Unit-1 Foundations for Systems Development

The Systems Development Environment

What is System?

A system is a collection of components (subsystems) that work together to realize some objective. For example, the library system contains librarians, books, and periodicals as components to provide knowledge for its members.

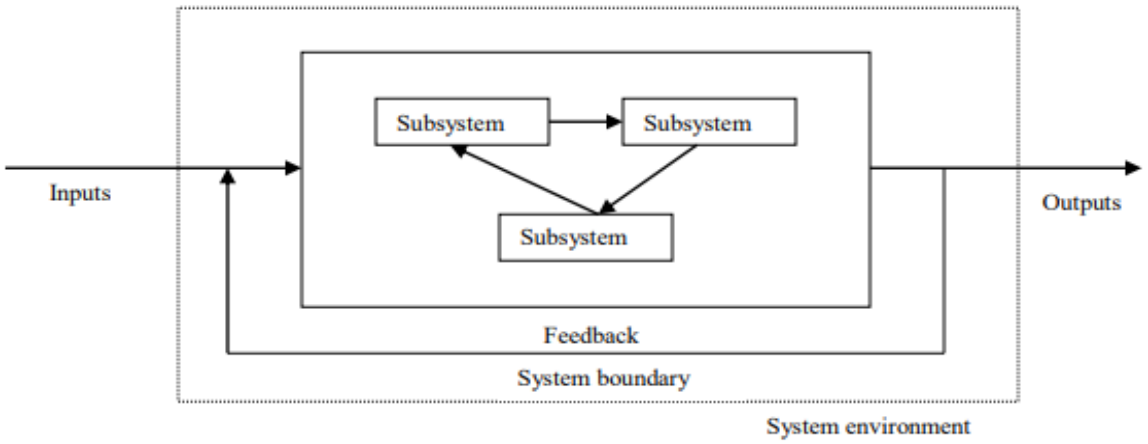


Fig: Basic System Model

Elements of the system

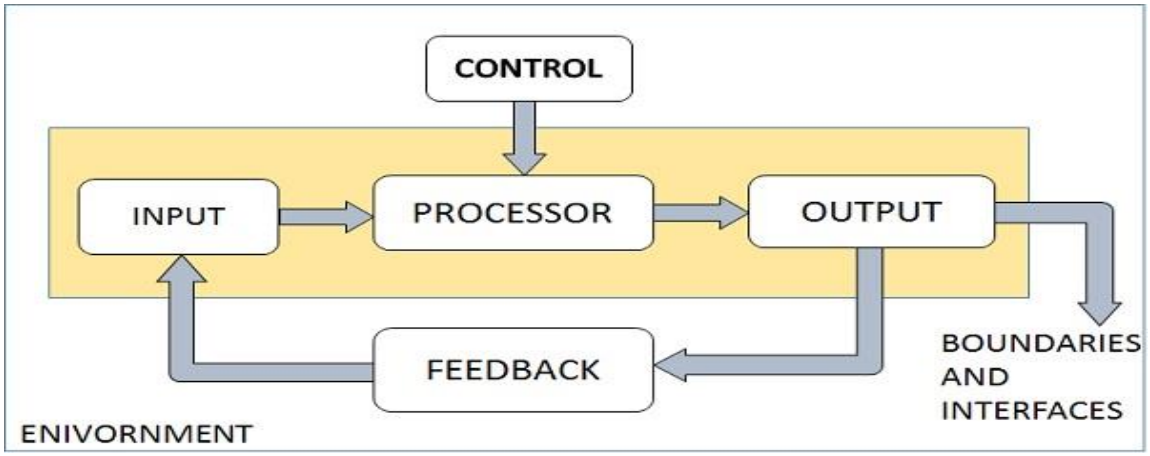


Fig: Elements of system

The above figure shows the elements of a system and elements are:

Outputs and Inputs

- The main aim of a system is to produce an output which is useful for its user.
- Inputs are the information that enters into the system for processing.
- Output is the outcome of processing.

Processor(s)

- The processor is the element of a system that involves the actual transformation of input into output.
- It is the operational component of a system. Processors may modify the input either totally or partially, depending on the output specification.
- As the output specifications change, so does the processing. In some cases, input is also modified to enable the processor for handling the transformation.

Control

- The control element guides the system.
- It is the decision–making subsystem that controls the pattern of activities governing input, processing, and output.
- The behavior of a computer System is controlled by the Operating System and software. In order to keep system in balance, what and how much input is needed is determined by Output Specifications.

Feedback

- Feedback provides the control in a dynamic system.
- Positive feedback is routine in nature that encourages the performance of the system.
- Negative feedback is informational in nature that provides the controller with information for action.

Environment

- The environment is the “super system” within which an organization operates.
- It is the source of external elements that strike on the system.
- It determines how a system must function. For example, vendors and competitors of organization’s environment may provide constraints that affect the actual performance of the business.

Boundaries and Interface

- A system should be defined by its boundaries. Boundaries are the limits that identify its components, processes, and interrelationship when it interfaces with another system.
- Each system has boundaries that determine its sphere of influence and control.
- The knowledge of the boundaries of a given system is crucial in determining the nature of its interface with other systems for successful design

The characteristics of a system are:

Organization

Organization implies structure and order. It is the arrangement of components that helps to achieve predetermined objectives.

Interaction

It is defined by the manner in which the components operate with each other. For example, in an organization, purchasing department must interact with production department and payroll with personnel department.

Interdependence

Interdependence means how the components of a system depend on one another. For proper functioning, the components are coordinated and linked together according to a specified plan. The output of one subsystem is the required by other subsystem as input.

Integration

Integration is concerned with how system components are connected together. It means that the parts of the system work together within the system even if each part performs a unique function.

Central Objective

The objective of system must be central. It may be real or stated. It is not uncommon for an organization to state an objective and operate to achieve another. The users must know the main objective of a computer application early in the analysis for a successful design and conversion.

Information System

Information system is an integrated set of components for collecting, storing, and processing data and for providing information, knowledge, and digital products. Business firms and other organizations rely on information systems to carry out and manage their operations, interact with their customers and suppliers, and compete in the marketplace. Information systems are used to run inter organizational supply chains and electronic markets. For instance, corporations use information systems to process financial accounts, to manage their human resources, and to reach their potential customers with online promotions. Many major companies are built entirely around information systems. These include eBay, a largely auction marketplace; Amazon, an expanding electronic mall and provider of cloud computing services; Alibaba, a business-to-business e-marketplace; and Google, a search engine company that derives most of its revenue from keyword advertising on Internet searches. Governments deploy information systems to provide services cost-effectively to citizens. Digital goods—such as electronic books, video products, and software—and online services, such as gaming and social networking, are delivered with information systems. Individuals rely on information systems, generally Internet-based, for conducting much of their personal lives: for socializing, study, shopping, banking, and entertainment.

Information Systems are classified by organizational levels, mode of data, processing, system objectives and type of support provided.

