Chapter 1: Information Technology Project Initiation

Information technology (IT) projects receive attention in the computer industry because they influence almost everyone. Whether the IT project manager is managing projects for business, financial, academic, government, military, or nonprofit organizations, he or she needs accurate computerized information to make decisions in a short amount of time. This computerized information is as good as the design and management of the IT project systems.

Effective IT Project Management Techniques

A modern IT project manager's task is threefold: to supervise IT computer professionals, understand state−of−the−art techniques, and make the IT project successful. The manager should enhance his or her ability to understand project management techniques, modern technologies, and system development methods and tools. A manager cannot effectively manage a technical team unless he or she understands the basics of what the team members are doing and the technical aspects of the organization. In this way, he or she increases the efficiency of the operation, maintains a positive environment, and reduces turnover.

A good IT project manager is hard to find. A manager must be able to handle his or her team well to deliver a quality product within the defined budget and schedule. A manager who is well versed in the state−of−the−art techniques can analyze the user's requirements, design the IT system, develop the software, and deliver the finished product to the satisfaction of the users.

A good manager must recognize what tools are required and use them in a knowledgeable manner to plan, estimate, schedule, and develop the IT project's complete work breakdown structure (WBS) without guesswork or reliance upon another individual's memory or experience. The manager is a leader who increases interpersonal relationships with his or her group, communicates effectively, and guides with the vision necessary to lead them to success. He or she is practical in making decisions and recognizes the importance of objectivity, vision, and initiative in arriving at sound, quality decisions.

A good manager also identifies and develops talent in others. He or she arranges special IT technical training to foster individual growth and efficiently manages time and resources through the delegation of tasks to those who are the most suitable. The manager is constantly assessing skills in others and escalating responsibilities accordingly. He or she gives due respect, promotions, and raises to maintain high morale of team members.

The manager selects the methodology and techniques that are the most suitable for development of the IT system. He or she staffs the project in phases as needed and provides the necessary technical training to bring himself or herself and the team up to standard. The manager understands users' acceptance criteria of the system products. He or she stresses that the team members should develop a prototype model and simulate the system to aid in understanding the users' requirements for the development and maintenance of the system. The manager encourages the users to examine cases at the completion of the project to confirm that the users' requirements are testable. The manager appoints other supporting personnel as shown in Box 1−1. Figure 1−1 graphically presents this system of support personnel and various phases

Box 1−1: List of Support Personnel

- Technical training consultant
- System engineers and analysts
- System testers

- Quality evaluation experts
- Configuration management personnel
- Internet and reuse experts
- Database and data dictionary administrators
- System designers
- Software designers and analysts
- Software engineers and programmers
- User's manual writers
- User training personnel
- System operational personnel
- System administrative personnel

