



sasi INSTITUTE OF
TECHNOLOGY &
autonomous ENGINEERING

Accredited by **NAAC** with **"A"** Grade
Recognised by **UGC** under section 2(f) & 12(B)
Approved by **AICTE** - NEW Delhi
Permanently Affiliated to **JNTUK, SBTET**
Ranked as **"A"** Grade by Govt. of A.P.

Department of Computer Science & Engineering

Academic Year 2023 – 24

DevOps MID I Sets III BTech II Sem CSE All Sections

Set 1

1. Explain importance of Agile software development. [10M]
2. Explain DevOps architecture and its features with a neat sketch. [10M]
3. Describe various features and capabilities in agile. [10M]

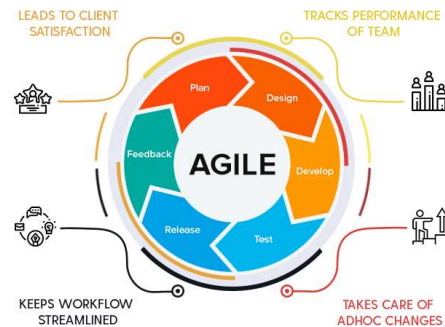
ANSWERS

1. Explain importance of Agile software development?

ANS: Agile: Agile software development is important because it allows software development teams to deliver high-quality projects on time and within budget. Agile enhances communication and collaboration among professionals, and it helps to minimize risks associated with complex projects.[2M]

Key benefits and characteristics of Agile software development

WHY COMMUNICATION IS ALL IMPORTANT DURING
AGILE DEVELOPMENT



promatics

[2M]

High-Quality Products: Breaking projects into units allows for high-quality development, testing, and collaboration. That also improved each time, with a focus on quality.

Collaboration and Communication : Increased communication and collaboration with clients ensures everyone is aligned .

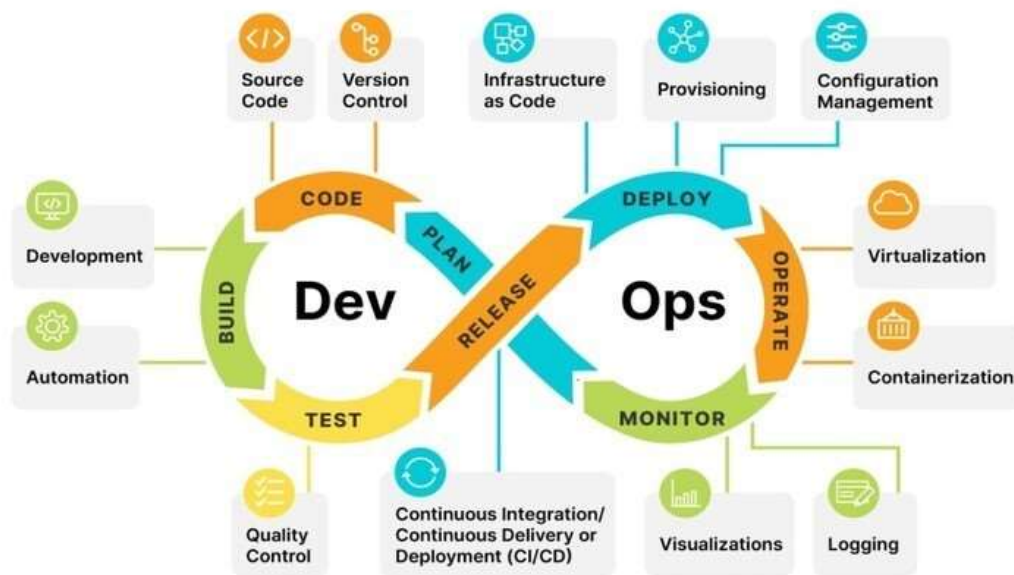
Adaptability : Agile provides the ability to respond quickly to changes at any time, adjusting timelines and prioritizing different features or tasks .

Customer Satisfaction : Agile development helps companies figure out customer needs. Agile software development values customer collaboration, and transparency over process.

Flexibility : Agile anticipates the need for flexibility and applies a level of pragmatism to the delivery of the finished product [6M]

2. Explain DevOps architecture and its features with a neat sketch ?

Ans: DevOps : DevOps architecture integrates software development (Dev) and IT operations (Ops) to enhance collaboration, efficiency, and quality across the software development lifecycle . It promotes continuous integration and delivery (CI/CD), enabling faster and more reliable software releases .[1M]



[2M]

DevOps Architecture Tools : [3M]

Plan: Initial stage for project planning.