Following are the type of information system:

1. Transaction Processing System (TPS):

- Transaction Processing System are information system that processes data resulting from the occurrences of business transactions
- Their objectives are to provide transaction in order to update records and generate reports i.e. to perform store keeping function
- The transaction is performed in two ways: Batching processing and online transaction processing.
- **Example:** Bill system, payroll system, Stock control system.

2. Management Information System (MIS):

- Management Information System is designed to take relatively raw data available through a Transaction Processing System and convert them into a summarized and aggregated form for the manager, usually in a report format. It reports tending to be used by middle management and operational supervisors.
- Many different types of report are produced in MIS. Some of the reports are a summary report, on-demand report, ad-hoc reports and an exception report.
- **Example:** Sales management systems, Human resource management system.

3. Decision Support System (DSS):

- Decision Support System is an interactive information system that provides information, models and data manipulation tools to help in making the decision in a semi-structured and unstructured situation.
- Decision Support System comprises tools and techniques to help in gathering relevant information and analyzes the options and alternatives, the end user is more involved in creating DSS than an MIS.
- **Example:** Financial planning systems, Bank loan management systems.

4. Experts System (ES):

- Experts systems include expertise in order to aid managers in diagnosing problems or in problem-solving. These systems are based on the principles of **artificial intelligence research.**
- Experts Systems is a knowledge-based information system. It uses its knowledge about a specify are to act as an expert consultant to users.
- Knowledgebase and software modules are the components of an expert system. These modules perform inference on the knowledge and offer answers to a user's question.

5. Knowledge Work System

- There are different knowledge management systems that an organization implements to ensure a continuous flow of new and updated knowledge into the company and its processes. A knowledge work system (KWS) is one of the knowledge management systems that ease the integration of new information or knowledge into the business process.
- Furthermore, KWS also offers support and resources to various knowledge creation techniques, artificial intelligence applications, and group collaboration systems for knowledge sharing, among others. It also uses graphics, visuals, etc., to disseminate new information. Below are some of the applications that work on the core fundamentals of KWS.

- Designers often use computer-aided design systems (CAD) to automate their design process.
- Financial workstations are used to analyze huge amounts of financial data with the help of new technologies.
- Virtual reality systems are found in the scientific, education, and business fields for using graphics and different systems to present data.

6. Office Automation System

- An office automation system is an information system that automates different administrative processes like documenting, recording data, and office transactions, among others. The office automation system is divided into managerial and clerical activities. Here are some of the business activities that are done under this type of information system:

- Email
- Voice mail
- Word processing

7. Executive Support System

- An Executive Support System or ESS helps top-level executives to plan and control workflow and make business decisions. It is very similar to Management Information System or MIS.
- Here are some of the unique characteristics of ESS:

- It provides great telecommunication, better computing capabilities, and effective display options to executives.
- It enables them with information through static reports, graphs, and textual information on demand.
- It helps monitor performances, track competitors' strategies, and forecast future trends, among others.

Introduction to System Analysis and Design

System analysis and design (SAD) deal with planning the development of information systems through understanding and specifying in detail what a system should do and how the components of the system should be implemented and work together. System analysts solve business problems through analyzing the requirements of information systems and designing such systems by applying analysis and design techniques.

Systems Analysis and Design (SAD) is an active field in which analysts repetitively learn new approaches and different techniques for building the system more effectively and efficiently. The primary objective of systems analysis and design is to improve organizational systems.

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 a sequence of step-by-step approaches that help develop your final product: the information system. Most methodologies incorporate several development techniques, such as direct observations and interviews with users of the current system.

Tools are computer programs, such as computer-aided software engineering (CASE) tools, that make it easy to use specific techniques.

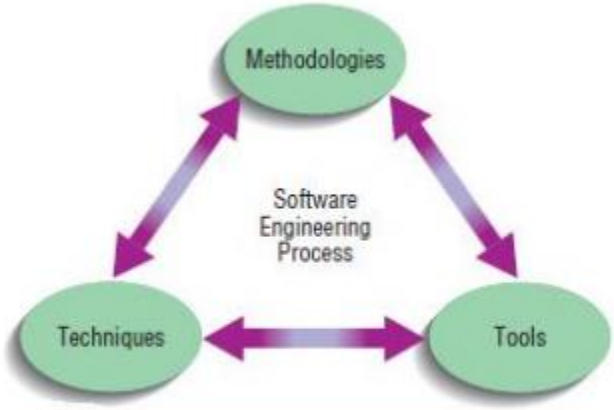


Figure: Methodologies, techniques, and tools.

Techniques are processes that you, as an analyst, will follow to help ensure that your work is well thought-out, complete, and comprehensible to others on your project team. Techniques provide support for a wide range of tasks, including conducting thorough interviews with current and future users of the information system to determine what your system should do, planning and managing the activities in a systems development project, diagramming how the system will function, and designing the reports, such as invoices, your system will generate for its users to perform their jobs. These three elements - methodologies, techniques, and tools - work together to form an organizational approach to systems analysis and design.

Systems development is systematic process which includes phases such as planning, analysis, design, deployment, and maintenance. Here, we will primarily focus on:

- Systems analysis
- Systems design

Systems Analysis

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Analysis specifies **what the system should do**.

