The Systems Development Environment

What is an Information System?

In a simplest sense, a system that provides information to people in an organization is called **information system (IS)**. Information systems in organizations capture and manage data to produce useful information that supports an organization and its employees, customers, suppliers and partners. So, many organizations consider information system to be the essential one.

Information systems produce information by using data about significant *people*, *places*, and *things* from within the organization and/or from the external environment to *make decisions*, *control operations*, *analyze problems*, and *create new products or services*. **Information** is the data shaped into a meaningful form. **Data**, on the other hand, are the collection of raw facts representing events occurring in organizations or the environment before they have been organized and arranged into a form that people can understand and use.

The three activities to produce information in an information system are *input*, *processing*, and *output*. **Input** captures or collects row data from within the organization or from its external environment for processing. **Processing** converts these row data into the meaningful information. **Output** transfers this information to the people who will use it or to the activities for which it will be used. Information systems also require **feedback**, which is used to monitor the current information system output and compare it to the system goal.

The two types of information systems are **formal** and **informal**. **Formal information systems** are based on accepted and fixed definitions of data and procedures for collecting, storing, processing, disseminating, and using these data with predefined rules. **Informal information systems**, in contrast, relay on unstated rules.

Formal information systems can be **manual** as well as **computer based**. **Manual information systems** use paper-and-pencil technology. In contrast, **computer-based information systems (CBIS)** relay on computer hardware and software for processing and disseminating information.

Types of Information Systems

In practice there are several classes of information systems in organizations. Each class serves the needs of different types of users. These are *transaction processing system (TPS), management information system (MIS), decision support system (DSS), executive information system (EIS), expert system, communication and collaboration system*, and *office automation system*.

1. Transaction Processing System (TPS)

These are the computerized systems that perform and records the daily routine transactions necessary to conduct business. These systems serve the operational level of the organization. Some examples include *sales order entry*, *hotel reservation systems*, *payroll*, *employee record keeping*, and *shipping*.

Transaction processing systems are central to a business. TPS failure for a few hours can cause a firm's demise and perhaps other firms linked to it. Managers need TPS to monitor the status of internal operations and the firm's relations with external environment. TPS are also major producers of information for the other types of systems.

Online transaction processing systems (OLTPS) is an interactive data processing system that involves a direct connection between TPS programs and users. As soon as a single transaction is entered into a computer system, the program interacts immediately with the user for that transaction. It is often known as the live system where there is no time lag

between data creation and its processing. A good example of this system is *online ticket reservation system*.

2. Management Information System (MIS)

These are the information systems at the management level of an organization and serve management-level functions like planning, controlling, and decision-making. These systems provide reports that are usually generated on a predetermined schedule and appear in prearranged format. Typically, these systems use internal data provided by the transaction processing systems. These systems are used for *structured decision-making* and in some cases for *semi-structured decision making* as well. Salary analysis and sales reporting are the examples in which MIS can be used.

3. Decision Support System (DSS)

These systems also serve at the management level of the organization. These systems combine data and sophisticated analytical models or data analysis tools to support *semi-structured* and *unstructured decision-making*. These systems use internal information from TPS and MIS, and often information from external sources, such as current stock prices or product prices of competitors. DSS have more analytical power than other systems. *Contract cost analysis* is an example in which DSS can be used.

4. Executive Information System (EIS)

These systems are also called **executive support systems (ESSs)** and serve the strategic level of the organization. These systems are designed to address *unstructured decision making* through advanced graphics and communication. These systems incorporate data about external events such as new tax laws or competitors, but they also draw summarized information from internal MIS and DSS.

These systems are not designed to solve a specific problem but they provide a generalized computing and telecommunication capacity that can be applied to a changing array of problems. 5-year operating plan is an example in which EIS can be used.

5. Expert System

An expert system is an **extension of DSS** that captures and reproduces the knowledge and expertise of an expert problem solver or decision maker and then simulates the "thinking" or "actions" of that expert. These systems imitate the logic and reasoning of the experts within their respective fields.

Expert systems are implemented with **artificial intelligence (AI) technology** that captures, stores, and provides access to the reasoning of the experts.

6. Communication and Collaboration Systems

These systems enable more effective communications between workers, partners, customers and suppliers to enhance their ability to collaborate. These systems use network technology that allows companies to coordinate with other organizations across great distances. These systems create new efficiencies and new relationships between an organization, its customers and suppliers, and business partners redefining organizational boundaries.

7. Office Automation Systems

Office automation (OA) is more than word processing and spreadsheet applications. Office automation systems support the wide range of business office activities for improved work flow and communication between workers, regardless of whether or not those workers are located in the same office.

Office automation functions include **word processing**, **spreadsheet applications**, **e-mails**, **work group computing**, **fax processing**, **work flow management** etc.

Office automation systems can be designed to support both individuals and work groups. **Personnel information systems** are those designed to meet the needs of a single user. They are designed to boost an individual's productivity. **Work group information systems**, on the other hand, are designed to meet the needs of a work group. They are designed to boost the group's productivity.

Systems Analysis and Design

Information systems analysis and design is a complex, challenging, and stimulating organizational process that a team of business and systems professionals uses to develop and maintain computer-based information systems. Information systems analysis and design is an organizational improvement process. Systems are built and rebuilt for organizational benefits. Benefits result from adding value during the process of creating, producing, and supporting the organization's products and services.

Careers in information technology (IT) present a great opportunity for you to make a significant and visible impact on business. The demand for skilled information technology workers is growing.

An important (but not the only) result of systems analysis and design is **application software**, software designed to support a specific organizational function or process. In addition to application software, the total information system includes the hardware and systems software on which the application software runs, documentation and training materials, the specific job roles associated with the overall system, controls, and the people who use the software along with their work methods. Although we will address all of these various dimensions of the overall system, we will emphasize application software development.

In systems analysis and design, we use various *methodologies*, *techniques* and *tools* that have been developed, tested, and widely used over the years to assist people during system analysis and design.

Methodologies are comprehensive, multiple-step approaches to systems development that will guide your work and influence the quality of your final product – the information system. Methodologies use a standard set of steps. A methodology adopted by an organization will be consistent with its general management style. Most methodologies incorporate several development techniques.

Techniques are particular processes that will help to ensure that your work is well thought-out, complete, and comprehensible to other on the project team. Techniques also provide support for a wide range of tasks like conducting interviews, planning and managing the activities in a system development project, diagramming the system's logic, and designing the reports that the system will generate.

Tools are typically computer programs that make it easy to use and benefit from the techniques and to faithfully follow the guidelines of the overall development methodology.

To be effective, both techniques and tools must be consistent with an organizations system development methodology. Techniques and tools make easy for system developers to conduct the steps in methodology. These three elements (methodologies, techniques, and tools) work together to form an organizational approach to systems analysis and design.

Although many people in organizations are responsible for systems analysis and design, in most organizations the **systems analyst** has the primary responsibility. The primary role of a systems analyst is to study the problems and needs of an organization in order to determine how people, methods, and information technology can best be combined to bring about improvements in the organization. A systems analyst helps system users and other business managers define their requirements for new or enhanced information services. As such, a systems analyst is an agent of change and innovation.

