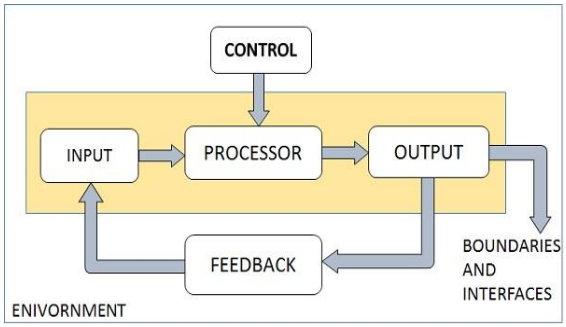
Unit 1
System Development Fundamentals

System Development

- Systems development is systematic process which includes phases such as:

a. Planning
b. Analysis
c. Design
d. Implementation, deployment and maintenance

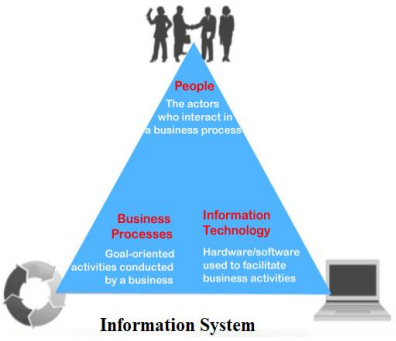
Elements of a System



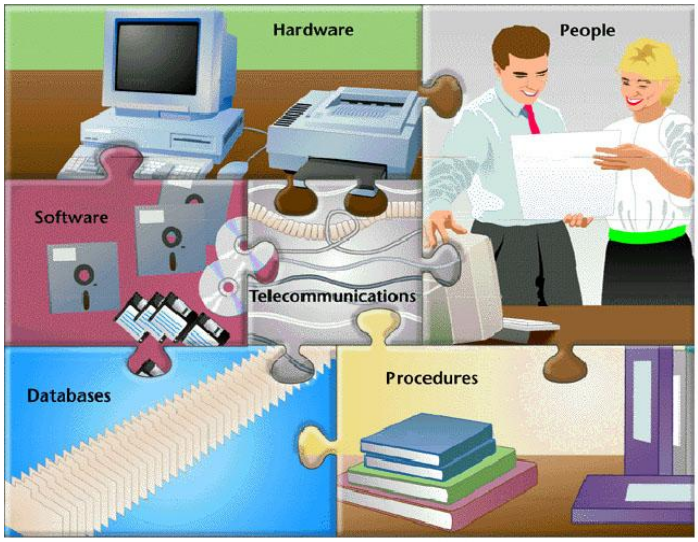
a. Processor: It is the operational component of a system that involves the actual transformation of input into output.
b. Control: It is the decision–making subsystem that controls the pattern of activities governing input, processing, and output.
c. Feedback: Feedback provides the control in a dynamic system.
d. Environment: The environment is the "supersystem" within which the system operates.
e. Boundaries and Interface: Boundaries are the limits that identify its components, processes, and interrelationship when it interfaces with another system.

Information System

- A system is an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal.
- Information system, an integrated set of components for collecting, storing, and processing data and for providing information, knowledge, and digital products.
- It is essential and vital to business, the development of an information system in the organization becomes crucial for the success of the organization



Computer Based Information System



System Analysis

- It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.
- Analysis specifies **what the system should do.**
- System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives.
- The requirements of the system are also analyzed.

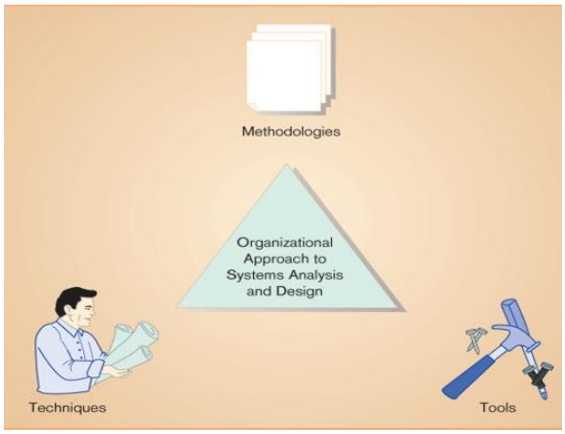
System Design

- System Design is a process that focuses on how to accomplish the objective of the system.
- It is the process of defining the architecture, product design, modules, interfaces, and data for a system to satisfy specified requirements.
- Some of the design methods are Architectural Design, Logical Design, Physical Design, rtc.

What is SAD then?

- Systems Analysis and Design (SAD) is a broad term for describing methodologies for developing high quality Information System which combines Information Technology, People and Data to support business requirement.
- Systems analysis and design, as performed by systems analysts, seeks to understand what humans need to analyze data input or data flow systematically, process or transform data, store data, and output information in the context of a particular business.
- Information systems development covers a wide range of technical areas including business process modeling, data modeling and database design, networking design, computer programming, and computer hardware and operating systems.

Organizational approach to system analysis and design



a. **Methodologies** are a sequence of step-by-step approaches that help develop our final product: the information system.

b. **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.

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

These three elements— methodologies, techniques, and tools—work together to form an organizational approach to systems analysis and design

A Modern Approach to Systems Analysis and Design

- The analysis and design of computer-based information systems began in the1950s.
- Since then, the development environment has changed dramatically, driven by organizational needs as well as by rapid changes in the technological capabilities of computers.
- **1950s:focus on efficient automation of existing processes**
  - development focused on the processes the software performed.
  - computer power was a critical resource, efficiency of processing became the main goal.
  - Computers were large, expensive, and not very reliable.
  - Emphasis was placed on automating existing processes, such as purchasing or paying, 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 for data storage as possible.

A Modern Approach to Systems Analysis and Design

- **1960s: advent of 3GL (Third Generation Programming Language), faster and more reliable computers**
  - development of smaller, faster, less-expensive computers—mini-computers
  - the beginnings of the software industry.
  - Most organizations still developed their applications from scratch using their in-house development staffs.
  - Systems development was more an art than a science.
- **1970s: system development becomes more like an engineering discipline**
  - 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.

A Modern Approach to Systems Analysis and Design

- **1980s: major breakthrough with 4GL (Fourth Generation Programming Language) as as microcomputers became key organizational tools.**
  - CASE tools, object oriented methods
  - computers continued to get smaller, faster, and cheaper,
  - 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.
- **1990s: focus on system integration, GUI applications, client/server platforms, Internet**
  - System development focused on systems integration.
  - Developers used visual programming environments, such as PowerBuilder or Visual Basic, to design the user interfaces for systems that run on client/server platforms.
  - The database, which may be relational or object oriented, and which may have been developed using software from firms such as Oracle, Microsoft, or Ingres, resided on the server..

A Modern Approach to Systems Analysis and Design

- In many cases, the application logic resided on the same server.
- Alternatively, an organization may have decided to purchase its entire enterprise-wide system from companies such as SAP AG or PeopleSoft Inc.
- 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.
- systems development efforts focused on the Internet, especially the Web.

- **The new century: Web application development, wireless PDAs (Personal Digital Assistants), component-based applications**
  - continued focus on developing systems for the Internet and for firms' intranets and extranets.
  - Development of three-tier design,
  - Wireless devices, such as cellphones and personal digital assistants (PDAs) (e.g., Palm Pilots or Pocket PCs), are increasingly able to access Web-based applications from almost anywhere.
  - The trend continues toward assembling systems from programs and components purchased off the shelf.
  - In many cases, organizations not only 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 an application service provider (ASP).

Types of Information Systems

- Transaction Processing Systems (TPS)
  - Automate handling of data about business activities (transactions)
  - Process orientation
- Management Information Systems (MIS)
  - Converts raw data from transaction processing system into meaningful form
  - Data orientation
- Decision Support Systems (DSS)
  - Designed to help decision makers
  - Provides interactive environment for decision making
  - Involves data warehouses, executive information systems (EIS)
  - Database, model base, user dialogue
- Expert Systems (ES)
  - Replicates decision making process
  - Knowledge representation describes the way an expert would approach the problem
- Office Automation System(OAS)
  - concerned with performing day to day business transactions of the organization.
  - Structured Decision
  - Examples of users at this level of management include cashiers at a point of sale, bank tellers, nurses in a hospital, customer care staff, etc.

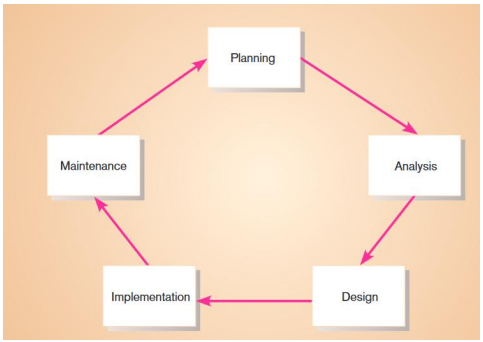
Types of Information Systems and Systems Development

TABLE 1-1 Systems Development for Different IS Types

IS Type	*IS Characteristics*	*Systems Development Methods*
Transaction processing system	High-volume, data capture focus; goal is efficiency of data movement and processing and interfacing different TPSs	Process orientation; concern with capturing, validating, and storing data and with moving data between each required step
Management information system	Draws on diverse yet predictable data resources to aggregate and summarize data; may involve forecasting future data from historical trends and business knowledge	Data orientation; concern with understanding relationships among data so data can be accessed and summarized in a variety of ways; builds a model of data that supports a variety of uses
Decision support system	Provides guidance in identifying problems, finding and evaluating alternative solutions, and selecting or comparing alternatives; potentially involves groups of decision makers; often involves semi-structured problems and the need to access data at different levels of detail	Data and decision logic orientations; design of user dialogue; group communication may also be key, and access to unpredictable data may be necessary; nature of systems requires iterative development and almost constant updating

Developing Information Systems and the System Development Life Cycle (SDLC)

- System Development Methodology is a standard process followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain information systems.
- System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.



