SOFTWARE ENGINEERING

1. What is software engineering? Explain Software Engineering as a Layered Technology. OR What is software engineering? Explain the need of Software Engineering. OR Define Software Engineering.(S-2018,S-2017,W-2016,W-2017,S-2016)

- Software engineering is a branch in computer science that deals with developing applications. It covers the technical part of building software systems through designing, implementing, and modifying software. It also covers software management issues, such as directing programming teams, scheduling, and budgeting.
- Software engineering may be defined as the systematic design and development of software products and the management of the software process.

NEED OF SOFTWARE ENGINEERING

- Software engineering is methodology provides the framework that guides engineers to developing the software. This framework defines different phases of software development such as requirements analysis, designing, implementation, testing, maintenance etc.
- Software engineering describes how the software is designed, what types of their requirements and what types of different phases are needed to complete the software product.
- The main goal of software engineering is to provide process models that produce well-documented, maintainable software.
- Using process model we can determine in advance how much time, cost and efforts are required to produce the final complete product.
- The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process.
- SDLC includes requirement phase, design phase, implementation phase, testing phase and maintenance phase.
- Different process models are used in different condition and situations. This process model specifies a general process, set of stages in which a project should be divided, the order in which the stages should be executed.

SOFTWARE ENGINEERING: A LAYERED TECHNOLOGY

- Software engineering is a layered technology.
- Various layers are listed below:

Quality

- The main focus of software engineering is to develop quality product.
- Quality of product refers to the characteristics that engineer specify for the product.
- In software development, quality of product refers to the output meets the functions and features specified in the requirement model.

Process

- The base layer is process layer. It is the heart of the software engineering approach.
- Software process is a set of activities together with ordered and performed properly to produce the desired result.
- Main objective of this layer is to deliver software in time.

Method

- The layer after process is method. It describes how to build the software product.
- Method includes different tasks such as user communication, requirement analysis, design modeling, coding, testing and maintenance.
- Method gives the exact way to build the software.
- Method is a way to execute processes in proper manner.

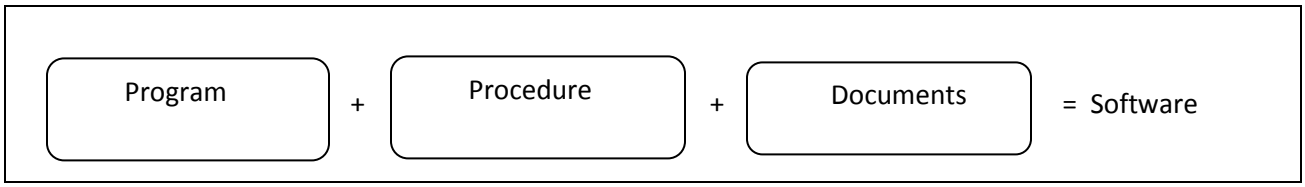
Tools

- Software engineering tools provide automated and semi-automated support to method and process. When tools are integrated the information created by one tool can be used by another.
- Computer aided software engineering (CASE) combines software, hardware, and software engineering database (information about analysis, design, program construction and testing) to create a software engineering environment.

SOFTWARE

2. What is Software? Explain characteristics of Software. OR
How the software product is differing than hardware product? OR
Explain: "Software doesn't wear out, but it does deteriorate."(S-2018,W-2017,W-2015)

- Software is a set of programs, which is designed to perform a well defined function.
- Software consists of lines of code written by computer programmers.



CHARACTERISTICS OF WELL ENGINEERED SOFTWARE

1. Software is engineered, not manufactured like hardware

- In the case of software, a good design will introduce good software. In the manufacturing of hardware may introduce quality problem and this does not exist in the case of software.
- Once hardware is manufactured it is not possible to correct it, change it, and enhance it without temper of hardware. But software can easily.
- In case of hardware numbers of copies generate a cost due to raw material and other manufacturing process but in case of software you can create number of copies of software.

2. Software does not wear out

- The "bathtub curve" shown in Figure 1 shows failure rate of hardware as a function of time.
- In first phase failure rate is much more. But after testing and fixing bugs failure rate will come down and stable after certain time. Second phase is useful life phase of hardware component where failure rate is low and constant. As time passes hardware component suffer from effects of dust, vibration, misuse, temperature and many other environmental effects. So the failure rate rises. This is the wear out of hardware.

