III. B.Tech I – Semester End Exams (Reg) Dec-2025

Devops Key

1. a) List any two advantages of Devops :

Ans —> Faster development and deployment applications.

> Faster response to the market changes to improve business growth.

> Improves customer experience and satisfaction.

b) Name any two Agile frameworks.

Ans. > Scrum methodology

> Kanban

> Extreme programming (XP)

> Lean software Development.

c) Define resilience in devops.

Ans. Devops resilience refers to the ability of a Devops system to withstand and recover from failures and disruptions. this means ensuring that the systems and processes used in Devops are robust, scalable, and able to adapt to changing conditions.

d) Mention any one benefit of using Devops in software development.

Ans > Faster software delivery.
> Improved collaboration and communication
> Enhanced software quality.
> Increased efficiency and productivity.

e) what is Docker?

Ans: Docker is an open-source containerization platform. that allows you to build, test, and deploy applications quickly. package applications and all their dependencies into standardized units called containers.

f) Define source code migration.

Ans: Source code management systems are used to manage code migrations, which are the process of moving code from one environment to another. where code is moved from a development environment to a testing environment and finally to a production environment.

g) Define Jenkins.

Ans: Jenkins is an open source automation server used for Continuous integration and continuous delivery to automate the building, testing and deployment of software projects.

h) what is the purpose of a host server?

Ans: host server runs applications, services, and the tools themselves, providing essential resources like CPU, memory and storage for development, testing and production. there are different types of host servers including physical, virtual and cloud-based infrastructure.

i) Name any two deployment tools.

Ans:
- > Jenkins
- > Circle CI
- > Travis CI
- > Docker
- > Ansible
- > Chef
- > Saltstack
- > Puppet.

J) Mention one advantage of automated testing.

Ans: > Automated testing has less chances of error hence more reliable.

> Automated testing takes far less resources in execution as Compared to manual testing.

> Automated tests can be run all time in 24x7.

2. Analyze a real-world example of Devops implementation highlighting challenges and outcomes.

Ans: Devops implementation for Netflix's example.

- Netflix transitioned from a DVD-by-mail service to a global video-streaming platform. This shift required high availability, massive scalability, and rapid feature delivery.

Implementation strategy:

1) Cloud & Infrastructure Automation.
   - migrated data centers to Aws.
   - used Infrastructure as Code to provision resources automatically.

2) Microservices Architecture.
   - Broke the monolithic system into hundreds of independent services.
   - Each service owned by a small team.
   - Allowed independent development, testing and deployment.

3) CI/CD pipelines.

- Automated build, test and deployment pipelines
- Developers could deploy changes frequently without centralized approval.
- Reduced deployment lead time significantly.

4) Resilience & Monitoring

- Built advanced monitoring and logging systems
- proactively identified weaknesses before real outages occurred.

Challenges:

- managing system complexity
- Reliability at scale.
- Skill Requirements

Out Comes:

- Deployment speed
- scalability & Availability
- operational Resilience
- Business Impact.

3) Explain the stages involved in the Release management process in Devops.

Ans: Release management is the process of planning, building, testing, deploying and monitoring software releases in a reliable and repeatable way.

> Release planning: It defines what will be released and when will be released.

- Define release goals and scope
- Identify features, bug fixes and improvements.
- Decide release schedule and milestones.
- Assess risks and dependencies.
- coordinate with development, testing and operations team.

> Source code management: All changes are managed using version control systems.

- Developers commit code changes.
- Branching and merging strategies
- Code reviews and approvals.

> Build and Continuous Integration : The application is built automatically whenever code changes are committed.

- Compile code
- Run automated unit tests
- Generate build artifacts
- Detects errors early.

> Testing and validation :- Ensures the release is stable and meets quality standards.

- Automation & Manual testing.
- Performance and security testing.
- User Acceptance Testing.

> Release Approval and Deployment : The application is deployed to target environments.

- Review test results.
- Compliance and security checks.
- Deploy to staging & production.
- Automate deployment using CI/CD pipelines.

> monitoring and feedback: Tracks system performance and user experience after release.

