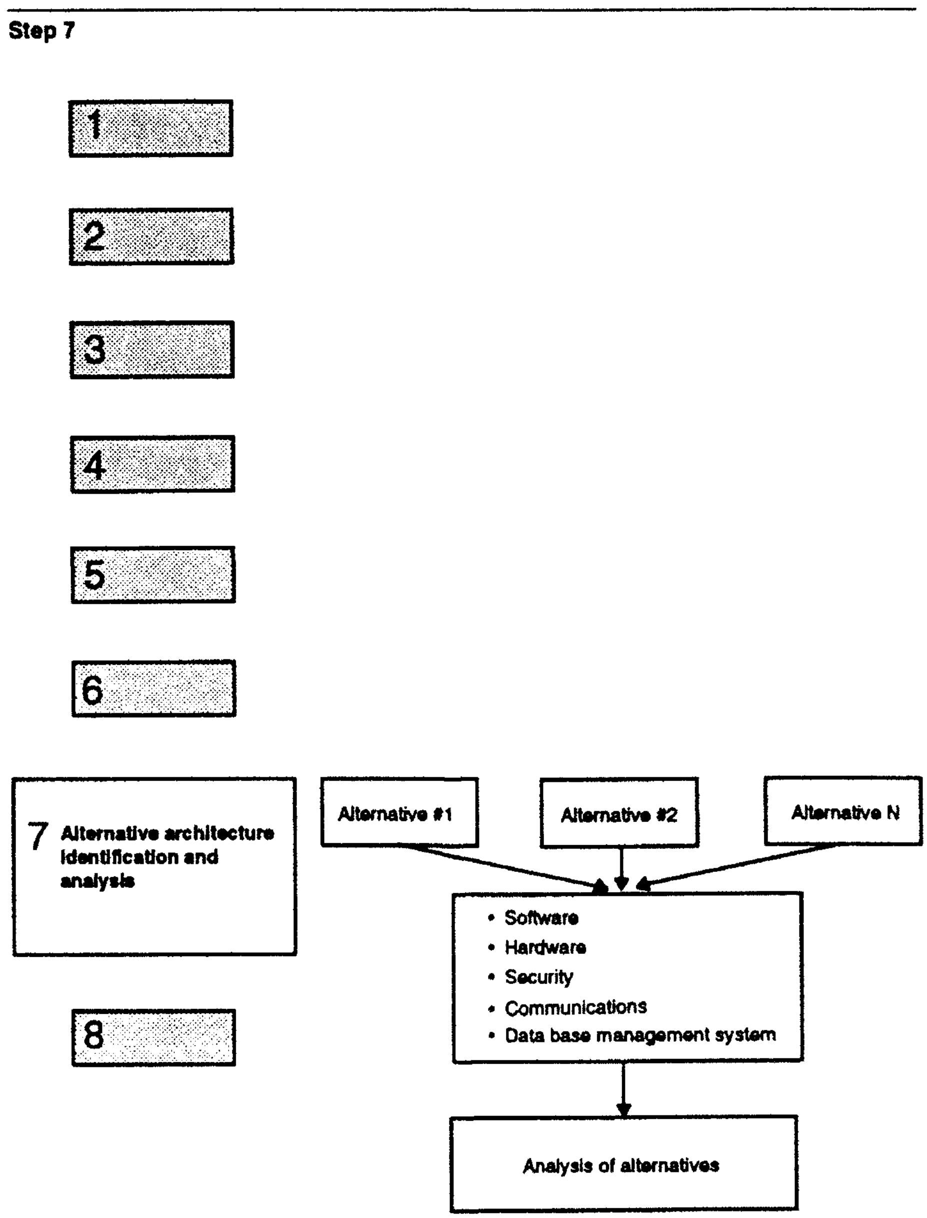
**Figure 7: Step 6: Logical System Definition**



## Logical System Definition

The logical system definition is a composite of all the steps discussed to this point—the functional architecture, the information architecture, the data architecture, and the applications architecture. All of these views of the system are interrelated, and each one builds upon its predecessor. The logical view of the system provides a profile of how the applications and their related data are grouped to form subsystems and how those subsystems support the various functions required to achieve the mission. For example, spare parts data and the application programs used to compute inventory would be grouped, possibly with other related data elements and applications, to form a spare parts supply subsystem. This subsystem would provide information to a vehicle repair function that must keep vehicles used by assault troops ready for use at all times. The logical system definition describes the relationships among and between the functional, information, data, and application architectures and provides criteria for analyzing various alternative hardware, software, communications, security, and data management architectures.