A Modern Approach to Systems Analysis and Design

The analysis and design of computer-based information systems began in the 1950s. In the 1950s, development focused on the processes the software performed. Because computer power was a critical resource, efficiency of processing became the main goal. Emphasis was placed on automating existing processes often within single departments. All applications had to be developed in machine language or assembly language, and they had to be developed from scratch because there was no software industry. Because computers were so expensive, computer memory was also at a premium, so system developers conserved as much memory as possible for data storage.

1960s saw important breakthroughs in technology that enabled the development of smaller, faster, less expensive computers – minicomputers – and the beginnings of the software industry. Most organizations still developed their applications from scratch using their in-house development staff. Systems development was more an art than a science. This view of systems development began to change in the 1970s, however, as organizations started to realize how expensive it was to develop customized information systems for every application. Systems development came to be more disciplined as many people worked to make it more like engineering. Early database management systems, using hierarchical and network models, helped bring discipline to the storage and retrieval of data. The development of database management systems helped shift the focus of systems development from processes first to data first.

The 1980s were marked by major breakthroughs in computing in organizations, as microcomputers became key organizational tools. The software industry expanded greatly as more and more people began to write off-the-shelf software for microcomputers. Developers began to write more and more applications in fourth-generation languages, which, unlike procedural languages, instructed a computer on what to do instead of how to do it. Computer-aided software engineering (CASE) tools were developed to make systems developers' work easier and more consistent. As computers continued to get smaller, faster, and cheaper, and as the operating systems for computers moved away from line prompt interfaces to windows- and icon-based interfaces, organizations moved to applications with more graphics. Organizations developed less software in-house and bought relatively more from software vendors. The systems developer's job went through a transition from builder to integrator.

The systems development environment of the late 1990s focused on systems integration. Developers used visual programming environments to design the user interfaces for systems that run on client/server platforms. The database, which may be relational or object-oriented, resided on the server. In many cases, the application logic resided on the same server. Alternatively, an organization may have decided to purchase its entire enterprise-wide system. Enterprise-wide systems are large, complex systems that consist of a series of independent system modules. Developers assemble systems by choosing and implementing specific modules. Starting in the middle years of the 1990s, more and more systems development efforts focused on the Internet, especially the web.

Today there is continued focus on developing systems for the Internet and for firms' intranets and extranets. As happened with traditional systems, Internet developers now rely on computer-based tools to speed and simplify the development of web-based systems. Many CASE tools directly support web application development. More and more, systems implementation involves a three-tier design, with the database on one server, the application on a second server, and client logic located on user machines. Another important development is the move to wireless system components. Wireless devices can access web-based applications from almost anywhere. Finally, the trend continues toward assembling systems from programs and components purchased off the shelf. In many cases, organizations

do not develop the application in-house. They don't even run the application in-house, choosing instead to use the application on a per-use basis by accessing it through the cloud.

Developing Information Systems and the Systems Development Life Cycle

Most organizations use a standard set of steps, called a **systems development methodology** to develop and support their information systems. It is a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems.

The **systems development life cycle (SDLC)** is a common methodology for systems development in many organizations. SDLC is the traditional methodology used to develop, maintain, and replace information systems. The life cycle can be thought of as a circular process in which the end of the useful life of one system leads to the beginning of another project that will develop a new version or replace an existing system altogether. At first glance, the life cycle appears to be a sequentially ordered set of phases, but it is not. The specific steps and their sequence are meant to be adapted as required for a project, consistent with management approaches. For example, the project can return to an earlier phase if necessary, it is also possible to complete some activities in one phase in parallel with some activities of another phase, the life cycle is iterative, and the life cycle is spiral. However conceived, the systems development life cycle used in an organization is an orderly set of activities conducted and planned for each development project. Software is the most obvious end product of the life cycle. Other essential outputs include documentation about the system and how it was developed, as well as training for users.

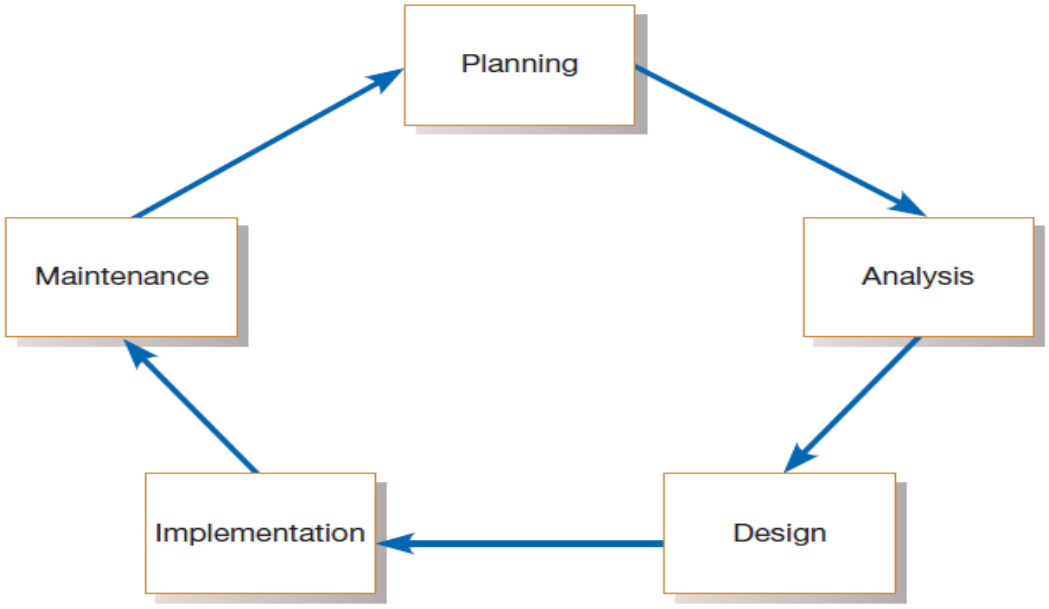


Fig: Systems development life cycle

Each phase of SDLC has specific outcomes and deliverables that feed important information to other phases. At the end of each phase, a systems development project reaches a milestone and, as deliverables are produced, they are often reviewed by parties outside the project team.

Phase	Products, Outputs, or Deliverables
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and information systems management are the result of associated systems Detailed steps, or work plan, for project

	Specification of system scope and planning and high-level system requirements or features Assignment of team members and other resources System justification or business case
Analysis	Description of current system and where problems or opportunities exist, with a general recommendation on how to fix, enhance, or replace current system Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs) Technical, detailed specifications of all system elements (programs, files, network, system software, etc.) Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

Table: Products of SDLC phases

The first phase is called **planning**. In this phase, someone identifies the need for a new or enhanced system. These needs are then analyzed, prioritized and arranged into a plan for the IS department, including schedule for developing new major systems. Here, a potential information systems project is explained and an argument for continuing or not continuing with the project is presented; a detailed plan is also developed for conducting the remaining phases or the SDLC for the proposed system.