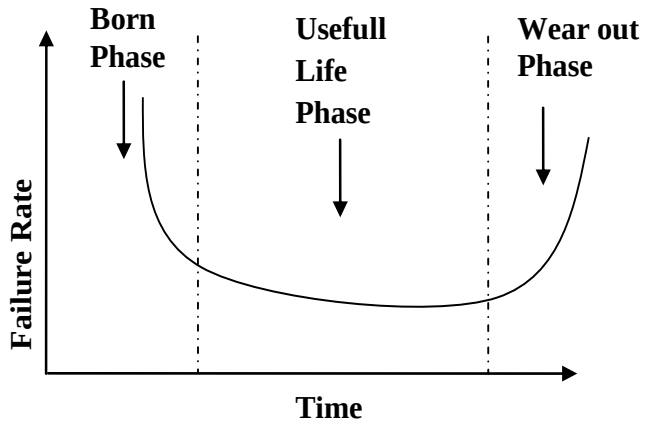


Figure 1

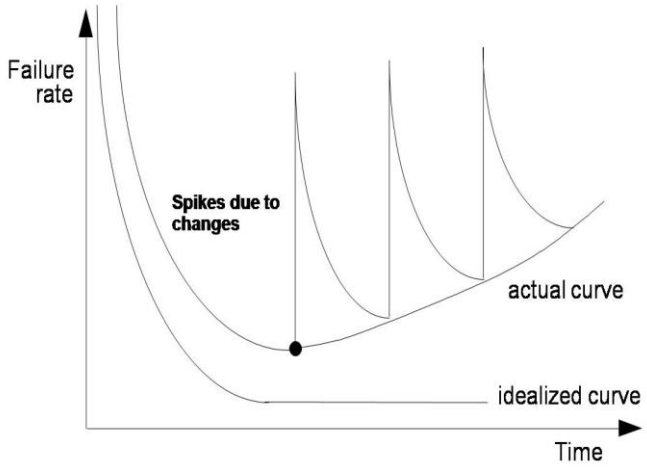


Figure 2

- Undiscovered errors will cause high failure rates early in the life of a program. However, these are corrected and the curve becomes flat as shown. The "Idealized curve" is shown in Figure 2.
- During its life, software will change. As changes are made, some new defects will be introduced, causing the failure rate curve to spike. The "Actual curve" is shown in Figure 2.
- When hardware component wears out, it is just replaced by spare parts. There is no spare part to replace in software component. It requires changes in existing product or component.

3. Software gives component-based construction, it gives reusability of components

- A software component should be designed and implemented so that it can be reused in many different programs.
- Graphical User Interfaces are built using reusable components that enables the creation of graphics window and animated menus.
- Reuse of components from libraries help in reduction of effort.

SOFTWARE MYTHS

3. Explain Software Myths OR Discuss any two software Myths.(S-2018,W-2017)

- Software myths propagate false beliefs and confusion in the minds of management, users and developers.

Management Myths

Myths	Reality
✓ Members of the organization know all the information containing procedures, principles and standards.	✓ Developers not know about standards. ✓ Standards are incomplete and outdated. ✓ Developers not follow standards.
✓ If the project is behind schedule,	✓ New workers take long time to learn about the

increasing the number of programmers can reduce the time gap.	project. So further delays the project.
✓ If the project is outsourced to a third party, the management can relax because third party develop software.	✓ If organization does not understand how to manage and control software project. It will suffer when it outsources the software projects.

User Myths

Myths	Reality
✓ Brief requirement is enough to start development; detailed requirements can be added at the later stages.	✓ Incomplete requirements lead to software failure. ✓ Adding requirements at a later stage repeating the development process.
✓ Software is easy to change because software is flexible.	✓ Changes at later stage require redesigning and extra resources.

Developer Myths

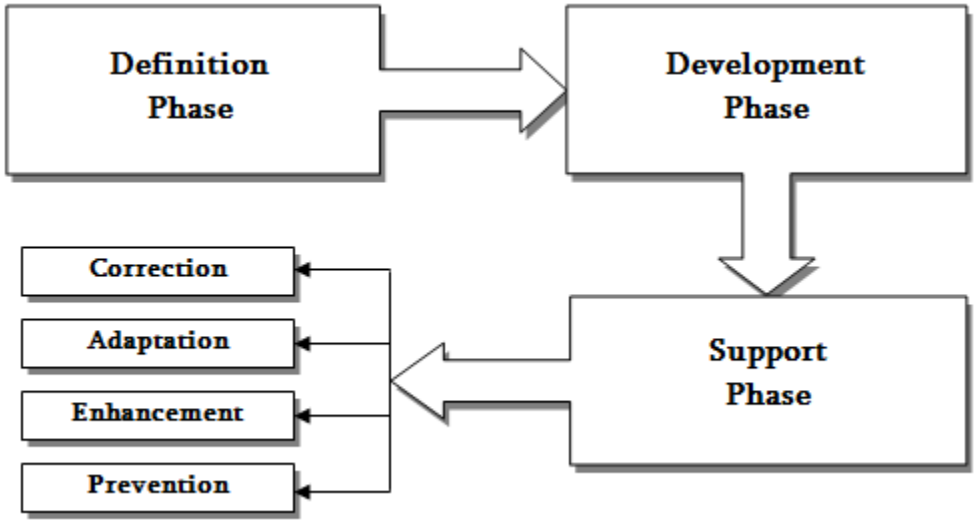
Myths	Reality
✓ Once the program is written, software can be called complete.	✓ More than 60% of the efforts are required after the software is delivered to the user.
✓ Software quality can be measure only after the program is executed.	✓ Using quality assurance quality of software measured during any phase of development process.
✓ The product, which is delivered after the project's completion can be called working program.	✓ Successful project includes not only the working program but also the documentation to guide the users for using the software.
✓ Unnecessary documentation slows down the project.	✓ Proper documentation increase quality which results in reducing the amount of rework.

SOFTWARE PROCESS

- Software process means methods of developing software. A software process is a set of activities, together with ordering among them, such that if the activities are performed properly, the desired result is produced.

4. Generic View of Software Engineering OR Explain three phases of generic view of a software engineering.

- The work associated with software engineering can be categorized into three generic phases.



Definition Phase

- The definition phase focuses on "what".
- During definition, the software engineer identify what information is to be processed, what function and performance are desired, what system behavior can be expected, what interfaces are to be established, what design constraints exist, and what validation criteria are required to define a successful system.

