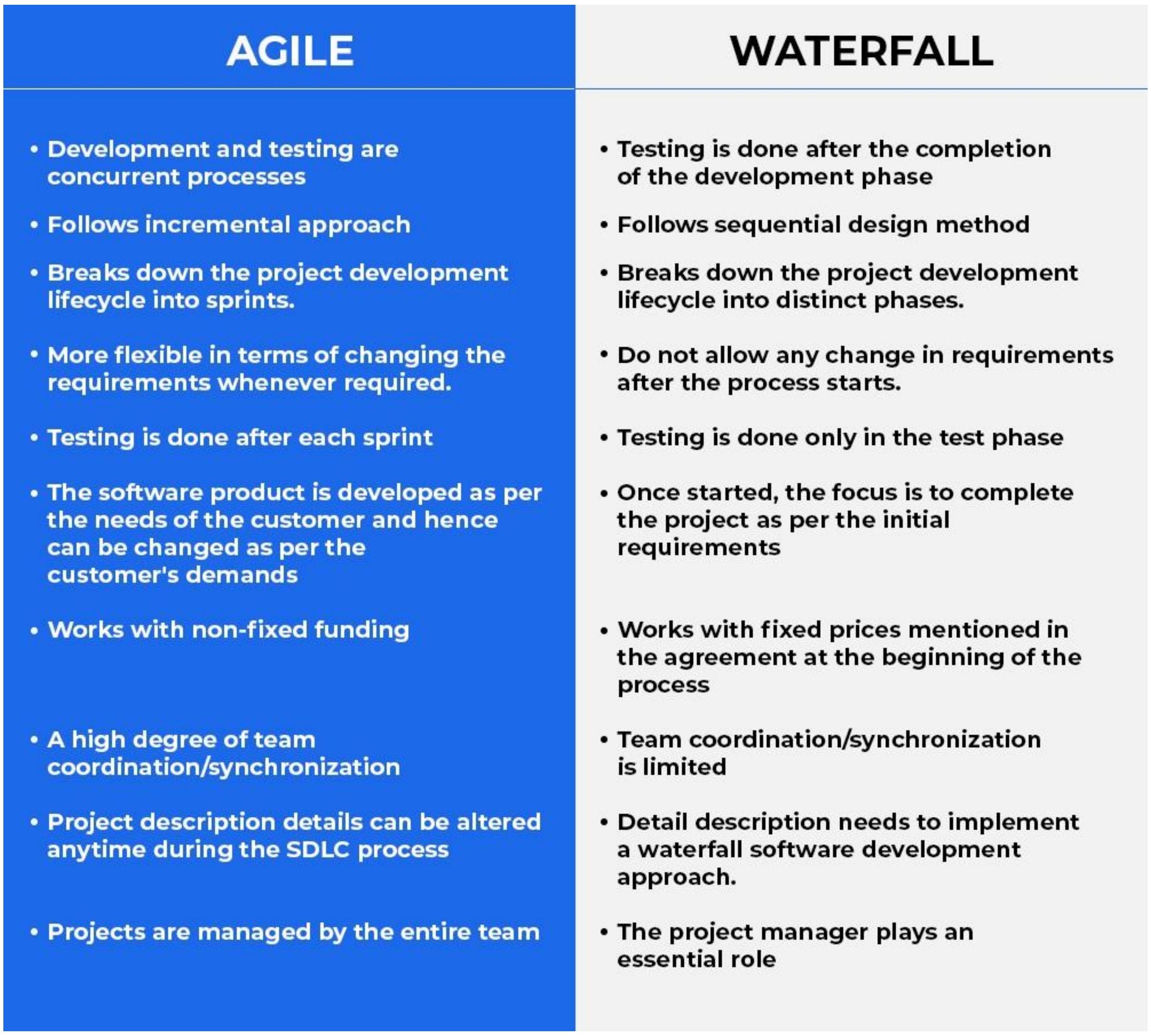
Module 4

Marks 5

1. **Explain the main differences between Agile and Waterfall methodologies.**

Ans :



AGILE	WATERFALL
Development and testing are concurrent processes	Testing is done after the completion of the development phase
Follows incremental approach	Follows sequential design method
Breaks down the project development lifecycle into sprints.	Breaks down the project development lifecycle into distinct phases.
More flexible in terms of changing the requirements whenever required.	Do not allow any change in requirements after the process starts.
Testing is done after each sprint	Testing is done only in the test phase
The software product is developed as per the needs of the customer and hence can be changed as per the customer's demands	Once started, the focus is to complete the project as per the initial requirements
Works with non-fixed funding	Works with fixed prices mentioned in the agreement at the beginning of the process
A high degree of team coordination/synchronization	Team coordination/synchronization is limited
Project description details can be altered anytime during the SDLC process	Detail description needs to implement a waterfall software development approach.
Projects are managed by the entire team	The project manager plays an essential role

2. **Describe the four core values of the Agile Manifesto.**

Ans: • **Individuals and Interactions Over Processes and Tools:**

- Emphasizes the importance of people and their interactions in the development process.

- Encourages collaboration and communication within the team, rather than relying solely on formal processes and tools.

- **Working Software Over Comprehensive Documentation:**

  - Prioritizes delivering functional software that meets user needs over creating extensive documentation.
  - Documentation is still important but should not overshadow the primary goal of producing working software.

