

MAJLIS ARTS AND SCIENCE COLLEGE PG DEPARTMENT OF COMPUTER SCIENCE

(Affiliated to the University of Calicut, approved by the Government of Kerala)

Majlis Nagar, Puramannur-P.O 676552 Malappuram Dt, Kerala.



FIFTH SEMESTER ONLINE STUDY CAMP

SCAN QR CODE TO JOIN STUDY CAMP
WHATSAPP GROUP



EXPECT TO

- *UNIT WISE REVISION
- *IMPORTANT TOPIC DISCUSSION
- *PREVIOUS YEAR QUESTION PAPER DISCUSSION
- *ASSIGNMENTS

"Get ready to exam
through online"

masc.majliscomplex.org

SOFTWARE ENGINEERING (BSc CS AND BCA)

MODULE 1

1. What you mean by software engineering?

software engineering as an engineering branch associated with the development of software product using well-defined scientific principles, methods and procedure.

2. Explain about the characteristics of a good software?

A good software must satisfy on the following grounds:

- Operational
- Transitional
- Maintenance

Operational

This tells us how well software works in operations. It can be measured on:

- Budget
- Usability
- Efficiency
- Correctness
- Functionality
- Dependability
- Security
- Safety

Transitional

This aspect is important when the software is moved from one platform to another:

- Portability
- Interoperability
- Reusability
- Adaptability

Maintenance

This aspect briefs about how well a software has the capabilities to maintain itself in the everchanging environment:

- Modularity
- Maintainability
- Flexibility
- Scalability

3. Explain about software process?

A software process is a collection of various activities.

There are five generic process framework activities:

- Communication:

The software development starts with the communication between customer and developer.

- Planning:

It consists of complete estimation, scheduling for project development and tracking.

- Modeling:

Modeling consists of complete requirement analysis and the design of the project like algorithm, flowchart etc. The algorithm is the step-by-step solution of the problem and the flow chart shows a complete flow diagram of a program.

- Construction:

Construction consists of code generation and the testing part. Coding part implements the design details using an appropriate programming language. Testing is to check whether the flow of coding is correct or not. Testing also check that the program provides desired output.

- Deployment:

Deployment step consists of delivering the product to the customer and take feedback from them. If the customer wants some corrections or demands for the additional capabilities, then the change is required for improvement in the quality of the software.

4. Explain umbrella activities?

The generic view of software engineering are complemented by a number of umbrella activities.

Umbrella Activities

- Software project tracking and control
- Risk Management
- Software Quality Assurance
- Technical review
- Measurement
- Software Configuration Management
- Reusability management
- Work Product preparation and production

Typical activities in this category include:

- Software project tracking and control :

Tracking and Control is the dual process of detecting when a project is drifting off-plan, and taking corrective action to bring the project back on track.

- Formal technical reviews :

This includes reviewing the techniques that has been used in the project.

- Software quality assurance :

This is very important to ensure the quality measurement of each part to ensure them.

- Software configuration management :

In software engineering, software configuration management (SCM or S/W CM) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of configuration management.

- Document preparation and production :

All the project planning and other activities should be hardly copied and the production get started here.

- Reusability management :

This includes the backing up of each part of the s/w project for the future work(eg: updation)

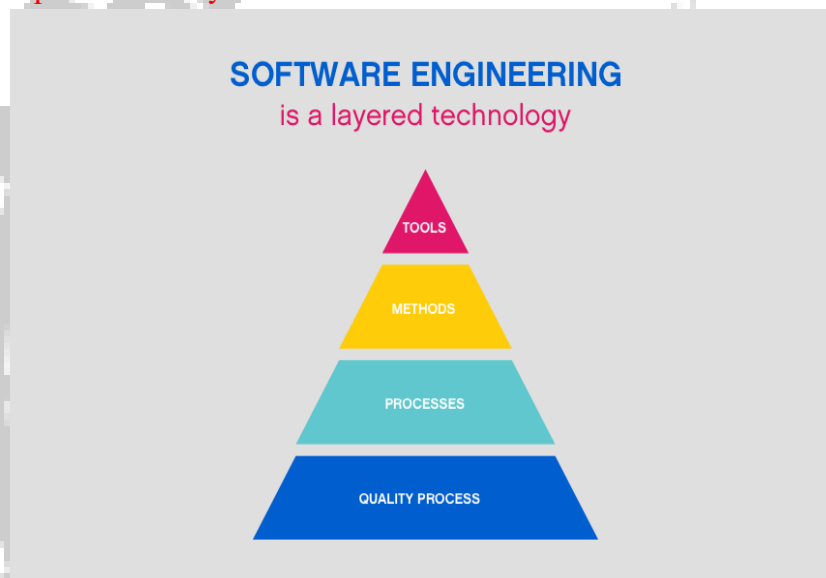
- Measurement :