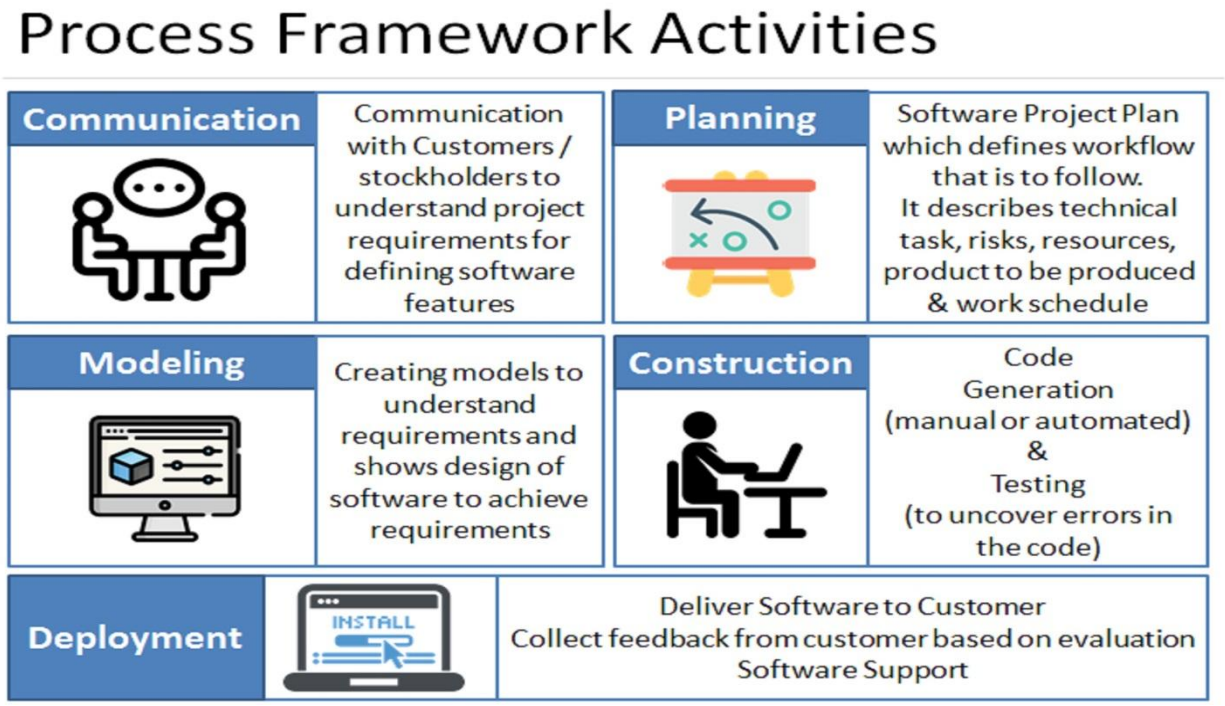
Development Phase

- The development phase focuses on "how".
- During development a software engineer define how data are to be structured, how function is to be implemented, how interfaces are to be categorize, how the design will be translated into a programming language, and how testing will be performed.

Support Phase

- The support phase focuses on "change" associated with error correction.
- Types of change are associated with support phase:
  - ✓ **Corrective maintenance**: Correcting errors that were not discovered during the product development phase.
  - ✓ **Perfective maintenance:** Improving and enhancing the functionalities of the system according to the customer's requirements.
  - ✓ **Adaptive maintenance:** Porting the software to work in a new environment. For example, porting may be required to get the software to work on new operating system.

5. Explain Process Framework Activity. OR Generic framework activities (S-2017)

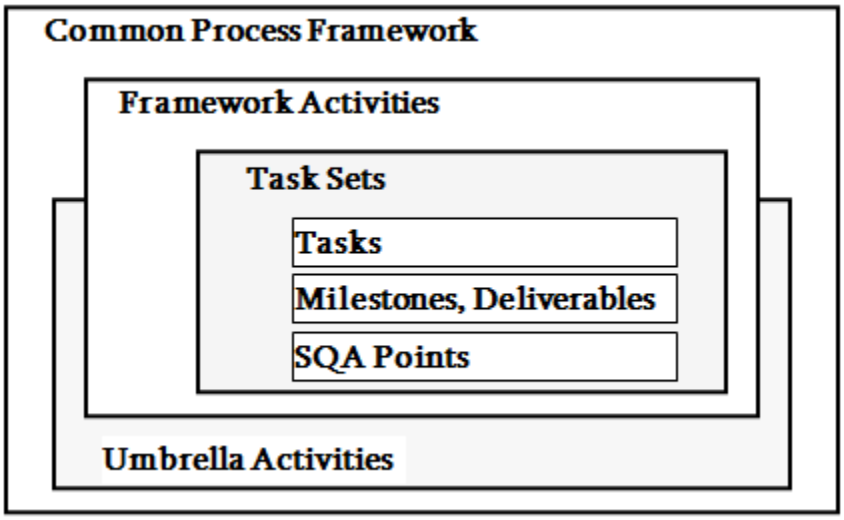


Generic Framework Activities OR Process Framework Activity.

- Process model define a set of framework activities that are always applicable to any type of project is called generic framework activities.
- There are five process framework activities:
  - ✓ **Communication:** This activity involves communication with customers to gather requirements and other related activities.
  - ✓ **Planning:** it required to define resources, timelines, describing technical and management risks.
  - ✓ **Modeling:** A model will be created to better understand the requirements and design to achieve these requirements.
  - ✓ **Construction:** Here the code will be generated and tested.
  - ✓ **Deployment:** Here, a complete or partially complete version of the software is represented to the customers to evaluate and they give feedbacks based on the evaluation.
- These above described five activities can be used in any kind of software development.
- Each framework activity contain number of task set that identifies work task that are to be completed, milestones that will be used to indicate progress, deliverables that will be produced and quality assurance points that will be required to get quality product at each stage.