The next phase is called **analysis**. During this phase, the analyst studies the current system and proposes alternative replacement systems. Here, the analyst thoroughly studies the organization's current procedures and the information systems used to perform organizational tasks. The analyst work with users to determine what the users want from a proposed system. The analyst carefully studies any current systems, manual and computerized, that might be replaced or enhanced as part of this project. The analyst studies the requirements and structures them according to their interrelationships and eliminates any redundancies; generates alternative initial designs to match the requirements; compare these alternatives to determine which best meets the requirements within the cost, labor, and technical levels the organization is willing to commit to the development process. The output of this phase is a description of the recommended alternative solution. Once the recommendation is accepted by owners, you can begin to make plans to acquire any hardware and system software necessary to build or operate the system as proposed.

The next phase is called **design**. During this phase, you convert the description of the recommended alternative solution into logical and then physical system specification. Here, you must design all aspects of the system form input and output screens to reports, databases, and computer processes. **Logical design** is the part of the design process that is independent of any specific hardware or software platform. Theoretically, the system could be implemented on any hardware and systems software. **Physical design** is the part of the design phase in which the logical specifications of the system form logical design are transformed into technology-specific details from which all programming and system construction can be accomplished.

The next phase is called **implementation**. In this phase, the information system is coded, tested, installed, and supported in the organization. During coding, programmers write the programs that make up the information system. During testing, programmers and analysts test

individual programs and the entire system in order to find and correct errors. During installation, the new system becomes a part of the daily activities of the organization. Implementation activities also include initial user support such as the finalization of documentation, training programs, and ongoing user assistance.

The final phase of SDLC is called **maintenance**. In this phase, information system is systematically repaired and improved. When a system is operating in an organization, users sometimes find problems with how it works and often think of better ways to perform its functions. Also the organization's needs with respect to the system change over time. In maintenance, you make the changes that users ask for and modify the system to reflect changing business conditions.

The Heart of System Development Process

Although almost all systems development projects adhere to some type of life cycle, the exact location of activities and the specific sequencing of steps can vary greatly from one project to the next.

Current practice combines the activities belonging to analysis, design, and implementation into a single iterative and parallel process. Current practice combines all of these activities into a single **analysis–design–code–test** process. These activities are the heart of systems development. This combination of activities is typical of current practices in Agile Methodologies.

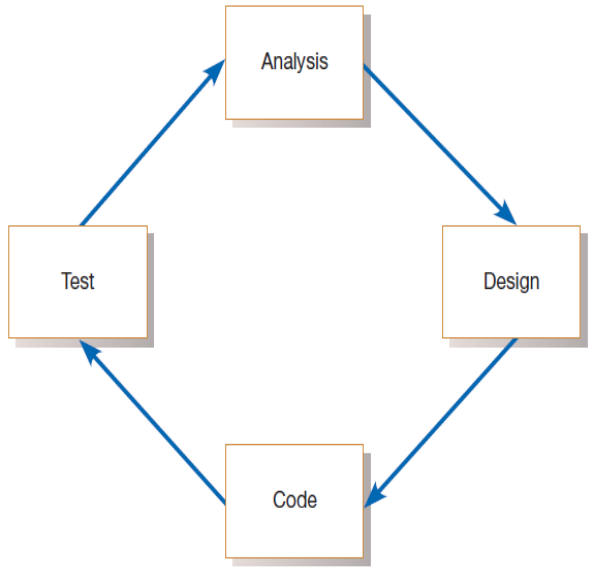


Fig: Analysis-design-code-test loop

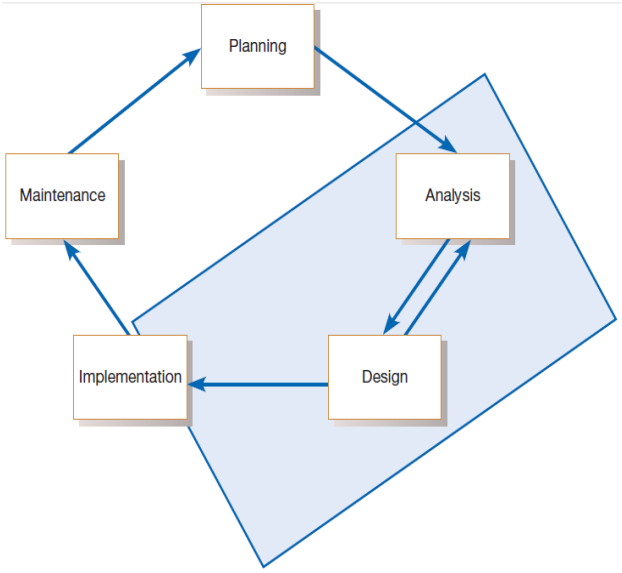


Fig: Heart of system development

Traditional Waterfall SDLC

In this approach flow of the project begins in the planning phase and from there runs "downhill" to each subsequent phase, just like a stream that runs off a cliff. Although the original developer of the waterfall model, W. W. Royce, called for feedback between phases in the waterfall, this feedback came to be ignored in implementation. This approach treat each phase as complete unto itself, never to be revisited once finished.

Waterfall model is the oldest approach for information systems development. While it does have weaknesses, it is significantly better than a haphazard approach. This model is suitable for the projects in which user requirements are certain and precise. The problems that are sometimes encountered with this approach are:

- System requirements "locked in" after being determined and hence changes in requirements can cause confusion as the project team proceeds.
- This approach makes it difficult to respond to changing customer requirements.
- A working version of the system will be available to customers late in the project. A major blunder, if undetected until the working program is reviewed, can be disastrous.
- The linear nature of this approach leads to "blocking states" in which some project team members must wait for other members of the team to complete dependent tasks.
- User involvement is limited. The role of system users or customers was narrowly defined.