**Figure 8: Step 7: Alternative Architecture Identification and Analysis**



## Alternative Architecture Identification and Analysis

A given logical architecture can be implemented using many different system architectures. It is important, therefore, to consider a broad range of alternatives (each discussing hardware, software, communications, data management, and security considerations) before selecting a specific target architecture. Each alternative architecture should address the current and potential future missions and be analyzed in terms of its present and future operational effectiveness, flexibility, maintainability, and cost effectiveness.

## Hardware Considerations

The types, configuration, and capabilities of hardware components proposed in each architectural alternative should be identified and analyzed. The advantages and disadvantages of the hardware types (processors, mass storage devices, printers, etc.); configuration (distributed, centralized, etc.); and capabilities (real-time or batch, processing speed, etc.) should be fully explained to provide a basis for comparing the various alternatives. In addition, each alternative should discuss system characteristics in terms of flexibility, expandability, supportability, reliability, maintainability, and fault tolerance, as well as cost over the entire system life cycle. A key consideration in choosing system hardware is its flexibility and expandability to meet additional requirements.

## Software Considerations

Systems and applications software should include characteristics that minimize software development costs and risks, and provide software that is effective, efficient, and open and flexible for meeting future mission and information requirements without costly redesign. Desirable software characteristics include reliability, efficiency, testability, flexibility, portability (to the maximum extent possible), reusability (where feasible), maintainability, and adherence to open systems standards such as Portable Operating System Interface for Computer Environments (POSIX); each alternative should address how these characteristics are to be provided. (See the glossary for definitions.)

Developing software with these characteristics requires adopting and institutionalizing methodologies and standards for designing, coding, testing, and documenting software projects. For example, coding software modules in a standard language, limiting the number of software languages, and using consistent development tools, such as computer-aided software engineering (CASE) tools and programming design languages, across the development environment are a few techniques that foster effective and efficient software development and reduce development and maintenance costs and risks. The languages to be used and how the software is to be structured should be identified and analyzed in each alternative architecture. The larger and more complex the system, the greater the need for a structured, standardized development approach; how this is to be achieved should be identified and evaluated.

## Communications Considerations

The need to exchange information has become an important factor in federal agencies' operations as they are increasingly sharing information not only among themselves, but also with state and local governments, international agencies, and private industry. In the past, vendor-unique solutions to data communications needs have resulted in systems that are isolated islands of processing that cannot communicate with each other. If systems are to be interconnected in the future, information exchange needs must be thoroughly identified and analyzed, and open and flexible communications alternatives must be identified.

These analyses should address requirements for local and wide-area communications, survivability, security, and reliability, as well as approaches to standardized communication protocols, such as Government Open Systems Interconnection Profile (GOSIP), and message formats. These analyses must also include detailed assessments of current and future work loads in terms of data volume, types, formats, and transmission speed to arrive at a comprehensive and cohesive communication architecture.