System Development Life Cycle (SDLC) Phases-Planning

- In Planning Phase, an organization's total information system needs are identified, analyzed, prioritized, and arranged.
- The outcome of the project identification and selection process is a determination of which system development project should be undertaken by the organization, at least in terms of initial study.
- Major activities during planning are:
  i. Investigation of the system problem or opportunity.
  ii. presentation of reasons why the system should or should not be developed by the organization.
  iii. Determining the scope of the proposed system
- At the end of planning phase, a specific plan for the proposed project the team will follow is produced which specifies the time and resources needed for execution.
- Also we identify whether the costs of developing system would give benefit in this phase.

System Development Life Cycle (SDLC) Phases-Analysis

- The second phase in the SDLC is analysis.
- During this phase, the analyst thoroughly studies the organization's current procedures and the information systems used to perform organizational tasks.
- Analysis has two sub phases:

a. The first is requirements determination. In this sub phase, analysts work with users to determine what the users want from a proposed system. The requirements determination process usually involves a careful study of any current systems, manual and computerized, that might be replaced or enhanced as part of the project.

b. In the second part of analysis, analysts study the requirements and structure them according to their interrelationships and eliminate any redundancies. The output of the analysis phase is a description of (but not a detailed design for) the alternative solution recommended by the analysis team. Once the recommendation is accepted by those with funding authority, the analysts can begin to make plans to acquire any hardware and system software necessary to build or operate the system as proposed.

System Development Life Cycle (SDLC) Phases-Design

- The third phase in the SDLC is design.
- During design, analysts convert the description of the recommended solution into logical and then physical system specifications.
- The analysts must design all aspects of the system, from input and output screens to reports, databases, and computer processes.
- The analysts must then provide the physical specifications of the system they have designed, either as a model or as detailed documentation, to guide those who will build the new system.
- That part of the design process that is independent of any specific hardware or software platform is referred to as **logical design**.
- Once the overall high-level design of the system is worked out, the analysts begin turning logical specifications into physical ones. This process is referred to as **physical design**.
- As part of physical design, analysts design the various parts of the system to perform the physical operations necessary to facilitate data capture, processing, and information output.

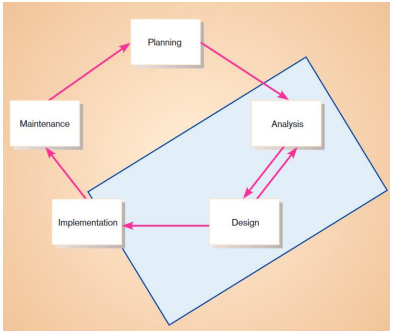
System Development Life Cycle (SDLC) Phases-Implementation

- The fourth phase in the SDLC is implementation.
- The physical system specifications, whether in the form of a detailed model or as detailed written specifications, are turned over to programmers as the first part of the implementation phase.
- During implementation, analysts turn system specifications into a working system that is tested and then put into use.
- Implementation includes coding, testing, and installation.
    a. During coding, programmers write the programs that make up the system. Sometimes the code is generated by the same system used to build the detailed model of the system.
    b. During testing, programmers and analysts test individual programs and the entire system in order to find and correct errors.
    c. During installation, the new system becomes part of the daily activities of the organization. Application software is installed, or loaded, on existing or new hardware, and users are introduced to the new system and trained.

System Development Life Cycle (SDLC) Phases-Maintenance

- The fifth and final phase in the SDLC is maintenance.
- 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, programmers make the changes that users ask for and modify the system to reflect evolving business conditions.
- These changes are necessary to keep the system running and useful.
- The amount of time and effort devoted to maintenance depends a great deal on the performance of the previous phases of the life cycle.

The Heart of the Systems Development Process

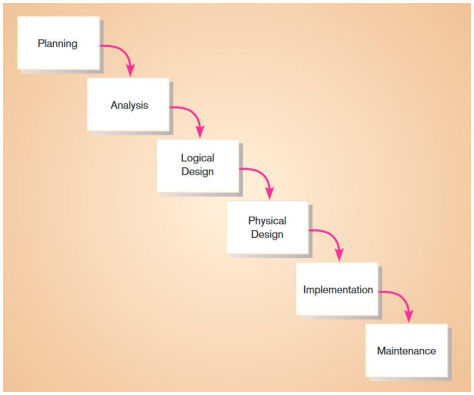


- Current practice combines the activities traditionally thought of as belonging to analysis, design, and implementation into a single process.

- Instead of systems requirements being produced in analysis, systems specifications being created in design, and coding and testing being done at the beginning of implementation, 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 started with Rapid Application Development (RAD) and is seen in such current practices as the Agile Methodologies.

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**.
- The waterfall model emphasizes a logical progression of steps.
- In this model, the whole process of software development is divided into separate phases and typically, the outcome of one phase acts as input for next phase.
- All the phases in this model are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases.
- The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model".
- This means that any phase in the development process begins only if the previous phase is complete.



When is waterfall model appropriate?

Waterfall model is appropriate when:

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

Advantages of Waterfall Model

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

- The waterfall model progresses through easily understandable and explainable phases and thus it is easy to 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.
- In this model, phases are processed and completed one at a time and they do not overlap.
- It is easy to arrange tasks in waterfall model.
- Process and results are well documented.

Disadvantages of Waterfall Model

- 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.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty. Not suitable for projects whose requirements change frequently.
- Not a good model for complex and object-oriented projects. Poor model for long and ongoing projects..
- 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.

Why waterfall model failed?

a. **Overlapping:** The waterfall model has lacked an overlapping among phase. The waterfall model recommends that new phase can start only after the completion of the previous phase. But in real projects, this can't be maintained.
b. **Single way mechanism:** This model is just like the one-way street. Once a phase is completed and next phase has started then there is no way to go back on the previous phase.
c. **Not Flexible:** Difficult to accommodate change requests. The waterfall model assumes that all the customer requirements can be completely and correctly defined at the beginning of the project, but actually customers' requirements keep on changing with time.
d. **Lack of interaction:** The waterfall model has lacked interaction among phase. Feedback is not taken during development. After a development process starts, changes can not accommodate easily.

Other Approaches: Alternatives to Traditional Waterfall SDLC

- Evolutionary\Iterative Model\Approach
  - Prototyping Model
  - Incremental Model
- Rapid Application Development (RAD)
- Spiral Development
- Agile Methodologies
- eXtreme Programming
- Joint Application Design (JAD)
- CASE tools

Evolutionary Model

- Evolutionary Model are iterative.
- Once the requirements are analyzed, they pass through a series of iterations till the complete software is developed.
- After each release, based on the review given by the reviewers, further iterations are performed.
- It involves more user interaction in every iteration, and thereby increasing reliability.
- The iterative lifecycle is based on the successive enlargement and refinement of a system through multiple iterations, with cyclic feedback and adaptation.
- **The result of each iteration is executable but incomplete system** i.e. it is not ready deliver into production.

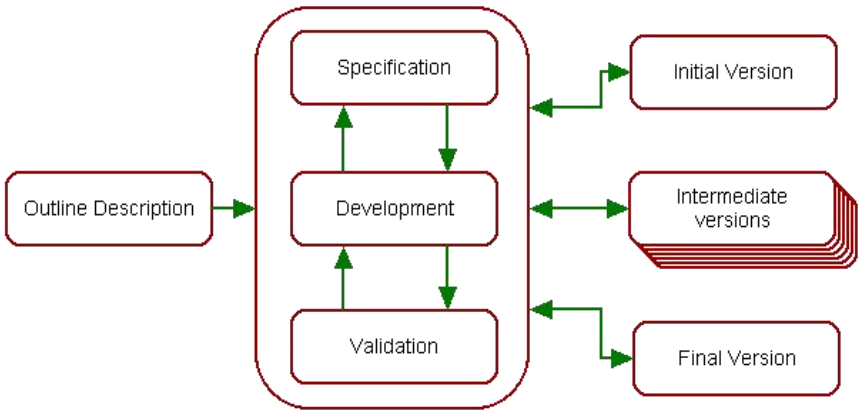
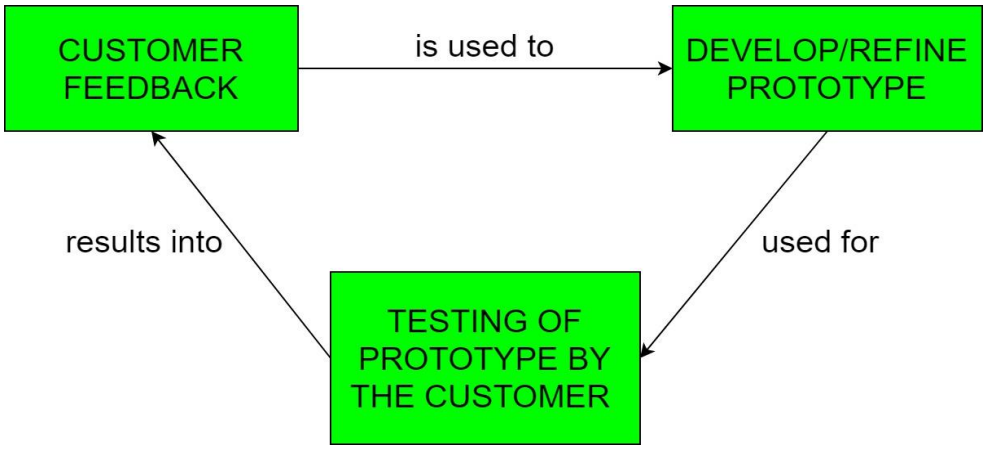


Figure: Evolutionary Model

Prototyping Model

- Prototype is a working model of system/software with some limited functionality.
- Prototyping Model is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved.
- This model is used when the customers do not know the exact project requirements beforehand.
- A prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.