Systems Design

It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

System Design focuses on **how to accomplish the objective of the system**

Developing Information Systems and the Systems Development Life Cycle

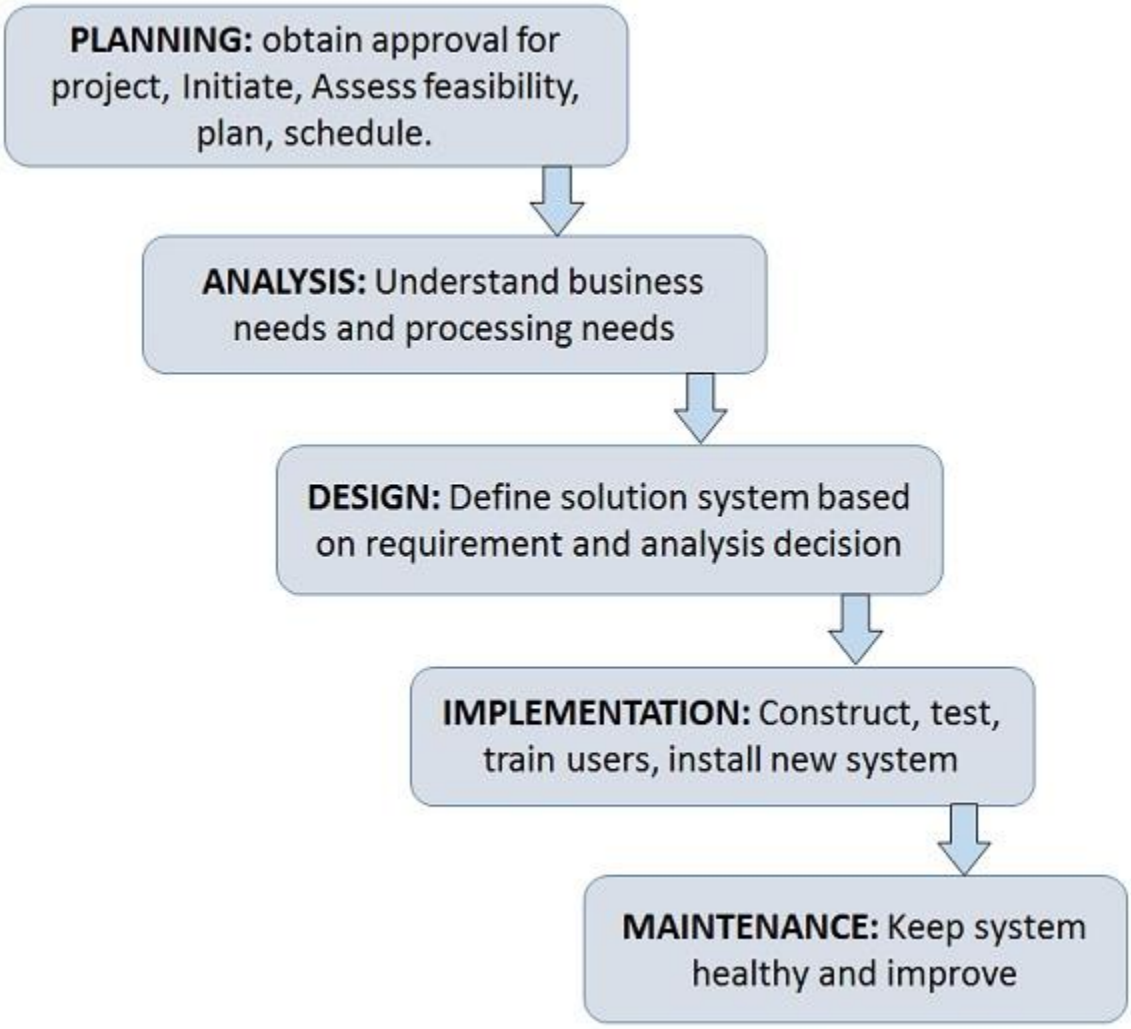
Most organizations use a standard set of steps, called a systems development methodology to develop and support their information systems. It is a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems. And **systems development life cycle (SDLC)** is the traditional methodology used to develop, maintain, and replace information systems. SDLC is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

An effective System Development Life Cycle (SDLC) should result in a high quality system that meets customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure.

Phases of SDLC

Systems Development Life Cycle is a systematic approach which explicitly breaks down the work into phases that are required to implement either new or modified Information System.

Phases of SDLC are shown by following diagram:



Feasibility Study or Planning

- Define the problem and scope of existing system.
- Overview the new system and determine its objectives.
- Confirm project feasibility and produce the project Schedule.
- During this phase, threats, constraints, integration and security of system are also considered.
- A feasibility report for the entire project is created at the end of this phase.

Analysis and Specification

- Gather, analyze, and validate the information.
- Define the requirements and prototypes for new system.
- Evaluate the alternatives and prioritize the requirements.
- Examine the information needs of end-user and enhances the system goal.
- A Software Requirement Specification (SRS) document, which specifies the software, hardware, functional, and network requirements of the system, is prepared at the end of this phase.

System Design

- Includes the design of application, network, databases, user interfaces, and system interfaces.
- Transform the SRS document into logical structure, which contains detailed and complete set of specifications that can be implemented in a programming language.
- Create a contingency, training, maintenance, and operation plan.
- Review the proposed design. Ensure that the final design must meet the requirements stated in SRS document.
- Finally, prepare a design document which will be used during next phases.

Implementation

- Implement the design into source code through coding.
- Combine all the modules together into training environment that detects errors and defects.
- A test report which contains errors is prepared through test plan that includes test related tasks such as test case generation, testing criteria, and resource allocation for testing.
- Integrate the information system into its environment and install the new system.

Maintenance/Support

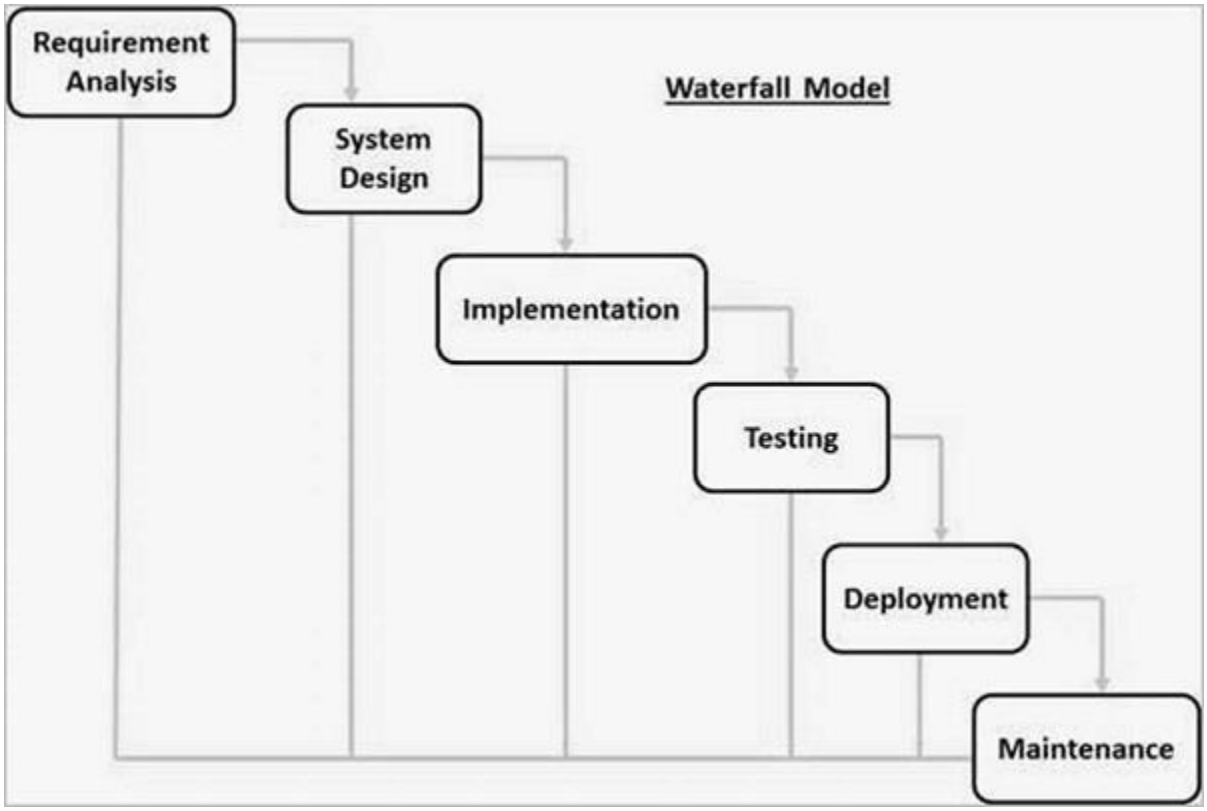
- Include all the activities such as phone support or physical on-site support for users that is required once the system is installing.
- Implement the changes that software might undergo over a period of time, or implement any new requirements after the software is deployed at the customer location.
- It also includes handling the residual errors and resolves any issues that may exist in the system even after the testing phase.
- Maintenance and support may be needed for a longer time for large systems and for a short time for smaller systems.