This will include all the measurement of every aspects of the software project.

- Risk management :

Risk management is a series of steps that help a software team to understand and manage uncertainty.

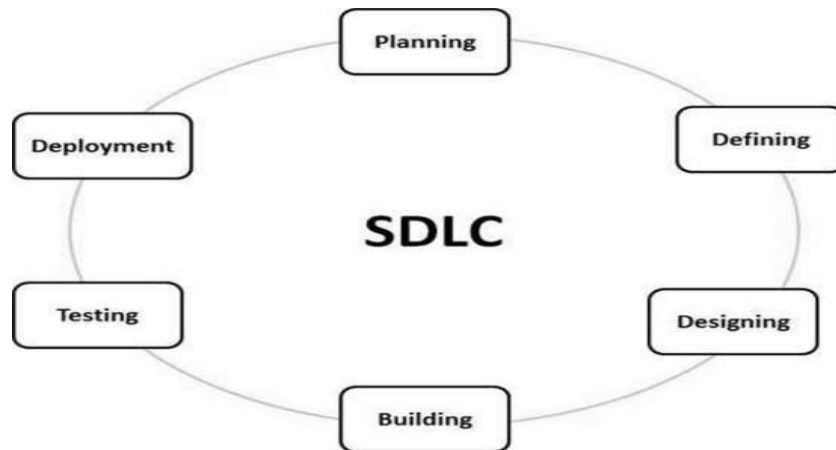
5. Explain about layered structure of SE?



- Tools: This layer contains automated or semi-automated tools that offer support for the framework and the method each software engineering project will follow.
- Method: This layer contains the methods, the technical knowledge and “how-tos” in order to develop software.
- Process: This layer consists of the framework that must be established for the effective delivery of software.
- Quality Focus: This layer is the fundamental layer for software engineering. As stated above it is of great importance to test the end product to see if it meets its specifications. Efficiency, usability, maintenance and reusability are some of the requirements that need to be met by new software.

6. Explain about SDLC?

SDLC is the acronym of Software Development Life Cycle. It is also called as Software Development Process. SDLC is a framework defining tasks performed at each step in the software development process.



A typical Software Development Life Cycle consists of the following stages –

Stage 1: Planning and Requirement Analysis/preliminary investigation

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry.

Stage 2: Defining Requirements/software analysis

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture/software design

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification. Eg: DFD, ER diagram etc...

Stage 4: Building or Developing the Product/software coding

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Stage 5: Testing the Product/software testing

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance/software maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

7. What is prescriptive process model?

A prescriptive process model is a model that describes "how to do" according to a certain software process system. A prescriptive model prescribes how a new software system should be developed. Prescriptive models are used as guidelines or frameworks to organize and structure how software development activities should be performed, and in what order.