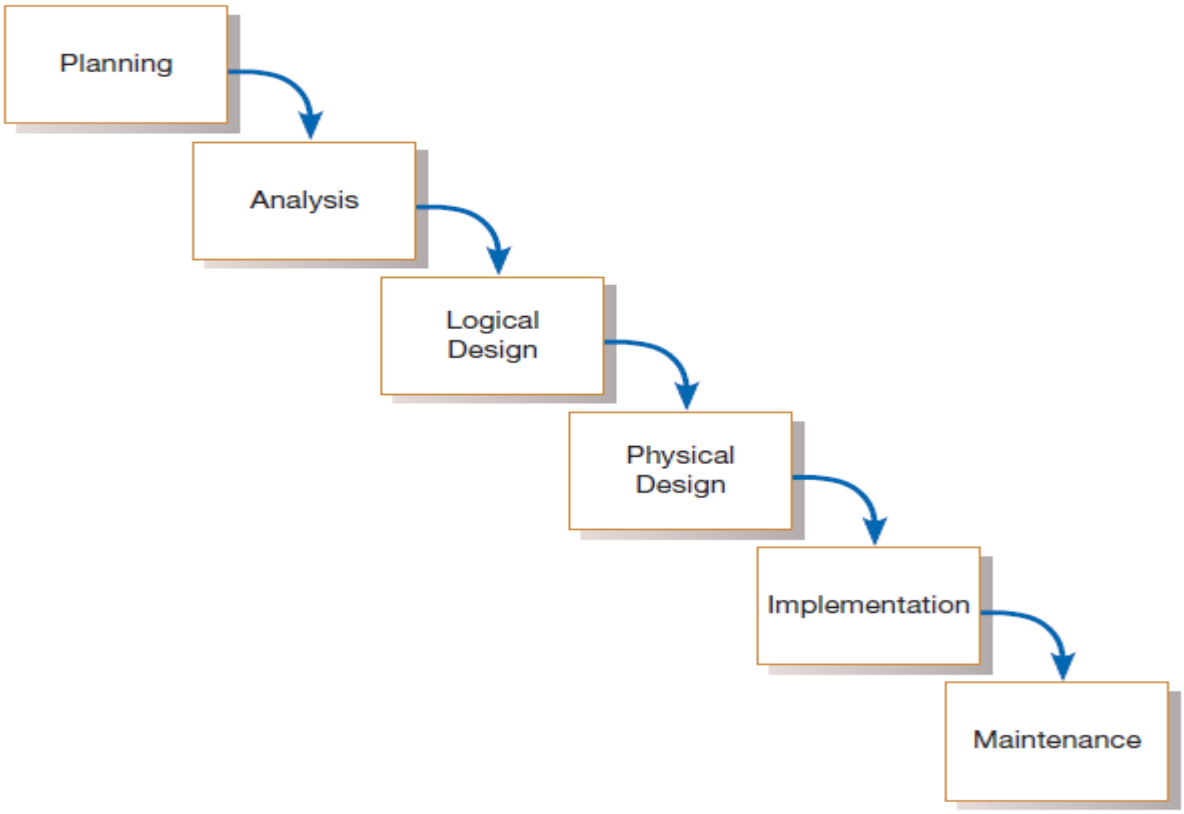


Fig: Traditional waterfall SDLC

CASE Tools

Computer-aided systems engineering (CASE) tools are the software programs that help the development team do their jobs more efficiently and more effectively. These tools support the drawing and analysis of system models. Some CASE tools also provide prototyping and code generation capabilities. The best known CASE tools are the series of Rational tools made by IBM.

At the center of any CASE tool's architecture is a developer's database called a *CASE repository*. **CASE repository** is a system developer's database where developers can store system models, detailed description and specification, and other products of system development.

Around the CASE repository is a collection of tools or facilities for creating system models and documentation. These facilities generally include:

- **Diagramming tools** enable system process, data, and control structures to be represented graphically.

- **Computer display and report generators** help prototype how systems "look and feel." Display (or form) and report generators make it easier for the systems analyst to identify data requirements and relationships.
- **Analysis tools** automatically check for incomplete, inconsistent, or incorrect specifications in diagrams, forms, and reports.
- A **central repository** enables the integrated storage of specifications, diagrams, reports, and project management information.
- **Documentation generators** produce technical and user documentation in standard formats.
- **Code generators** enable the automatic generation of program and database definition code directly from the design documents, diagrams, forms, and reports.

Today's CASE tools provide two distinct ways to develop system models – *forward engineering* and *reverse engineering*. **Forward engineering** requires the system analyst to draw system models, either from scratch or from templates. The resulting models are subsequently transformed into program code. **Reverse engineering**, on the other hand, allows a CASE tool to read existing program code and transform that code into a representative system model that can be edited and refined by the systems analyst. CASE tools that allow for bi-directional, forward and reverse engineering are said to provide for **"round-trip engineering"**.

Other Approaches

Prototyping Model

Prototyping is a form of **rapid application development (RAD)**. Prototyping is a *rapid*, *iterative*, and *incremental* process of systems development in which requirements are converted to a working system that is continually revised through close work between the development team and the users. We can build a prototype with any computer language or development tool, but special prototyping tools have been developed to simply the process. A prototype can be developed with some fourth-generation language (4GL), with the query and screen and report design tools of a database management system, and with tools called *computer-aided software engineering (CASE)* tools.

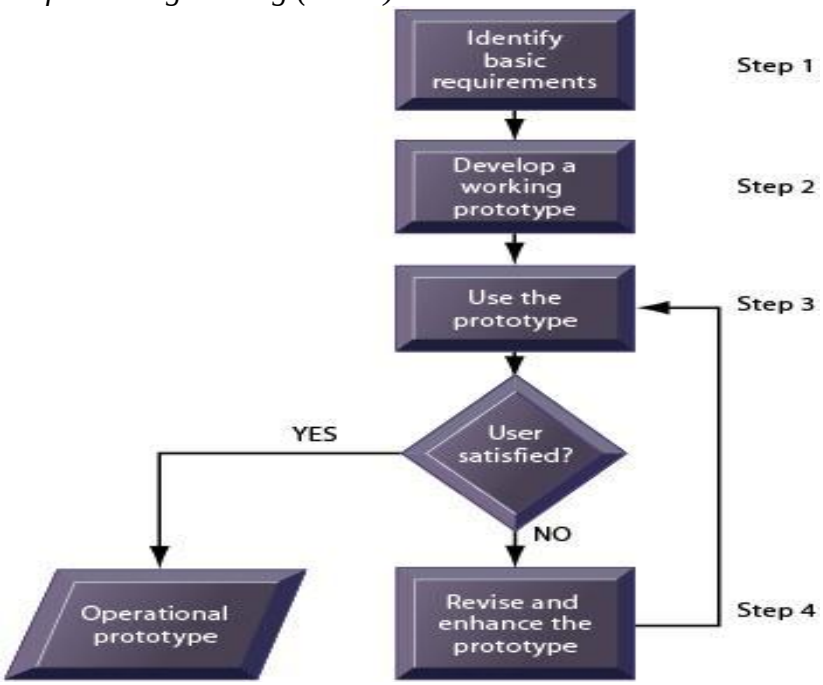


Fig: Prototyping Model

In prototyping, the analyst works with users to determine the initial or basic requirements for the system. The analyst then quickly builds a prototype. When the prototype is completed, the users work with it and tell the analyst what they like and do not like about it. The analyst uses this feedback to improve the prototype and takes the new version back to the users. This iterative process continues until the users are relatively satisfied with what they have seen.

Advantages:

- Useful for projects in which user requirements are uncertain or imprecise.
- It encourages active user and management participation.
- Projects have higher visibility and support because of the extensive user involvement.
- Users and management see working, software based solutions more rapidly.
- Errors and omissions tend to be detected earlier in prototypes.
- Testing and training are natural by-products.
- It is more natural process.
- It is most popular for small to medium-size projects.

Disadvantages:

- It increases lifetime cost to operate, support and maintain the system.
- It can solve the wrong problems since problem analysis is abbreviated or ignored.
- The product may have less quality because of speed in development.