- **Customer Collaboration Over Contract Negotiation:**

  - Focuses on continuous collaboration with customers to ensure their needs are met.
  - Encourages direct communication and feedback from customers rather than strictly adhering to contract terms.

- **Responding to Change Over Following a Plan:**

  - Values the ability to adapt and respond to changes as they arise during the project.
  - Recognizes that requirements may evolve, and being flexible is more important than sticking rigidly to an initial plan.

3. Summarize the principles of Agile that focus on customer satisfaction.

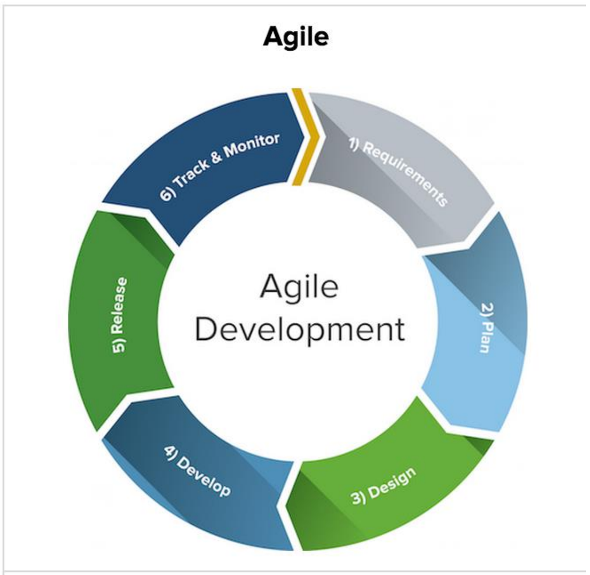
Ans:

1. **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
   - This principle emphasizes the importance of delivering working software frequently to ensure that customers can see progress and provide feedback regularly. It ensures that the software developed meets the customer's needs and expectations.
2. **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
   - Agile embraces change and encourages flexibility in responding to evolving customer requirements. This adaptability ensures that the final product remains relevant and valuable to the customer.
3. **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**
   - Frequent delivery of working software allows customers to experience and use the product early and often, providing opportunities for feedback and adjustments. This iterative approach ensures continuous improvement and alignment with customer needs.
4. **Business people and developers must work together daily throughout the project.**
   - Continuous collaboration between business stakeholders and the development team ensures that the project stays aligned with business goals and customer requirements. Regular interaction fosters better understanding and faster decision-making.
5. **Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**

    - Ensuring that the team is motivated and empowered to make decisions enhances their ability to deliver high-quality software that meets customer expectations. A supportive environment contributes to team effectiveness and customer satisfaction.
6. **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**
    - Direct communication helps in quickly resolving issues and clarifying requirements, ensuring that the team fully understands the customer's needs and expectations.

4. Illustrate how iterative development is implemented in Agile.

Ans:



- Continuous cycles
- Small, high-functioning, collaborative teams
- Multiple methodologies
- Flexible/continuous evolution
- Customer involvement

5. Discuss the importance of communication in Agile methodology.

Ans:

- Flexibility
- Collaboration
- Customer satisfaction
- Continuous improvement
- Faster delivery

6. Identify the role of a Scrum Master in a Scrum team.

Ans:

The Scrum Master's major responsibility is to ensure that scrum is understood and practiced by every team member in the true spirit.

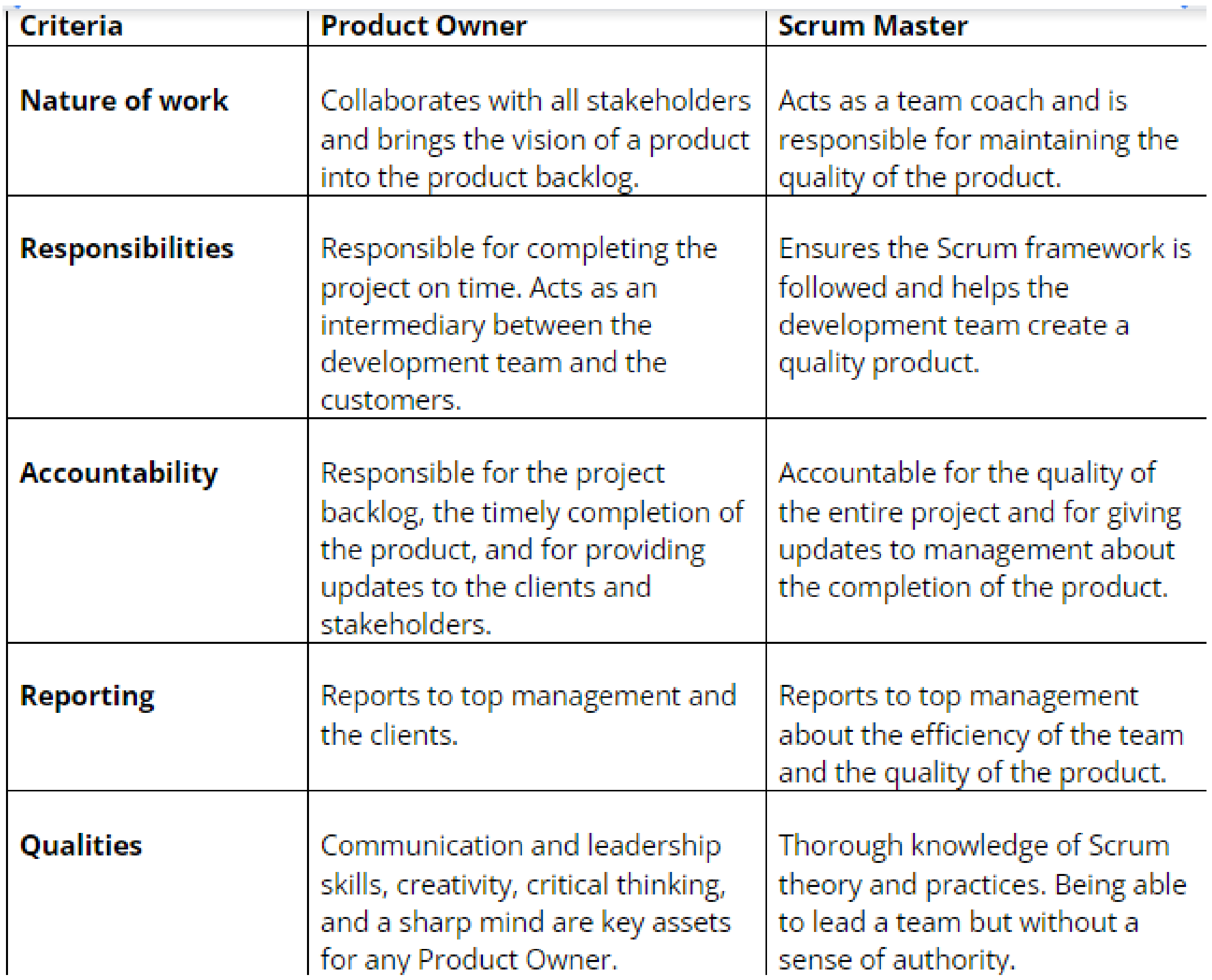
The following are the key responsibilities of the Scrum Master :