8. Give some examples of prescriptive process model?

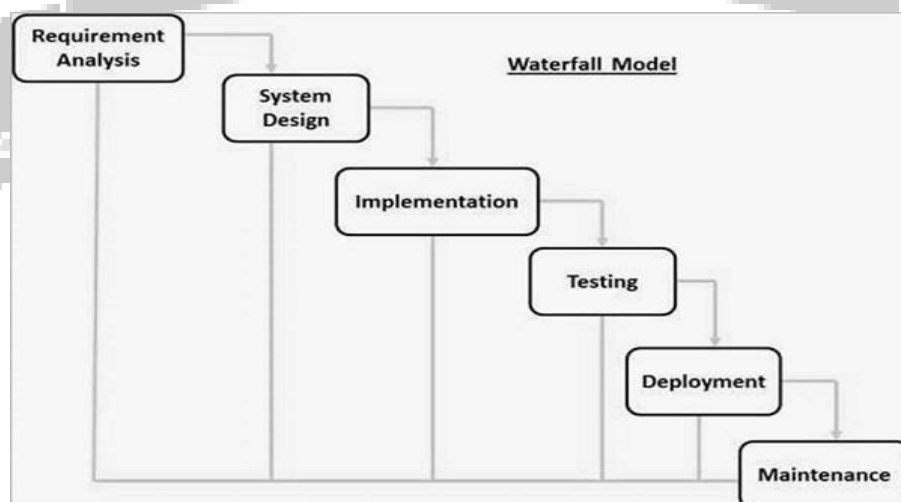
- Waterfall model
- V model
- Incremental model
- Evolutionary model
- Prototyping
- Spiral model
- Win-Win Spiral model
- Concurrent development

9. Explain about waterfall model?

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

Characteristics of waterfall model:-

- It also known as Linear sequential model, Classic life cycle model.
- Simplest of all models
- Systematic sequential approach
- Real project rarely follows
- A working version is available only at the end
- Difficult for customer to state all requirements



- Requirement Gathering and analysis – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- System Design – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- Implementation – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- Integration and Testing – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- Deployment of system – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- Maintenance – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

10. List the advantages and disadvantages of waterfall model?

Advantages

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

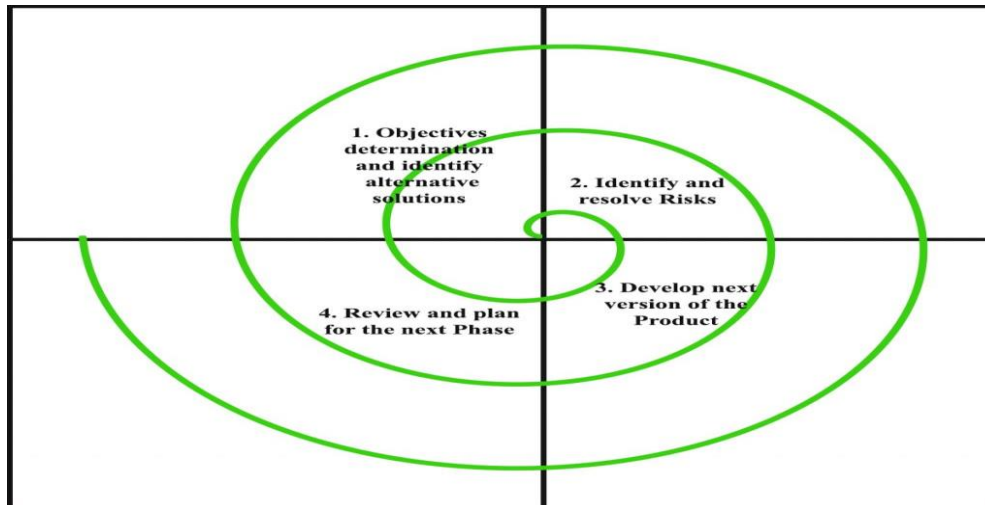
Disadvantages

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

11. Explain about spiral model?

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

manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model.



- Objectives determination and identify alternative solutions: Requirements are gathered from the customers and the objectives are identified, elaborated and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
- Identify and resolve Risks: During the second quadrant all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution.
- Develop next version of the Product: During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
- Review and plan for the next Phase: In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