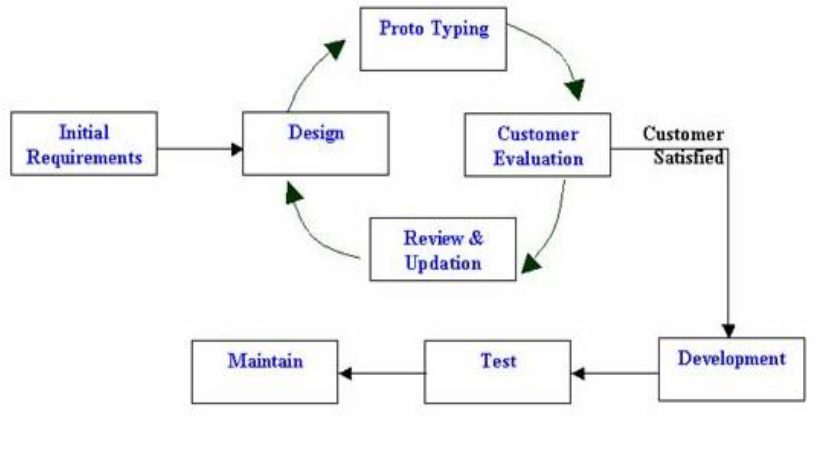


Figure: Prototyping Model

Pros and Cons of Prototyping Model

Pros:

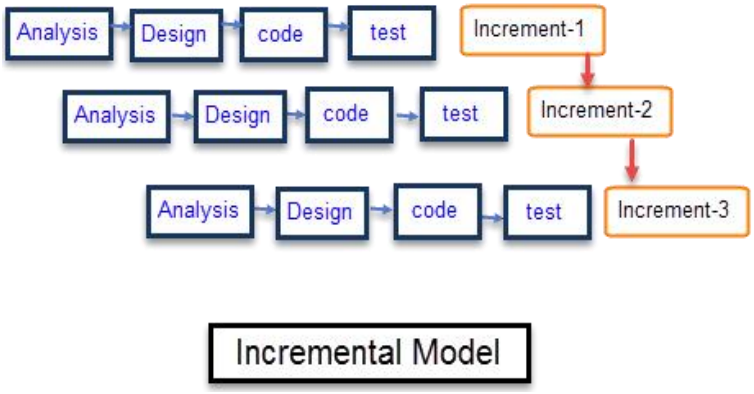
1. Increased user involvement in the product even before its implementation.
2. Since a working model of the system is displayed, the users get a better understanding of the system being developed.
3. Reduces time and cost as the defects can be detected much earlier.
4. Quicker user feedback is available leading to better solutions.
5. Missing functionality can be identified easily.
6. Confusing or difficult functions can be identified.

Cons:

1. Users may get confused in the prototypes and actual systems.
2. Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
3. Developers may try to reuse the existing prototypes to build the actual system, even when it is not technically feasible.
4. The effort invested in building prototypes may be too much if it is not monitored properly

Incremental Development Model

- Incremental: a little more is added each time/adding new functionality in small chunks
- It is a type of evolutionary model.
- Incremental Model is a process of software development where requirements are broken down into multiple standalone modules of software development cycle.
- Incremental development is done in steps from analysis design, implementation, testing/verification, maintenance.
- First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is realized.
- As each successive version of the software is constructed and delivered, now the feedback of the Customer is to be taken and these were then incorporated in the next version.



Incremental Model

Rapid Application Development(RAD) Model

- It targets at developing software in a short span of time.
- It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects.
- The main features of RAD model are that it focuses on the reuse of templates, tools, processes, and code.

Rapid Application Development(RAD) Model

- Rapid Application Development (RAD) model has the following phases −
  - **Requirements Planning phase** − In the requirements planning phase, a workshop needs to be conducted to discuss business problems in a structured manner.
  - **User Description phase** − In the User Description phase, automated tools are used to capture information from users.
  - **Construction phase** − In the Construction phase, productivity tools, such as code generators, screen generators, etc. are used inside a time-box, with a "Do until Done" approach.
  - **Cut Over phase** − In the Cut over phase, installation of the system, user acceptance testing (testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment) and user training are performed

When to use RAD Methodology?

- When a system needs to be produced in a short span of time (2-3 months)
- When the requirements are known
- When the user will be involved all through the life cycle
- When technical risk is less
- When there is a necessity to create a system that can be modularized in 2-3 months of time
- When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation

Pros and Cons

Pros:

- Reduced development time.
- Increases reusability of components
- Quick initial reviews occur
- Encourages customer feedback
- Integration from very beginning solves a lot of integration issues.

Cons:

- Depends on strong team and individual performances for identifying business requirements.
- Only system that can be modularized can be built using RAD
- Requires highly skilled developers/designers.
- High dependency on modeling skills
- Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

Agile Development Model

- Agile is the ability to create and respond to change.
- It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment.
- The Agile model was primarily designed to help a project to adapt to change requests quickly. So, the main aim of the Agile model is to facilitate quick project completion.
- Agility is achieved by fitting the process to the project, removing activities that may not be essential for a specific project. Also, anything that is wastage of time and effort is avoided.
- Agile approaches to software development consider design and implementation to be the central activities in the software process.
- Specification, Design, Implementation and testing are interleaved.

Agile Principles

1. Customer Involvement: Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
2. Incremental Delivery: The software is developed in increments with the customer specifying the requirements to be included in each increment.
   - Frequent delivery of working software
3. People not process: The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes
   - Support, trust, and motivate the people involved
4. Embrace Change: Expect the system requirements to change and so design the system to accommodate these changes.
5. Maintain simplicity: Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

Agile Principles

6. Enable face-to-face interactions
7. Working software is the primary measure of progress
8. Agile processes to support a consistent development pace
9. Attention to technical detail and design enhances agility
10. Regular reflections on how to become more effective
11. Customer satisfaction through early and continuous software delivery

Existing Agile Development Methodologies

- Extreme Programming (XP)
- Scrum
- Crystal Family of Methodologies
- Feature Driven Development
- Dynamic Systems Development Methods
- Adaptive software development
- Open source software development
- Agile Modeling
- Pragmatic Programming

Advantages of Agile Model:

- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Enables concurrent development and delivery within an overall planned context.
- Frequent Delivery
- Face-to-Face Communication with clients.

Disadvantages:

- Not suitable for handling complex dependencies.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- lack of documentation.

Extreme Programming (XP)

- XP: like iterative but taken to the extreme
- An agile software development framework that aims to produce higher quality software, and higher quality of life for the development team
- Based on the iterative-and-incremental approach
- Extreme Programming (XP) is a discipline of software development based on values of simplicity, communication and feedback.
- XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.
- EXtreme Programming (XP) was conceived and developed to address the specific needs of **software development by small teams in the face of vague and changing requirements.**
- It provides values and principles to guide the team behavior.

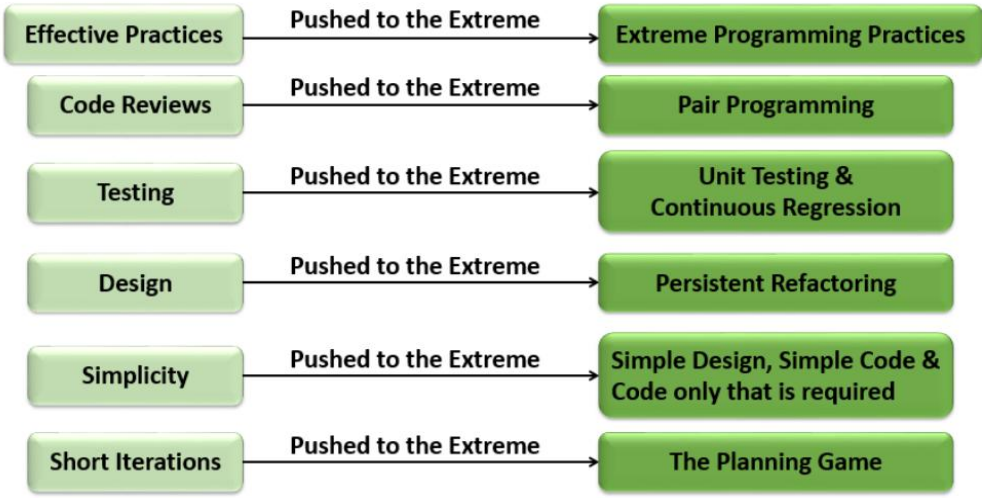
Extreme Programming

- Extreme Programming involves-

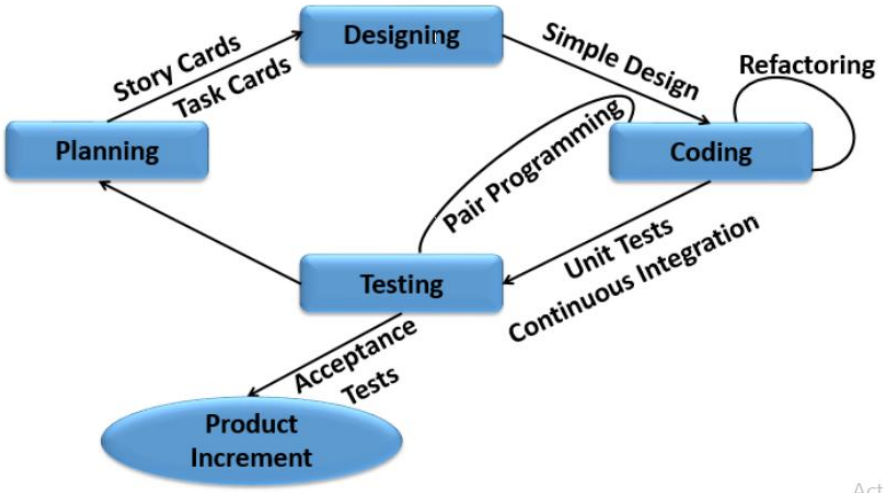
1. Writing unit tests before programming and keeping all of the tests running at all times. The unit tests are automated and eliminates defects early, thus reducing the costs.
2. Starting with a simple design just enough to code the features at hand and redesigning when required.
3. **Programming in pairs (called pair programming),** with two programmers at one screen, taking turns to use the keyboard. While one of them is at the keyboard, the other constantly reviews and provides inputs.
4. Integrating and testing the whole system several times a day.
5. Putting a minimal working system into the production quickly and upgrading it whenever required.
6. Keeping the customer involved all the time and obtaining constant feedback.

Why is it called Extreme?

➤Extreme Programming takes the effective principles and practices to extreme levels.



Extreme Programming



Joint Application development

- It is the process which is used to design and develop computer based system/solutions.
- **JAD involves the client or end-users in designing and development process.**
- As compared to other primitive SDLC models, the JAD model leads to faster progression of the system development, which has better client approval.
- It also comprises of approaches for improving the quality of specification and user participation through a successive collaborative workshops called JAD sessions.