- Is a servant leader
- Helps remove obstacles
- Facilitates collaboration

- Teaches scrum to the team.
- Protects the teams from external disruptions such as changes to stories in the current sprint.

7. Compare the responsibilities of the Product Owner and the Scrum Master.

Ans:



Criteria	Product Owner	Scrum Master
Nature of work	Collaborates with all stakeholders and brings the vision of a product into the product backlog.	Acts as a team coach and is responsible for maintaining the quality of the product.
Responsibilities	Responsible for completing the project on time. Acts as an intermediary between the development team and the customers.	Ensures the Scrum framework is followed and helps the development team create a quality product.
Accountability	Responsible for the project backlog, the timely completion of the product, and for providing updates to the clients and stakeholders.	Accountable for the quality of the entire project and for giving updates to management about the completion of the product.
Reporting	Reports to top management and the clients.	Reports to top management about the efficiency of the team and the quality of the product.
Qualities	Communication and leadership skills, creativity, critical thinking, and a sharp mind are key assets for any Product Owner.	Thorough knowledge of Scrum theory and practices. Being able to lead a team but without a sense of authority.

8. Define the purpose of the Sprint Review in Scrum.

Ans:

Sprint Review: The team meets to review the work completed during the sprint and demonstrate the product increments to stakeholders. This is an opportunity for the team to receive feedback and make improvements for the next sprint.

9. Explain the significance of a Product Backlog in Scrum.

Ans:

A product backlog is a dynamic list of functionalities the product might include, such that it provides value to users.

These are a few unique characteristics of a product backlog:

- It is dynamic in nature as it evolves based on changing market needs
- Lists all the features and capabilities that will be taken up in iteration and delivered as a product increment
- It is refined on a continuous basis. The Product Owner and Development team collaborate and update the details, estimate, and prioritize based on business value and size

10. Describe the process of a Sprint Planning meeting.

Ans:

Sprint Planning: The team meets to plan the upcoming sprint, review the product backlog, and determine which items can be completed during the sprint. This is a collaborative effort between the development team and the product owner.

11. **Summarize the key artifacts in Scrum.**

Ans:

Scrum is an Agile framework for managing software development projects. It involves a set of practices and principles designed to help teams deliver high-quality products in a fast and efficient manner.
The following are the three main artifacts in Scrum:

1. Product Backlog
2. Sprint Backlog
3. Increment

1.Product backlog

A product backlog is a dynamic list of functionalities the product might include, such that it provides value to users.
These are a few unique characteristics of a product backlog:

- It is dynamic in nature as it evolves based on changing market needs
- Lists all the features and capabilities that will be taken up in iteration and delivered as a product increment
- It is refined on a continuous basis. The Product Owner and Development team collaborate and update the details, estimate, and prioritize based on business value and size

2. Sprint backlog:

Sprint backlog is a subset of the entire product backlog that the scrum team plans to implement in **one iteration or sprint.**
Sprint backlog has:

- Subset of product backlog items that the teams commit to implement in one sprint
- Items broken into smaller pieces of work as tasks
- A focus on 'HOW' the team does the work and delivers the value in one sprint
- A story or task board that is used by the teams to view backlog and what the individuals sign up for work after backlog prioritization
- Provision for the Development teams to track the sprint progress and check their alignment with sprint goals

3. Increment:

- An increment is the work delivered at the end of every sprint. Typically, after every iteration there will be a Product Increment (PI) that delivers value and the final product will be a working software.
- This increment is a sum of all the capabilities that were delivered in the previous sprints as a part of the PI. At the end of every sprint, the Product Owner decides whether to release the working product increment or wait until the next release.

12. **Discuss the role of Daily Stand-ups in Scrum.**

Ans:

1.**Empirical Process Control**: Scrum is based on the principle of empirical process control, which means that it relies on experience and experimentation to continuously improve the process.