12. List the advantages and disadvantages of spiral model?

Advantages

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

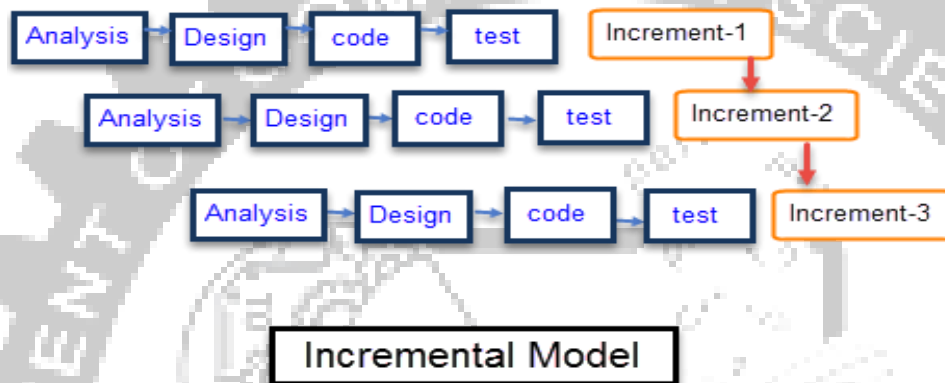
Disadvantages

- **Complex**: The Spiral Model is much more complex than other SDLC models.
- **Expensive**: Spiral Model is not suitable for small projects as it is expensive.
- **Too much dependable on Risk Analysis**: The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced expertise, it is going to be a failure to develop a project using this model.

- Difficulty in time management: As the number of phases is unknown at the start of the project, so time estimation is very difficult.

13. Explain about incremental 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. Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionality has been implemented.



The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.

- Analysis: Requirement and specification of the software are collected
- Design: Some high-end function are designed during this stage
- Code: Coding of software is done during this stage
- Test: Once the system is deployed, it goes through the testing phase

14. Characteristics of incremental model?

Analysis: Requirement and specification of the software are collected

Design: Some high-end function are designed during this stage

Code: Coding of software is done during this stage

Test: Once the system is deployed, it goes through the testing phase

15. What are the advantages and disadvantages of incremental model?

Advantages

- The software will be generated quickly during the software life cycle
- It is flexible and less expensive
- Throughout the development stages changes can be done
- This model is less costly compared to others
- A customer can respond to each building
- Errors are easy to be identified

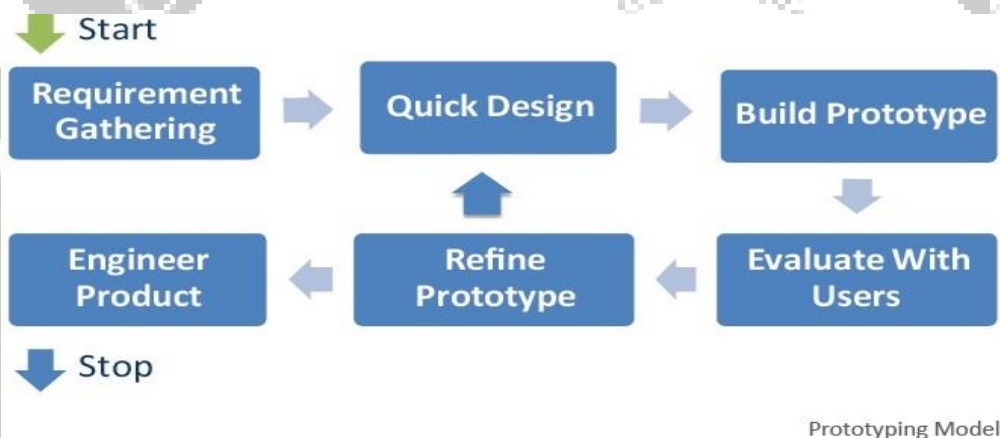
Disadvantages

- It requires a good planning designing

- Problems might cause due to system architecture as such not all requirements collected up front for the entire software lifecycle
- Each iteration phase is rigid and does not overlap each other
- Rectifying a problem in one unit requires correction in all the units and consumes a lot of time.

16. Explain about evolutionary prototyping model?

Evolutionary models are iterative type models. They allow to develop more complete versions of the software. It is software working model of limited functionality. Prototype methodology is defined as a Software Development model in which a prototype is built, test, and then reworked when needed until an acceptable prototype is achieved. First of all, we will develop the most visual aspect of the system. You present a portion of the system to the customer and continue to develop prototypes based on the feedback received. At some point, you and the customer agree that the prototype is “good enough” and release the prototype as the final product.