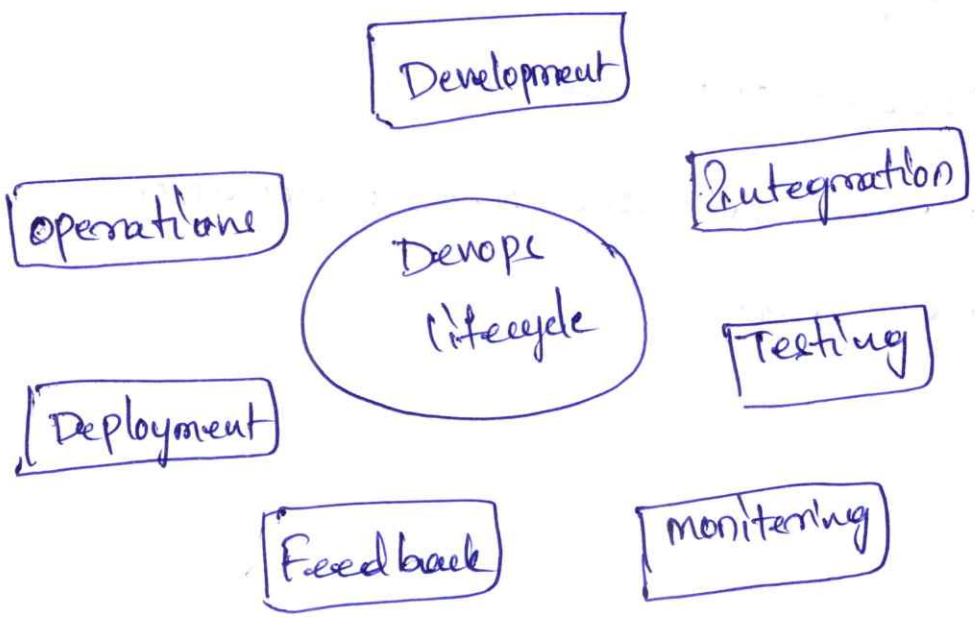
- monitor logs, metrics and alerts.
- Detect failures and performance issues
- Gather user feedback.

> Continuous Improvement: Evaluates the success of the release.

- Analyze issues and incidents.
- Conduct post-release review
- Improve processes and automation.
- plan next release.

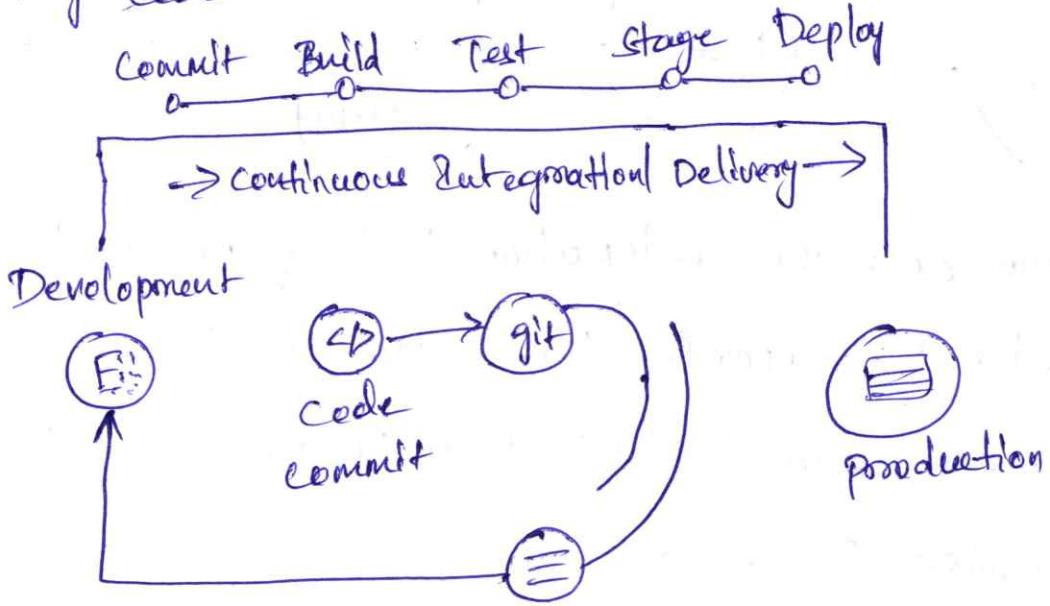
4) Explain the Devops lifecycle for business agility with a neat diagram and example.