2. **Cross-functional Teams**: Scrum teams are cross-functional, meaning they consist of individuals with a range of skills and expertise.

3. **Sprint**: A sprint is a time-boxed period, usually one to four weeks, during which the team works on completing a set of tasks.

4. **Incremental Delivery**: Scrum allows for the delivery of a usable product increment at the end of each sprint.

5. **Transparency:** Scrum emphasizes transparency in all aspects of the process, from the work being done by the team to the progress being made.

13. Explain how Continuous Integration is achieved using Jenkins.

Ans:

Jenkins

Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly

Features

- Jenkins increases the scale of automation.
- It can easily set up and configure via a web interface.
- It can distribute the tasks across multiple machines, thereby increasing concurrency.
- It supports continuous integration and continuous delivery.
- It offers 400 plugins to support the building and testing any project virtually.
- It requires little maintenance and has a built-in GUI tool for easy updates.

14. Illustrate the use of Docker in DevOps for containerization.

Ans:

Docker

- Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

Features

- It configures the system more comfortably and faster.
- It increases productivity.
- It provides containers that are used to run the application in an isolated environment.
- It routes the incoming request for published ports on available nodes to an active container. This feature enables the connection even if there is no task running on the node.
- It allows saving secrets in the swarm itself.

15. Describe the purpose of Kubernetes in managing containers.

Ans:

Kubernetes

Kubernetes is a powerful and feature-rich platform for managing containerized applications

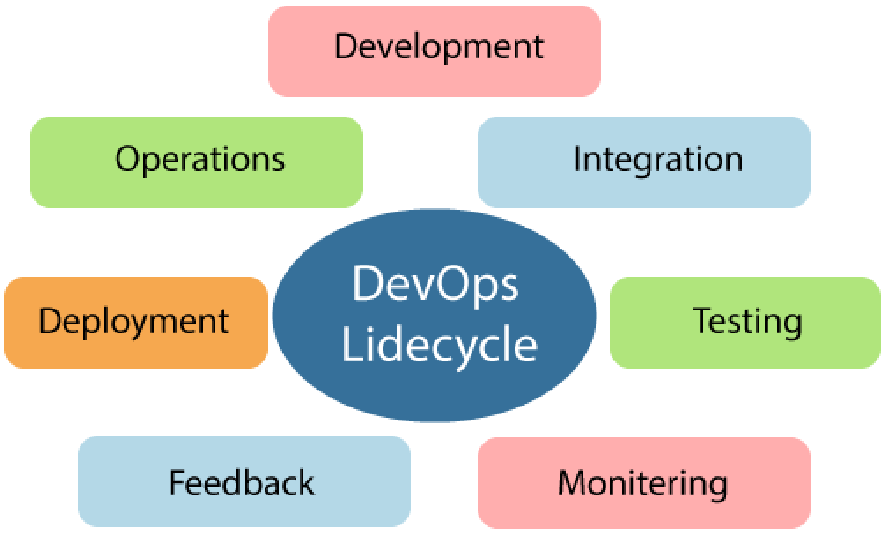
Features

- Container orchestration: Kubernetes provides a platform for deploying, scaling, and managing containers, making it easier to manage and scale applications.
- Self-healing: Kubernetes has built-in mechanisms for automatically replacing failed containers, ensuring that applications remain available and running even in the face of failures.
- Automatic scaling: Kubernetes can automatically scale the number of containers running based on demand, providing a way to ensure that applications can handle increased traffic and load.
- Load balancing: Kubernetes provides built-in load balancing capabilities, making it easy to distribute incoming traffic across multiple containers.

16. Explain the DevOps lifecycle and its significance.

Ans:

DevOps defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from the beginning to the final stage of the product



- The DevOps lifecycle consists of **seven phases: Continuous Development, Continuous Integration, Continuous Testing, Continuous Monitoring, Continuous Feedback, Continuous Deployment, and Continuous Operations**. Each phase plays a crucial role in the software development process and contributes to the overall efficiency and effectiveness of the DevOps approach.
- **Continuous Development, involves planning and coding the software** where the vision of the project is decided and the developers begin coding the application.
- Continuous Integration is the **heart of the DevOps lifecycle** and involves **committing changes to the source code frequently and building the code**, including unit testing, integration testing, code review, and packaging.
- **Continuous Testing involves constantly testing the developed software for bugs using automation testing tools** such as TestNG, JUnit, and Selenium.
- **Continuous Monitoring involves monitoring the operational factors** of the DevOps process and recording important information about the use of the software.
- Continuous Feedback involves analyzing the results from the operations of the software to continuously improve the application development. Continuous Deployment involves deploying the

code to the production servers and ensuring that the code is correctly used on all servers. Configuration management tools play a crucial role in executing tasks frequently and quickly in this phase.

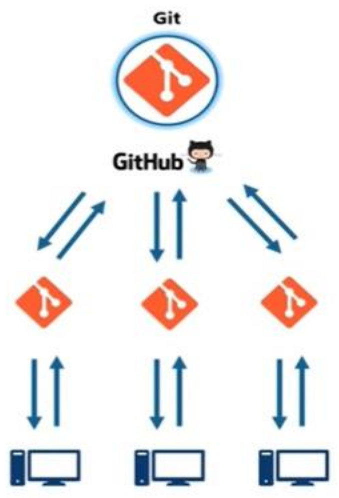
- Continuous Operations are based on continuity with complete automation of the release process and allow organizations to accelerate the overall time to market. With DevOps, the software product becomes more efficient and increases the overall count of interested customers.