Traditional Waterfall SDLC

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model:



The sequential phases in Waterfall model are:

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Each phase of development proceeds in strict order.

Some of the major **advantages** of the Waterfall Model are as follows:

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.

- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

The major disadvantages of the Waterfall Model are as follows:

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early

Computer-aided Software Engineering (CASE) Tools

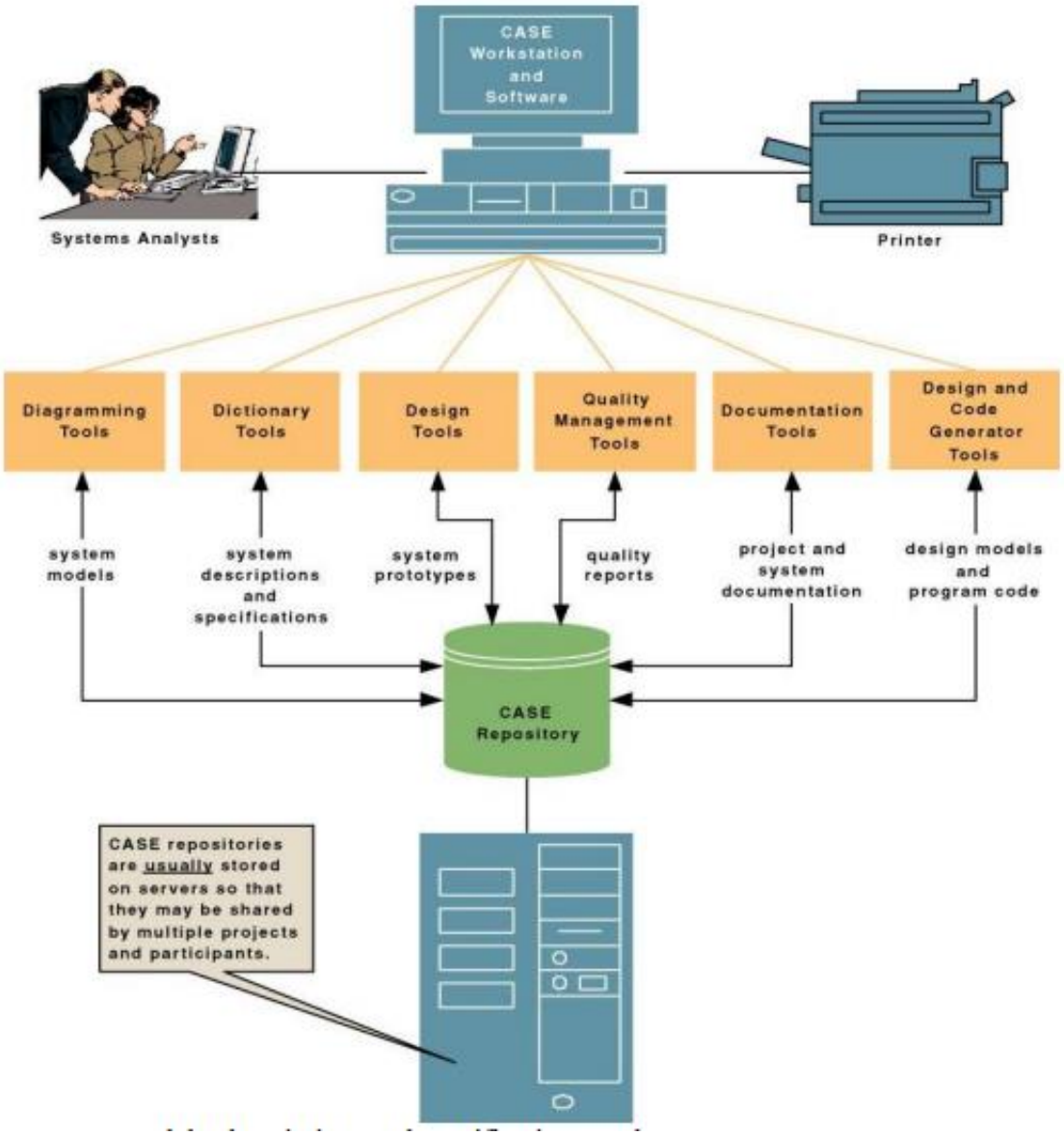
Computer-aided systems engineering (CASE) tools are the software programs that help the development team do their jobs more efficiently and more effectively. These tools support the drawing and analysis of system models. Some CASE tools also provide prototyping and code generation capabilities. Some examples are: Oracle’s Designer 2000, Rational’s Rose, Platinum’s Erwin, Popkin’s System Architect 2001, and Visible System’s Visible Analyst. 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. It is also called dictionary or encyclopedia. Around the CASE repository is a collection of tools or facilities for creating system models and documentation. These facilities generally include:

- **Diagramming tools** –These tools are used to draw system models.
- **Dictionary tools** – These tools are used to record, delete, edit, and output detailed documentation and specification.
- **Design tools** – These tools are used to construct system components including system inputs and outputs. These are also called prototyping tools.
- **Documentation tools** – These tools are used to assemble, organize, and report on system models, descriptions and specifications, and prototypes.
- **Quality management tools** – These tools are used to analyze system models, descriptions and specifications, and prototypes for completeness, consistency, and conformance to accepted rules of methodologies.
- **Design and code generator tools** – These tools automatically generate database designs and application programs or significant portions of those programs.

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".

The figure below shows CASE tool architecture.



Other Approaches

In the efforts to improve the systems analysis and design processes, different approaches have been developed. The traditional waterfall approaches focuses on compartmentalizing project into several phases. The agile approach focuses on self adaptive processes with an emphasis on individual talents. The object oriented approach focuses on combining data and processes into objects and shares the iterative development approach of the agile method. These approaches all have different pros and cons in a way that they could be used to fit and optimize different kinds of projects. **Some approaches are explained below**:

Prototyping Approach

- The prototyping model is a model in which the prototype is developed prior to the actual software.
- Prototype models have limited functional capabilities and inefficient performance when compared to the actual software. Dummy functions are used to create prototypes. This is a valuable mechanism for understanding the customers' needs.
- Software prototypes are built prior to the actual software to get valuable feedback from the customer. Feedbacks are implemented and the prototype is again reviewed by the customer for any change. This process goes on until the model is accepted by the customer.