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 beginning to the final stage of the product.



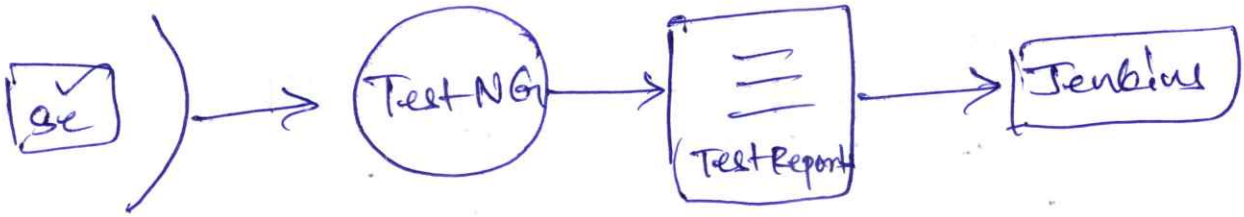
1) Continuous Development: This phase involves the planning and coding of the software. The vision of the project is decided during the planning phase. the developers begin developing the code for application. There are several tools for maintaining the code.

2) Continuous Integration : This stage is the heart of the entire Devops lifecycle. the developers require to commit changes to the source code more frequently. Every commit is built, and this allows early detection of problems if they are present. the code supporting new functionality is continuously integrated with the existing code.



Jenkins is a popular tool used in this phase. whenever there is a change in the Git Repository then Jenkins fetches the updated code and prepares a build of that code, which is an executable file in the form of 'Jar'. this build is forwarded to the test server or production server.

3) Continuous Testing: The developed software is continuously testing for bugs. For constant testing, automation testing tools such as TestNG, JUNit, Selenium. these tools allows to test multiple code bases thoroughly in parallel to ensure that there is no flaw in the functionality.

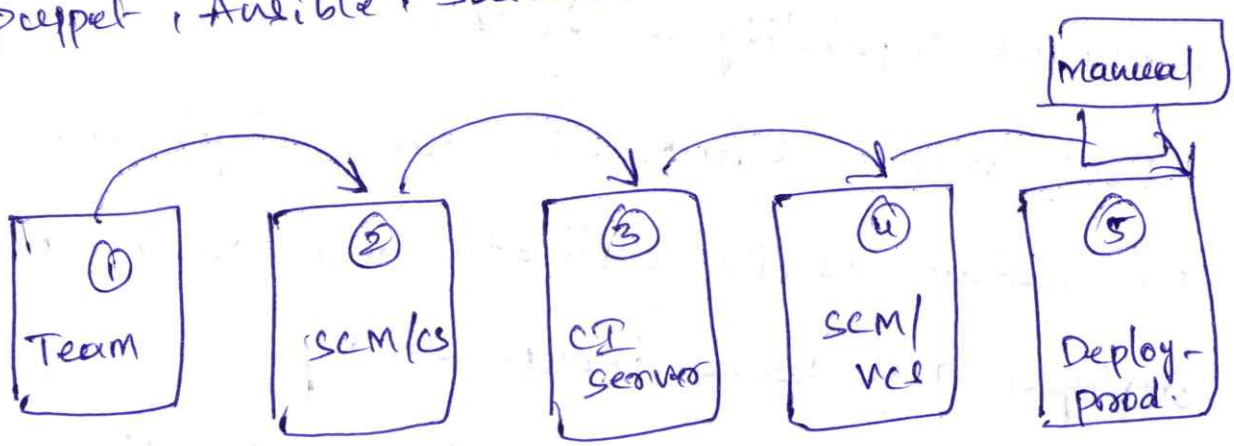


Selenium does the automation testing, and TestNG generates the reports. This entire testing phase can automate with the help of a continuous Integration tool called Jenkins.