Ideally, the prototype serves as a mechanism for identifying information system requirements. In this case, we throw away the prototype (also called **throwaway prototype**) after identifying requirements. The actual information system is developed with an eye towards quality and maintainability based on the requirements.

Spiral Model

Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model.

Spiral model is represented as a spiral with each round of the spiral incorporating different phases of software development. As you move from one round of the spiral to another, you repeat all of the phases of software development. So, it couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model.

The Radius of the spiral at any point represents the expenses (cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

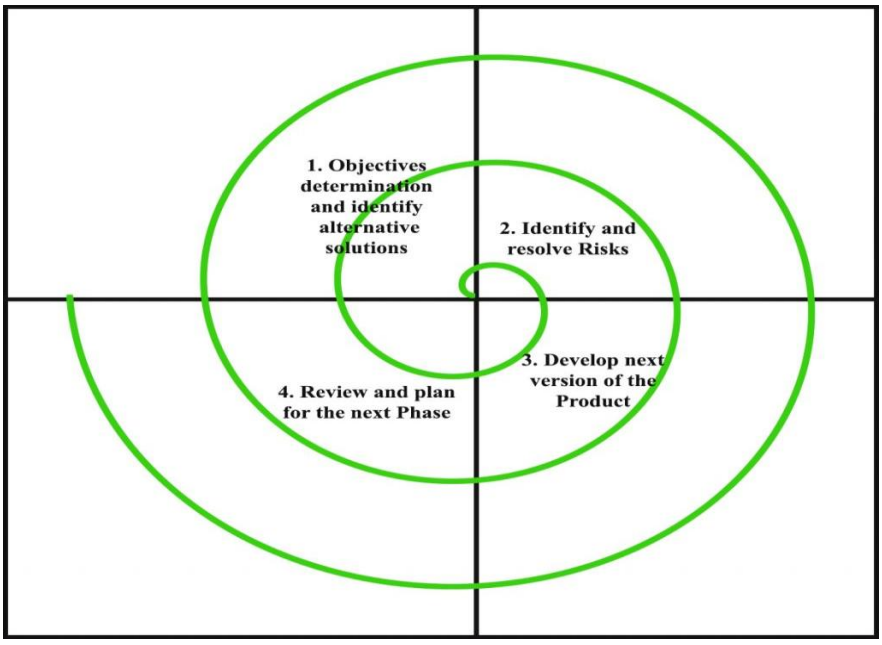


Fig: Spiral Model

Each phase of Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants are discussed below.

1. **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
2. **Identify and resolve Risks:** During the second quadrant all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution.

3. **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
4. **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. At the end, planning for the next phase is started.

Advantages:

- **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
- **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
- **Flexibility in Requirements:** Change requests in the Requirements at later phase can be incorporated accurately by using this model.
- **Customer Satisfaction:** Customer can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

Disadvantages:

- **Complex:** The Spiral Model is much more complex than other SDLC models.
- **Expensive:** Spiral Model is not suitable for small projects as it is expensive.
- **Too much dependable on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced expertise, it is going to be a failure to develop a project using this model.
- **Difficulty in time management:** As the number of phases is unknown at the start of the project, so time estimation is very difficult.

Rapid Application Development

Rapid Application Development (RAD) is a systems development methodology created to radically decrease the time needed to design and implement information systems.

RAD-based methodologies attempt to address both weaknesses of waterfall design methodologies by adjusting the SDLC phases to get some part of the system developed quickly and into the hands of the users. In this way, the users can better understand the system and suggest revisions that bring the system closer to what is needed.

RAD-based methodologies recommend that analysts use special techniques and computer tools to speed up the analysis, design, and implementation phases, such as computer-aided software engineering (CASE) tools, joint application design (JAD) sessions, fourth-generation or visual programming languages that simplify and speed up programming, and code generators that automatically produce programs from design specifications. The combination of the changed SDLC phases and the use of these tools and techniques improves the speed and quality of systems development.

However, there is one possible subtle problem with RAD-based methodologies: managing user expectations. In these methodologies, User expectations change dramatically. As a user better understands the information technology (IT), the systems requirements tend to expand. This problem is less when using methodologies that spent a lot of time thoroughly documenting requirements.

Agile Development

Agile is the ability to create and respond to change. Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the "**Manifesto for Agile Software Development**". All agile development methodologies are based on the **agile manifesto** and a set of twelve principles behind it. The

emphasis of the manifesto is to focus the developers on the working conditions of the developers, the working software, the customers, and addressing changing requirements instead of focusing on detailed systems development processes, tools, all-inclusive documentation, legal contracts, and detailed plans. These programming-centric methodologies have few rules and practices, all of which are fairly easy to follow. These methodologies are typically based only on the twelve principles of agile software. These principles include the following:

1. Software is delivered early and continuously through the development process, satisfying the customer.
2. Changing requirements are embraced regardless of when they occur in the development process.
3. Working software is delivered frequently to the customer.
4. Customers and developers work together to solve the business problem.
5. Motivated individuals create solutions; provide them the tools and environment they need, and trust them to deliver.
6. Face-to-face communication within the development team is the most efficient and effective method of gathering requirements.
7. The primary measure of progress is working, executing software.
8. Both customers and developers should work at a pace that is sustainable. That is, the level of work could be maintained indefinitely without any worker burnout.
9. Agility is heightened through attention to both technical excellence and good design.
10. Simplicity, the avoidance of unnecessary work, is essential.
11. Self-organizing teams develop the best architectures, requirements, and designs.
12. Development teams regularly reflect on how to improve their development processes.

Based on these principles, agile methodologies focus on streamlining the system-development process by eliminating much of the modeling and documentation overhead and the time spent on those tasks. Instead, projects emphasize simple, iterative application development. All agile development methodologies follow a simple cycle through the traditional phases of the systems development process.

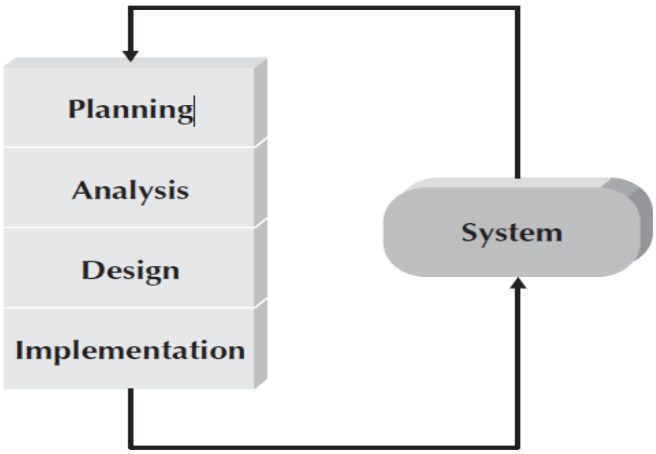


Fig: Typical Agile Development Methodology

Agile Methodologies are not for every project. It is recommended to use an agile or adaptive process if your project involves

- unpredictable or dynamic requirements,
- responsible and motivated developers, and
- customers who understand the process and will get involved.