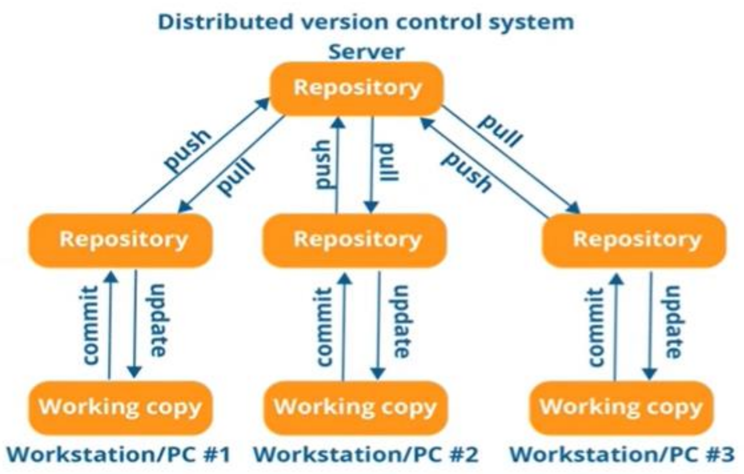
17. Describe the workflow of Git for source control.

Ans:

- Git is a widely used open-source distributed version control system that helps track changes made to software source code. It was created by Linus Torvalds in 2005 and has since become one of the most popular version control systems in use today.
- Git offers a number of benefits over other version control systems, including:

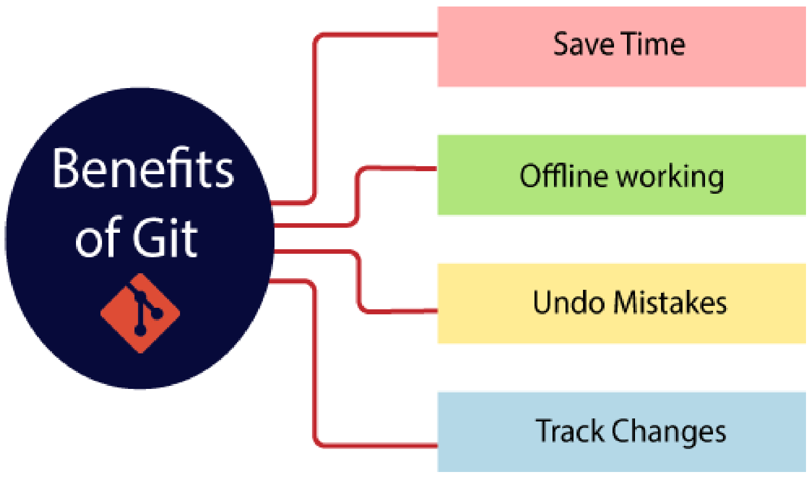
1. Distributed Architecture.
2. Branching and Merging





18. Summarize the benefits of using Git for version control in software development.

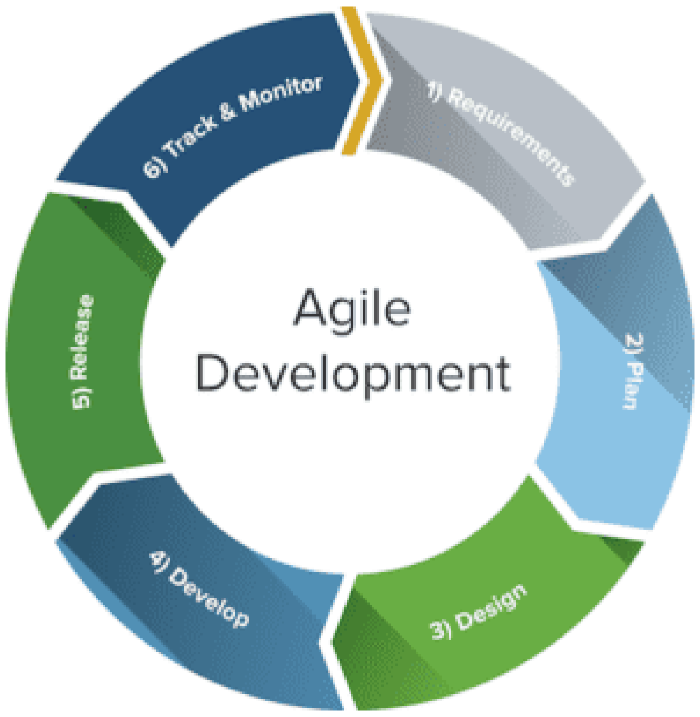
Ans:

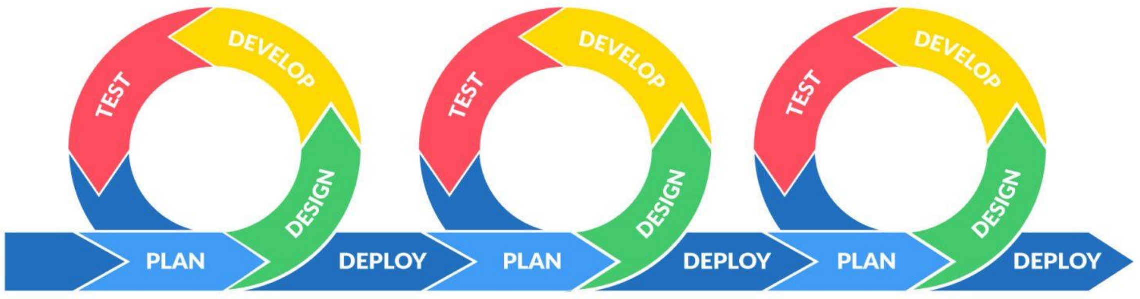


10 marks:

1. **Explain in detail how Agile methodology enhances flexibility in project management.**

Ans:





2. Discuss how the Agile principles support adaptive planning and evolutionary development.

Ans:

We follow these principles:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- **Working software is the primary measure of progress.**
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

3. Describe the process and benefits of conducting a Sprint Retrospective.

Ans:

Process of Conducting a Sprint Retrospective:

1. **Set the Stage**:
   - The Scrum Master facilitates the retrospective, ensuring the team has a conducive environment for open communication and reflection.
   - The team gathers in a comfortable space where they can focus on discussing the sprint without interruptions.
2. **Review the Sprint**:

    - The team reviews the sprint's goals, commitments, and outcomes. This includes looking at what was accomplished, any incomplete work, and any deviations from the sprint plan.
    - Metrics such as sprint velocity, burn-down charts, and any other relevant data are reviewed to provide a factual basis for the discussion.
3. **Gather Data**:
    - Each team member shares their perspective on what went well during the sprint and what could be improved.
    - The Scrum Master may facilitate this discussion using techniques like round-robin sharing or sticky notes to gather feedback anonymously.
4. **Generate Insights**:
    - The team identifies patterns, themes, or recurring issues from the feedback gathered. This may include identifying bottlenecks, communication gaps, technical challenges, or process inefficiencies.
    - Insights are discussed openly, focusing on understanding the root causes rather than assigning blame.
5. **Generate Action Items**:
    - Based on the insights gained, the team collaboratively generates specific action items for improvement. These action items should be actionable, measurable, and achievable within the next sprint.
    - Each action item is assigned to a team member or a subgroup responsible for implementation.
6. **Decide on Improvements**:
    - The team prioritizes the action items based on their impact and feasibility. They decide which improvements to implement immediately and which may need further discussion or planning.
    - Consensus is reached on the action items to ensure commitment from all team members.
7. **Close the Retrospective**:
    - The Scrum Master summarizes the outcomes of the retrospective, including the agreed-upon action items and any decisions made.
    - The team reaffirms their commitment to implementing the agreed improvements and acknowledges the value of the retrospective in fostering continuous improvement.

Benefits of Conducting a Sprint Retrospective:

1. **Continuous Improvement**: Retrospectives promote a culture of continuous improvement by encouraging teams to reflect on their processes, practices, and interactions regularly.
2. **Team Collaboration**: Retrospectives facilitate open communication and collaboration among team members. They provide a safe space for sharing feedback, ideas, and concerns.
3. **Learning and Knowledge Sharing**: Teams learn from both successes and failures. They can identify what worked well and replicate it while addressing areas for improvement.
4. **Ownership and Accountability**: By identifying action items and assigning responsibilities, retrospectives foster ownership and accountability among team members for implementing changes.

5. **Adaptation and Flexibility**: Teams can adapt their processes and practices based on feedback and changing circumstances. This adaptability improves responsiveness to customer needs and market demands.
6. **Team Morale**: Addressing challenges and celebrating successes during retrospectives boosts team morale and cohesion. It reinforces a sense of achievement and progress.
7. **Productivity and Efficiency**: Implementing improvements identified in retrospectives can lead to increased productivity and efficiency in subsequent sprints. It reduces waste and optimizes the team's workflow.
8. **Empowerment**: Retrospectives empower teams to take ownership of their work environment and process improvements. This empowerment enhances job satisfaction and motivation.

4. Compare and contrast Scrum with other Agile frameworks like Kanban.

Ans:

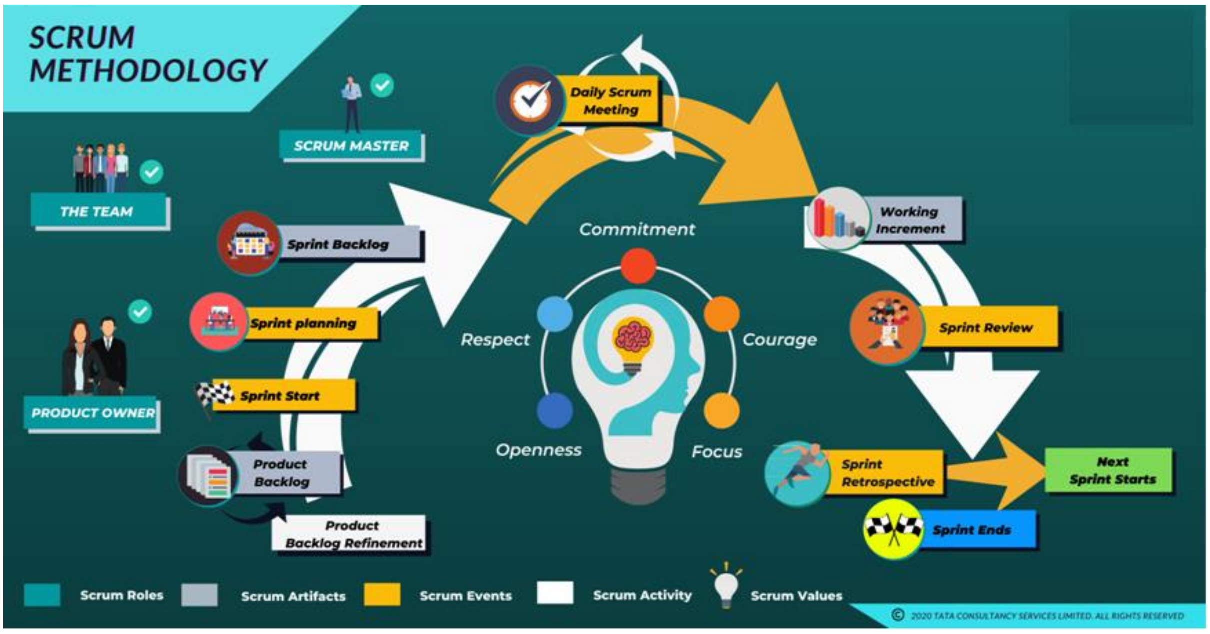
1.**Empirical Process Control**: Scrum is based on the principle of empirical process control, which means that it relies on experience and experimentation to continuously improve the process.