Code: Selecting the right tools for the project.

Build: Code is committed to a shared repository where builds and tests are run.

Test: Automated testing in a staging environment.

Release: Tested code is prepared for deployment

Deploy: Seamless deployment to the production environment.

Operate: Collecting feedback and scaling the platform as needed.

Monitor: Reviewing the cycle to avoid errors and make improvements

Key Features and Components:[4M]

Automation: Automates repetitive tasks like code testing, deployment, and monitoring, allowing teams to focus on innovation and problem-solving.

CI/CD Pipeline: A core component that includes continuous integration (CI) for merging code changes into a shared repository and continuous delivery (CD) for automating the deployment process.

Collaboration: Emphasizes effective communication and teamwork between development and operations teams, utilizing collaboration tools and practices to enhance coordination.

Configuration Management: Manages and automates the configuration of infrastructure and applications using tools like Ansible, Chef, and Puppet to ensure consistent settings across different environments.

Infrastructure as Code (IaC): Manages infrastructure using code, simplifying setup and scaling operation.

Monitoring and Feedback Loops: Ensures issues are detected and addressed promptly to enhance the stability and performance of applications.

3.Describe various features and capabilities in agile?

Ans : Agile: Agile is a flexible and iterative software development methodology that focuses on customer satisfaction, adaptability, and high-quality software delivery. **[1M]**

Features of Agile[6M]

1. **Iterative Development :** Agile follows short development cycles called sprints. Each sprint results in a working product increment.
2. **Customer Collaboration :** Customers are involved throughout the development process. Regular feedback is collected to improve the product.
3. **Adaptive Planning :** Agile allows teams to adjust plans based on new requirements. Priorities are continuously updated based on business needs.
4. **Lightweight Documentation :** Agile emphasizes working software over detailed documentation. Only necessary documents are maintained.

5. **Cross-Functional Teams** : Teams consist of developers, testers, and business analysts. Encourages shared responsibilities and knowledge exchange.

6. **Frequent Releases** : Agile delivers software in small, incremental releases. Customers get usable features early.

Capabilities of Agile[3M]

1. **Sprint-Based Work** : Agile divides work into time-boxed iterations (sprints). o Helps manage workloads and track progress.

2. **Continuous Feedback** : Regular feedback loops ensure customer needs are met.

3. **Risk Management** : Frequent testing and iterations reduce project risks.

4. **Enhanced Team Productivity** : Agile teams work collaboratively, leading to better efficiency.

5. **Better Quality Assurance** :Continuous testing improves software quality.

Set 2

1. What is SDLC? Explain various phases involved in SDLC. [10M]
2. Explain briefly about various stages involved in the DevOps pipeline. [10M]
3. Describe the phases in DevOps life cycle. [10M]

ANSWERS

1.What is SDLC? Explain various phases involved in SDLC?

Ans: **SDLC**: The Software Development Life Cycle (SDLC) is a framework that outlines the steps involved in creating software, from initial planning to deployment and maintenance. It provides a structured approach for developing high-quality software in a systematic and cost-effective manner.[1M]

Key Phases of the SDLC:[6M]

Planning and Requirement Gathering: This initial stage involves understanding the project's goals, target users, and specific requirements. It includes gathering input from stakeholders to define the software's purpose and functionalities.