Two of the more popular examples of agile development methodologies are **extreme programming (XP)** and **Scrum**.

Managing the Information Systems Project

Introduction

Project management (PM) may be the most important aspect of systems development. Throughout the SDLC, the project manager is responsible for **initiating**, **planning**, **executing**, and **closing** down the systems development project. Project management is arguably the most important aspect of an information systems development project. Effective project management helps to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.

The demand for project managers in the information systems community is strong. Typically, information system project managers come from the ranks of experienced information system developers such as systems analysts. In some cases, the organizations tend to hire and/or develop professional project managers.

Managing the Information Systems Project

Project management is an important aspect of the development of information systems and a critical skill for a systems analyst. The focus of project management is to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.

The **project manager** is normally a well experienced systems analyst with a diverse set of skills – management, leadership, technical, conflict management, and customer relationship – who is responsible for initiating, planning, executing, and closing down a project. As a project manager, your environment is one of continual change and problem solving. A project manager is often thought of as a juggler keeping aloft many balls, which reflect the various aspects of a project's development.

Fig: A project manager juggles numerous activities

Creating and implementing successful projects requires managing the resources, activities, and tasks needed to complete the information systems project. A **project** is a planned

undertaking of a series of related activities to reach an objective that has a beginning and an end.

In most cases, systems development projects are undertaken for two primary reasons: to take advantage of business opportunities and to solve business problems. Taking advantage of an opportunity might mean providing an innovative service to customers through the creation of a new system. Solving a business problem could involve modifying the way an existing system processes data so that more accurate or timely information is provided to users.

Once a potential project has been identified, an organization must determine the resources required for its completion. This is done by analyzing the scope of the project and determining the probability of successful completion. The organization can then determine whether the project is feasible within time and resource constraints. If deemed feasible, a more detailed project analysis is then conducted.

To successfully orchestrate the construction of a complex information system, a project manager must have interpersonal, leadership, and technical skills.

Activity	**Description**	**Skill**
Leadership	Influencing the activities of others toward the attainment of a common goal through the use of intelligence, personality, and abilities	Communication; liaison between management, users, and developers; assigning activities; monitoring progress
Management	Getting projects completed through the effective utilization of resources	Defining and sequencing activities; communicating expectations; assigning resources to activities; monitoring outcomes
Customer relations	Working closely with customers to ensure that project deliverables meet expectations	Interpreting system requests and specifications; site preparation and user training; contact point for customers
Technical problem solving	Designing and sequencing activities to attain project goals	Interpreting system requests and specifications; defining activities and their sequence; making trade-offs between alternative solutions; designing solutions to problems
Conflict management	Managing conflict within a project team to assure that conflict is not too high or too low	Problem solving; smoothing out personality differences; compromising; goal setting
Team management	Managing the project team for effective team performance	Communication within and between teams; peer evaluations; conflict resolution; team building; self-management
Risk and change management	Identifying, assessing, and managing the risks and day-to-day changes that occur during a project	Environmental scanning; risk and opportunity identification and assessment; forecasting; resource redeployment

Table: Common activities and Skills of a Project Manager

Project management is a controlled process of initiating, planning, executing, and closing down a project. So, project management process involves four phases: **initiating the project**, **planning the project**, **executing the project** and **closing down the project**.

1. **Initiating a Project:** During **project initiation**, the project manager performs several activities to assess the size, scope, and complexity of the project and to establish procedures to support subsequent activities. The types of activities you will perform when initiating a project are:
    - *Establishing the project initiation team*: This activity involves organizing an initial core of project team members to assist in accomplishing the project initiation activities.
    - *Establishing a relationship with the customer*: A thorough understanding of your customer builds stronger partnerships and higher levels of trust.
    - *Establishing the project initiation plan*: This step defines the activities required to organize the initiation team while it is working to define the goals and scope of the project.
    - *Establishing management procedures*: Successful projects require the development of effective management procedures. When establishing procedures, you are concerned with developing team communication and reporting procedures, job assignments and roles, project change procedures, and determining how project funding and billing will be handled.
    - *Establishing the project management environment and project workbook*: The focus of this activity is to collect and organize the tools that you will use while managing the project and to construct the project workbook. Project workbook serves as a repository for all project correspondence, inputs, outputs, deliverables, procedures, and standards established by the project team. The **project workbook** can be stored as an online electronic document or in a large three-ring binder. The project workbook is used by all team members and is useful for project audits, orientation of new team members, communication with management and customers, identifying future projects, and performing post-project reviews.

      Keeping the project workbook online has many advantages for keeping the project team on track and efficient. Online documents can be easily accessed by all team members. Additionally, everyone is always working with the most up-to-date information. The best feature of using the web as your repository is that it enables project members and customers to review a project's status and all related information continually.
    - *Developing the project charter*: The **project charter** is a short (typically one page), high-level document prepared for the customer that describes what the project will deliver and outlines many of the key elements of the project. A project charter can vary in the amount of detail it contains, but it often includes the following elements:
        - Project title and date of authorization
        - Project manager name and contact information
        - Customer name and contact information
        - Projected start and completion dates
        - Key stakeholders, project role, and responsibilities
        - Project objectives and description
        - Key assumptions or approach
        - Signature section for key stakeholders

      The project charter ensures that both you and your customer gain a common understanding of the project. It is also a very useful communication tool; it helps to announce to the organization that a particular project has been chosen for development.

Before moving on to the next phase of the project, the work performed during project initiation is reviewed at a meeting attended by management, customers, and project team members. An outcome of this meeting is a decision to continue, modify, or abandon the project.

If the scope of the project is modified, it may be necessary to return to project initiation activities and collect additional information. Once a decision is made to continue the project, a much more detailed project plan is developed during the project planning phase.

2. **Planning the Project:** The next step in the project management process is **project planning**. Project planning involves defining clear, discrete activities and the work needed to complete each activity within a single project. It is much easier to plan nearer-term activities than those occurring in the future. In actual fact, you often have to construct longer-term plans that are more general in scope and nearer-term plans that are more detailed. The repetitive nature of the project management process requires that plans be constantly monitored throughout the project and periodically updated (usually after each phase), based upon the most recent information.

   As with the project initiation process, varied and numerous activities must be performed during project planning. The types of activities that you can perform during project planning are:

   - *Describing project scope, alternatives, and feasibility*: The purpose of this activity is to understand the content and complexity of the project. During this activity, you should reach agreement on the following questions:
     - What problem or opportunity does the project address?
     - What are the quantifiable results to be achieved?
     - What needs to be done?
     - How will success be measured?
     - How will we know when we are finished?