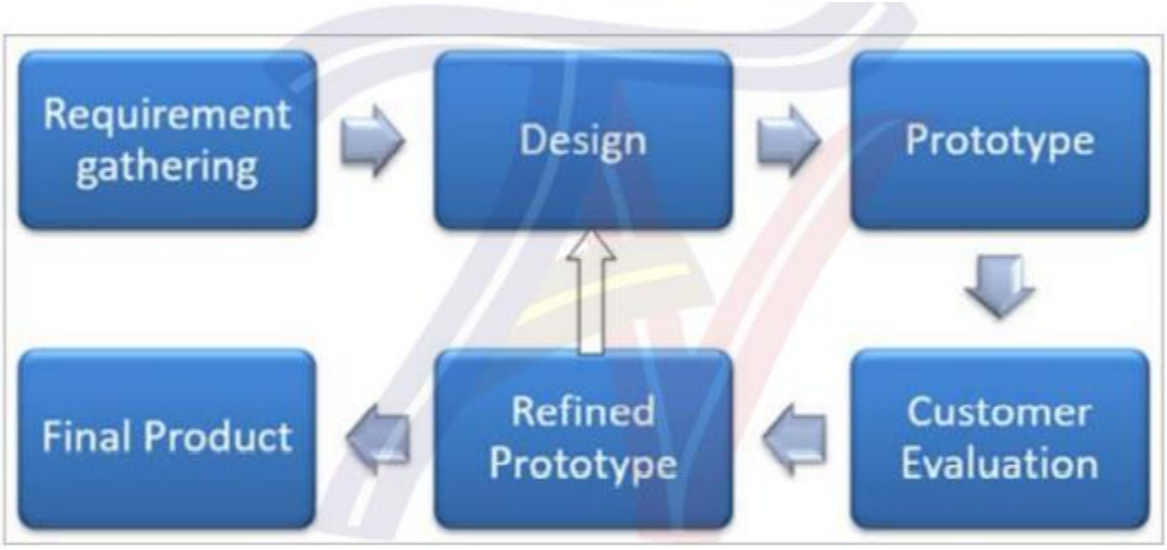


Fig: Prototyping model

- Once the requirement gathering is done, the quick design is created and the prototype which is presented to the customer for evaluation is built.
- Customer feedback and the refined requirement is used to modify the prototype and is again presented to the customer for evaluation. Once the customer approves the prototype, it is used as a requirement for building the actual software. The actual software is build using the Waterfall model approach.

Advantages of Prototype Model:

- Prototype model reduces the cost and time of development as the defects are found much earlier.
- Missing feature or functionality or a change in requirement can be identified in the evaluation phase and can be implemented in the refined prototype.
- Involvement of a customer from the initial stage reduces any confusion in the requirement or understanding of any functionality.

Disadvantages of Prototype Model:

- Since the customer is involved in every phase, the customer can change the requirement of the end product which increases the complexity of the scope and may increase the delivery time of the product.

Spiral Approach

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.

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.

Below diagram shows the different phases of the Spiral Model:

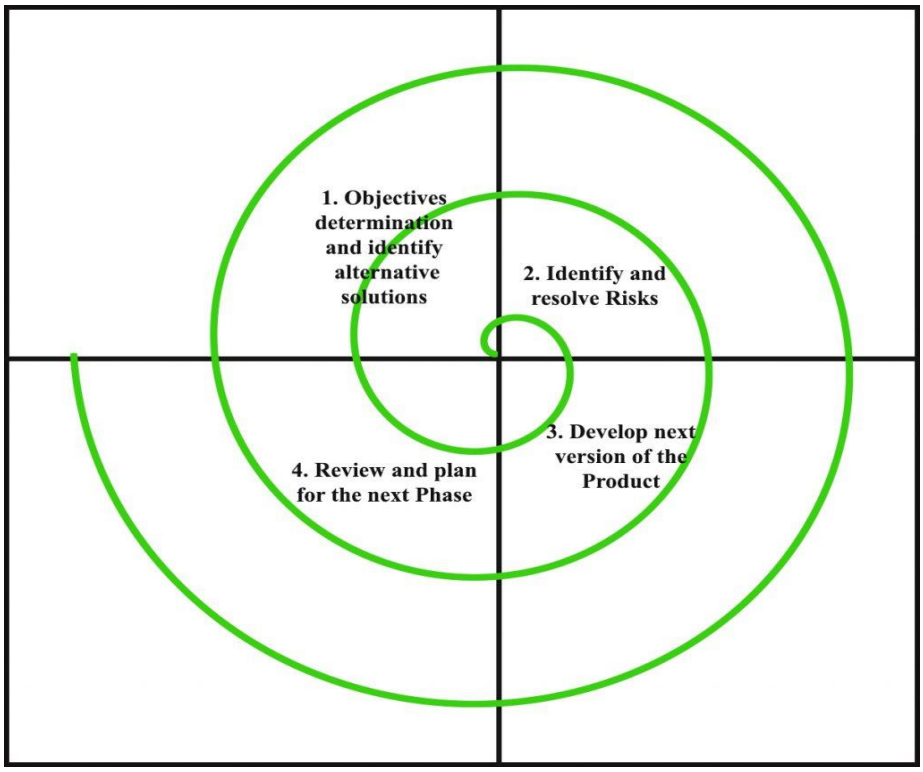


Fig: Different phases of spiral model

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. In the end, planning for the next phase is started.

Advantages of Spiral Model:

- Risk Analysis is done extensively using the prototype models.
- Any enhancement or change in the functionality can be done in the next iteration.

Disadvantages of Spiral Model:

- The spiral model is best suited for large projects only.
- The cost can be high as it might take a large number of iterations which can lead to high time to reach the final product.