Joint Application Design (JAD)

- **Phases of JAD Model:**
  - **Define Specific Objectives**: The facilitator, in partnership with stakeholders, sets all the objectives and a list of items, which is then distributed to other developers and participants to understand and review.
  - **Session Preparation**: The facilitator is solely responsible for this preparation, where all relevant data is collected and sent to other members before time.
  - **Session Conduct**: Here, the facilitator is accountable for identifying those issues that have to be working out to make the system error-free.
  - **Documentation**: After the product is developed, the records and published documents are put forward into the meeting so that the stakeholders and consumers can approve it through the meeting.
- **Benefits using JAD**
  - **Faster Delivery time**
  - **Better understanding**: professional interaction among each other makes better understandable.
  - **Cost minimization**: stakeholders will make less effort to develop the system
  - **Quality improvements**: all the key decision-makers and stakeholders of the project are involved in the development of the project, so there is the least chance of error, and hence the product quality becomes better and more accurate.
- **Weaknesses:**
  - Difficult to align goals and maintain focus because of different options within team
  - Require a significant commitments

Spiral Development Model

- Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis.
- This model is best used for large projects which involves continuous enhancements.
- There are specific activities which are done in one iteration (spiral) where the output is a small prototype of the large software.
- The same activities are then repeated for all the spirals till the entire software is build.
- Each loop of the spiral is called a Phase of the software development process.
- The exact number of phases needed to develop the product depends upon the project risk.

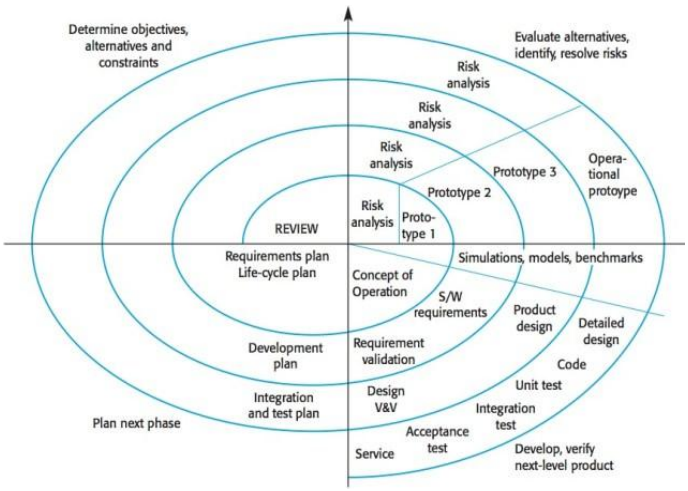


Figure: Spiral Model

- Spiral model, generally involves four phases, which are followed repeatedly, in each round of the model, until no further requirements are needed to be implemented in the software product.
- 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.

When is Spiral Model applicable?

- When there is a budget constraint and risk evaluation is important.
- For medium to high-risk projects.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

Pros and Cons of Spiral Model

Pros:

1. Changing requirements can be accommodated.
2. Allows extensive use of prototypes.
3. Requirements can be captured more accurately.
4. Users see the system early.
5. Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

Cons:

1. Management is more complex.
2. End of the project may not be known early.
3. Not suitable for small or low risk projects and could be expensive for small projects.
4. Process is complex
5. Spiral may go on indefinitely.
6. Large number of intermediate stages requires excessive documentation.

Computer-Aided Software Engineering(CASE)

- Computer-aided software engineering (CASE) is software to support software development and evolution processes.
- Computer aided software engineering (CASE) is the implementation of computer facilitated tools and methods in software development.
- CASE is used to ensure a high-quality and defect-free software.
- It generates a framework for organizing projects and to be helpful in enhancing productivity.

Types of CASE Tools

1. Diagram tools

- These tools are used to represent system components, data and control flow among various software components and system structure in a graphical form.
- For example, Flow Chart Maker tool for creating state-of-the-art flowcharts.

2. Process Modeling Tools

- Process modeling is method to create software process model, which is used to develop the software.
- Process modeling tools help the managers to choose a process model or modify it as per the requirement of software product.
- For Example: EPF Composer(an open-source tool platform designed for process engineers and project managers)

3. Project Management Tools

- These tools are used for project planning, cost and effort estimation, project scheduling and resource planning.
- For example, Creative Pro Office, Trac Project, Basecamp.

4. Documentation Tools

- Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project.
- Documentation tools generate documents for technical users and end users.
- Technical users are mostly in-house professionals of the development team who refer to system manual, reference manual, training manual, installation manuals etc.
- The end user documents describe the functioning of the system such as user manual.
- For example, Doxygen, DrExplain, Adobe RoboHelp for documentation.

5. Analysis Tools

- These tools help to gather requirements, automatically check for any inconsistency, inaccuracy in the diagrams, data redundancies, etc.
- For example, Accept 360, Accompa, CaseComplete for requirement analysis, Visible Analyst for total analysis.

6. Design Tools

- These tools help software designers to design the block structure of the software, which may further be broken down in smaller modules using refinement techniques.
- These tools provides detailing of each module and interconnections among modules.
- For example, Animated Software Design

7. Configuration Management Tools

- Configuration Management tools deal with:
  - Version and revision management
  - Baseline configuration management
  - Change control management

- CASE tools help in this by automatic tracking, version management and release management.
- For example: Fossil, Git, Accu REV.

8. Change Control Tools

- These tools are considered as a **part of configuration management tools.**
- They deal with changes made to the software after its baseline is fixed or when the software is first released.
- CASE tools automate change tracking, file management, code management and more.

9. Programming Tools

- These tools consist of programming environments like IDE (Integrated Development Environment), in-built modules library and simulation tools.
- For example: Eclipse, Netbeans.

10. Prototyping Tools

- Prototype provides **initial look and feel** of the product and simulates few aspect of actual product.
- Prototyping CASE tools essentially come with graphical libraries.
- They can create hardware independent user interfaces and design.
- These tools help us to build rapid prototypes based on existing information.
- For example, Serena prototype composer, Mockup Builder, etc.

11. Web Development Tools

- These tools assist in designing web pages with all allied elements like forms, text, script, graphic and so on.
- Web tools also provide live preview of what is being developed and how will it look after completion.
- For example, Fontello, Adobe Edge Inspect, Foundation 3, Brackets.

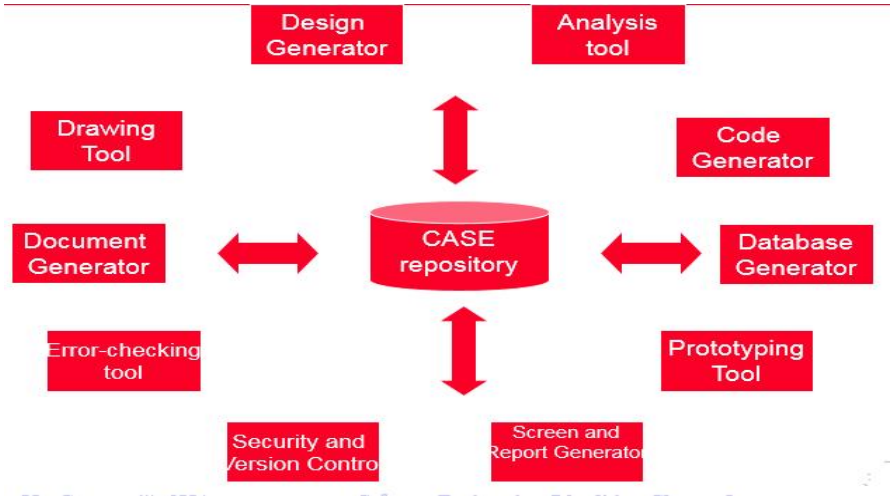
12. Quality Assurance Tools

- Quality assurance in a software organization is monitoring the engineering process and methods adopted to develop the software product in order to ensure conformance of quality as per organization standards.
- QA tools consist of configuration and change control tools and software testing tools.
- For example: SoapTest, AppsWatch, Jmeter, Selenium.

13. Maintenance Tools

- Software maintenance includes modifications in the software product after it is delivered.
- Automatic logging and error reporting techniques, automatic error ticket generation and root cause Analysis are few CASE tools.
- For example: Bugzilla for defect tracking, HP Quality Center.

Components of CASE



Advantages of the CASE approach

1. Cost Reduction: As special emphasis is placed on redesign as well as testing, the servicing cost of a product over its expected lifetime is considerably reduced.
2. Quality Improvement: The overall quality of the product is improved as an organized approach is undertaken during the process of development.
3. Meets user requirements more likely: Chances to meet real-world requirements are more likely and easier with a computer-aided software engineering approach.
4. CASE indirectly provides an organization with a competitive advantage by helping ensure the development of high-quality products.

Object Oriented Analysis and Design (OOAD)

- It is a technical approach for analyzing and designing an application, system, or business by applying Object Oriented Concepts.
- In OOAD, analysis is object-oriented and design is also object-oriented. So OOAD is a design approach that emphasizes on a logical solution based on objects.
- The object-oriented approach combines data and processes(called methods) into single entities called objects.
- Goals of OOAD:
  - Identifying the concepts and objects of a system
  - Identify the relationships existing between the objects
  - Generate a design which can be converted into applications using OO languages
  - Provide a logical and optimal solution based on objects
  - Analyzing, Designing and Developing a fully functioning system or application.
  - Increase Reusability and Improve system quality

Object Oriented Analysis and Design (OOAD) Based Development

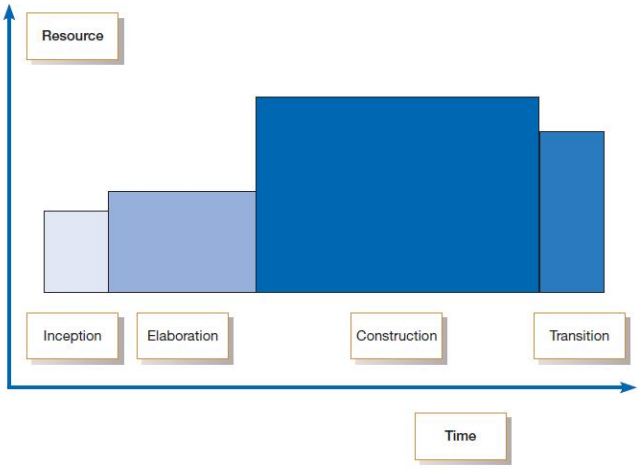


FIGURE 1-11
Phases of OOAD-based development

Object Oriented Analysis and Design (OOAD) Based Development-Phase

a. **Inception Phase:** Inception is the initial short step to establish a common vision and basic scope for the project which involves the following activities:

- Requirements understanding and requirements description
- Understand the business case of the system being developed.
- Describes the scope of the system.
- Describes the goals of the system.
- Feasibility study is performed as well as size of the project is estimated during the inception phase.
- The actors of the system and their interaction with the system are analyzed at a high level.
- To identify the entities that interact with the system like people, organizations, external systems.

b. Elaboration Phase

➤ The project's architecture and required resources are further evaluated.