2. **Cross-functional Teams**: Scrum teams are cross-functional, meaning they consist of individuals with a range of skills and expertise.

3. **Sprint**: A sprint is a time-boxed period, usually one to four weeks, during which the team works on completing a set of tasks.

4. **Incremental Delivery**: Scrum allows for the delivery of a usable product increment at the end of each sprint.

5. **Transparency:** Scrum emphasizes transparency in all aspects of the process, from the work being done by the team to the progress being made.



5. Explain the role and importance of each Scrum artifact (Product Backlog, Sprint Backlog, Increment).

Ans:

Scrum is an Agile framework for managing software development projects. It involves a set of practices and principles designed to help teams deliver high-quality products in a fast and efficient manner.
The following are the three main artifacts in Scrum:

4. Product Backlog
5. Sprint Backlog
6. Increment

1.Product backlog
A product backlog is a dynamic list of functionalities the product might include, such that it provides value to users.
These are a few unique characteristics of a product backlog:

- It is dynamic in nature as it evolves based on changing market needs
- Lists all the features and capabilities that will be taken up in iteration and delivered as a product increment
- It is refined on a continuous basis. The Product Owner and Development team collaborate and update the details, estimate, and prioritize based on business value and size

2. Sprint backlog:
Sprint backlog is a subset of the entire product backlog that the scrum team plans to implement in **one iteration or sprint.**
Sprint backlog has:

- Subset of product backlog items that the teams commit to implement in one sprint
- Items broken into smaller pieces of work as tasks
- A focus on 'HOW' the team does the work and delivers the value in one sprint
- A story or task board that is used by the teams to view backlog and what the individuals sign up for work after backlog prioritization
- Provision for the Development teams to track the sprint progress and check their alignment with sprint goals

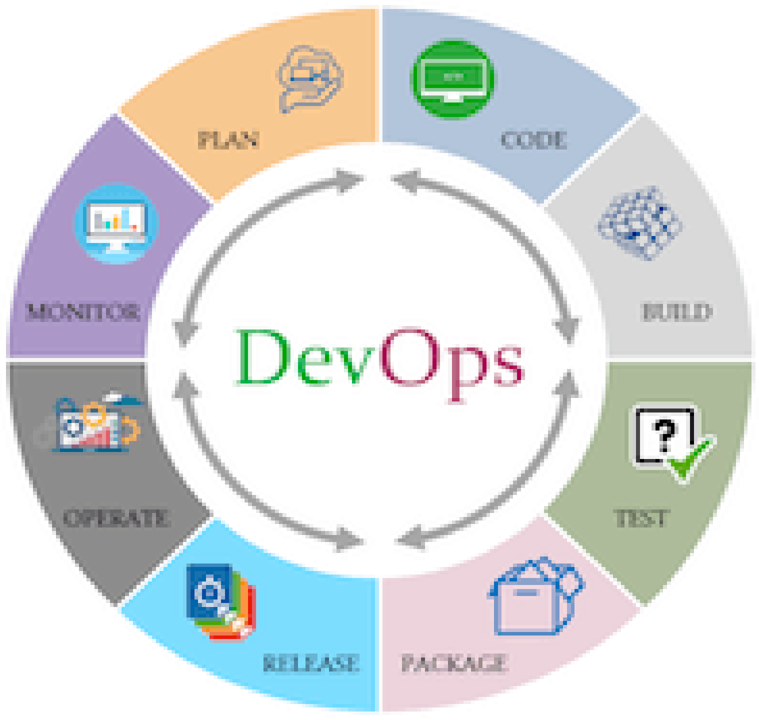
3. Increment:

- An increment is the work delivered at the end of every sprint. Typically, after every iteration there will be a Product Increment (PI) that delivers value and the final product will be a working software.