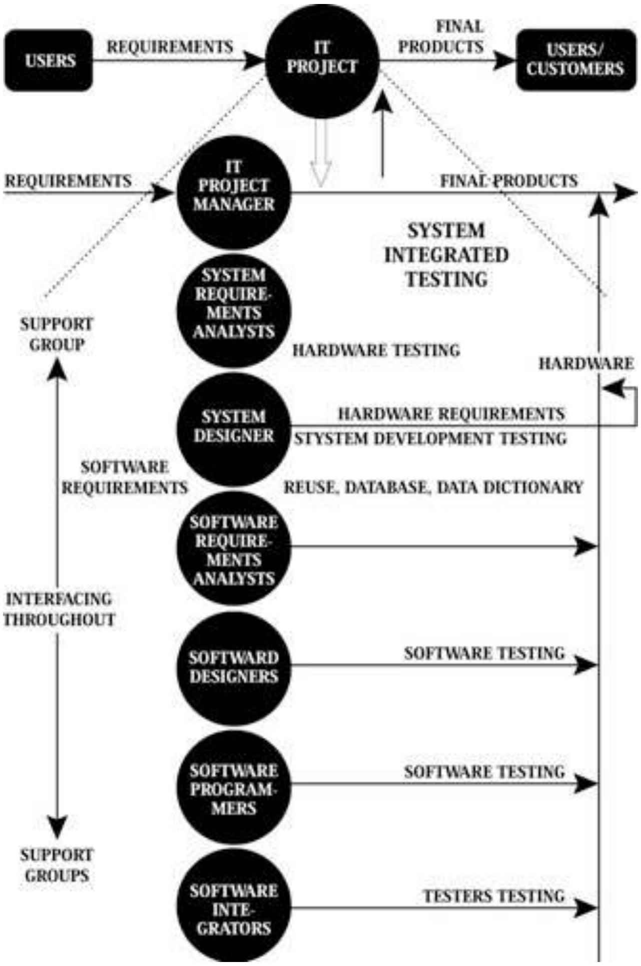

Figure 1−1: Various phases of the IT project

The manager sets standards for the documentation that is to be presented to the user and uses these standards to interface with the user throughout the development of the system. The manager guarantees that the user will be fully trained to accept the required deliverable and maintains healthy communication between his or her team members and the user. The manager ensures that graphics tools are used wherever necessary to maintain good communication and for the success of the project.

Above all, the manager remembers that a good system design is the key to producing readable, understandable, reliable, and easily maintained system products. The manager understands that he or she is responsible for setting the standards and controls that guide his or her team in the successful completion of the project.

IT Project Management Objectives

The IT project manager must understand the users' requirements, hire quality team members, and manage the team members well. He or she must be able to complete the project successfully on time, within the budget, and to the satisfaction of the users.

System development has faced problems in the IT industry since its inception. The cost of hardware is declining with time, whereas the cost of software development is steadily increasing with time (Figure 1−2). Many IT projects are either challenged or not completed, and challenged projects are often completed with cost overrun and delays in delivery schedules. The percentage of application development failures is greater than the number of successfully completed projects for many reasons.

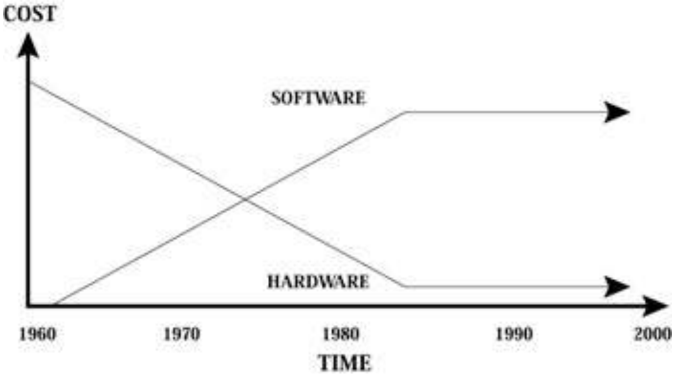

Figure 1−2: Cost of hardware and software trends with time

Since the inception of computers, there has been a desire for system development to reduce cost and delivery schedules while producing quality systems with few errors that contain system development goals and principles. Developers in the computer industry are continually striving to find an efficient method to achieve such goals. They have tried many methods and techniques, such as structured programming, structured analysis and design, and an object−oriented approach, but they have not achieved success.

Traditional methods, such structured analysis and design, contain functional decomposition, hierarchy charts, data flow diagrams, and state transition diagrams. Data−oriented analysis and design, events−oriented analysis and design, and their variations deal with operations and data as distinct and loosely coupled. Operations determine the structure of the system, and data are of secondary importance.

The object−oriented approach to system software development is preferable because it leads to construction of an application that consists of reusable classes and their objects. Objects are discrete software components that contain data and operation procedures together. Software systems are partitioned based on service objects, which interact between the client and server machines. Data determine the structure of the software.