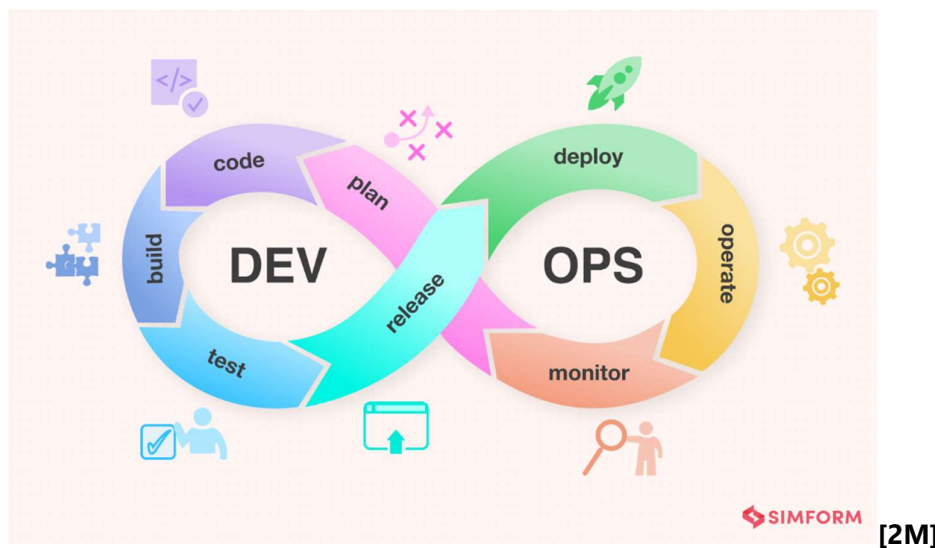
Design: In this phase, developers determine the best way to implement the solution, considering dependencies, performance needs, and the existing system architecture.The output is a design document that outlines the architecture and implementation details.

Implementation (Coding): This involves writing the actual code based on the design specifications. Developers create new code or modify existing code to meet the project's requirements.

Testing and Integration: After the code is written, it undergoes thorough testing to ensure it functions correctly and meets the defined requirements. This includes unit testing and end-to-end integration tests.

Deployment: The tested code is deployed to the production environment, making it accessible to users. Deployment tools and processes are used to ensure a smooth and reliable release.

Operationalization and Maintenance: This final stage involves monitoring the software's performance, addressing any issues or errors that arise, and providing ongoing support. Maintenance also includes updates and enhancements to adapt to changing needs and new use cases.



8 phases of DevOps Life Cycle :[1M]

Planning: Define goals and requirements.

Analysis: Assess feasibility and document needs.

Design: Plan the software architecture.

Coding: Write the software code.

Testing: Ensure code functions correctly.

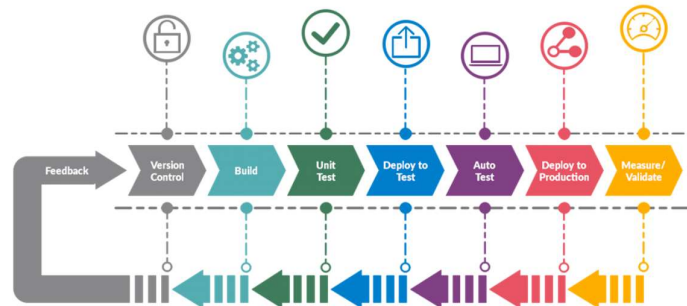
Integration: Combine modules and designs.

Deployment: Release software to users.

Maintenance: Monitor, fix, and update.

2.Explain briefly about various stages involved in the DevOps pipeline?

Ans: The **DevOps pipeline** is a series of automated processes and tools that streamline software development, from code integration to deployment and monitoring. It combines the functions of both the Development and Operations teams. The goal is to deliver value to end-users faster through high-quality software solutions.[2M]



[2M]

stages involved in a DevOps pipeline: [6M]

Plan : Development teams discuss and define project requirements, timelines, and goals. This stage involves planning the workflow, development roadmap, and gathering feedback.

Code: Developers write the code, using version control systems to manage and track changes. Code is then pushed into a source code repository.

Build: The code is compiled, and deployable artifacts are created using a build system. This stage also merges the code base and initiates automated integration and unit tests.

Test: Performance is verified, and bugs are identified through various automated tests, including unit, functional, and integration tests. Testers execute functional, system, and unit tests on the build.

Release: The software is prepped for deployment, which includes compiling documentation and release notes, and ensuring compatibility with the target environment. The operations team confirms the project is ready to be released and builds it into the production environment.

Deploy: The software is released to the production environment, either manually or automatically. Approved changes are deployed for end-users to access.

Operate: The software is actively used by end-users. This involves managing and maintaining the software, ensuring its availability and performance, and addressing user support requests. The operations team configures and manages the project in the production environment, often relying on automation.

Monitor: The project's performance, user response, and overall success are monitored. This stage provides insights into user behaviour and helps identify areas for improvement

3. Describe the phases in DevOps life cycle?

Ans: The **DevOps life cycle** includes multiple phases that integrate development and operations, ensuring continuous software improvement and faster deployments. **[2M]**

Phases of DevOps Life Cycle **[6M]**

1. **Plan** : Requirements are collected, and project tasks are defined.