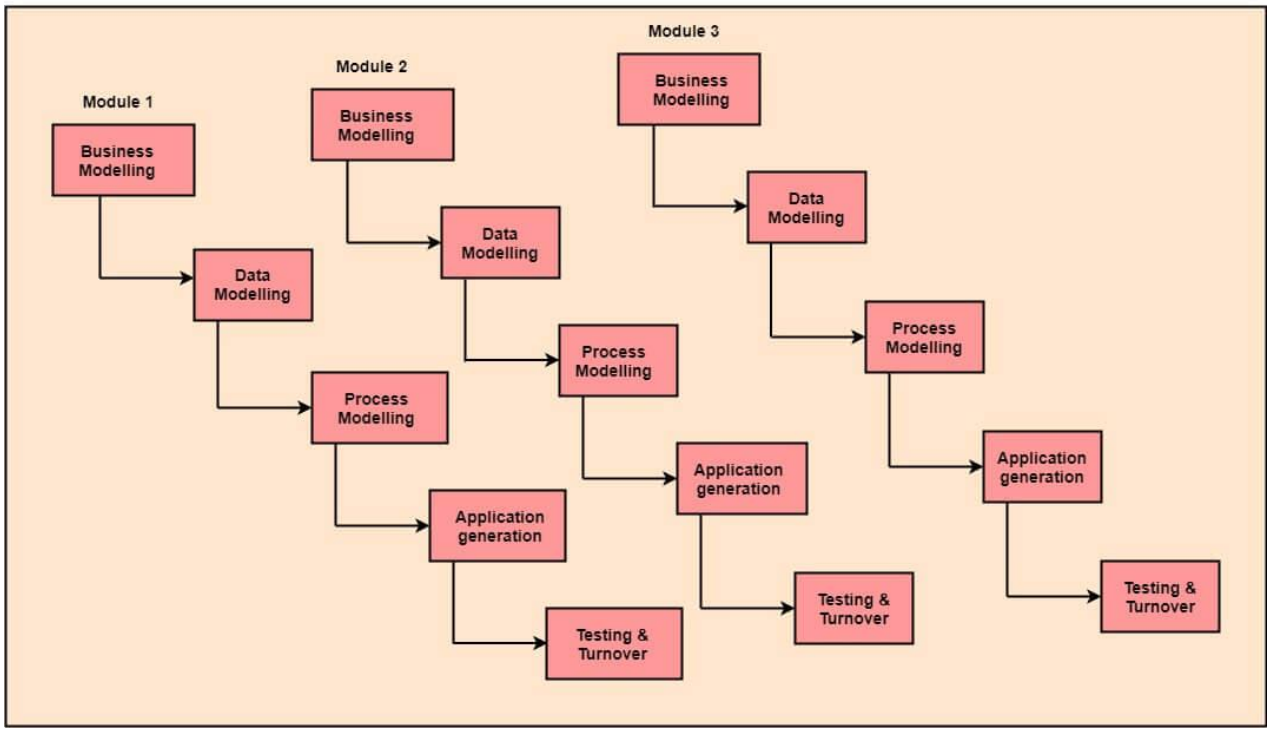
Rapid Application Development Approach

RAD is a linear sequential software development process model that emphasizes a concise development cycle using an element based construction approach. If the requirements are well understood and described, and the project scope is a constraint, the RAD process enables a development team to create a fully functional system within a concise time period.

RAD (Rapid Application Development) is a concept that products can be developed faster and of higher quality through:

- Gathering requirements using workshops or focus groups
- Prototyping and early, reiterative user testing of designs
- The re-use of software components
- A rigidly paced schedule that refers design improvements to the next product version
- Less formality in reviews and other team communication

Fig: RAD Model



The various phases of RAD are as follows:

1.Business Modelling: The information flow among business functions is defined by answering questions like what data drives the business process, what data is generated, who generates it, where does the information go, who process it and so on.

2. Data Modelling: The data collected from business modeling is refined into a set of data objects (entities) that are needed to support the business. The attributes (character of each entity) are identified, and the relation between these data objects (entities) is defined.

3. Process Modelling: The information object defined in the data modeling phase are transformed to achieve the data flow necessary to implement a business function. Processing descriptions are created for adding, modifying, deleting, or retrieving a data object.

4. Application Generation: Automated tools are used to facilitate construction of the software; even they use the 4th GL techniques.

5. Testing & Turnover: Many of the programming components have already been tested since RAD emphasis reuse. This reduces the overall testing time. But the new part must be tested, and all interfaces must be fully exercised.

When to use RAD Model?

- When the system should need to create the project that modularizes in a short span time (2-3 months).
- When the requirements are well-known.
- When the technical risk is limited.
- When there's a necessity to make a system, which modularized in 2-3 months of period.
- It should be used only if the budget allows the use of automatic code generating tools.

Advantage of RAD Model

- This model is flexible for change.
- In this model, changes are adoptable.
- Each phase in RAD brings highest priority functionality to the customer.
- It reduced development time.
- It increases the reusability of features.

Disadvantage of RAD Model

- It required highly skilled designers.
- All application is not compatible with RAD.
- For smaller projects, we cannot use the RAD model.
- On the high technical risk, it's not suitable.
- Required user involvement.

Introduction to Agile Development

Agile software development refers to methods and practices that provide value quickly, efficiently, and consistently to customers as it relates to the software development lifecycle (SDLC). The ability to build and react to change is called *Agile*. It is a method for coping and, ultimately, succeeding in an uncertain and turbulent development environment. Within the Agile software development model, self-organizing and cross-functional teams work together to build and deploy solutions. Some of the popular Agile methodologies include Scrum, Kanban, and Lean.

The Agile software development paradigm is a software development methodology comprising practices and approaches that thrive on iterative and incremental software development. In this development methodology, the requirements – as well as the solutions – evolve through collaboration amongst self-organizing, cross-functional programmer teams that may not or may not be collocated or remote.

The Agile approach fosters adaptability, evolutionary development, and delivery, as well as a time-bound, iterative approach and quick response to change. Agile promotes adaptive planning, evolutionary growth, early delivery, and continual improvement.

The Agile Lifecycle

A typical Agile lifecycle consists of the following steps:

- Project Planning – This helps your team understand the goals, the value to be delivered to the stakeholders, and define the project scope.
- Product Roadmap – This helps to define a breakdown of all the features that are needed as part of the final deliverable.
- Release Planning – This helps plan the future releases and revisit and reevaluate the release plans before a sprint starts.

- Sprint Planning – This helps plan how the tasks in a sprint should be accomplished, by whom and the time it would take to complete those tasks.
- Daily Scrum Meetings – These meetings are usually short and help the team know the tasks to be accomplished on a particular day, the roadblocks (if any) and assess if any changes are required.
- Sprint Review / Retrospective Meetings – These meetings are usually held after every sprint to discuss what went well in the sprint and what did not or what could have been done better.

Managing the Information System Project

In this we focus on the system analyst's role as project manager of an information system project. 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 projects. Effective **project management** helps to ensure that systems development projects meet customer expectations and are delivered within budget and time constraints.

A project management information system gathers, organizes, and uses project data via one or more software applications. These programs assist project managers in planning, executing, and closing their projects and organizing the information stream. There are various PMIS software types, most with similar features like information collection, work authorization, tools for scheduling and delivery, etc. Some additional features - gathering and reporting of automated key performance indicators (KPI).

A Project Management Information System is designed to assist all elements of project management, including information that must be monitored or collected. PMIS can be used for project scope management, integration management, project time management, project cost management, project quality management, project risk management, communications management, and more.

Phases of Project Management

Project management refers to the knowledge, skills and techniques you use to execute and complete projects. Whether it's a one-time project or an ongoing project, it's typically led by a project manager. While responsibilities vary by business and industry, most project managers help define the goals of the project and determine its components as well as the resources necessary to implement it.

Project management is a controlled process of initiating, planning executing, monitoring controlling, and closing down a project. Five phases of project management process are explained below:

Initiating: Before you begin a project, it's important to determine whether it's worth pursuing in the first place. That's where the initiation phase comes in. During the initiation phase, you'll determine the objectives of the project, the problems it's trying to solve and the outcome you hope to achieve. You'll also hone in on how you'll measure success and which internal and external stakeholders will be involved. Put simply, the initial stage is designed to turn abstract ideas into meaningful goals.

Planning: Once the project is approved, you'll want to develop the outline or road map your team will follow to complete it. You'll take the goals you defined in the initiation phase and determine how you'll achieve them. In addition, you'll predict and prepare for any obstacles and nail down the cost, time frame and scope of the project. Unless you're following the agile project management methodology, you should expect the planning phase to take almost half the project's entire time span.

Executing: Now, it's time to put your plan into action! In most cases, there will be a kickoff meeting with all the stakeholders to clarify who is doing what and when. The work that occurs during the execution phase is derived from the project plan. A project manager will usually check in with everyone involved and keep them focused on their assigned tasks while ensuring they collaborate with others so that things run smoothly and everyone stays on the same page.