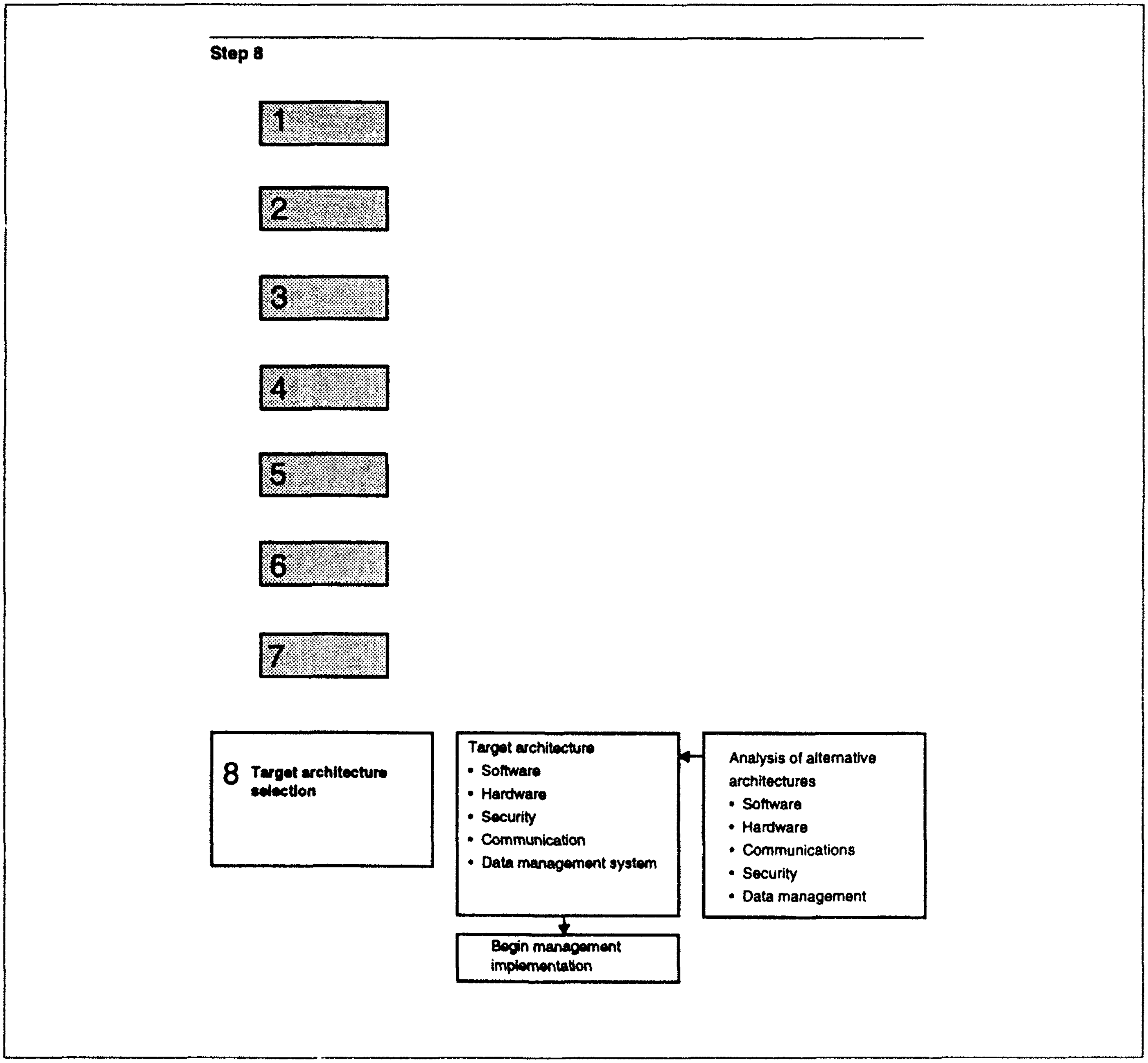
## Data Management Considerations

The proliferation of computing and data storage capabilities has resulted in a corresponding proliferation of redundant and inconsistent data, but increasingly, organizations are viewing data and information as resources that must be managed. Each alternative architecture should address how data will be collected, structured, and stored; how data flows into, within, out of, and among subsystems; and how access to data will be provided so that it can be used by the organization. To maximize the data sharing potential, a data management system should be as open and flexible as practicable. To accomplish such a system effectively, standards for data elements, naming conventions, a data dictionary, and data base management and access methods should be identified and analyzed for each alternative. Further, commercial off-the-shelf data management solutions should be identified and their capabilities evaluated in relation to the needs of the system in which they will be implemented (for example, distributed versus single site systems, real-time versus batch processors, secure versus nonsecure systems).

## Security Considerations

Security is a critical and potentially costly computer system element. Potential costs include money, schedule, and system performance. It is not possible to meet stringent requirements for trust (security, integrity, high reliability, and safety) without affecting system performance, cost, and operability. A detailed, up-front analysis should be performed to identify the security benefits to be derived by investing in hardware and software to satisfy security requirements versus the performance degradation and increased development cost and time that will result from attempting to implement stringent security features. However, to provide a basis for conducting such a detailed analysis, the organization must have in place an approved and well-designed security policy and security concept of operations.

Figure 9: Step 8: Target Architecture Selection



## Target Architecture Selection

Historically, information system architectures and resulting designs have been selected primarily on the basis of acquisition cost; less emphasis has been given to operational effectiveness, maintainability, and flexibility. However, at this point in strategic information planning, several alternative architectures should have been analyzed. These alternatives should have included different approaches to meeting the organization's mission and may have included new technologies (hardware and software) that must be developed; existing technologies that could be purchased from one or more vendors; all or portions of existing systems that could be upgraded, modified, or expanded; or some combination of each. The architecture selected from these alternatives should be as open and flexible as possible to allow for future change and growth.

Selecting a target architecture—the architecture that will be used as a blueprint to guide information system development, upgrade, and expansion over time—from the alternatives identified requires that the organization consider how well each architecture meets the organization's near- and far-term information needs; the maintainability, operability, and expandability of each alternative throughout the system's life; the availability and maturity of the technologies being proposed and the risk involved in using a new or relatively immature technology; and the costs of implementing each architecture, including acquisition, expansion or upgrade, and maintenance costs over the system's life. This selection process must include all organizational participants (managers, users, operators, maintainers, and designers) and consider organizational policy, current and anticipated future budgets, and all aspects of system life cycle management.