Tools: Jira, Azure DevOps.

2. **Develop** : Developers write and commit code. Version control tools help in tracking changes.

Tools: Git, GitHub, GitLab.

3. **Build** : Converts source code into executable files. Automated builds ensure consistency.

Tools: Maven, Gradle

4. **Test** : Automated and manual testing to detect defects.

Tools: Selenium, JUnit.

5. **Release** : Software is packaged for deployment.

Tools: Docker, Helm.

4. **Deploy** : Code is moved to production environments.

Tools: Kubernetes, AWS CodeDeploy.

5. **Operate** : Ensures smooth functioning of the application.

Tools: Prometheus, Nagios.

6. **Monitor** : Tracks application performance and logs errors.

Tools: ELK Stack, Datadog.

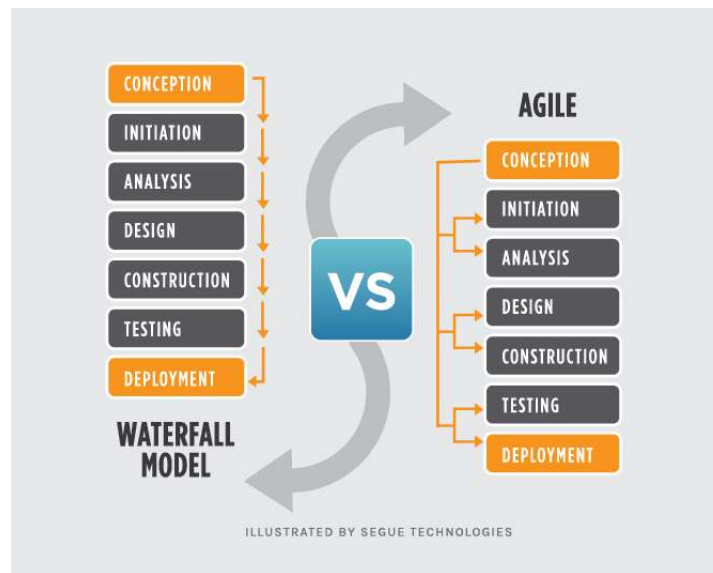
Set 3

1. Write the difference between Waterfall and Agile models. [10M]
2. Discuss in detail about DevOps eco system. [10M]
3. List and explain the steps followed for adopting DevOps in IT projects. [10M]

ANSWERS

1. Write the difference between Waterfall and Agile models?

Ans : The Agile and Waterfall models are two distinct approaches to software development. Agile is iterative and flexible, whereas Waterfall is sequential. [1M]



[2M]

Agile Methodology	Waterfall Methodology
Divides process into sprints	Divides process into phases
Incremental approach	Sequential approach
Flexible	Well-structured and rigid
It is like a collection of several different projects	The entire process is a part of one single project
Allows making changes later on as well	Almost impossible to make changes in the later stages
Iterative approach as planning, designing, prototyping, development, and testing is done more than once.	Planning, designing, prototyping, development, and testing are done just once.
Testing is carried out during the development process	Testing is carried out after the build phase

[7M]

2. Discuss in detail about DevOps eco system?

Ans: **DevOps ecosystem:** The DevOps ecosystem comprises various interconnected components, practices, and tools that streamline software development and deployment. It emphasizes collaboration, automation, and continuous improvement across the software development lifecycle.[2M]

Key components of the DevOps ecosystem:[8M]

- **Automation:** Automating repetitive tasks like testing, deployment, and monitoring reduces errors and accelerates software delivery.
- **CI/CD:** Continuous Integration (CI) involves frequently integrating code changes into a shared repository for early error detection. Continuous Delivery (CD) automates the deployment process, enabling frequent and reliable software releases.
- **Collaboration:** Fosters communication and teamwork between development and operations teams, ensuring everyone is aligned.
- **Configuration Management:** Ensures consistency and up-to-date systems and environments. Tools like Puppet and Ansible automate server and application configuration, reducing manual errors.
- **Containerization:** Packages applications and their dependencies into containers (e.g., Docker), ensuring portability and consistent deployment across different environments.
- **Infrastructure as Code (IaC):** Manages and provisions infrastructure resources using code and automation, providing greater agility, scalability, and consistency.
- **Microservices:** An architectural approach that breaks down applications into smaller, independent services for easier development, deployment, and scaling.
- **Monitoring and Health Checks:** Automated monitoring tools continuously check the performance, availability, and overall health of software systems and infrastructure, alerting teams to potential issues.
- **Version Control Systems (VCS):** Tools like Git track changes in source code, facilitate team collaboration, and maintain a history of modifications.