Step 1: Requirements gathering and analysis

- A prototyping model starts with requirement analysis. In this phase, the requirements of the system are defined in detail. During the process, the users of the system are interviewed to know what is their expectation from the system.

Step 2: Quick design

- The second phase is a preliminary design or a quick design. In this stage, a simple design of the system is created. However, it is not a complete design. It gives a brief idea of the system to the user. The quick design helps in developing the prototype.

Step 3: Build a Prototype

- In this phase, an actual prototype is designed based on the information gathered from quick design. It is a small working model of the required system.

Step 4: Initial user evaluation

- In this stage, the proposed system is presented to the client for an initial evaluation. It helps to find out the strength and weakness of the working model. Comment and suggestion are collected from the customer and provided to the developer.

Step 5: Refining prototype

- If the user is not happy with the current prototype, you need to refine the prototype according to the user's feedback and suggestions. This phase will not over until all the

requirements specified by the user are met. Once the user is satisfied with the developed prototype, a final system is developed based on the approved final prototype.

Step 6: Implement Product and Maintain

- Once the final system is developed based on the final prototype, it is thoroughly tested and deployed to production. The system undergoes routine maintenance for minimizing downtime and prevent large-scale failures.

17. What are the advantages and disadvantages of evolutionary prototyping model?

Advantages

- Prototype model need not know the detailed input, output, processes, adaptability of operating system and full machine interaction.
- In the development process of this model users are actively involved.
- The development process is the best platform to understand the system by the user.
- Errors are detected much earlier.
- Gives quick user feedback for better solutions.
- It identifies the missing functionality easily. It also identifies the confusing or difficult functions.

Disadvantages

- The client involvement is more and it is not always considered by the developer.
- It is a slow process because it takes more time for development.
- Many changes can disturb the rhythm of the development team.
- It is a thrown away prototype when the users are confused with it.

18. What you mean by specialized process model?

Special process models take on many of the characteristics of one or more of the conventional models. However, specialized models tend to be applied when a narrowly defined software engineering approach is chosen.

- Component based development
- Formal methods model

19. What is Agile model in software engineering?

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. To accomplish this task agility is required. 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. An agile process is adaptable and it should adapt incrementally S/w increments must be delivered in short time periods

Agile principles

- a. satisfy customer
- b. welcome changing requirements
- c. deliver working software frequently

20. what are the principles of Agile model development in software engineering?

- Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- It welcomes changing requirements, even late in development.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shortest timescale.

- Build projects around motivated individuals. Give them the environment and the support they need, and trust them to get the job done.
- Working software is the primary measure of progress.
- Simplicity the art of maximizing the amount of work not done is essential.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

21. What are the different types of Agile development models?

- Extreme Programming(XP)
- Adaptive Software Development
- Scrum
- Dynamic Systems Development Method
- Feature Driven Development
- Lean Software Development
- Agile Modelling
- Agile Unified Process

22. Explain about Extreme programming(XP) in Agile model?

It is the most widely used approach of agile development

The XP Process includes

- Planning
- Design
- Coding
- Testing
- ▶ XP design follows KIS(Keep It Simple) principle
- ▶ Encourages refactoring
- ▶ Pair programming concept is used in coding
- Applications of Extreme Programming (XP): Some of the projects that are suitable to develop using XP model are given below:
 - Small projects: XP model is very useful in small projects consisting of small teams as face to face meeting is easier to achieve.
 - Projects involving new technology or Research projects: This type of projects face changing of requirements rapidly and technical problems. So XP model is used to complete this type of projects.

23. Explain ASD in agile model development?

Adaptive Software Development is cyclical like the Evolutionary model, with the phase names reflecting the unpredictability in the complex systems.