➤Developers consider possible applications of the software and costs associated with the development.

➤The project plan is developed and risk assessment is also performed. Some of the types of risks are requirement risks, technological risks, skills risks, political risks.

➤It ensures the followings things/factors for the next phase(construction phase)

- Standards
- Guidelines
- Use case model
- Revised Risk List
- Processes
- Tools

c. Construction Phase

➤In this phase, project is developed and completed. The software is designed, written, and tested.

➤This phase involves:

- Completely understanding and describing the remaining requirements that were miss during inception and elaboration.
- Preparing the design of the system.
- Ensuring the successful validation of the customer.
- Optimizing the resources.
- Minimizing the cost.
- Developing the versions of the system.

➤At the end of this phase :

- The product should be stable and mature for release
- Actual versus planned expenditure should be acceptable

d. Transition phase

➤This phase involves beta testing and product deployment. The software is released to the public.

➤Final adjustments or updates are made based on feedback from end users.

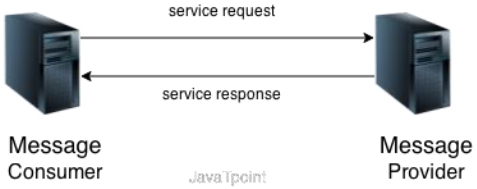
➤Transition phase involves:

- Delivery of the overall system to the customer.
- Defects findings
- System modifications.
- Product installation
- Product release

➤The main objectives of transition phase are customer satisfaction and achieving final product baseline in cost effective manner.

Service-Oriented Architecture (SOA)

- SOA is an architectural pattern which is designed to build distributed systems that deliver services to other applications through the protocol (set of rules for communication).
- Here service is a well-defined, self-contained function that represents a unit of functionality.
- A service can exchange information from another service.
- It uses a loosely coupled (less dependent on other), message-based communication model to communicate with applications and other services.



Service-Oriented Architecture (SOA)-Advantages

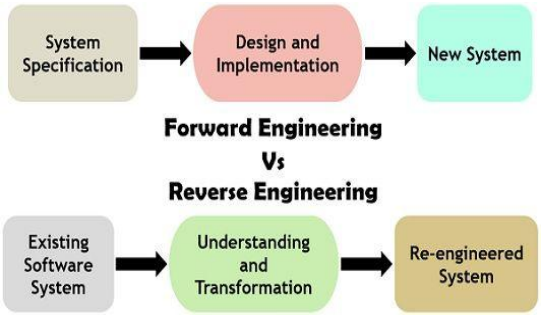
- SOA allows reuse the service of an existing system alternately building the new system.
- It allows plugging in new services or upgrading existing services to place the new business requirements.
- It can enhance the performance, functionality of a service and easily makes the system upgrade.
- SOA has capability to adjust or modify the different external environments and large applications can be managed easily.
- The companies can develop applications without replacing the existing applications.
- It provides reliable applications in which you can test and debug the independent services easily as compared to large number of code.

Service-Oriented Architecture (SOA)-Disadvantages

- SOA requires high investment cost (means large investment on technology, development and human resource).
- There is greater overhead when a service interacts with another service which increases the response time and machine load while validating the input parameters.
- SOA is not suitable for GUI (Graphical User Interface) applications which will become more complex when the SOA requires the heavy data exchange

Forward and Reverse Engineering

- Forward Engineering is the mode of creation or development in which the development is done on the basis of given requirements from client/consumer.
- On other hand Reverse Engineering is the mode of creation or development in which the development is done on the basis of requirements gathered from the developed application (existing application) or the changes/enhancements that are provided from the client/consumer.

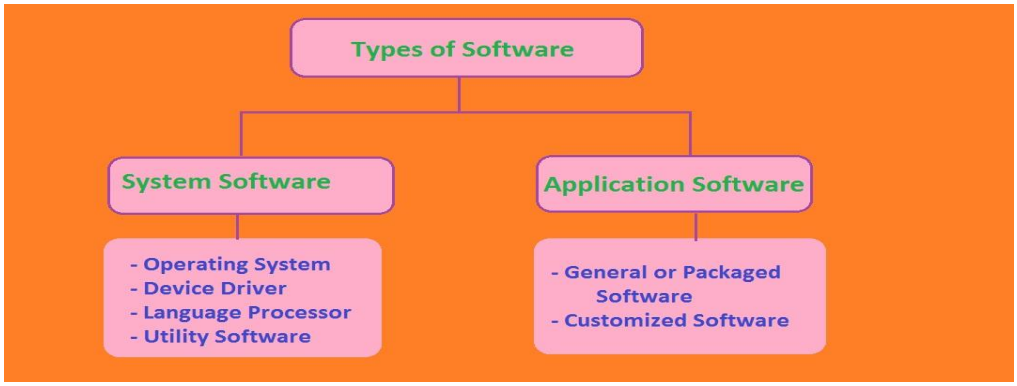


The Origins Of Software

- Software is a set of instructions or programs that tells a computer what to do.
- Software is collection of computer programs and associated documentation such as requirements, design models and user manuals.
- Software products may be developed for a particular customer or may be developed for a general market.
- New software can be created by developing new programs, configuring generic software systems or reusing existing software.
- Software products may be:
  - Generic - developed to be sold to a range of different customers e.g. PC software such as Excel or Word.
  - Bespoke (custom) - developed for a single customer according to their specification.

Types of Software

➤Software can be categorized according to what it is designed to accomplish.



System Software

➤The type of software that supports and manages the computer resources and operations of the computer system is known as system software.

➤It creates a communication link between the computer and user.

➤It also provides and maintains an environment or platform for running application software.

➤Futher divided into four sub types:

a. Operating System: Controls and coordinates overall operations of the computer system
b. Device Driver: Controls the devices attached to the computer system.
c. Language Processor: Language processor is a type of system software that translates the programming language code into machine code.
d. Utility Software-software used for maintenance of computer system like Antivirus, Network utilities, Screen savers, File managers, Disk defragmenters, Disk cleaners etc.

Functions of System Software

1. Monitoring the use of all the hardware components
2. Providing an easy and efficient interaction between the computer and user.
3. Communication with the peripheral devices such as a scanner, printer, webcam etc.
4. Managing files and folders
5. Providing security
6. Providing platform for application software

Application Software

➤Application software is a type of software that is designed to solve a specific problem of the user or to perform the specific task.

➤The application software can be sub-divided into two types.

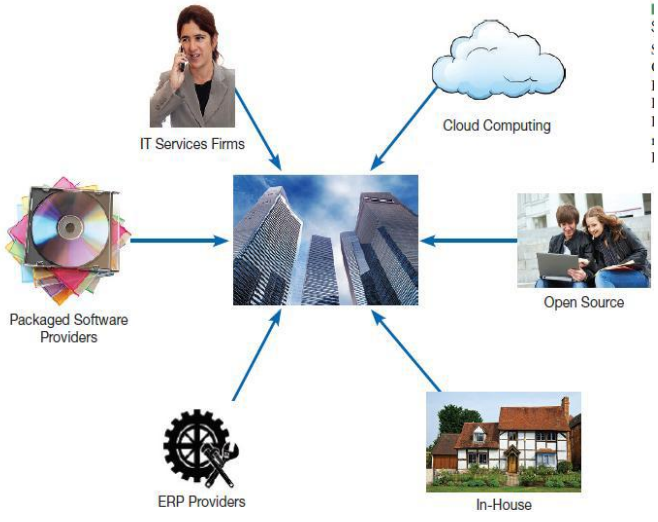
a. Tailored or customized Software: Sometimes companies, organization or an individual needs specific types of **software to solve specific problems**, so they ask the software development companies or individual programmers to develop a custom software that meets the requirement to solve their problem.
For instance, Payroll system, Result processing, Library management system, Banking system, School management system etc. are the example of customized software.

b. General or Packaged software: General software are those types of software that are made by software companies **to solve the general problems of people**.
Some examples of these types of software are MS-Excel, MS-Access, Photoshop, Dreamweaver, Notepad, etc.

System Acquisition

- Acquisition → an asset or object bought or obtained
- Internal Corporate Information Systems departments now spend a smaller and smaller proportion of their time and effort on developing systems from scratch (very beginning).
- Companies continue to spend relatively little time and money on traditional software development and maintenance.
- Instead, they invest in **packaged software**, **open-source software**, and **outsourced services**.

Sources of Application Software



Sources of Software

- We can group the sources of software into six major categories:
  - **In-house Development**
  - **Information Technology Services Firms**
  - **Packaged Software Producers**
  - **Enterprise-wide Solutions**
  - **Cloud Computing Vendors**
  - **Open-source Software**

TABLE 2-1 Leading Software Firms and Their Development Specializations

Specialization	Example Firms or Websites
IT Services	Accenture Computer Sciences Corporation (CSC) IBM HP
Packaged Software Providers	Intuit Microsoft Oracle SAP AG Symantec
Enterprise Software Solutions	Oracle SAP AG
Cloud Computing	Amazon.com Google IBM Microsoft Salesforce.com
Open Source	SourceForge.net

Sources of Software

1. In-house Development

- In-house refers to conducting an activity or operation within a company, instead of relying on outsourcing.
- Development work that takes place in and for organizations.
- In-house development can lead to a larger maintenance burden than other development methods, such as packaged applications, outsourcing.