3.List and explain the steps followed for adopting DevOps in IT projects ?

Ans: Adopting DevOps in IT projects requires a structured approach, ensuring collaboration, automation, and continuous delivery.[1M]

Steps for Adopting DevOps[9M]

1. **Assess Current Processes :** Evaluate existing development and operations workflows. Identify areas needing improvement.
2. **Establish a DevOps Culture :** Encourage collaboration between development, operations, and QA teams. Promote shared responsibility and automation.
3. **Implement Version Control :** Use Git-based repositories for managing code.

Tools: GitHub, GitLab, Bitbucket.

4. Adopt Continuous Integration (CI) : Automate code integration and testing.

Tools: Jenkins, Travis CI.

5.Enable Continuous Deployment (CD) : Automate software release and deployment.

Tools: Docker, Kubernetes, AWS CodeDeploy.

6.Automate Infrastructure Management : Implement Infrastructure as Code (IaC).

Tools: Terraform, Ansible, Chef.

7.Integrate Monitoring & Logging : Use monitoring tools for performance tracking.

Tools: Prometheus, ELK Stack.

8.Implement Security Practices (DevSecOps): Automate security testing within CI/CD pipelines.

Tools: SonarQube, OWASP ZAP.

9.Measure & Improve : Continuously analyze performance and optimize processes. Use feedback loops for improvement.

Set 4

1. Explain the values and principles of Agile model. [10M]
2. Write a short notes on the DevOps Orchestration. [10M]
3. What is the difference between Agile and DevOps models? [10M]

ANSWERS

1.Explain the values and principles of Agile model?

Ans : **Agile model** : The Agile model is guided by twelve principles that promote a flexible, collaborative, and customer-focused approach to software development. These principles aim to delivery high-quality products.[1M]

The 12 Agile Principles:[9M]

1. **Customer satisfaction:** Deliver valuable software early and continuously to satisfy the customer.
2. **Embrace change:** Welcome changing requirements, even late in development, for the customer's competitive advantage.
3. **Frequent delivery:** Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. Agile

project management focuses on shortening the distance between planning and delivery.

4. **Collaboration:** Business people and developers must work together daily throughout the project.
5. **Motivated individuals:** Build projects around motivated individuals, giving them the environment, support, and trust to get the job done.
6. **Face-to-face conversation:** The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. **Working software:** Working software is the primary measure of progress. Agile focuses more on creating software rather than just planning for it.
8. **Sustainable development:** Agile processes promote sustainable development, where sponsors, developers, and users should be able to maintain a constant pace indefinitely. Project managers and leaders should set realistic and clear expectations.
9. **Technical excellence:** Continuous attention to technical excellence and good design enhances agility. Code should get better with each iteration.
10. **Simplicity:** Maximize the amount of work not done. Agile optimizes efficiency by emphasizing simplicity.
11. **Self-organizing teams:** The best architectures, requirements, and designs emerge from self-organizing teams.
12. **Regular reflection:** At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2. Write a short notes on the DevOps Orchestration?

Ans : **DevOps orchestration** is the automated coordination and management of tasks and processes in the software development and deployment lifecycle. [1M]

Key aspects: [4M]

- **Automation:** Orchestration relies on automation to reduce manual intervention, enabling rapid and error-free software deployment.
- **Coordination:** It manages the sequence and timing of automated tasks across multiple systems, ensuring they run in the correct order.
- **Consistency, Scalability, and Reliability:** Orchestration ensures these qualities across different environments

- **Goal:** The aim is to minimize production problems and shorten the time to introduce new products to the market

Benefits:[2M]

- Simplifies and manages CI/CD pipelines.
- Reduces manual activities, speeding up development cycles.
- Maximizes the potential of automation tools by ensuring functions work together