4) Continuous Monitoring: Monitoring is a phase that involves all the operational factors of the entire Devops process, where important information about the use of software is recorded and carefully processed to find out trends and identify problem areas. the monitoring is integrated within the operational capabilities of the software application.

5) Continuous Feedback: the application development is consistently improved by analyzing the results from the operations of the software. This is carried out by placing the critical phase of constant feedback between the operations and the development of the next version of the current software application.

6) Continuous Deployment: the code is deployed to the production servers. Also it is essential to ensure that the code is correctly used on all the servers. the new code is deployed continuously. and configuration management tools play an essential role in executing task frequently and quickly. there are some popular tools which are used in this phase, such as chef, puppet, Ansible, saltstack.



Example: E-commerce Application.

5) Compare and contrast monolithic and microservices architectures with suitable examples:

Ans: monolithic architecture:- The entire program is constructed as a single invisible unit.

- Any changes or updates to the application require modifying and redeploying the entire monolith.
- monolithic architectures are often characterized by their simplicity and ease of development, especially for small to medium-sized applications.
- they can become complex and difficult to maintain as the size and complexity of the application grows.

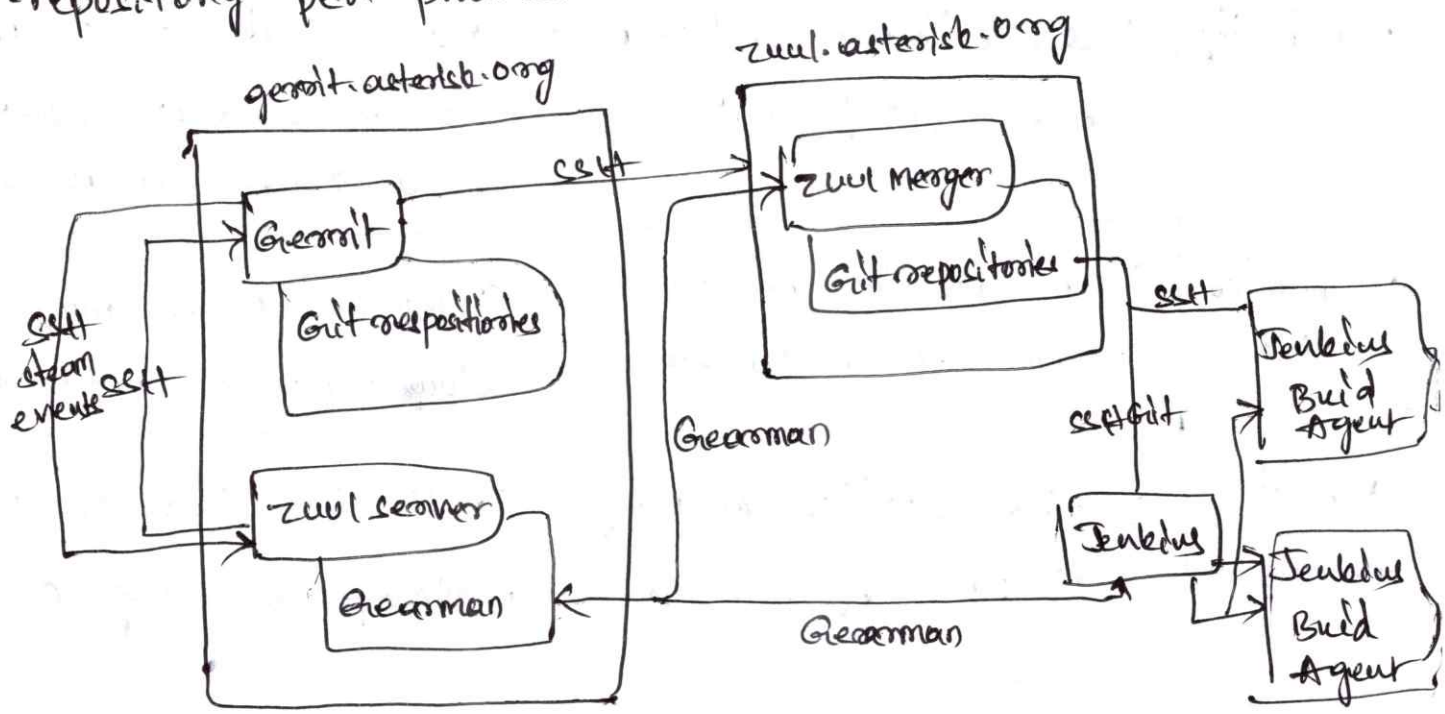
Microservices architecture:- An application designed as a set of small, independent services. Each one represents a business capability in itself.