2. Information Technology Services Firms

- If a company needs an information system but does not have the expertise or the personnel to develop the system in-house the company will likely consult an information technology services firm.
- IT services firms help companies develop custom information systems for internal use, or they develop, host, and run applications for customers, or they provide other services according to need.

Sources of Software

3. Packaged Software Producers

- Some of the largest computer companies in the world are companies that produce software exclusively.
- Software companies develop what are sometimes called prepackaged or off-the-shelf (off-the-shelf-software are standardized software applications that are mass-produced, available to the general public, and fit for immediate use) systems.
- Software companies develop software to run on many different computer platforms, from microcomputers to large mainframes.
- The companies range in size from just a few people to thousands of employees.
- Software companies consult with system users after the initial software design has been completed and an early version of the system has been built. The systems are then tested in actual organizations to determine whether there are any problems or if any improvements can be made.
- **A good example is Microsoft**, probably the best-known software company in the world. Almost 87 percent of Microsoft's revenue comes from its software sales, mostly for its Windows operating systems and its personal productivity software, the Microsoft Office Suite.

Sources of Software

4. Enterprise Solutions Software

- Many firms have chosen complete software solutions, called enterprise solutions or enterprise resource planning (ERP) systems, to support their operations and business processes.
- These ERP software solutions consist of a series of integrated modules. Each module supports an individual, traditional business function, such as accounting, distribution, manufacturing, or human resources.
- The difference between the modules and traditional approaches is that the modules are integrated to focus on business processes rather than on business functional areas.
- For example, a series of modules will support the entire order entry process, from receiving an order, to adjusting inventory, to shipping to billing, to after-the-sale service.
- Using enterprise software solutions, a firm can integrate all parts of a business process in a unified information system.
- All aspects of a single transaction occur seamlessly within a single information system, rather than as a series of disjointed, separate systems focused on business functional areas.

Sources of Software

4. Enterprise Solutions Software

- The benefits of the enterprise solutions approach include a single repository of data for all aspects of a business process and the flexibility of the modules.
- A single repository ensures more consistent and accurate data, as well as less maintenance.
- The modules are flexible because additional modules can be added as needed once the basic system is in place.
- Added modules are immediately integrated into the existing system.
- Examples: Customer Relationship Management (CRM) Software, Project Management Tools, Marketing Automation Tools, etc.

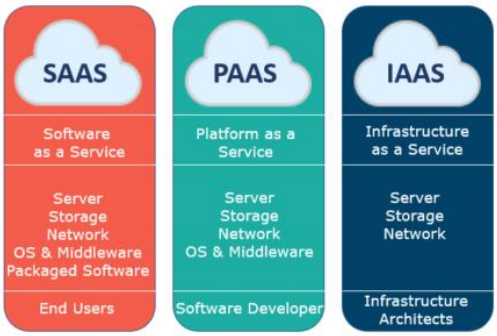
Disadvantages

- The systems are very complex, so implementation can take a long time to complete.
- Organizations typically do not have the necessary expertise inhouse to implement the systems

Sources of Software

5. Cloud Computing

- Another method for organizations to obtain applications is to rent them or license them from third-party providers who run the applications at remote sites.
- Users have access to the applications through the Internet or through virtual private networks.
- The application provider buys, installs, maintains, and upgrades the applications.
- **Users pay on a per-use basis** or they license the software, typically month to month.
- A well-known example of cloud computing is **Google Apps**, where users can **share** and **create documents, spreadsheets, and presentations**



Sources of Software

6. Open Source Software

- The term open source refers to something people can modify and share because its design is publicly accessible.
- Open source software is code that is designed to be publicly accessible—anyone can see, modify, and distribute the code as they see fit.
- Open source software is often cheaper, more flexible.
- Some of the most well-known and popular open-source software names are Linux, an operating system; mySQL, a database system; and Firefox, a web browser.
- Companies and individuals can make money with open source in two primary ways:
  1. by providing maintenance and other services or
  2. by providing one version of the software free and selling a more fully featured version

Reuse

- Reuse is the use of previously written software resources in new applications.
- Because so many bits and pieces of applications are relatively generic across applications, it seems intuitive that great savings can be achieved in many areas if those generic bits and pieces do not have to be written a new each time they are needed.
  - **Reuse reduces the need of software development from scratch**
- Reuse should increase programmer productivity because being able to use existing software for some functions means they can perform more work in the same amount of time.
- Reuse should also decrease development time, minimizing schedule overruns.
- Since existing pieces of software have already been tested, reusing them should also result in higher-quality software with lower defect rates, decreasing maintenance costs.

Project Management

- Project management is the use of specific knowledge, skills, tools and techniques to deliver something of value to people.
- Project management involves the planning and organization of a company's resources to move a specific task, event, or duty towards completion.
- Project management is often associated with fields in engineering and construction and, more lately, healthcare and Information Technology (IT).

Basic Concept In Project Management

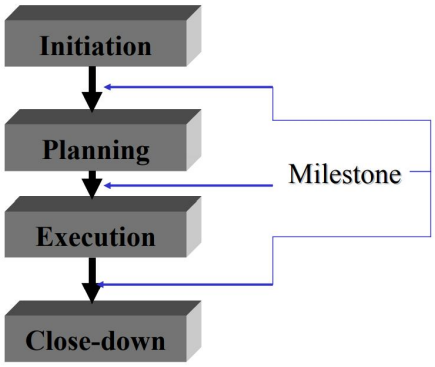
- **Project** is a planned undertaking of a series of related activities to reach an objective that has a beginning and an end
- **Objective of a project**
  - Solve a business problem (develop a MIS)
  - Take advantage of a business opportunities (develop BIS)
  - Other non rational reason: spend existing available resources, training and enhancing skills of employees
- **Project manager** is an individual with a diverse set of skills who is responsible for managing the project process when the project is accepted.
- **Milestone:** A milestone is a specific point within a project's life cycle used to measure the progress toward the ultimate goal. Milestones in project management are used as signal posts for a project's start or end date, external reviews or input, budget checks, submission of a major deliverable, etc.
- **Deliverables:** These are project results delivered to customers.

Basic Concept In Project Management

- **Responsibilities of project manager include**:
  - initiating, planning, executing and closing-down the project
- **Skills of project manager includes:**
  - Leadership
  - Management (resources, materials, funding)
  - Customer relationships
  - Technical problem solving
  - Conflict management
  - Customer relationship Management
  - Team management
  - Risk and change management

Project Management Process

- Project management is the controlled process of initiating, planning, executing and closing-down a project.
- Project Management Process Mainly Consists four Phases:
  1. Initiation
  2. Planning
  3. Execution
  4. Closing down



Project Management Process

- **Phase 1:Project Initiation**
  - It is the phase where activities are performed to assess the size, scope and the complexity of the project & to establish procedures to support later project activities.
  - **Activities in Project Initiations are:**
    1. Establish project initiation team
       - Size: identify project team member who will work within the project
    2. Establish relationship with customer
       - Develop good work relationship & trust between customer or users of the project & the IS development group (project team) before the project
       - Helps customers to understand their problems they might face & propose improvement
    3. Establish project initiation plan
       - Define the scope (objectives) of the project (even objectives are fuzzy)
       - Assign objectives to project team members
       - Define roles of each member in the project

4. Establishing management procedure
   - Concern procedures to manage the project such as communication and reporting procedures,
   - Communication procedure
   - Funding and billing procedure
   - Conflict management
   - Regulatory procedure
   - Procedures exist or need to be created
5. Establishing project management environment and project workbook
   - Establish an environment that includes data related project, called workbook
   - Performing the workbook is one important task of project manager

Project Workbook: It is a collection of templates that allows project managers, team members, sponsors and stakeholders to easily track and monitor project activities.

Phase 2-Project Planning

- It consists to define clear discrete activities and the work needed to complete the activities within a single project (identify time, input and output)
- Project planning is different than Information System Planning (ISP)
- ISP focuses on assessing the information system needs of the entire organization
- Following activities involves in project Planning Phases:
  1. Describing project scope, alternatives and feasibility
  2. Dividing the project into manageable tasks and logical order
  3. Estimating resources and creating a resources plan
  4. Developing a preliminary schedule
  5. Developing a communication plan
  6. Determine project standards and procedures
  7. Identifying and assessing risks
  8. Creating a preliminary budget
  9. Developing a statement of work (for customer)
  10. Setting a baseline project plan

1. Describing project scope, alternatives and feasibility

- Identify project scope through answering question such as
  - What problem or opportunity does the project address?
  - What quantifiable results to be achieved
  - How success will be measured
  - What criteria to be used in order to ensure the project is completed?
  - How will we know when we are finished?
- Identify alternatives solution for current business problem
- Assess feasibility of solutions
- Make decision about the planed solution

2. Dividing the project into manageable tasks and logical order

– Is called “work breakdown” structure

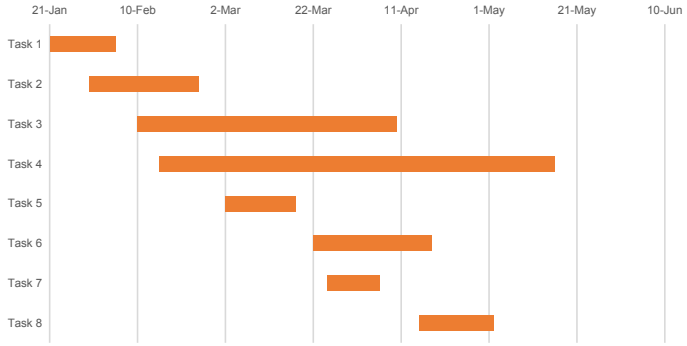
– Require to decompose SDLC phases into activities and activities in to tasks

– Each activities involves many tasks. For example:

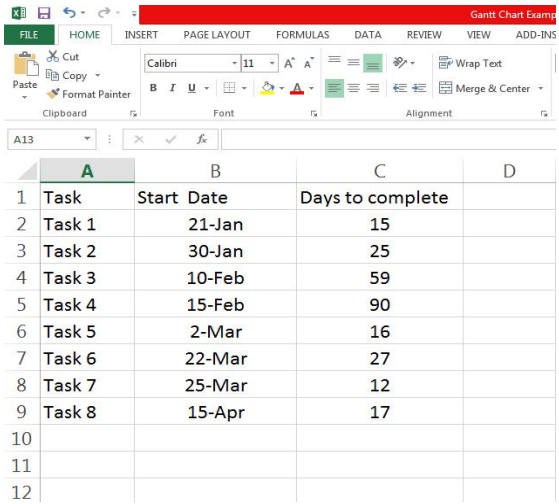
- Activities = develop data and process flow.
- Tasks = interviewing manager, identifying process and data inflow, outflow, and transformation, etc.