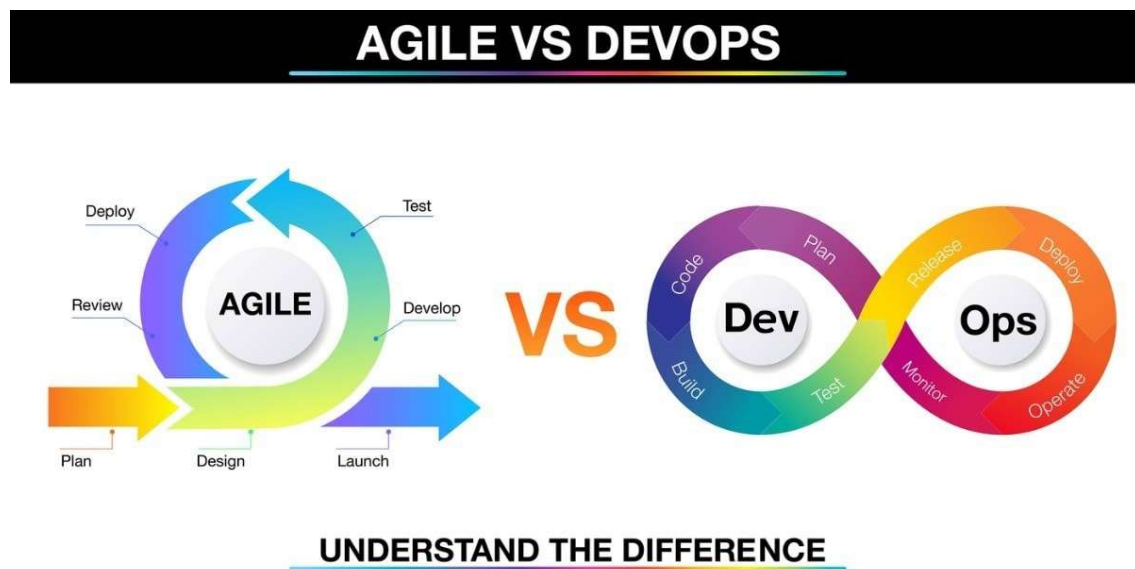
Tools:[2M]

- Containerization technologies like Docker.
- Orchestration platforms such as Kubernetes.
- Jenkins pipelines

In essence, DevOps orchestration streamlines complex workflows, enhances efficiency, and fosters collaboration between development and operations teams.[1M]

3.What is the difference between Agile and DevOps models?










Ans : Agile and DevOps are both approaches aimed at improving software development and delivery, but they operate at different levels and focus on different aspects of the process. Agile is a development *philosophy*, while DevOps is a *practice*. [2M]



[2M]

Difference Between Agile and DevOps

This template covers difference between agile and devOps based on some parameters such as Purpose, Task, Implementation, Target Areas, duration, team skill sets etc.

Parameter	Agile	DevOps
 What is it?	<ul style="list-style-type: none"> Agile refers to an iterative method that relies on teamwork, input from consumers, and limited, fast updates. 	<ul style="list-style-type: none"> DevOps is considered a method of grouping together teams for growth and operations.
 Purpose	<ul style="list-style-type: none"> Agile helps to manage complex projects. 	<ul style="list-style-type: none"> DevOps central concept is to manage end-to-end engineering processes.
 Task	<ul style="list-style-type: none"> Agile process focusses on constant changes. 	<ul style="list-style-type: none"> DevOps focuses on constant testing and delivery.
 Implementation	<ul style="list-style-type: none"> Under a number of tactical structures such as a jog, secure and scrum, agile approach can be applied. 	<ul style="list-style-type: none"> DevOps' main purpose is to concentrate on cooperation, since it has no widely agreed structure.
 Team Skill Set	<ul style="list-style-type: none"> Agile growth stresses educating all members of the organization to have a wide range of common and equivalent talents. 	<ul style="list-style-type: none"> The expertise set of the production and service teams is split and shared by DevOps.
 Team Size	<ul style="list-style-type: none"> The center of Agile is a tiny team. As the squad is smaller, the less people in it, the easier they can pass. 	<ul style="list-style-type: none"> Relatively larger team size as it involves all the stack holders.
 Duration	<ul style="list-style-type: none"> In units of "sprints," agile development is handled. With each sprint, this period is much shorter than a month. 	<ul style="list-style-type: none"> For major releases, DevOps looks for targets and metrics. The best aim is to distribute code to Everyday output or a few hours.
 Feedback	<ul style="list-style-type: none"> Feedback is given by the customer. 	<ul style="list-style-type: none"> Feedback comes from the internal team.
 Target Areas	<ul style="list-style-type: none"> Software Development 	<ul style="list-style-type: none"> End-to-end business solution and fast delivery.

[6M]