Characteristics of a Good IT Project Manager

Personal Characteristics

- Is technically qualified
- Is a decision maker
- Is honest and creates a relaxed atmosphere
- Possesses the art of saying no without offending others
- Believes in managing time and people

Project−Related Characteristics

- Achieves the objectives and goals of the project within the established schedule, budget, and procedures
- Develops IT projects on budget and on time to the complete satisfaction of the users
- Has experience in related or similar projects
- Can control project outcomes by measuring and evaluating performance against established objectives and standards
- Develops and executes contingency plans to meet unforeseen circumstances and problems
- Develops and implements decisions relating to planning
- Is willing to redefine goals, responsibilities, and schedules as necessary to get the project back on track in case the schedule slips or the project is over budget
- Establishes and meets real priorities and deadlines
- Believes in good planning to reduce pressure and stress and increase productivity
- Establishes long−term and short−term planning

Team−Related Characteristics

- Has good communication and managerial skills
- Is able to plan, organize, lead, motivate, and delegate proper responsibilities to team members
- Respects team members and has their confidence and respect
- Shares success with the team members
- Selects the right person for the right job
- Shows appreciation to good workers
- Gets others in the organization to accept his or her ideas and carry out his or her plans
- Delegates duties and maintains control
- Believes in professionally training people for their delegated jobs
- Considers himself or herself as a part of the team
- Creates structured discipline
- Recognizes individual differences and takes advantage of individual strengths
- Provides work that stimulates a feeling of personal respect and professional growth
- Allows sufficient time for ideas to develop and mature
- Allows free time and encourages openness
- Understands the team members and creates effective communication
- Monitors his or her team members on a regular basis for the following types of people and takes necessary actions:

    - ♦ People who waste their time and that of others
    - ♦ Opportunists who steal others' ideas
    - ♦ Critical people who find only mistakes in others' work
    - ♦ Idle people who are unproductive
    - ♦ Egotistic people who brag about themselves
    - ♦ Gossips who spread rumors

IT Project Initiation

A project starts with the acceptance of a proposal by a customer who is willing to fund the project. A project can start within an IT organization with a statement of work or requirements by users. At this point the senior executives of the IT organization give a go−ahead signal to initiate the project. The executives select a suitable project manager who starts planning for the project.

Planning covers coordination of available resources in an organization. The plan focuses on integration of system development goals and principles throughout an organization. To save cost and time, the plan includes adoption of a hardware and software reuse approach to systems development and maintenance throughout the life cycle. Changing old habits of a developing system via stovepipe practices is difficult. Practice of system reuse via the Internet requires time to change culture and mindset.

The manager develops a system infrastructure model, which is defined as the architecture of a domain. Domain architecture is defined as the high−level graphic model that concerns the main assets of a domain and their roles and relationships. The architecture specifies a generic description from the user's viewpoint and interface, input and output operations, memory organization, and control of a system. The architecture describes the nature, configuration, and interconnection of the major players of the domain. Architectures are necessary to develop reusable assets that adequately meet domain−specific requirements and are properly designed to interface and interact with other domain assets. Architectures are categorized as follows:

- Domain−specific architecture captures the commonality and variance of related systems within the domain and promotes the development of reusable assets to conform to those architectures. Domain−specific reuse exploits those domains to support building of the specific architectures.
- Process−driven architecture is the transparent assimilation of reuse and is an integral part of the system development process.
- Product line architecture is a set of systems that share a common infrastructure and satisfy the requirements of one or more domains. The product line approach involves the development and application of reusable assets for system development and maintenance among systems that possess similar infrastructures. Architecture centric is the reuse−oriented architecture in high−leverage domains to spur investment in the creation of reusable assets that comply with those architectures. The infrastructure of an organization manages reuse policies, standards, processes, business practices, roles and responsibilities, technology, training, and coordination across product lines.

Preparing for a cultural change is a continuous process in an organization. Almost everyone who has developed an IT system has practiced system reuse, and its benefits have been known for many years since the inception of computers. Much of that practice has been informal and ad hoc. Developers now recognize that the real payoff from reuse is achieved by moving to a more systematic approach. Preparations for cultural change are categorized as follows:

- ***Paradigm changes*** are defined as the overall structure of the process and are driven by the product line characteristics and the product line life cycle, organization readiness, and constraints.
- ***Process changes*** are defined as the activities and flows that fit into a given paradigm.
- ***Method changes*** are defined as the detailed techniques for activities within the processes.

Tools provide automated support to methods and processes. A reuse standard includes guidelines for requirements, reusability metrics, and certification.