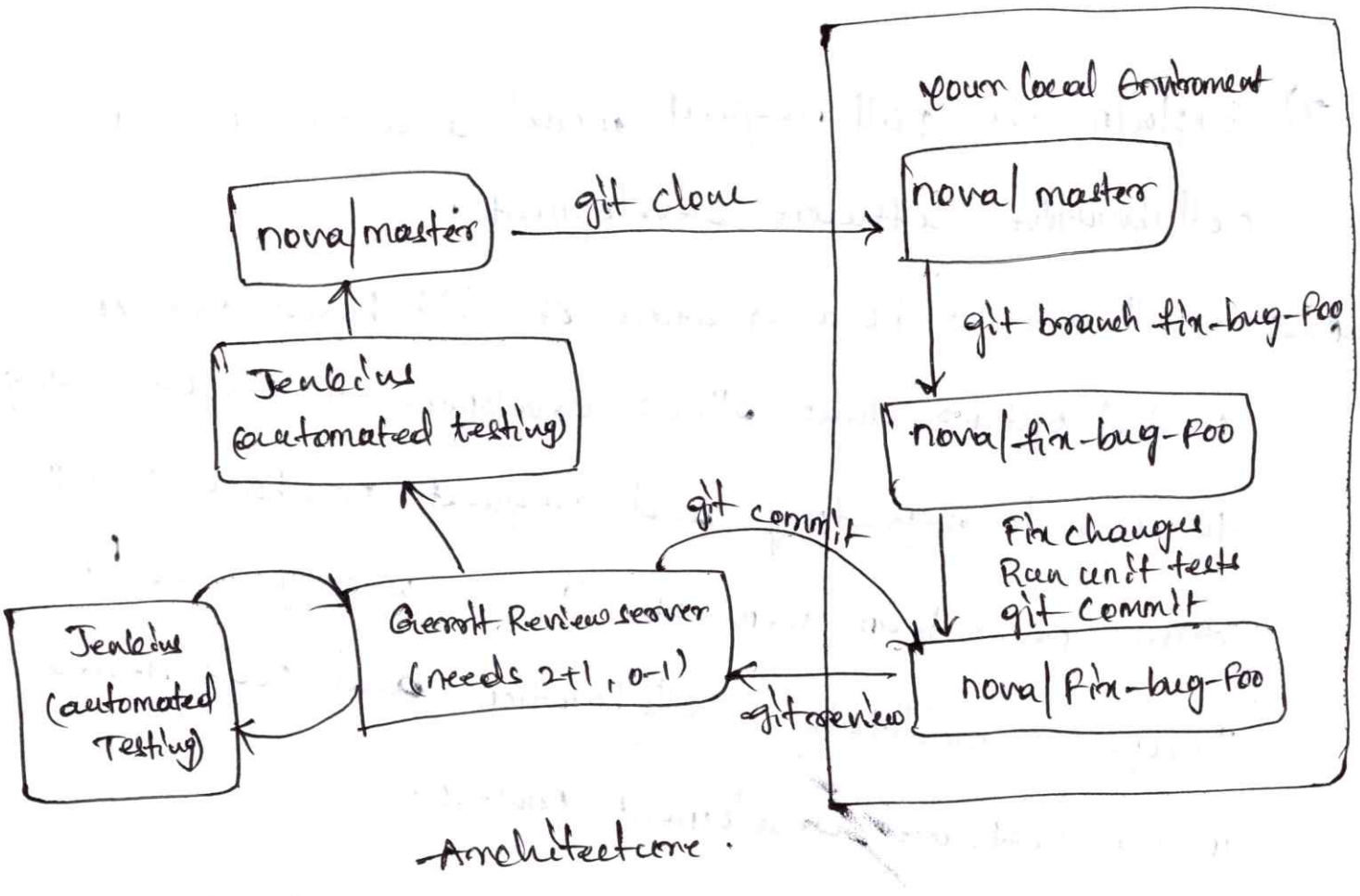
- Each service is responsible for a single functionality or feature of the application and can be developed, deployed and scaled independently.
- The microservices architecture has a significant impact on the relationship between the application and the database.