UMBRELLA ACTIVITIES

6. Explain Umbrella activities with diagram. OR What is Umbrella Activities(S-2018,W-2017,S-2017,S-2016)



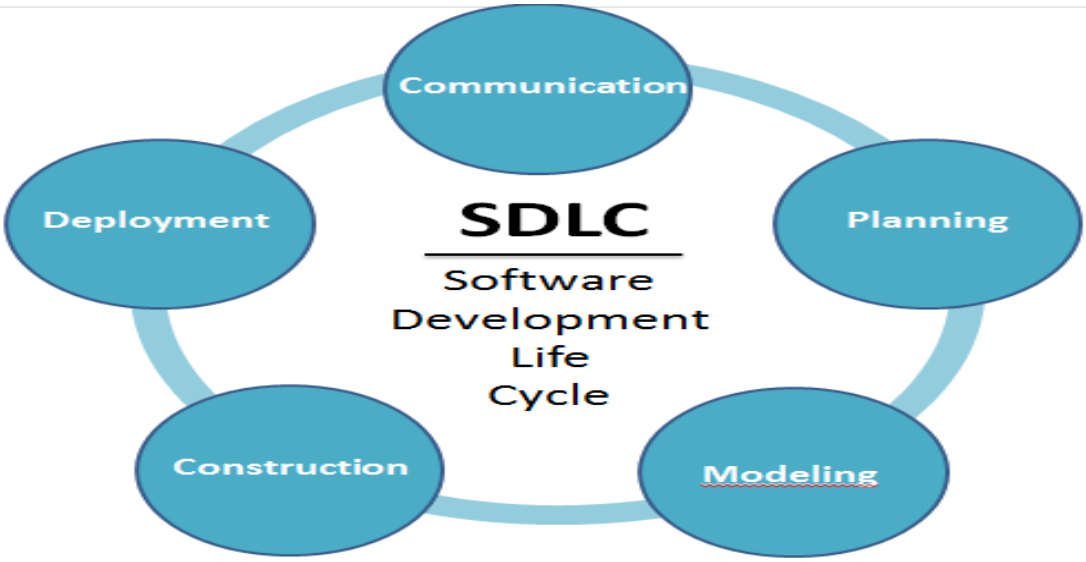
Umbrella Activities

- Umbrella activities are independent of anyone framework activity.
- Typical activities in this category include:
  - ✓ **Project tracking and control**: allows the team to track progress against the project plan and take necessary action to maintain schedule.
  - ✓ **Risk Management**: Identify the risks that may affect the outcome of the project or the quality and manage that risk.
  - ✓ **Software quality assurance**: It defines the activities to ensure the software quality.
  - ✓ **Formal Technical Review**: The FTR (Formal Technical Review) is software quality assurance activities to remove errors before they go to the next activity.
  - ✓ **Software configuration management**: Software configuration management (SCM) is the required to controlling the changes that take place during development.
  - ✓ **Reusability management**: It defines criteria for work product reuse and establishes mechanism to achieve reusable components.
  - ✓ **Work product preparation and production**: It create work products such as models, documents, etc.
  - ✓ **Measurement**: This will include all the measurement of every phase of the software project.

SOFTWARE DEVELOPMENT LIFE CYCLE MODEL (SOFTWARE PROCESS MODELS)

7. Explain Software Development life cycle (W-2017,S-2017)

- A software life cycle model (also called process model) is a descriptive and diagrammatic representation of the software life cycle.
- A life cycle model represents all the activities required to make a software product.
- Software process models are adjusted to meet the need of software engineers and managers for specific project.



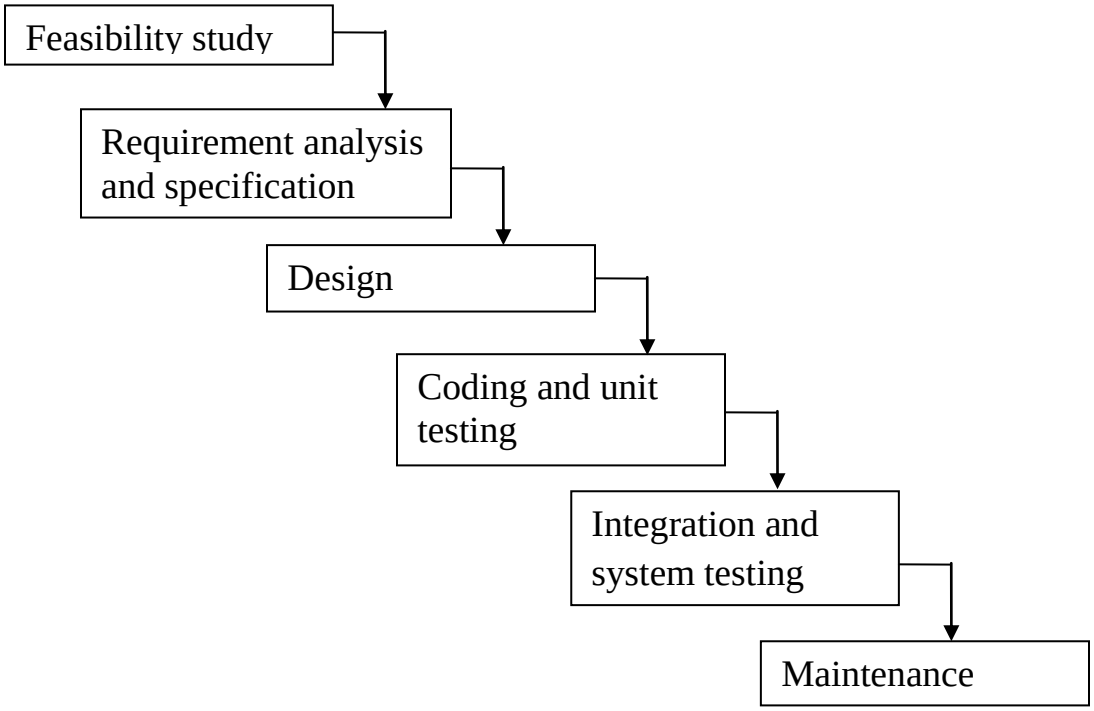
- ✓ **Communication:** This activity involves communication with customers to gather requirements and other related activities.
- ✓ **Planning:** it required to define resources, timelines, describing technical and management risks.
- ✓ **Modeling:** A model will be created to better understand the requirements and design to achieve these requirements.
- ✓ **Construction:** Here the code will be generated and tested.
- ✓ **Deployment:** Here, a complete or partially complete version of the software is represented to the customers to evaluate and they give feedbacks based on the evaluation.