Assessment of potential problems for implementation of system reuse in an organization is perpetual. Reuse is not a stand−alone technology. Developers are getting close to assessing problems that only raise more questions about reuse. The practice of reuse requires that reuse considerations be built into the system's development and maintenance paradigm together with the supporting processes, methods, and tools. Reuse is built into the system's development process, especially for a product line approach, because reuse decisions influence the architecture decision and requirement specifications. If developers introduce reuse in the later phases of system development, major reuse opportunities are likely to be precluded. Different types of domains may require different paradigms, processes, methods, and tools. Developers must define alternative practices and environments to accommodate different requirements.

Standards and practices of system reuse establish the guidelines for use of reusable assets during system development and maintenance. These standards include the guidelines for practice during system requirements analysis, domain analysis, system design, software requirements analysis, software design, software implementation, documentation, repository management, and assets certification and validation processes. An organization establishes such a system reuse standard to be followed by all the developers. The requirement analysis standard establishes guidelines for reusable assets during the software requirements phase. During this requirement models phase, documentation is created in compliance with the established system reuse standard. The main features of requirement standards are the context diagram, requirements model, data dictionary, communication model, and domain identification.

Identification of Project External Interfaces

Identification of project external interfaces is important for design and estimation of resources and schedule. The context diagram (Figure 1−3) establishes external interfaces with other systems and objects. This diagram is a high−level view reflecting the project.

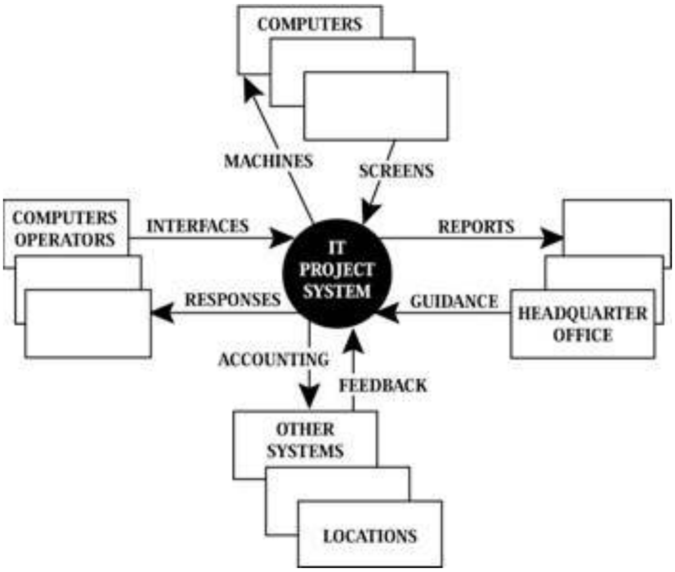

Figure 1−3: Context diagram

The system receives input to start from human operators. The system communicates externally with other systems and computers and submits reports to the headquarters office. To design a system, the developer must know with whom the system will be externally interfacing. The developer should state this information clearly in the system requirements to estimate resources and schedule.

Figure 1−3 shows that more than one human operator, computer, other system, and headquarters office can be involved. At this time the developer must know the external interfaces for the system under design.

Identification of Users, Customers, and Stakeholders

The IT project manager must identify the users, customers, and stakeholders at the beginning of the project. Figure 1−4 establishes the interfaces of users, customers, and stakeholders.