Aspect	Monolithic Architecture	Microservice Architecture
Architecture	Single tier architecture	Multi tier architecture
Size	Large, all components tightly coupled	Small, loosely coupled components
Deployment	Deployed as a single unit	Individual services can be deployed independently
Scalability	Horizontal scaling can be challenging	Easier to scale horizontally
Development	Development is simpler	Complex due to managing multiple services
Technology	Limited technology choices.	Freedom to choose the best Technology for each service.
maintenance	Easier to maintain due to its simplicity	Requires more effort to manage multiple services
Flexibility	less flexible as all components are tightly coupled	More flexible as components can be developed, deployed and scaled independently
Communication	Communication between Components is faster.	Communication may be slower due to network calls
Fault Tolerance	Entire application may fail if a part fails.	Individual services can fail without affecting others.

6) Discuss the workflow of Gerrit and explain how it supports code review.

Ans: Gerrit is a web based code review tool which is integrated with Git and built on top of Git version control system. It allows to merge changes to Git repository when you are done with the code reviews. Gerrit is an exceptionally extensible and configurable apparatus for online code survey and storehouse the executives for projects utilizing the Git control framework. It is used to store the merged code base and the changes under review that have not being merged yet. Gerrit has the limitation of a single repository per project.





Architecture.

- The code review process allows new comers to see the code of other more experienced developers.
- Developers can get feedback on their suggested changes.
- Experienced developers can help to evaluate the impact on the whole code.
- shared code ownership: by reviewing code of other developers the whole team gets a solid knowledge of the complete code base.

7) Explain the pull request model and its role in collaborative software development.

Ans: Pull request is a feature of Git based version control systems that allows developers to propose changes to a Git repository, and request feedback or approval from other team members. It is widely used in Devops to facilitate collaboration and code review in the software development process.

In the pull request model, a developer creates a new branch in a Git repository, makes changes to the code, and then opens a pull request to merge the changes into the main branch. Other team members can then review the changes, provide feedback, and approve or reject the request.

Pull requests essentially provide convenient tooling for a development workflow that existed in many open-source projects. This workflow begins with a contributor creating a new logical branch, either by starting a new branch in the central repository,

cloning into a personal repository on both. The contributor then works on that branch, typically in the style of a feature branch, pulling any updates from mainline into their branch. When they are done they communicate with the maintainer of the central repository indicating that they are done.

Github's pull request mechanism makes this flow much easier. It keeps track of the clones through its fork mechanism, and automatically creates a message thread to discuss the pull request, together with behavior to handle the various steps in the review workflow.

The pull request model provides Benefits.

Improved Code Quality: pull requests encourage collaboration and code review, helping to catch potential bugs and issues before they make it into the main codebase.

Increased transparency: pull requests provide a clear audit trail of all changes made to the code, making it easier to understand how code has evolved over time.

Better collaboration: pull request allow developers to share their work and get feedback from others, improving collaboration and communication within the development team.

The pull request model is an important tool in the Devops toolkit, helping to improve the quality, transparency and collaboration of software development processes.

8) Compare alternative build servers to Jenkins and discuss their strengths and weaknesses.

Ans: Jenkins is a widely used open source CI/CD tool. Several alternative build servers offer better cloud integration, ease of use, and scalability.

Alternative Build servers:

> Gitlab CI/CD: Gitlab CI/CD is a part of Gitlab, a comprehensive Devops platform that provides version control, issue tracking, and CI/CD pipelines in a single application. Gitlab allows teams to define their build, test and deploy processes using a simple YAML file.