DIFFERENT PROCESS MODELS:

- Waterfall model (linear sequential model)
- Incremental process model
- Prototyping model
- The spiral model
- Rapid application development model (RAD model)

CLASSICAL WATERFALL MODEL

8. Define SDLC model. Provide a diagram representing waterfall model. Also, explain various phases of waterfall model.(S-2017, S-2016,W-2015)



Feasibility study

- The main aim of feasibility study is to determine whether it would be financially and technically feasible to develop the product.
- At first project managers or team leaders study different input data to the system and output data to be produced by the system. The feasibility study concentrates on the following area.

✓ **Operational Feasibility:** Operational feasibility study tests the operational scope of the software to be developed. The proposed software must have high operational feasibility. The usability will be high.

✓ **Technical Feasibility:** The technical feasibility study compares the level of technology available in the software development area and the level of technology required for the development of the product. Here the level of technology consists of the programming language, the hardware resources, other software tools etc.

✓ **Economic Feasibility:** The economic feasibility study compares the cost of the software development against the income or benefits gets from the developed system. There must be scopes for profit after the successful Completion of the project.

Requirements analysis and specification

- The aim of the requirements analysis and specification phase is to understand the exact requirements of the customer and to document them properly. This phase consists of two distinct activities, namely
  - ✓ Requirements gathering and analysis, and
  - ✓ Requirements specification
- **Requirements gathering:** The goal of the requirements gathering activity is to collect all information from the customer related to the product to be developed. This is done to clearly understand the requirements so that incompleteness and inconsistencies are removed.
- **Requirements analysis:** This activity is used to collecting all data related to the product from the users or customer through interviews and discussions.
- **Requirements specification:** During SRS activity, the user requirements are systematically organized into a Software Requirements Specification (SRS) document.

Design

- During the design phase the software architecture is derived from the SRS document.
- The customer requirements are broken down into logical modules for the implementation.
- Hardware and software requirements for every module are identified.
- Also the inter relation between the various logical modules is established at this stage.
- Algorithms and diagrams defining the objective of each logical model are developed.

Coding and unit testing (Implementation)

- The purpose of the coding and unit testing phase of software development is to translate the software design into source code.
- Each module of the design is implemented as a program module.
- Each module is unit tested for determine the correct working of all the individual modules.
- The result of this phase is a set of program modules that have been individually tested.

Integration and system testing

- Integration of different modules is done once they have been coded and unit tested.
- During the integration and system testing phase, the modules are integrated.
- All the modules have been successfully integrated and tested, system testing is carried out.
- The goal of system testing is to ensure that the developed system conforms to its requirements specifies in the SRS document.
- System testing usually consists of three different kinds of testing activities.
- **α – testing:** It is the system testing performed by the development team.
- **β – Testing:** It is the system testing performed by a friendly set of customers.

- **Acceptance testing:** It is the system testing performed by the customer himself after the product delivery to determine whether to accept or reject the delivered product.

Maintenance

- Maintenance involves following three kinds of activities:
- Correcting errors that were not discovered during the product development phase. This is called **corrective maintenance**.
- Improving and enhancing the functionalities of the system according to the customer's requirements. This is called **perfective maintenance**.
- Porting the software to work in a new environment. For example, porting may be required to get the software to work on new operating system. This is called **adaptive maintenance.**

Advantages:

- Simple to implements and manage

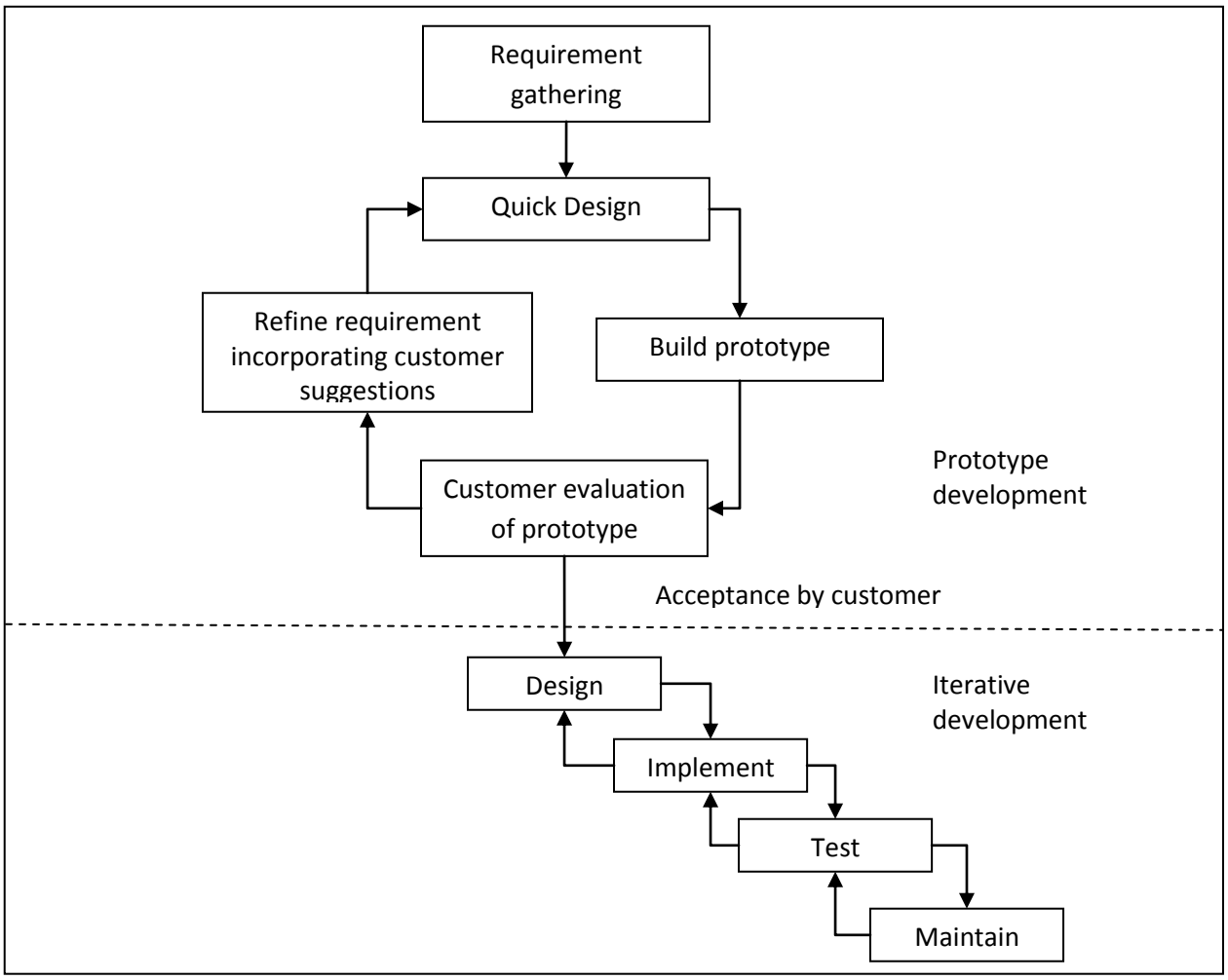
Disadvantages:

- Unable to accommodate changes at later stages, that is required in most of the cases.
- Working model not available during development. This can lead the development with major mistakes.
- Not suitable for large projects.

PROTOTYPE MODEL

9. Define SDLC model. Provide a diagram representing prototyping model. Also, explain various phases of prototyping Model. OR describe Prototyping Model and list out its disadvantages. OR When Prototype Model is used(S-2018,W-2017,S-2017,S-2016)

- A prototype is the sample implementation of the real system.
- A prototype includes limited functionality, low reliability, and inefficient performance compared to the actual software.
- An important purpose of prototype is to illustrate the input data formats, messages, reports, and the interactive dialogues to the customer.
- This model is used to understanding of the customer's requirement:
  - ✓ how the screens might look like
  - ✓ how the user interface would behave
  - ✓ how the system would produce outputs



- Prototyping model can be used when requirement are unclear to the development team.
- As shown in figure the first phase is prototype development. This is followed by an iterative development cycle. In this model prototyping start with an initial requirements gathering phase. A quick design is carried out and a prototype is built.
- The developed prototype is submitted to the customer for evaluation. Based on the customer feedback the requirements are refined and the prototype is modified.
- This cycle of obtaining customer feedback and modifying the prototype continues till the customer approves the prototype. Once the customer approves prototype the actual system is developed using the iterative waterfall approach.

Advantages

- Errors can be detected much earlier.
- User are actively involved in the system , so more accurate user requirements are obtained.

- Since in this methodology a working model of the system is provided , the users get better understating of the system being developed.

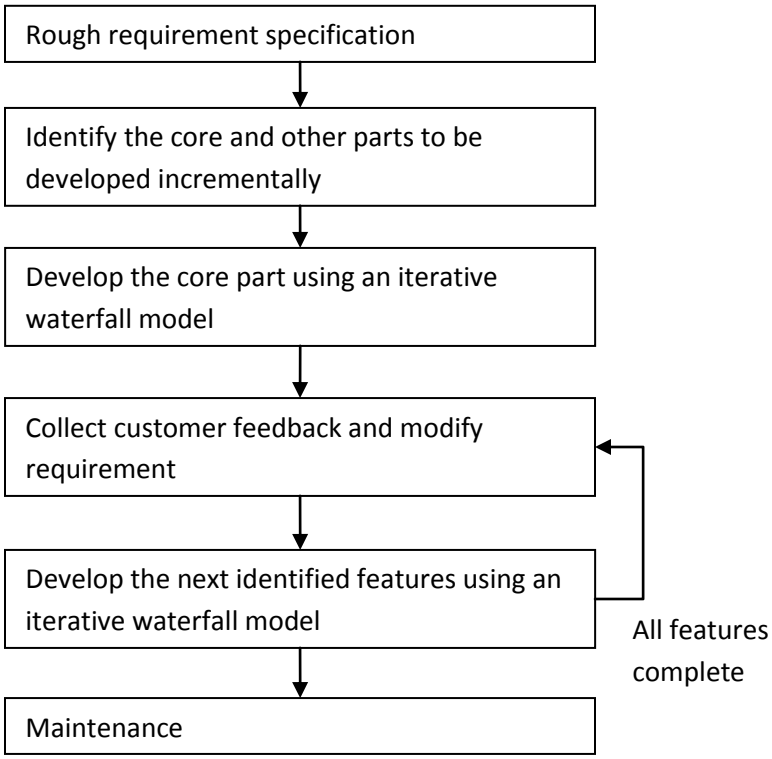
Disadvantages

- It requires extensive participation and involvement of the customer, which is not always possible.
- Leads to Implementing and then repairing way of building systems.

EVOLUTIONARY MODEL (INCREMENTAL MODEL)

10. Describe Incremental Model for Software Development with diagram. OR write a Short note on Incremental Model.(S-2018,W-2017,S-2017)

- In the evolutionary life cycle model the software requirement is first broken down into several modules (functional units).



- The development team first develops the core modules of the system.
- The core modules are those that do not need services from other modules.
- Non-core modules need services from the core modules.
- This initial product is refined into increasing levels of capability by adding new functionalities in successive version.
- Each evolutionary version may be developed using an iterative waterfall model of development.

- Each successive version of the product is fully functioning software capable of performing more work than the previous version.

Advantages

- Less cost and time required to develop first increment called core product.
- Less Risk is incurred to develop the smaller systems represented by the increments.