     After defining the scope of the project, your next objective is to identify and document general alternative solutions for the current business problem or opportunity. You must then assess the feasibility of each alternative solution and choose which to consider during subsequent SDLC phases.
   - *Dividing the project into manageable tasks*: This is a critical activity during the project planning process. Here, you must divide the entire project into manageable tasks and then logically order them to ensure a smooth evolution between tasks. A **task** can be done by one person or a well-defined group, has a single and identifiable deliverable (the task is, however, the process of creating the deliverable), has a known method or technique, has well-accepted predecessor and successor steps, and is measurable so that the percentage completed can be determined.

     The definition of tasks and their sequence is referred to as the **work breakdown structure**. Some tasks may be performed in parallel, whereas others must follow one another sequentially. Task sequence depends on which tasks produce deliverables needed in other tasks, when critical resources are available, the constraints placed on the project by the client, and the process outlined in the SDLC. A work breakdown for these activities is represented in a Gantt chart. A **Gantt chart** is a graphical representation of a project that shows each task as a horizontal bar whose length is proportional to its time for completion. Different colors, shades, or shapes can be used to highlight each kind of task.

- *Estimating resources and creating a resource plan*: The goal of this activity is to estimate resource requirements for each project activity and to use this information to create a project resource plan. The resource plan helps assemble and deploy resources in the most effective manner. Project managers use a variety of tools to assist in making estimates of project size and costs. The most widely used method is called COCOMO (**COnstructive COst MOdel**), which uses parameters that were derived from prior projects of differing complexity. The Constructive Cost Model (COCOMO) is an automated software estimation model that uses historical project data and current as well as future project characteristics to estimate project costs.
  People are the most important, and expensive, part of project resource planning. Project time estimates for task completion and overall system quality are significantly influenced by the assignment of people to tasks. Resource estimates may need to be revised based upon the skills of the actual person (or people) assigned to a particular activity.
- *Developing a preliminary schedule*: During this activity, you use the information on tasks and resource availability to assign time estimates to each activity in the work breakdown structure. These time estimates will enable you to create target starting and ending dates for the project. Target dates can be revisited and modified until a schedule is produced that is acceptable to the customer. The schedule may be represented as a Gantt chart or as a network diagram. A **network diagram** is a graphical depiction of project tasks and their interrelationships. As with a Gantt chart, each type of task can be highlighted by different features on the network diagram. The distinguishing feature of a network diagram is that the ordering of tasks is shown by connecting tasks – depicted as rectangles or ovals – with their predecessor and successor tasks.
- *Developing a communication plan*: The goal of this activity is to outline the communication procedures among management, project team members, and the customer. The communication plan includes when and how written and oral reports will be provided by the team, how team members will coordinate work, what messages will be sent to announce the project to interested parties, and what kinds of information will be shared with vendors and external contractors involved with the project. It is important that free and open communication occur among all parties with respect to proprietary information and confidentiality with the customer.
  In this plan, a summary of communication documents, work assignments, schedules, and distribution methods will be outlined. Additionally, a project communication matrix can be developed that provides a summary of the overall communication plan.
- *Determining project standards and procedures*: During this activity, you will specify how various deliverables are produced and tested by you and your project team. For example, the team must decide which tools to use, how the standard SDLC might be modified, which SDLC methods will be used, documentation styles, how team members will report the status of their assigned activities, and terminology.
- *Identifying and assessing risk:* The goal of this activity is to identify sources of project risk and estimate the consequences of those risks. Risks might arise from the use of new technology, prospective users' resistance to change, availability of critical resources, competitive reactions or changes in regulatory

actions due to the construction of a system, or team member inexperience with technology or the business area. You should continually try to identify and assess project risk.

- *Creating a preliminary budget*: During this phase, you need to create a preliminary budget that outlines the planned expenses and revenues associated with your project. The project justification will demonstrate that the benefits are worth these costs.
- *Developing a Project Scope Statement*: An important activity that occurs near the end of the project planning phase is the development of the Project Scope Statement. Developed primarily for the customer, this document outlines work that will be done and clearly describes what the project will deliver. The Project Scope Statement is useful to make sure that you, the customer, and other project team members have a clear understanding of the intended project size, duration, and outcomes.
- *Setting a Baseline Project Plan*: Once all of the prior project planning activities have been completed, you will be able to develop a Baseline Project Plan. This baseline plan provides an estimate of the project's tasks and resource requirements and is used to guide the next project phase – execution. As new information is acquired during project execution, the baseline plan will continue to be updated. At the end of the project planning phase, a review of the Baseline Project Plan is conducted to double – check all information in the plan. As with the project initiation phase, it may be necessary to modify the plan, which means returning to prior project planning activities before proceeding.

3. **Executing the Project: Project execution** puts the Baseline Project Plan into action. Within the context of the SDLC, project execution occurs primarily during the analysis, design, and implementation phases. The five key activities during project execution are:
    - *Executing the Baseline Project Plan*: As project manager, you oversee the execution of the baseline plan. This means that you initiate the execution of project activities, acquire and assign resources, orient and train new team members, keep the project on schedule, and ensure the quality of project deliverables. You may want to plan social events, regular team project status meetings, team-level reviews of project deliverables, and other group events to mold the group into an effective team.
    - *Monitoring project progress against the Baseline Project Plan*: While you execute the Baseline Project Plan, you should monitor the progress. If the project gets ahead of (or behind) schedule, you may have to adjust resources, activities, and budgets. Monitoring project activities can result in modifications to the current plan.

      It is possible, with project schedule charts such as Gantt charts, to show progress against a plan, and it is easy with network diagrams to understand the ramifications of delays in an activity. Monitoring progress also means that the team leader must evaluate and appraise each team member, occasionally change work assignments or request changes in personnel, and provide feedback to the employee's supervisor.
    - *Managing changes to the Baseline Project Plan*: You will encounter pressure to make changes to the baseline plan. Numerous events may initiate a change to the Baseline Project Plan, including the following possibilities:
        - A slipped completion date for an activity

- A bungled activity that must be redone
- The identification of a new activity that becomes evident later in the project
- An unforeseen change in personnel due to sickness, resignation, or termination

When an event occurs that delays the completion of an activity, you typically have two choices: devise a way to get back on schedule or revise the plan. Devising a way to get back on schedule is the preferred approach because no changes to the plan will have to be made.

- *Maintaining the project workbook*: As in all project phases, maintaining complete records of all project events is necessary. The workbook provides the documentation new team members require to assimilate project tasks quickly. It explains why design decisions were made and is a primary source of information for producing all project reports.
- *Communicating the project status*: The project manager is responsible for keeping all stakeholders (system developers, managers, and customers) abreast of the project status. A broad variety of methods can be used to distribute information, each with strengths and weaknesses. Two types of information are routinely exchanged throughout the project: *work results* (the outcomes of the various tasks and activities that are performed to complete the project) and the *project plan* (the formal comprehensive document that is used to execute the project).