# Major Contributors to This Publication

## Information Management and Technology Division, Washington, D.C.

Michael T. Blair, Assistant Director
Dr. Rona Stillman, Chief Scientist
L. J. Latham, Technical Assistant Director
Sally Obenski, Senior Technical Adviser

## Denver Regional Office

Frederick Day, Evaluator-in-Charge
Sigrid McGinty, Senior Evaluator
David Powner, Staff Evaluator
Michael Gorin, Staff Evaluator

# Glossary

**Access Methods**
A method of transferring data between the computer's main storage and an input/output device.

**Application**
A system or problem to which a computer is applied; applications may be computational (arithmetic) or procedural.

**Application Software**
Computer programs that perform data processing functions rather than control functions.

**Architecture**
A description of all functional activities to be performed to achieve the desired mission, the system elements needed to perform the functions, and the designation of performance levels of those system elements. An architecture also includes information on the technologies, interfaces, and location of functions and is considered an evolving description of an approach to achieving a desired mission.

**Baseline Architecture**
The initial architecture that is or can be used as a starting point for subsequent architectures or to measure progress.

**CASE Tools**
Computer-Aided Software Engineering (CASE) tools.

**Coding**
Creating the software used by the computer from program flowcharts.

**Communications**
The transmission of data or information from one place or piece of equipment—a computer, for example—to another.

**Configuration**
The relative or functional arrangement of components in a system.

**Data**
A general term used to describe facts, including those that refer to or describe an object, idea, condition, situation, or other factor, numbers, letters, or symbols. It connotes basic elements of information that can be processed or produced by a computer.

| | |
|---|---|
| Data Fusion | The integration of data. |
| Data Structure | The logical relationships among data units and description of attributes or features of a piece of data (e.g., type, length). |
| Data Management | Providing or controlling access to data stored in a computer and the use of input/output devices. |
| Distributed Processing | Data processing that is performed by connected computer systems at more than one location. |
| Efficiency | The amount of computing resources and code required by a program to perform a function. |
| Fault Tolerance | The ability of a processor to maintain effectiveness after some subsystems have failed. |
| Flexibility | Effort required to modify an operational program. |
| Function | An activity or operation. |
| Hardware | The physical equipment or devices that form a computer and its peripheral components. |
| Information | Knowledge or intelligence created from or made up of data elements. |
| Information Flow | The sequence, timing, and direction of how information proceeds through an organization. |

| | |
|---|---|
| Integrity | The state achieved by maintaining and authenticating the accuracy and accountability of system data, hardware, and software. |
| Interoperate | To provide services to or accept services from other systems, subsystems, or components and to use the exchanged services effectively. |
| Life Cycle Management | The process of administering an automated information system throughout its expected life, with emphasis on strengthening early decisions that affect system costs and utility throughout the system's life. |
| Local Area Network | A communication system designed for intra-building data communications. |
| Logical System Definition | The planning of an automated information system prior to its detailed engineering design. This would include the synthesis of a network of logical elements that perform specific functions. |
| Maintainability | Effort required to locate and fix an error in an operational program. |
| Modular Software | Software that is in self-contained logical sections, or modules, which carry out well-defined processing actions. |
| Operating System | Software that controls the execution of computer programs and provides services such as scheduling and input/output control. |
| Portability | Degree to which a computer program can be transferred from one hardware configuration and/or software system environment to another. |
| Real Time Processing | Operations performed on a computer simultaneously with a physical process or activity such that the answers obtained through the computer operations can affect the process or activity. |

| | |
|---|---|
| Reliability | Extent to which a program can be expected to perform its intended function with the required precision on a consistent basis. |
| Reusability | Extent to which a program can be used in other applications; related to the packaging and scope of the function that programs perform. |
| Safety | Freedom from those conditions that can cause death or injury, or damage to or loss of data, hardware, or software. |
| Security | Preservation of the authenticity, integrity, confidentiality, and ensured service of any sensitive or nonsensitive system-valued function and/or information element. |
| Software | Computer program instructions and data definitions that enable the computer hardware to perform computational and control functions. |
| Systems Engineering | The systematic application of technical and managerial processes and concepts to transform an operational need into an efficient, cost-effective system using an iterative approach to define, analyze, design, build, test, and evaluate the system. |
| System Parameter | A factor or property whose value determines a characteristic or behavior of the system. |
| Testability | Effort required to test a program to ensure it performs its intended function. |
| Traffic Load | The number of messages input to a network during a specific time period. |