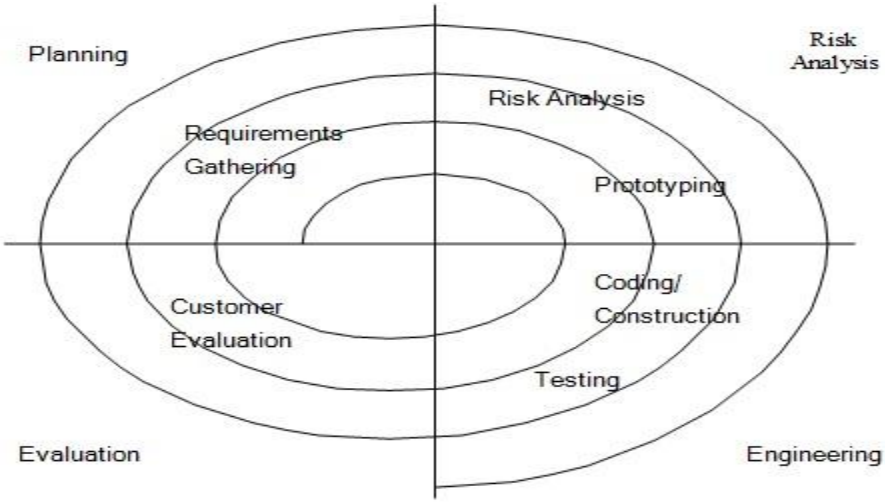
Disadvantages

- Managing the resulting cost, Schedule and Configuration complexity of all increments may exceed the capabilities of the organization.

SPIRAL MODEL

11. Describe Spiral model with necessary diagram.(W-2017,S-2017,S-2016,W-2016)

- The Spiral model of software development is shown in figure. The diagrammatic representation of this model appears like a spiral with many loops. The exact number of loops in the spiral is not fixed. Each loop of the spiral represents a phase of the software process.
- For example, the innermost loop might be concerned with feasibility study. The next loop with requirements specification, the next one with design, and so on.
- Each phase in this model is split into four sectors (or quadrants) as shown in figure. The following activities are carried out during each phase of a spiral model.



Planning

- In this phase software analyst communicate with the customer and necessary requirements are gathered.
- The objectives and alternatives of the project are determined and are documented.

Risk Analysis

- All possible alternatives, which can help in developing a cost effective project are analyzed.
- This phase identify and resolve all the possible risks in the project development.

Engineering

- In this phase all the necessary coding is done for development of software. Also testing done in this phase.
- The actual development of the project is carried out in this phase.

Evaluation

- In this phase customer evaluate the developed software.
- After evaluation if customer wants to add more feature in the software then the process of iteration is continued.

Why we use spiral model?

- For development of large scale / high risk projects.
- When cost and risk evolution is important.
- Users are unsure of their needs.
- Requirements are complex.

Advantages

- Strong approval and documentation control
- Additional functionality can be added at a later date.
- Software is produced early in software life cycle.