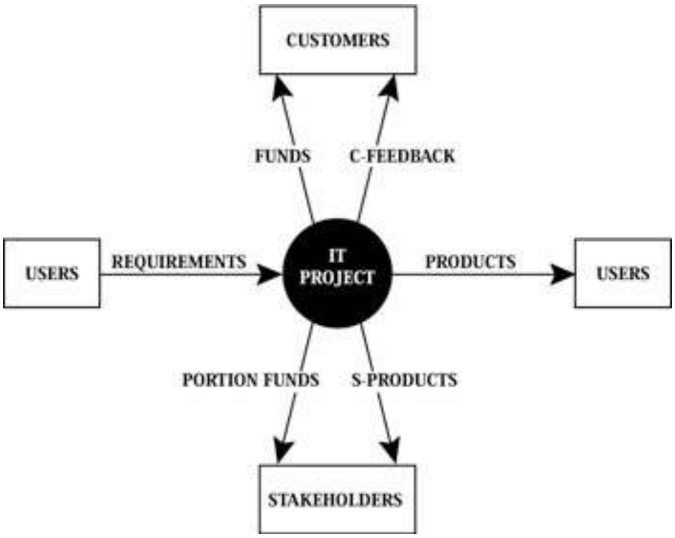

Figure 1−4: Relationship among users, customers, and stakeholders

Users

Users are those who provide requirements to design a system; they have need for that system and are going to use the system products. Users define their requirements and generate use cases to the system developers for testing of their requirements. Their involvement from the beginning of the project's development is necessary to reconfirm that their requirements are understood and that the system will be user friendly.

Users also state acceptance criteria. Acceptance testing involves the proper planning and execution of tests that demonstrate that the implemented system satisfies the requirements. The tests include functional, performance, regression, and stress tests. For their own satisfaction, users perform the acceptance tests to see for themselves that the result achieved is correct. Some of the tools used for the acceptance tests are the coverage analyzer, timing analyzer, and standards checker. Users detail the required documentation that is needed to support the system. The documentation relates to operation, training, and maintenance of the system. Users also require system design blueprints, hardware features, software design, source and object codes, and testing results.

Customers

Customers finance the project, which starts only if they have available funds. They have a vested interest in the project and desire a guarantee and warrantee of the accuracy of the system. Customers also keep track of the system development to ensure that it is completed on time and within the schedule. Sometimes one person or organization can be both the customer and the user.

Customers and users form a team with IT system developers to monitor the progress. They ensure that all their requirements are accurately included in the system. The data dictionary documents the elements and definitions of interfaces, objects, classes, data groups, and data elements in detail. This document standardizes the process so that the system developer uses standardized terms and definitions throughout the system development process. This process enhances the quality and efficiency of the system.

Stakeholders

Stakeholders also have a vested interest in the IT project and provide their share of funds to complete the project. They will use all or part of the system products. They also generate requirements and use cases. Stakeholders form a team with users and customers to monitor the success of the project, participating from the beginning to the end. They participate in all phases of system development and provide input for the success of the project. Stakeholders reuse the system products cost effectively.

IT Project Development Life Cycle Phases

IT project life cycle phases begin with the users' needs and requirements and continue until the successful completion of the project. The major life cycle phases include planning, prototype modeling, and the WBS of tasking, scheduling, milestone, budgeting, and system development. Most IT systems are driven by costly software; therefore the majority of time is allocated to software development and testing. Table 1−1 and Figure 1−5 illustrate well−managed IT project phases and allocation of resources.

Table 1−1: Allocation Of It Project Resources

Phase	Allocation (%)
Study and planning	8
System requirements analysis	6
System design	7
Infrastructure and prototype model	5
Reuse planning and Internet	6
Software requirements analysis	8
Software design	8
Software coding and testing	15
Software integration testing	10
System integration testing	15
System deployment	10
Audits	2

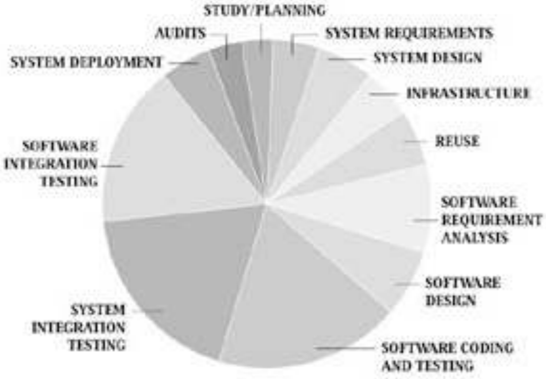

Figure 1−5: Major phases of an IT project

These percentage numbers are only for guidance. Depending on a project's size, the percentage of phases varies. Other variances are complexity of requirements, environment, unfamiliarity of software language and hardware, and manager and team member experience with the related projects. The manager provides the team with proper education and training that help in reducing risk and removing resistance among the team members.