Monitoring and control: Since project monitoring and adjustment usually occur in tandem, they are both part of the fourth phase. By carefully monitoring tasks, you can actively resolve any roadblocks that might pop up and calculate key performance indicators, or KPIs. Before you move forward with any changes, it's essential to determine how they'll impact the project and whether it's better to stay the original course or follow through with them. The main goal of the monitoring and adjustment phase is to ensure the project objectives and deliverables are met.

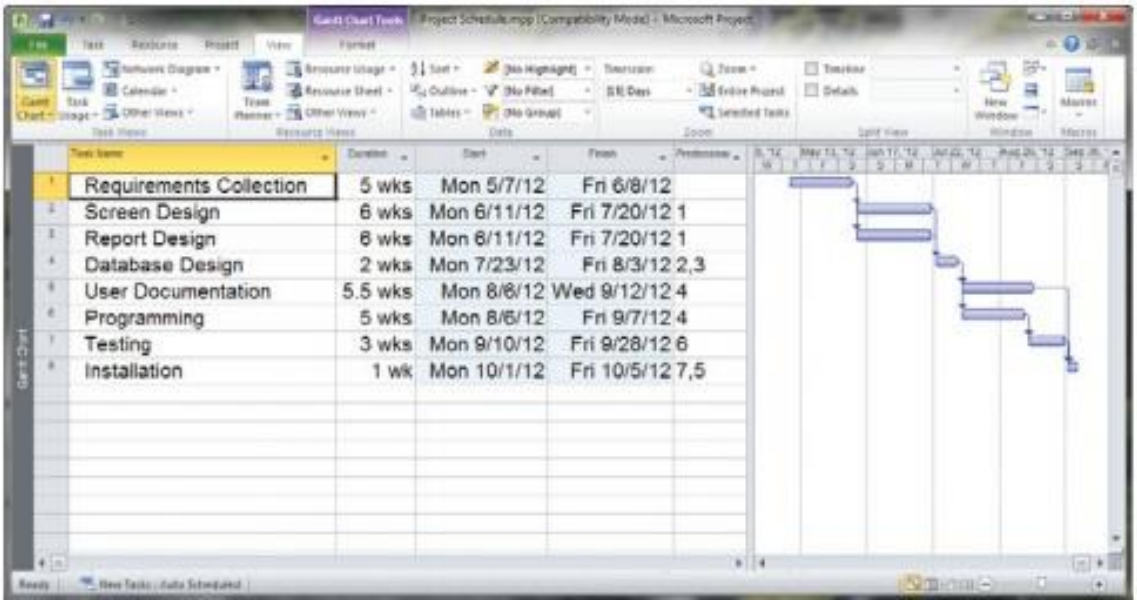
Closing: The closure phase happens at the end of the project after the project has been delivered. It gives your team the chance to assess and document the project and move on to the next one. It's also when the project manager hosts a reflection meeting and presents a final project report and budget. They'll also ask stakeholders to turn over any project documentation and end any contractor relationships. Ideally, you'd use the closure phase to identify the successes and shortcomings of the project to improve the way you plan and execute future projects.

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 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, the activities of a single worker, or for monitoring the progress of activities compared to scheduled completion dates. Recall that a Network diagram shows the ordering of activities by connecting a task to its predecessor and successor tasks. Sometimes a Network diagram is preferable; other times a Gantt chart more easily shows certain aspects of a project. **Here are the key differences between these two representations**:

- A Gantt chart shows the duration of tasks, whereas a Network diagram shows the sequence dependencies between tasks.
- A Gantt chart shows 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 show slack time available within an earliest start and latest finish date. A Network diagram shows these data within activity rectangles.



	Task Name	Duration	Start	Finish	Predecessors
1	Requirements Collection	5 wks	Mon 5/7/12	Fri 6/8/12	
2	Screen Design	6 wks	Mon 6/11/12	Fri 7/20/12	1
3	Report Design	6 wks	Mon 6/11/12	Fri 7/20/12	1
4	Database Design	2 wks	Mon 7/23/12	Fri 8/3/12	2,3
5	User Documentation	5.5 wks	Mon 8/6/12	Wed 9/12/12	4
6	Programming	5 wks	Mon 8/6/12	Fri 9/7/12	4
7	Testing	3 wks	Mon 9/10/12	Fri 9/28/12	6
8	Installation	1 wk	Mon 10/1/12	Fri 10/5/12	7,5

Figure: Gantt chart

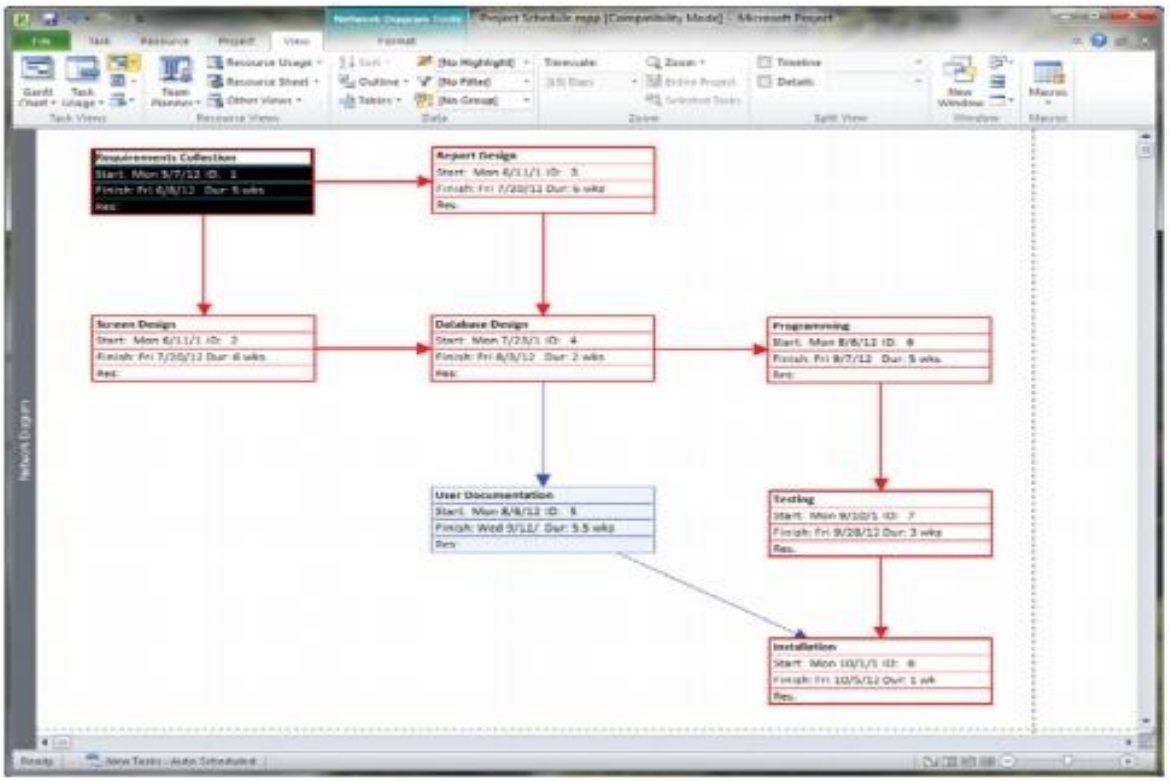


Figure: Network diagram

Representing Project Plans

Project scheduling and management requires 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. A Network diagram is one of the most widely used and best-known scheduling methods.