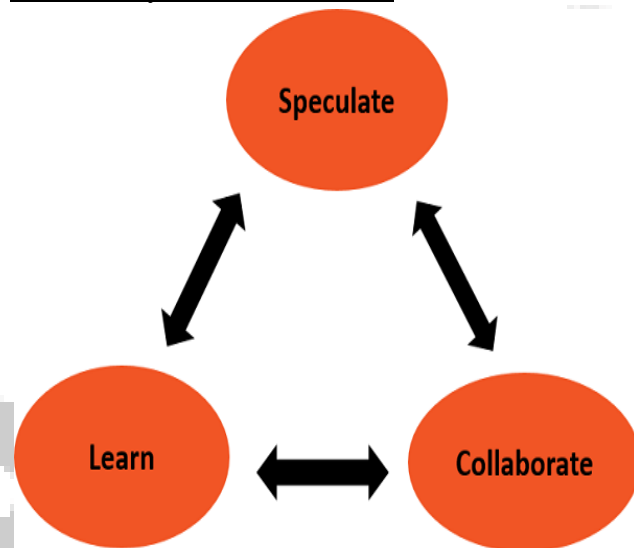
Phases of ASD Life Cycle

- Speculate
- Collaborate
- Learn

These three phases reflect the dynamic nature of Adaptive Software Development. The Adaptive Development explicitly replaces Determinism with Emergence. It goes beyond a mere change in lifecycle to a deeper change in management style. Adaptive Software Development has a dynamic Speculate-Collaborate-Learn Lifecycle.

The Adaptive Software Development Lifecycle focuses on results, not tasks, and the results are identified as application features.

ASD life cycle is shown below



Speculate

- In Adaptive Software Development, the term plan is replaced by the term speculate. While speculating, the team does not abandon planning, but it acknowledges the reality of uncertainty in complex problems. Speculate encourages exploration and experimentation. Iterations with short cycles are encouraged.

Collaborate

- Collaborate would require the ability to work jointly to produce results, share knowledge or make decisions.

Learn

- Team has to enhance their knowledge constantly, using practices such as –
- Technical Reviews
- Project Retrospectives
- Customer Focus Groups

24. What is Scrum framework in agile model?

Scrum is a lightweight, iterative and incremental framework for managing complex work.

Scrum framework activities:

- Requirements
- Analysis
- Design
- Evolution
- Delivery

Important concepts in scrum

- Backlog: a prioritized list of requirements or features. Within each framework activity work task occurs within a process pattern called sprint.
- Sprint: consists of work units required to achieve a requirement defined in backlog that can be done in a predefined time box(30days)
- Scrum meeting : short meeting(15 minutes) held daily by the scrum Team
- Demos: delivers the increment to the customer

25. Explain DSDM in Agile model?

- The Dynamic Systems Development method (DSDM) is an agile software development methodology. It is an iterative, incremental approach that is largely based on the Rapid Application Development (RAD) methodology.
- Principles
 - Focus on the business need
 - Deliver on time
 - Collaborate
 - Never compromise quality
 - Build incrementally from firm foundations
 - Develop iteratively
 - Communicate continuously and clearly
 - Demonstrate control

26. Explain FDD in Agile model?

Feature Driven Development (FDD) is an agile framework that, as its name suggests, organizes software development around making progress on features. Features in the FDD context, though, are not necessarily product features in the commonly understood sense. “complete the login process” might be considered a feature in the Feature Driven Development (FDD) methodology.

FDD was designed to follow a five-step development process, built largely around discrete “feature” projects. That project lifecycle looks like this:

- Develop an overall model
- Build a features list
- Plan by feature
- Design by feature
- Build by feature

27. What is lean software development(LSD)?

Lean Software Development (LSD) is an agile framework based on optimizing development time and resources, eliminating waste, and ultimately delivering only what the product needs. The Lean approach is also often referred to as the Minimum Viable Product (MVP) strategy, in which a team releases a bare-minimum version of its product to the market, learns from users what they like, don’t like and want to be added, and then iterates based on this feedback.

28. What is Agile modelling and AUP?

Agile modeling (AM) is a methodology for modeling and documenting software systems based on best practices. It is a collection of values and principles, that can be applied on an (agile) software development project. This methodology is more flexible than traditional modeling methods, making it a better fit in a fast changing environment.

Agile Unified Process (AUP) is a simplified version of the Rational Unified Process (RUP), which is an iterative software development process framework. It describes a simple, easy to understand approach to developing business application software using agile techniques and concepts.