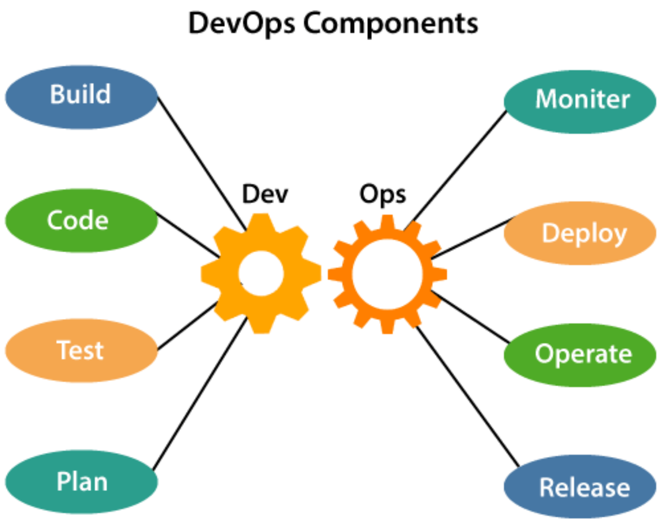
This increment is a sum of all the capabilities that were delivered in the previous sprints as a part of the PI. At the end of every sprint, the Product Owner decides whether to release the working product increment or wait until the next release.

6. **Discuss how DevOps practices improve collaboration between development and operations teams.**

Ans:
DevOps is a combination of two words, one is Software Development, and the second is Operations. This allows a single team to handle the entire application lifecycle, **from development to testing, deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.



The various components that are used in the DevOps architecture:



The various components that are used in the DevOps architecture:

- Build: Cloud and shared resources are used to control resource usage.
- Code: Good practices like Git make code use, tracking, and reusability easier.
- Test: Automated testing saves time and prepares the app for production.
- Plan: Agile methodology helps plan development and improve productivity.
- Monitor: Continuous monitoring reduces risk of failure and tracks app health.
- Deploy: Automated deployment captures insights and optimizes performance.
- Operate: DevOps collaborates throughout the service lifecycle.
- Release: Deployment to production is done manually to minimize impact.

7. Explain the integration of Jenkins, Docker, and Kubernetes in a DevOps pipeline.

Ans:

Docker

- Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

Features

- It configures the system more comfortably and faster.
- It increases productivity.
- It provides containers that are used to run the application in an isolated environment.
- It routes the incoming request for published ports on available nodes to an active container. This feature enables the connection even if there is no task running on the node.
- It allows saving secrets in the swarm itself.

Jenkins :

Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly

Features

- Jenkins increases the scale of automation.
- It can easily set up and configure via a web interface.
- It can distribute the tasks across multiple machines, thereby increasing concurrency.
- It supports continuous integration and continuous delivery.
- It offers 400 plugins to support the building and testing any project virtually.
- It requires little maintenance and has a built-in GUI tool for easy updates.

Kubernetes :

Kubernetes is a powerful and feature-rich platform for managing containerized applications

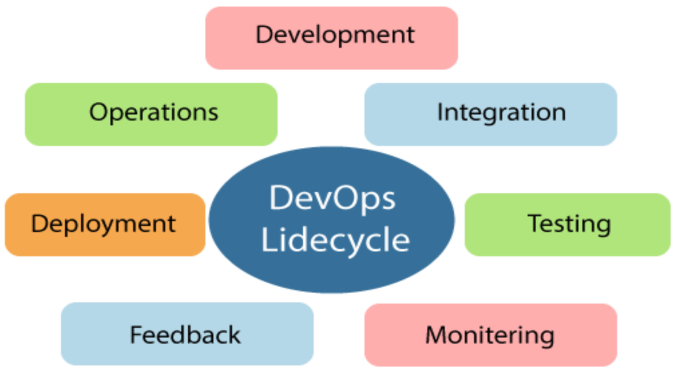
Features

- Container orchestration: Kubernetes provides a platform for deploying, scaling, and managing containers, making it easier to manage and scale applications.
- Self-healing: Kubernetes has built-in mechanisms for automatically replacing failed containers, ensuring that applications remain available and running even in the face of failures.
- Automatic scaling: Kubernetes can automatically scale the number of containers running based on demand, providing a way to ensure that applications can handle increased traffic and load.
- Load balancing: Kubernetes provides built-in load balancing capabilities, making it easy to distribute incoming traffic across multiple containers.

8. Describe the end-to-end workflow of a typical DevOps lifecycle.

Ans:

DevOps defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from the beginning to the final stage of the product



- The DevOps lifecycle consists of **seven phases: Continuous Development, Continuous Integration, Continuous Testing, Continuous Monitoring, Continuous Feedback, Continuous Deployment, and Continuous Operations**. Each phase plays a crucial role in the software development process and contributes to the overall efficiency and effectiveness of the DevOps approach.
- **Continuous Development, involves planning and coding the software** where the vision of the project is decided and the developers begin coding the application.

- Continuous Integration is the **heart of the DevOps lifecycle** and involves **committing changes to the source code frequently and building the code**, including unit testing, integration testing, code review, and packaging.
- **Continuous Testing involves constantly testing the developed software for bugs using automation testing tools** such as TestNG, JUnit, and Selenium.
- **Continuous Monitoring involves monitoring the operational factors** of the DevOps process and recording important information about the use of the software.
- Continuous Feedback involves analyzing the results from the operations of the software to continuously improve the application development. Continuous Deployment involves deploying the code to the production servers and ensuring that the code is correctly used on all servers. Configuration management tools play a crucial role in executing tasks frequently and quickly in this phase.
- Continuous Operations are based on continuity with complete automation of the release process and allow organizations to accelerate the overall time to market. With DevOps, the software product becomes more efficient and increases the overall count of interested customers.

The main principles of DevOps are Continuous delivery, automation, and fast reaction to feedback.

- **End to End Responsibility**
- **Continuous Improvement**
- **Automate Everything**
- **Custom Centric Action**
- **Monitor and test everything**
- **Work as one team**