> Circle CI: Circle CI is a CI/CD platform designed to help teams automate their build, test and deployment process. It supports various languages and can be integrated with Github, Bitbucket and Gitlab.

> Travis CI: Travis CI is a hosted CI/CD service that integrates seamlessly with Github. It allows developers to test and deploy their code automatically upon each commit

> Github : Github is a CI/CD service allowing users to automate their workflows directly within their Github repositories.

> Aws codepipeline CI/CD: Aws code pipeline is a fully managed CI/CD service provided by Amazon web services. It enables users to automate their build, test, and deploy processes using Aws services.

> Azure pipelines : Azure pipeline CI/CD solution within microsoft Azure that supports multiple platforms and programming languages.

> Bitbucket pipelines : A CI/CD solution within Bitbucket that allows for pipeline creation and management within the code repository.

9) Describe how quality measures are collated and monitored during the build process.

Ans: In Devops, collating quality measures is an important part of the continuous improvement process. the following are some common quality measures used in Devops to evaluate the quality of software systems.

> Continuous Integration metrics: metrics that track the success rate of automated builds and tests, such as build duration and test pass rate.

> Continuous Deployment metrics: metrics that track the success rate of deployments, such as deployment frequency and time to deployment.

> Code review metrics: metrics that track the effectiveness of code reviews, such as review completion time and code review feedback.

> performance metrics: measures of system performance in production, such as response time and resource utilization.

> User experience metrics: measures of how users interact with the system, such as click-through rate and error rate.

> Security metrics: measures of the security of the system, such as the no. of security vulnerabilities and the frequency of security updates.

> Incident response metrics: metrics that track the effectiveness of incident response, such as mean time to resolution and incident frequency.

By regularly collating these quality measures, Devops teams can identify areas for improvement, track progress over time, and make informed decisions about the quality of their systems.

10) Compare Test-Driven Development (TDD) and REPL driven development.

Ans:

Aspect	Test Driven Development	REPL Driven Development
Definition	TDD is a software development approach where tests are written before the actual code. Development follows a strict Red-Green-Refactor cycle.	REPL driven development is an interactive approach where developers write and test code incrementally in a Read-Eval-print loop environment
Code area	Test first, then code	code first, test interactively
Development	structured and planned	Exploratory and interactive
Feedback speed	slower	Immediate feedback.
Automation	Fully automated tests	mostly manual experiment
Error detection	Early and systematic	Early but informal
Refactoring	Strong	Limited.
Documentation	Tests act as documentation	minimal documentation.
Suitability	Large, long term projects	Prototyping, learning

11) Explain various types of testing used in software development.