Gantt Chart

- A Gantt Chart is a horizontal bar chart used in project management to visually represent a project plan over time.



Gantt Chart



	A	B	C	D
1	Task	Start Date	Days to complete	
2	Task 1	21-Jan	15	
3	Task 2	30-Jan	25	
4	Task 3	10-Feb	59	
5	Task 4	15-Feb	90	
6	Task 5	2-Mar	16	
7	Task 6	22-Mar	27	
8	Task 7	25-Mar	12	
9	Task 8	15-Apr	17	
10				
11				
12				

3. **Estimating resources and creating a resources plan**
   - Estimate resource requirements :
     - how much manpower, money, software tools, are required to complete the project?
   - Resources planning is the estimation of the resource, within each activity, needed to complete the project
   - Time allocated to tasks depend on people assignment to tasks
   - Remark: a person could be assigned to more than one tasks in his own area of expertise
4. **Developing a preliminary schedule**
   - Here we 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.
   - This schedule may be drawn using the Grant and the PERT (Program Evaluation and Review Technique)/Network chart
5. **Developing a communication plan**
   - Who are the stakeholders for this project?
   - What information does each stakeholder need?
   - When and how different roles will communicate
   - When and how written and oral presentation will be provided by the team
   - How many deliverables (official reports) and when should be written (set deadlines)
   - What communication medium will be most effective for delivering this information to the stakeholder?
   - Define agenda for meetings and set deadlines (small & big meetings)

6. **Determine project standards and procedures**
   - How standard SDLC must be modified?
   - What case tools to use?
   - What approaches will be used (JAD, prototyping, etc.)?
   - How different team will report (horizontal or vertical)?
7. **Identifying and assessing risks**
   - Identify sources of risks and their expected impact
   - Source: use of new technology, resistance to change, availability of insufficient resources, team member inexperience
8. **Creating a preliminary budget**
   - Estimate project cost of the and some times the revenue of the project (show revenue is great than cost)
9. **Developing a statement of work (for customer)**
   - Is a short description of all work to be done & expected deliverables
   - Give clear idea to all project team and customer about the project size
10. **Setting a baseline project plan and set Milestone with a review meeting**
    - Contains resources, times and manpower (resource requirement)
    - Is used to guide the executing phase or to update it when change happen
    - Propose modification and update if needed ->back to previous activities

Some components of Project Planning

- **Statement of Work (SOW)**
  - "Contract" between the IS (Information System) staff and the customer regarding **deliverables** and time estimates for a system development project
- **The Baseline Project Plan (BPP)**
  - Contains estimates of scope, benefits, schedules, costs, risks, and resource requirements
- **Preliminary Budget**
  - Cost-benefit analysis outlining planned expenses and revenues
- **Work Breakdown Structure (WBS)**
  - Division of project into manageable and logically ordered tasks and subtasks
- **Scheduling Diagrams**
  - Gantt chart: horizontal bars represent task durations
  - Network diagram: boxes and links represent task dependencies

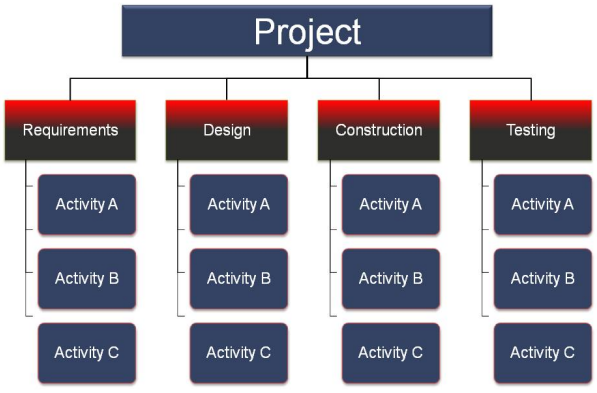
WBS(Work Breakdown Structure)

- A work breakdown structure (WBS), in project management and systems engineering, is a deliverable-oriented decomposition of a project into smaller components. A work breakdown structure is a key project deliverable that organizes the team's work into manageable sections.
- Dividing complex projects to simpler and manageable tasks is the process identified as Work Breakdown Structure (WBS).
- Usually, the project managers use this method for simplifying the project execution.
- In WBS, much larger tasks are broken down to manageable chunks of work. These chunks can be easily supervised and estimated.

WBS:Work Breakdown Structure

- Following are a few reasons for creating a WBS in a project:
  - Accurate and readable project organization and structure.
  - Accurate assignment of responsibilities to the project team.
  - Indicates the project milestones and control points.
  - Helps to estimate the cost, time and risk.
  - Illustrate the project scope, so the stakeholders can have a better understanding of the same.
- **Construction of a WBS**
  - The root of the tree is labelled by the problem name.
  - Each node of the tree is broken down into smaller activities that are made the children of the node.
  - Each activities is recursively decomposed into smaller sub-activities until at the leaf level.

WBS Structure of MIS Problem



Phase-3 : Executing the project

- It consists to put the planned baseline project into action
- Activities during executing phase:

1. Executing the base line project plan

- This means that we 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.
- Motivate people and increase the work team; think as one member

2. Monitoring the project progress against the actual progress work

- Enable modifications to current plans when needed
- Adjust resources, budget, and time to activities
- Evaluate efficiency of project team member
- E.g. if one dependence activity is changed, you have to undertake the impact on the other related activities

Phase-3 : Executing the project

3. Managing change to the baseline project plan

- Manage the change if it has to occur
  - Slipped completion dates
  - Changes in personnel
  - New activities
  - spoiled activities
- Managing change requires three steps
  - Request a change
  - Accept change
  - Apply change order
- PERT and Gantt charts can be used to undertaking changes

4. 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 (to take in, fit into, or become similar) project tasks quickly.
- It explains why design decisions were made and is a primary source of information for producing all project reports.

Phase-3 : Executing the project

5.Communicating the project status

- Keep involved roles informed about the latest development of the project
- There are three types of communication (communication is useful for)
  - Solving issues through oral presentation
  - Informing others through e-mails
  - Keeping permanent storage of records (of written documents)
- **Communication methods example:**
  - Project workbook
  - Meetings
  - Seminars and workshops
  - Newsletters and status reports
  - Specification documents
  - Minutes of meetings
  - Bulletin boards
  - Memos
  - Brown Bag launches and hallway discussions

Phase-4 : Closing down the project

- It is consists to bring the project to an end
- When does a project end?
  - If requirements have been all met (normal or natural end) or project stop(unnatural end)
  - If all objectives have been successfully achieved
  - Customers' need are not any more valid in the customer business environment; state-of-the-art technology is available on the market)
  - Running out of money
- Closing-down the project
  - Inform all members about the project end during a review meeting
  - Personnel Appraisal
  - Conducting post-project review
    - Set a review meeting with management and customers to assess project' strengths and weakness
    - Develop new idea for new projects
  - Closing the customer contract
    - Stop funding and further new projects

Representing and Scheduling Project Plans

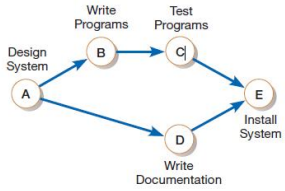
- A project manager has a wide variety of techniques available for depicting (to represent or show something in a picture or story) and documenting project plans.
- These planning documents can take the form of graphical or textual reports, although graphical reports have become most popular for depicting project plans.
- The most commonly used methods are **Gantt charts** and **network diagrams**.
- 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 (representing) relatively simple projects.
- 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.

Gantt Charts

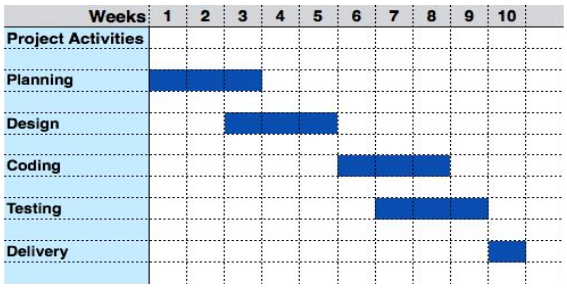
- Gantt chart was invented by a mechanical engineer named Henry Gantt in 1910. Since the invention, Gantt chart has come a long way.
- Gantt chart is a type of a bar chart that is used to allocate resources to activities including staff, hardware and software.
- Gantt charts allow project managers to track the progress of the entire project. Through Gantt charts, the project manager can keep a track of the individual tasks as well as of the overall project progression.
- In addition to tracking the progression of the tasks, Gantt charts can also be used for tracking the utilization of the resources in the project. These resources can be human resources as well as materials used.
- The bars in the Gantt are drawn along a time line and the length of each bar is proportional to the duration of time planned for the corresponding activity.
- The White Part in Gantt chart shows the slack time(latest time by which the task must be finished) and the shaded part shows the length of the time each task is estimated to take.
- Advantages and Disadvantages:
  - ability to grasp the overall status of a project and its tasks at once
  - helps to identify and maintain the critical path of a project schedule
  - For large projects, the information displayed in Gantt charts may not be sufficient for decision making.
  - it does not elaborate on the project size or size of the work elements