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, as illustrated in Figure below:

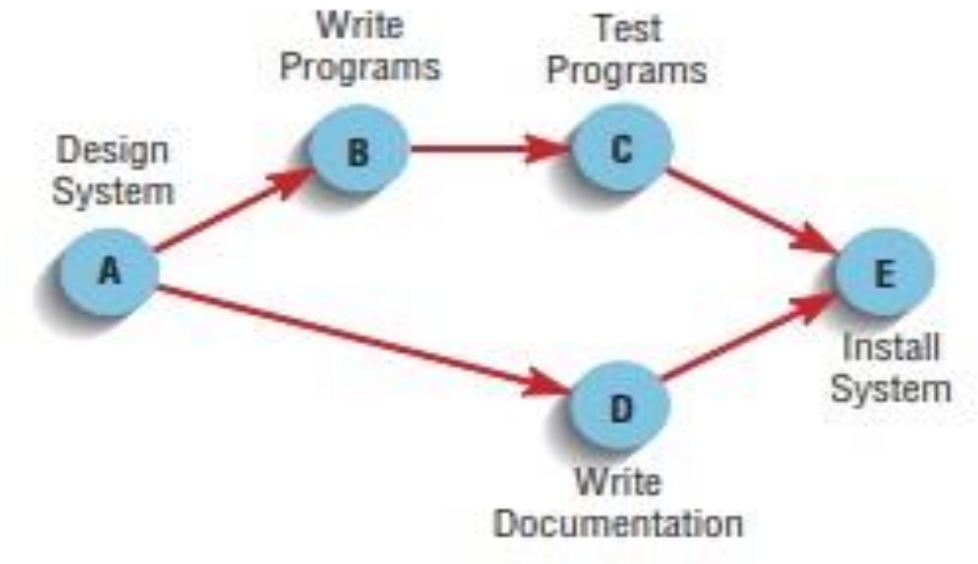


Figure: A Network diagram showing activities (represented by circles) and sequence of those activities (represented by arrows).

Calculating Expected Time Durations Using PERT

One of the most difficult and most error-prone activities when constructing a project schedule is the determination of the time duration for each task within a work breakdown structure. It is

particularly problematic to make these estimates when a high degree of complexity and uncertainty characterize a task. **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 helps you obtain a better time estimate when you are uncertain 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 time (r), or most likely time, reflects the project manager's "best guess" of the amount of time the activity will require for completion. Once each of these estimates is made for an activity, an expected completion time (ET) can be calculated for that activity. Because the expected completion time should be closer to the realistic time (r), the realistic time is typically weighted 4 times more than the optimistic (o) and pessimistic (p) times. Once you add these values together, it must be divided by 6 to determine the ET. This equation is shown in the following formula:

$$ET = \frac{o + 4r + p}{6}$$

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

For example, suppose that your instructor asked you to calculate an expected time for the completion of an upcoming programming assignment. For this assignment, you estimate an optimistic time of 2 hours, a pessimistic time of 8 hours, and a most likely time of 6 hours. Using PERT, the expected time for completing this assignment is 5.67 hours. Commercial project management software such as Microsoft Project assists you in using PERT to make expected time calculations. Additionally, many commercial tools allow you to customize the weighing of optimistic, pessimistic, and realistic completion times.

Using Project Management Software

A wide variety of automated project management tools are 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 common set of features that include the ability to define and order tasks, assign resources to tasks, and easily modify tasks and resources. Project management tools are available to run on Windows-compatible personal computers, the Macintosh, and larger mainframe and workstation-based systems. These systems vary in the number of task activities supported, the complexity of relationships, system processing and storage requirements, and, of course, cost. Prices for these systems can range from a few hundred dollars for personal computer–based systems to more than $100,000 for large-scale multiproject systems. Yet, a lot can be done with systems like Microsoft Project as well as public domain and shareware systems. For example, numerous shareware project management programs (e.g., OpenProj or EasyProjectPlan) can be downloaded from the World Wide Web (e.g., at www.download.com). Because these systems are continuously changing, you should comparison shop before choosing a particular package.

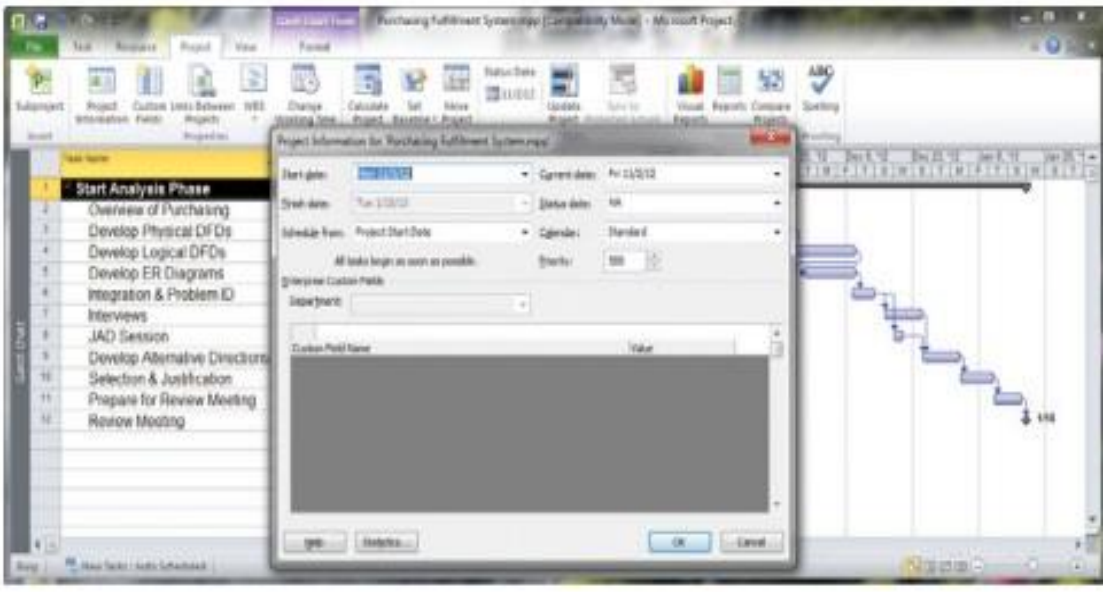


Figure: Establishing a project starting date in Microsoft Project for Windows.

We now illustrate the types of activities you would perform when using project management software. Microsoft Project for Windows is a project management system that has earned consistently high marks in computer publication reviews (see www.microsoft.com and search

for “project”—also, if you search the Web, you can find many useful tutorials for improving your Microsoft Project skills). When using this system 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.

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