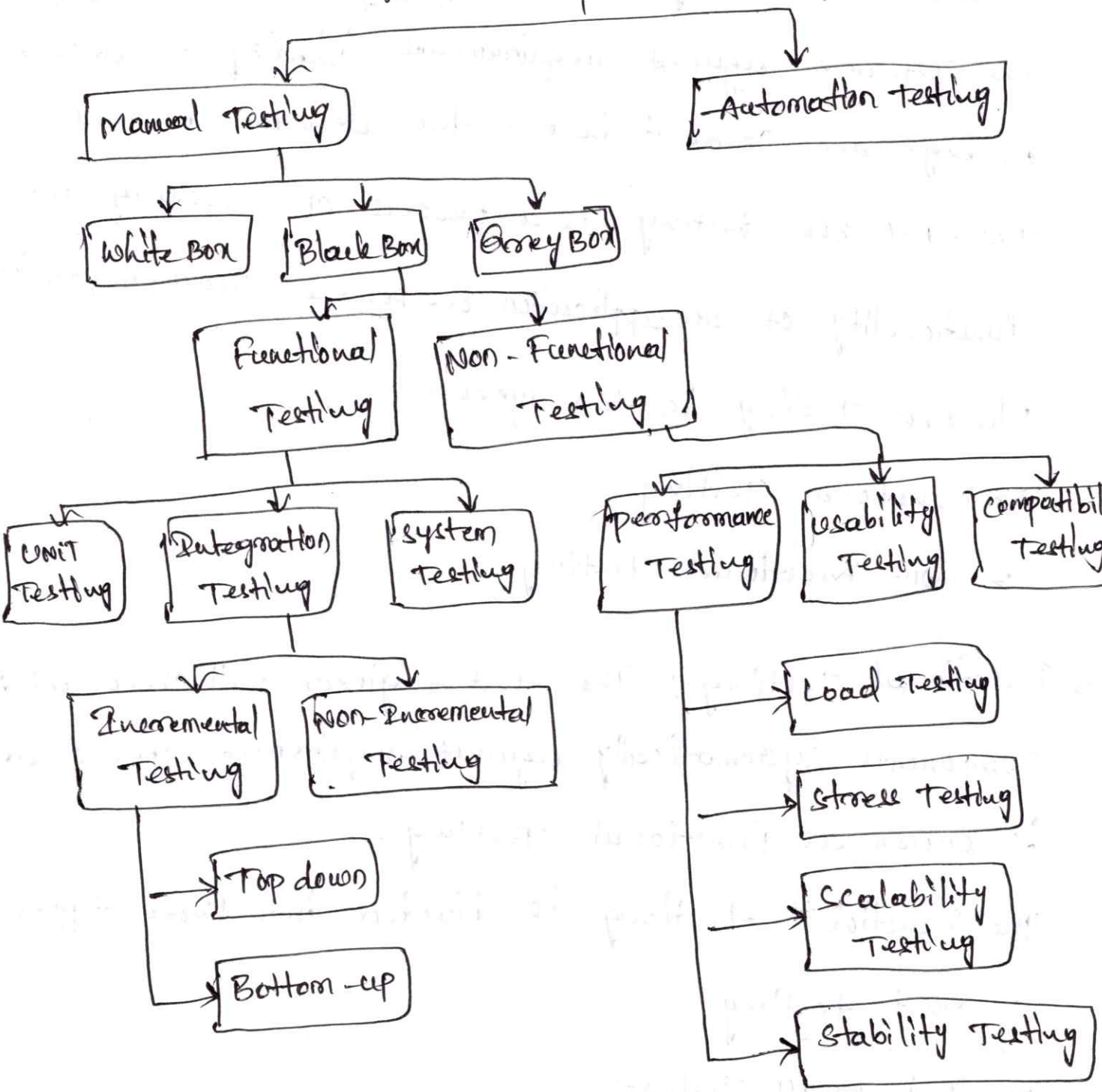
Ans: Software testing is the process of verifying and validating that a software application works as expected and meets user requirements. Different types of testing are used at different stages of the software development lifecycle to ensure quality. The software testing divided into two parts.

> Manual Testing
> Automation Testing.

> Manual Testing: Testing any software or an application according to the client's needs without using any automation tool is known as Manual Testing. Manual testing can be further classified into three different types of testing.

- white Box Testing.
- Black Box testing
- Grey Box Testing.

Types of software Testing.



> White Box Testing: The developer will inspect every line of code before handling it over to the testing team. The purpose of this testing is to emphasize flow of inputs and outputs over the software.

> Black Box Testing: The test engineer will analyze the software against requirements, identify the defects or bugs.. and sends it back to the development team.

The black box testing is a process of checking the functionality of an application as per the customer requirement

Blackbox Testing is two types:

> Functional Testing

> Non-Functional Testing.

> Functional Testing: The test engineer will check all the components systematically against requirement specification is known as Functional Testing.

The Functional testing is divided into three types

> unit testing

> Integration Testing

> System Testing.

> Non-Functional Testing: It provides detailed information on software product performance and used technologies. Non-functional testing will help us minimize the risk of production and related costs of the software.

Non functional testing categorized into different parts.

- > performance testing
- > Usability Testing
- > Compatibility Testing.

> Grey Box Testing : It is a collaboration of black box and white box testing. It includes access to internal coding for designing test cases. Grey box testing is performed by a person who knows coding as well as testing.

> Automation Testing : It uses specific tools to automate manual design test cases without any human interference. It is used to re-run the test scenarios, which were executed manually, quickly and repeatedly. Automation testing is the best way to enhance the efficiency, productivity, and coverage of software Testing.