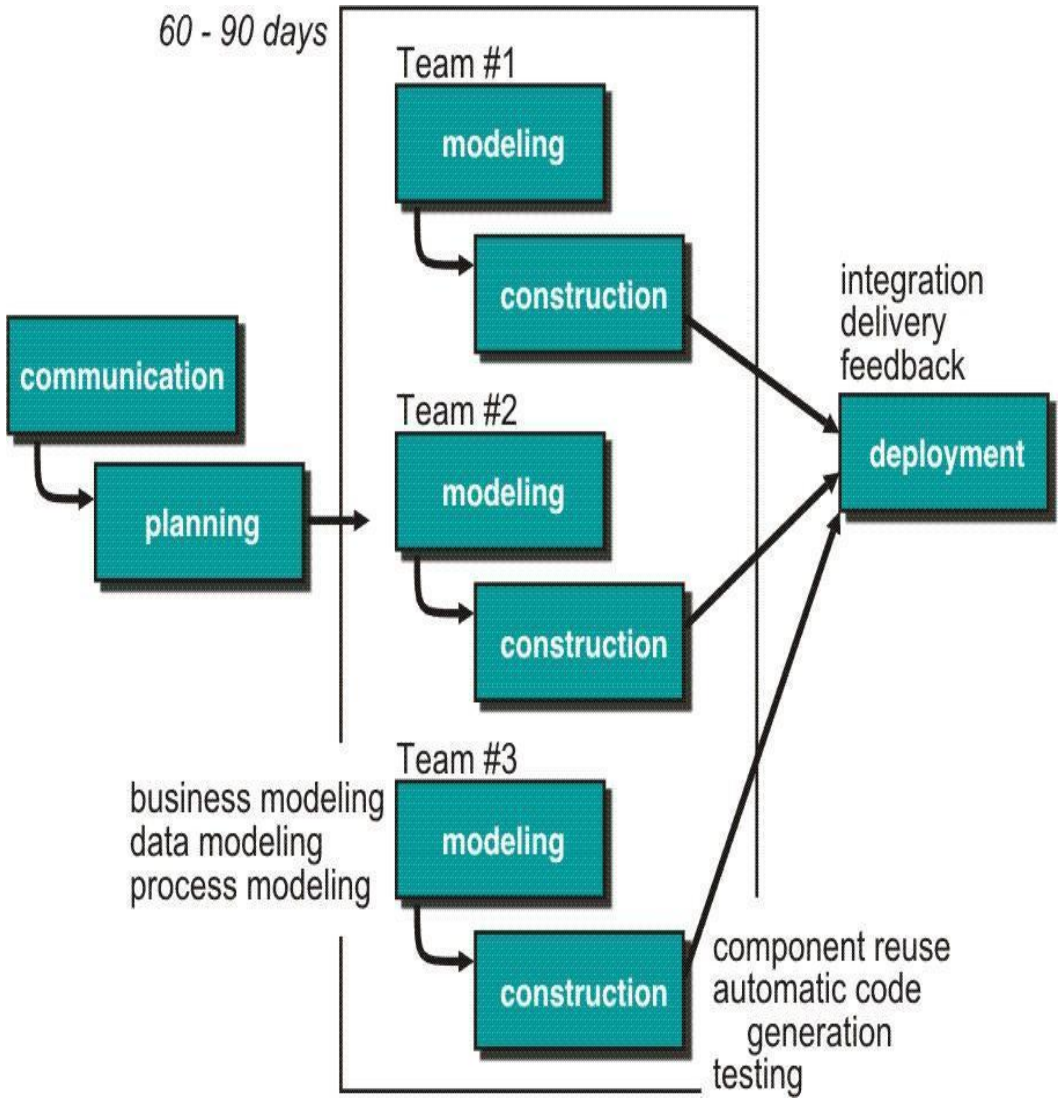
Disadvantages

- Can be costly model to use.
- Project success is highly dependent on the risk analysis phase.

RAD MODEL (RAPID APPLICATION DEVELOPMENT)

12. Explain RAD model.(W-2017,W-2016)

- Rapid Application Development (RAD) is an incremental software process model.
- It makes use of reusable software components.
- If requirements are well understood the RAD process enables a development team to create a fully functional system within a very short time period (e.g. 60 to 90 days).
- In RAD model the functional modules are developed in parallel and integrated to make the complete product for faster product delivery.
- **Communication** is an activity which works to communicate with the customer and necessary requirements are gathered.
- **Planning** is required because many software teams work in parallel on different system functions.



- **Modeling** includes three major phases -

  1. Business modeling
  2. Data modeling
  3. Process modeling

Business Modeling: The information flow is identified between various business functions.

- **Data Modeling:** Information gathered from business modeling is used to define data objects that are needed for the business. The attributes of all data objects is identified.
- **Process Modeling:** Any changes to the data object are defined in this phase. Process descriptions for adding, deleting, retrieving or modifying a data object are given.
- **Construction** focuses on the use of existing software components and the application of automatic code generation.
- **Deployment** Involves delivery of software to the customer for evaluation and feedback

Program Vs Software

13. Differentiate between program and software(W-2015)

Program	Software
Program is a sequence of logically related statements that are written to perform a specific task.	Software is a set of programs, which is designed to perform a well defined function.
Programs are developed by single user for their personal use.	Software is an application that is designed by one or more software developer.
Programs are small in size and have limited functionality.	Software products are extremely large and they have multiple users.
Author of the program himself uses his program so it is possible that there is not a proper user interface and proper documentation.	Software has multiple users and therefore it contains good user interface and good documentation.
A program consists of only program code.	Software consists of program code with specification document, design document, test document and user's manual.
A program can be developed according to the programmer's individual style of development	Software product must be developed using the software engineering principles.

2. MARKS QUESTION – ANSWER

1. **Explain System Software .(S-2018)**

   System software

   System software is a collection of programs to provide services to other programs.

   The system software area is characterized by heavy interaction with computer hardware, heavy usage by multiple users, concurrent operation that requires scheduling, resource sharing and process management.

2. **Explain Artificial Intelligence Software.(S-2018)**

   Artificial Intelligence Software

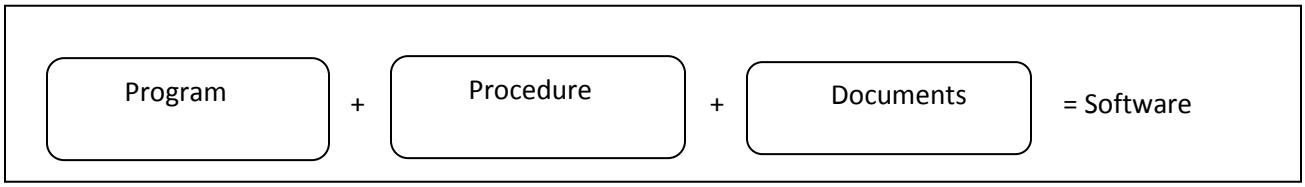
   Artificial intelligence software makes use of non numerical algorithms to solve complex problem that are not capable to computation or straight forward analysis. Examples are expert system , game playing etc.

3. **What is Software Scope.(S-2018)**

   First activity of project planning is determining software scope . software scope describes the function and features that are to be delivered to end users, the data that are input and output , the content that is presented to the user a consequence of using the software. It describes performance, constraints, interfaces and reliability of the system.

4. **Define Software(S-2018)**

   - Software is a set of programs, which is designed to perform a well defined function.
   - Software consists of lines of code written by computer programmers.



5. **Define Software Engineering.(W-2017)**

   - Software engineering is a branch in computer science that deals with developing applications. It covers the technical part of building software systems through designing, implementing, and modifying software. It also covers software management issues, such as directing programming teams, scheduling, and budgeting.
   - Software engineering may be defined as the systematic design and development of software products and the management of the software process.

6. **List Umbrella activities.(S-2016,W-2015,S-2015)**

   - Project tracking and control

- Risk Management
- Software quality assurance
- Formal Technical Review
- Software configuration management
- Reusability management
- Work product preparation and production
- Measurement

7. which are the tools used in Software Development (S-2018)

- Terminal
- Tmux
- Slack
- Chrome
- GitHub
- Stack Overflow
- Sublime Text

8. Compare Application Software and System Software(S-2018)

Application Software	**System Software**
Perform a single specific task.	Works as interface between user and hardware, perform process management , memory management, hardware device controlling and many more
Developed using JAVA , C++, C, Visual basic	Develop using C,C++,Assembly Languages
Run only when user request to run the application .	Boots up when the user switches on the computer and runs till he switches off the machine.
Cannot be installed without an operating system	Necessary for the proper functioning of the computer.
Ex: word, spreadsheet, Presentation, database management system	Ex: Windows, Unix, Linux, DOS