Case Study 1−1

The following is the current method of ticket purchase at the Modern Airlines Corporation:

A passenger normally reserves the ticket and picks it up at the counter. At the ticket window, the passenger tells the ticket agent his or her name, either the flight number or departure time, and the destination of the flight.

The ticket agent enters the information into the computer terminal. When the system is 'up,' the computer finds the customer's reservation and generates a ticket. Once the ticket agent gets the completed ticket, he or she asks the customer for payment. The agent can only accept payment in cash, checks, or credit cards with proper identification.

The ticket agent notes the method of payment on a copy of the ticket that he or she files with the stack of purchased ticket copies. He or she writes the ticket number on the check, cash receipt, or charge receipt and files it in the payment drawer. The agent puts the ticket into a ticket packet and hands it to the customer.

The Modern Airlines Corporation wants to fully automate the ticket generation system for their customers at various airports. Bob is selected to manage this project because he possesses almost all the qualities of a manager that have been defined in this chapter. Bob follows the following steps to initiate the project:

1. He studies the project statement and tries to understand the project objectives, requirements, and goals.
2. He discusses the project with senior management and tries to understand the scope of the project, resources, schedule (24 months), and funding ($400,000).
3. He identifies the customer, users, and stakeholders.
4. He draws a context diagram to show the roles of the customer, users, and stakeholders graphically (Figure 1−6).

   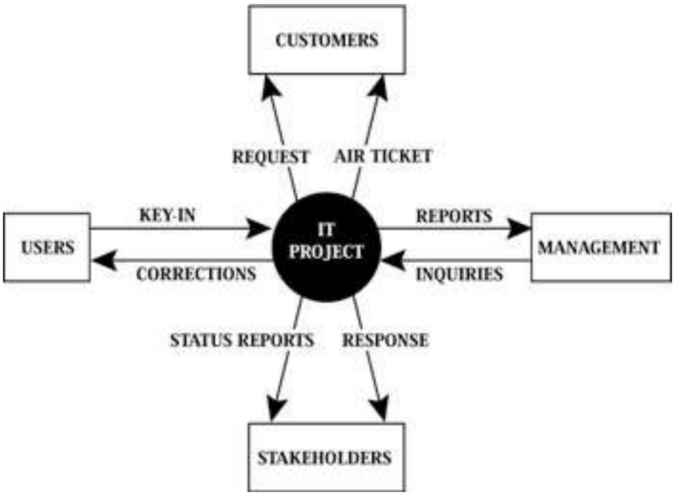

   Figure 1−6: Generation air−ticket relationship among customers, users, stakeholders, and management
5. He sets up a small team among the management, customer, users, and stakeholders to discuss the progress of the project on a regular basis.
6. He allocates the money as follows:

Study and planning	$32,000

System requirements analysis	$24,000
System design	$28,000
Infrastructure and prototype model	$20,000
Reuse planning and Internet	$24,000
Software requirements analysis	$32,000
Software design	$32,000
Software coding and testing	$60,000
Software integration testing	$40,000
System integration testing	$60,000
System deployment	$40,000
Audits	$8,000

This case study is revisited in later chapters.

Project Management Checklist

A good IT project manager must do the following:

- Identify users, stakeholders, and customers
- Communicate with users, stakeholders, and customers throughout the project
- Educate and train users, stakeholders, and customers about the project
- Understand the requirements of users, stakeholders, and customers
- Set up the project book
- Plan the project
- Staff the project
- Establish project phases
- Allocate the budget for various phases and activities
- Estimate resources
- Estimate the level of efforts
- Model and simulate the project
- Establish standards and controls
- Form project tasks teams
- Staff task teams
- Provide necessary training
- Staff support groups
- Provide a team concept
- Establish a schedule and milestones
- Provide a guideline to implement success
- Provide state−of−the−art tools
- Encourage the reuse of assets, the Internet, and telecommunications
- Track the deliverables
- Measure the success
- Create effective communication among the members of the project team
- Gain the users' confidence
- Show results
- Document the products
- Provide a smooth transition of the product to the users

- Provide effective training that enables the users to use the product
- Evaluate achievements
- Pass on due credits to the staff
- Share profits and satisfaction
- Use the Internet to learn about related projects