4. **Closing down the Project:** The focus of **project closedown** is to bring the project to an end. Projects can conclude with a natural or unnatural termination. A natural termination occurs when the requirements of the project have been met – the project has been completed and is a success. An unnatural termination occurs when the project is stopped before completion. Within the context of the SDLC, project closedown occurs after the implementation phase. The different activities are:
    - *Closing down the project*: During closedown, you perform several diverse activities. For example, if you have several team members working with you, project completion may signify job and assignment changes for some members. You will likely be required to assess each team member and provide an appraisal and salary determination. You may also want to provide career advice to team members, write letters to superiors praising special accomplishments of team members, and send thank-you letters to those who helped but were not team members. When closing down the project, it is also important to notify all interested parties that the project has been completed and to finalize all project documentation and financial records so that a final review of the project can be conducted. You should also celebrate the accomplishments of the team. Some teams will hold a party, and each team member may receive memorabilia like T-shirt.
    - *Conducting postproject reviews*: Once you have closed down the project, final reviews of the project should be conducted with management and customers. The objective of these reviews is to determine the strengths and weaknesses of project deliverables, the processes used to create them, and the project management process.
    - *Closing the customer contract*: The focus of this final activity is to ensure that all contractual terms of the project have been met. A project governed by a contractual agreement is typically not completed until agreed to by both parties, often in writing.

Representing and Scheduling Project Plans

A project manager has a wide variety of techniques available for depicting and documenting project plans. These planning documents can take the form of graphical or textual reports, although graphical reports have become most popular for depicting project plans.

The most commonly used methods are **Gantt charts** and **network diagrams**. Because **Gantt charts** do not (typically) show how tasks must be ordered (precedence) but simply show when a task should begin and when it should end, they are often more useful for depicting relatively simple projects or subparts of a larger project, showing the activities of a single worker, or monitoring the progress of activities compared to scheduled completion dates. A **network diagram** shows the ordering of activities by connecting a task to its predecessor and successor tasks. Some key differences between these two charts are:

- Gantt charts visually show the duration of tasks, whereas a network diagram visually shows the sequence dependencies between tasks.
- Gantt charts visually show the time overlap of tasks, whereas a network diagram does not show time overlap but does show which tasks could be done in parallel.
- Some forms of Gantt charts can visually show slack time available within an earliest start and latest finish duration. A network diagram shows this by data within activity rectangles.

Project managers also use textual reports that depict resource utilization by task, complexity of the project, and cost distributions to control activities. Most project managers use computer-based systems to help develop their graphical and textual reports.

A major activity of project planning focuses on dividing the project into manageable activities, estimating times for each, and sequencing their order.

Representing Project Plans:

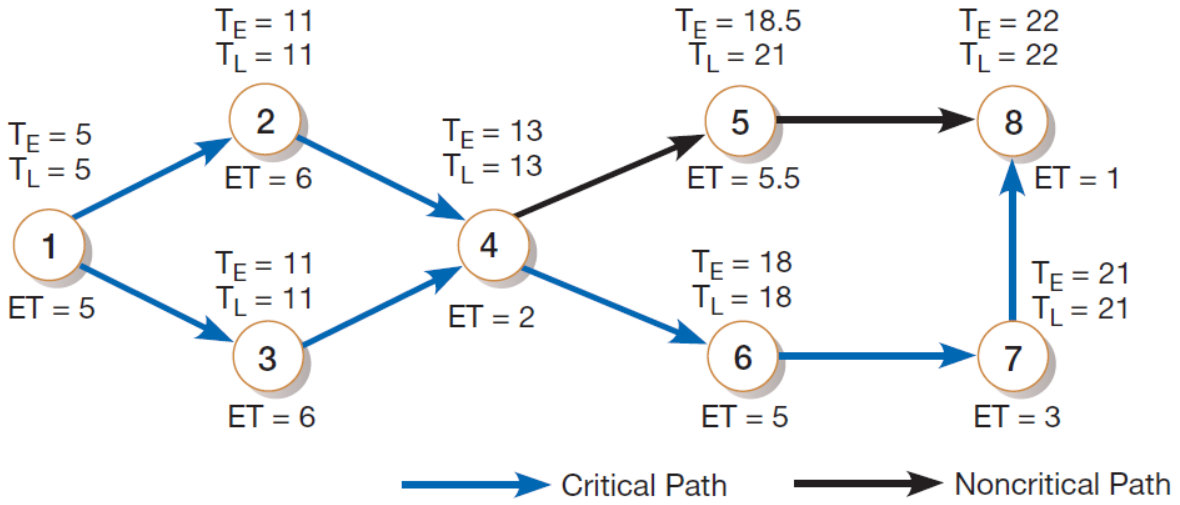
Project scheduling and management require that time, costs, and resources be controlled. **Resources** are any person, group of people, piece of equipment, or material used in accomplishing an activity. Network diagramming is a **critical path scheduling** technique used for controlling resources. A **critical path** refers to a sequence of task activities whose order and durations directly affect the completion date of a project. Nodes not on the critical path contain **slack time** and allow the project manager some flexibility in scheduling. The amount of time that an activity can be delayed without delaying the project is called slack time.

A network diagram is one of the most widely used and best-known scheduling methods. You would use a network diagram when tasks

- are well defined and have a clear beginning and end point,
- can be worked on independently of other tasks,
- are ordered, and
- serve the purpose of the project

A major strength of network diagramming is its ability to represent how completion times vary for activities. Because of this, it is more often used than Gantt charts to manage projects such as information systems development, where variability in the duration of activities is the norm. Network diagrams are composed of circles or rectangles representing activities and connecting arrows showing required work flows.

The figure below shows a network diagram showing estimated times (ET) for each activity, and the earliest expected completion time (T_E) and latest expected completion time (T_L) for each activity along critical and non-critical path.



Calculating expected time durations using Pert:

PERT (Program Evaluation Review Technique) is a technique that uses optimistic, pessimistic, and realistic time estimates to calculate the expected time for a particular task. This technique can help you to obtain a better time estimate when there is some uncertainty as to how much time a task will require to be completed. The optimistic *(o)* and pessimistic *(p)* times reflect the minimum and maximum possible periods of time for an activity to be completed. The realistic *(r)* time, or most likely time, reflects the project manager's "best guess" of the amount of time the activity actually require for completion. Once each of these estimates is made for an activity, an expected time *(ET)* can be calculated. Because the expected completion time should be closest to the realistic *(r)* time, it is typically weighted four times more than the optimistic *(o)* and pessimistic *(p)* times. Once you add these values together, it must be divided by six to determine the *ET*. This equation is shown in the following formula:

$ET = (o + 4r + p)/6$

Where,

ET = expected time for the completion for an activity
o = optimistic completion time for an activity
r = realistic completion time for an activity
p = pessimistic completion time for an activity

Using Project Management Software

A wide variety of automated project management tools is available to help you manage a development project. New versions of these tools are continuously being developed and released by software vendors. Most of the available tools have a set of common features that include the ability to define and order tasks, assign resources to tasks, and easily modify tasks and resources. Some common project management softwares are ProjectLibre, Microsoft Project, and OpenProj.

When using project management software to manage a project, you need to perform at least the following activities:

- Establish a project starting or ending date.
- Enter tasks and assign task relationships.
- Select a scheduling method to review project reports.