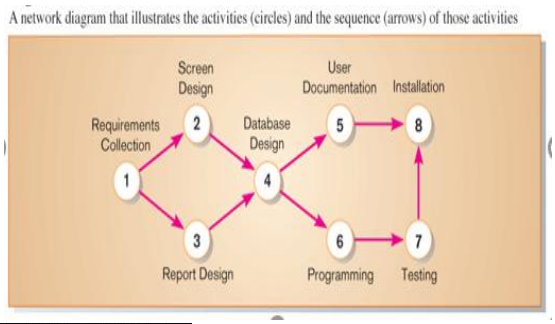
Network Diagrams

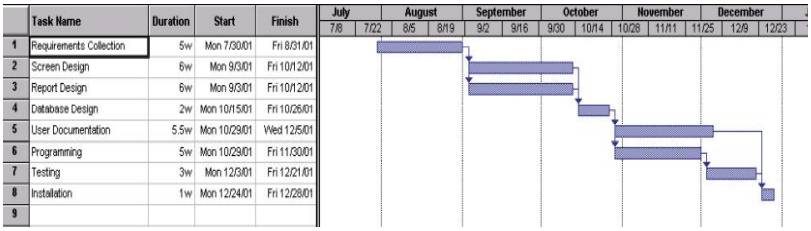
- A network diagram is a visual representation of the workflow of a project.
- A network diagram is a chart that is populated with boxes noting tasks and responsibilities, and then arrows that map the schedule and the sequence that the work must be completed.
- Network diagramming is a **critical path 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**.
- Network Diagrams are mainly used for:
  - Planning the structure of a project
  - Coordinating updates or changes automatically
  - Identifying potential risks or bottlenecks of a project
  - Documentation for external communication
  - Keeping track of project progress
  - When proposing changes to stakeholders



Gantt Chart and Network Diagram







	Task Name	Duration	Start	Finish
1	Requirements Collection	5w	Mon 7/30/01	Fri 8/31/01
2	Screen Design	6w	Mon 9/3/01	Fri 10/12/01
3	Report Design	6w	Mon 9/3/01	Fri 10/12/01
4	Database Design	2w	Mon 10/15/01	Fri 10/26/01
5	User Documentation	5.5w	Mon 10/29/01	Wed 12/5/01
6	Programming	5w	Mon 10/29/01	Fri 11/30/01
7	Testing	3w	Mon 12/3/01	Fri 12/21/01
8	Installation	1w	Mon 12/24/01	Fri 12/28/01
9				

Difference between Gantt Chart and Network Diagram

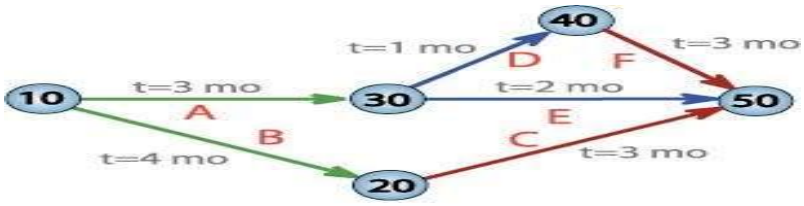
Gantt Chart	Network Diagram
Gantt Chart is a Pictorial **Representation of Project Schedules**	Network Diagram is a Pictorial **Representation of Project Work-flow**
Focus is on Tasks and Time Management	Focus is on Milestones **Work Sequential Order**
In-depth Plan to understand of Project Tasks	High-level Plan to Understand the Project Work-flow
Project Time lines are Clearly Defined. For example, 1-Jan-2017 to 25-July-2017 for Task A	Estimated Time frames are defined. For example, 4 days, 2 weeks, 1 month for Milestone A.
Effective Resource Allocation is can be done using Gantt Format	It is not possible using Network Diagram
Project Progress is captured	Can not Capture the % Progress
Project Status is updated. Like New, Open, In progress, Completed, On hold	We can not Updated Project Status
Project Complexity Details recorded. For Example, Level 2, Level 2, Level 3.	We will not record project complexity.
We can add Task description and Remarks for each task and update any time if required	We can not enter remarks or task description here
Easy to modify and edit the information	Difficult to change the information
Stacked Bar Chart is used to Create Gantt Chart	Process Flow Chart diagrams are used to create Network diagram

PERT(Program Evaluation and Review Techniques)

- PERT (Program Evaluation and Review Technique) is a project management tools used to schedule, organize and coordinate tasks within a project.
- PERT was initially created by the US Navy in the late 1950s. The pilot project was for developing Ballistic Missiles and there have been thousands of contractors involved.
- PERT chart consists of networks of boxes and arrows. The boxes represent activities and the arrows represents task dependencies.
- PERT chart represents the statistical variations in the project estimates assuming normal distribution. So instead of making single estimates for each task ,pessimistic(p),optimistic(o) and most likelihood(m) estimates are made. That is
  - Time Estimate(TE)=$\frac{(to+4*tm+tp)}{6}$
  - Where optimistic time(o) means the minimum possible period of time for an activity to be completed
  - Pessimistic time(p) means the maximum possible period of time for an activity to be completed
  - Most likelihood or realistic time(m) is the project manager's best guess of the amount of time that the activity actually will require for the completion

PERT(Program Evaluation and Review Techniques)

- Same as most of other estimation techniques, PERT also breaks down the tasks into detailed activities.
- PERT consists of Milestones and Activities.
- Advantages:
  - Simple graphical representations helps to show the task interrelationships.
  - It has the ability to highlights the project's critical path and slack time.
  - It also shows which tasks must be completed before other are begun.
  - It exposes all possible parallelism in the activities and thus help in allocating resources .
- Limitations:
  - Project task has to be clearly defined as well as their relationships
  - It does not deal very well with task overlapping case.
  - It does not help in deciding which activities are necessary or how long each will take.



PERT network chart for a seven-month project with five milestones (10 through 50) and six activities (A through F)

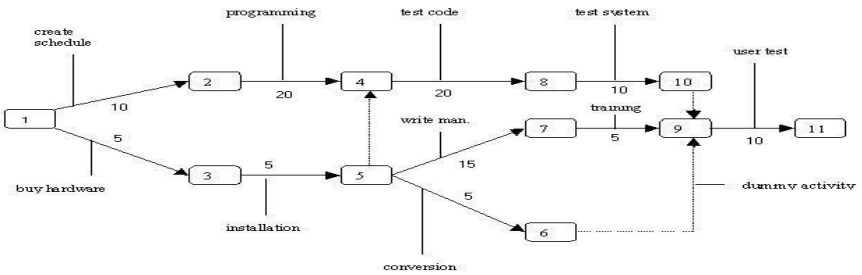


Fig. 1: PERT Chart

- Numbered rectangles are nodes and represent events or milestones.
- Directional arrows represent dependent tasks that must be completed sequentially.
- Diverging arrow directions (e.g. 1-2 & 1-3) indicate possibly concurrent tasks
- Dotted lines indicate dependent tasks that do not require resources.

CPM(Critical Path Methods)

- CPM allows us to monitor the achievements of the project goals and help to find the remedial actions need to be taken to get the project back on course.
- Critical path is the sequential activities from start to the end of a project i.e. the path of longest duration as determined on the project network diagram(total duration of the project).
- In the critical path method, the critical activities of a program or a project are identified. These are the activities that have a direct impact on the completion date of the project.
- Although many projects have only one critical path, some projects may have more than one critical paths depending on the flow logic used in the project.
- If there is a delay in any of the activities under the critical path, there will be a delay of the project deliverables.
- Most of the times, if such delay is occurred, project acceleration or re-sequencing is done in order to achieve the deadlines.

CPM(Critical Path Methods)

- Critical path method is based on mathematical calculations and it is used for scheduling project activities. This method was first introduced in 1950s as a joint venture between Remington Rand Corporation and DuPont Corporation.
- The critical task will have starting time and finishing time and task not in critical path can have flexibility of starting and finishing the task known as slack time or float time.
- Critical Path have four key elements:
  a. **Critical path analysis:** find the path with longest duration among different activities from start to finish.
  b. **Analyze the Earliest Start Time(EST) and Latest Finish Time (LFT)**
     EST→how fast/early an activity can start
     LFT→ Maximum time available within which the project must be finished
  c. **Calculate Earliest Finish Time(EFT) and Latest Start Time(LST) where**
     EFT = EST + activity duration
     LST = LFT – activity duration
  d. **Float determination:** float is an amount of time an activity can delayed before it causes whole project delayed that is slack.(no slack in CP)
     Total Float = LST – EST / LFT – EFT
     The path with total float =0 is the critical.

Basis for Comparison	PERT	CPM
Meaning	PERT is a project management technique, used to manage uncertain activities of a project.	CPM is a statistical technique of project management that manages well defined activities of a project.
What is it?	A technique of planning and control of time.	A method to control cost and time.
Orientation	Event-oriented	Activity-oriented
Evolution	Evolved as Research & Development project	Evolved as Construction project
Model	Probabilistic Model	Deterministic Model
Focuses on	Time	Time-cost trade-off
Estimates	Three time estimates	One time estimate
Appropriate for	High precision time estimate	Reasonable time estimate
Management of	Unpredictable Activities	Predictable activities
Nature of jobs	Non-repetitive nature	Repetitive nature
Critical and Non-critical activities	No differentiation	Differentiated
Suitable for	Research and Development Project	Non-research projects like civil construction, ship building etc.
Crashing concept	Not Applicable	Applicable

Using Project management software

- It's a tool to help us plan, organize, and manage our team's work, from start to finish.
- A wide variety of automated project management tools is available to help us 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
  - easily modify tasks and resources.

Using Project management software

- The top 5 key functionality aspects of project management software are:

1. **Task lists** – being able to assign and update the status of tasks so that everyone in your team is on the same page is critical
2. **Schedules** – many tools offer calendars, Gantt charts or milestone tools that help you understand where a task fits into the project as a whole and how much time there is to complete it.
3. **File sharing** – being able to share and organise key project documents eliminates time wasted searching for files
4. **Communication** – this is critical in project management as a smooth flow of communication means quick and easy problem solving
5. **Reporting** – this is important for all team members when it comes to updating themselves on the project as a whole. However this is also a huge plus for project managers who want to ensure that the project is progressing and tasks are being carried out in a timely

Using Project management software

- Some commonly used software project management software are:

1. **Jira**
2. **OpenProj**
3. **ActiveCollab**
4. **Bugzilla**
5. **Zoho Projects**
6. **Cage**
7. **eGroupWare**