DAY 5 Assignment:

Q) 1) Write 2 scripts for daily standup meeting

Script 1: Software Development Team

Team Lead (Mayur):

"Good morning, everyone. Let's get started with our daily standup. As usual, we'll go around and each person will share what they did yesterday, what they're working on today, and if they have any blockers. Let's keep it brief. Jyotsna, why don't you start?"

Jyotsna (Developer):

"Thanks, Mayur. Yesterday, I finished implementing the user authentication feature and started on the unit tests for it. Today, I'll be working on integrating the authentication with the front-end. No blockers at the moment."

Mayur:

"Great, thanks Jyotsna. Shivanya, you're up next."

Shivanya (Developer):

"Yesterday, I fixed the bug in the payment processing module and reviewed Jyotsna's authentication code. Today, I'll be working on the new reporting feature. I do have a blocker: I'm waiting on the API documentation from the backend team."

Mayur:

"Got it, Shivanya. I'll follow up with the backend team to get that documentation for you. Swami, how about you?"

Swami (Tester):

"Yesterday, I completed the regression tests for the latest build. Today, I'll be focusing on writing test cases for the new reporting feature. No blockers from my side."

Mayur:

"Thanks, Swami. And finally, Sanvi?"

Sanvi (Designer):

"Yesterday, I finished the UI design for the new dashboard. Today, I'll start working on the user flow for the new onboarding process. No blockers."

Mayur:

"Awesome, thanks Sanvi. Sounds like we're all set for today. If anyone encounters any issues, don't hesitate to reach out. Have a productive day, everyone!"

Script 2: Software Development Team (IT Infrastructure)

Team Lead (Rahul):

"Good morning, team. Let's start our daily standup. As always, we'll go around and share what we did yesterday, what we're working on today, and any blockers. Let's keep it brief and focused. Mohan, could you start us off?"

Mohan (DevOps Engineer):

"Sure, Rahul. Yesterday, I finished setting up the new CI/CD pipeline and tested it with our staging environment. Today, I'll be migrating our current builds to the new pipeline. No blockers from my end."

Rahul:

"Excellent, Mohan. Neha, you're next."

Neha (System Administrator):

"Yesterday, I completed the server migration to our new cloud provider and configured the firewall settings. Today, I'll be monitoring the server performance and starting the setup of our new database cluster. No blockers."

Rahul:

"Thanks, Neha. Gaurav, how about you?"

Gaurav (Network Engineer):

"Yesterday, I resolved the network latency issues and updated our network architecture documentation. Today, I'll be focusing on improving our network security protocols and starting a network audit. I have a blocker: I need access to some network logs from the security team."

Rahul:

"Understood, Gaurav. I'll coordinate with the security team to get you access to those logs. Varsha, your turn."

Varsha (Database Administrator):

"Yesterday, I optimized our database queries and performed routine maintenance. Today, I'll be working on the database backup strategy and helping Neha with the database cluster setup. No blockers."

Rahul:

"Great, Varsha. And finally, Raj?"

Raj (Security Specialist):

"Yesterday, I conducted a security audit on our new infrastructure and identified a few vulnerabilities. Today, I'll be working on patching those vulnerabilities and reviewing our overall security policies. No blockers."

Rahul:

"Thanks, Raj. It sounds like we're all set for today. If any issues arise, please let me know as soon as possible. Have a productive day, everyone, and we'll reconvene tomorrow."

Q) 2)Explain the responsibility of scrum roles: scrum master, product owner, development team.

In a Scrum framework, there are three primary roles: Scrum Master, Product Owner, and Development Team. Each role has distinct responsibilities that contribute to the success of the Scrum process. Each role is essential to the success of the Scrum framework. The Scrum Master ensures the process runs smoothly, the Product Owner ensures the team is building the right product, and the Development Team ensures the product is built correctly and incrementally. Together, they help deliver valuable and high-quality products.

A) Scrum Master :

The Scrum Master is responsible for ensuring that the Scrum framework is understood and enacted. They serve as a facilitator and coach for the team, helping them adhere to Scrum practices and principles. Key responsibilities include:

Facilitating Scrum Events: Ensuring that Scrum ceremonies (Daily Standups, Sprint Planning, Sprint Reviews, and Sprint Retrospectives) occur and are effective.

Removing Impediments: Identifying and removing obstacles that hinder the team's progress.

Coaching the Team: Guiding the team and organization on how to use Scrum effectively, and helping them improve their processes and practices.

Protecting the Team: Shielding the team from outside interruptions and distractions to maintain focus on the Sprint goals.

Fostering Collaboration: Encouraging collaboration within the team and with stakeholders.

B) Product Owner :

The Product Owner is responsible for maximizing the value of the product resulting from the work of the Development Team. They manage the product backlog and ensure that the team is working on the highest-priority tasks. Key responsibilities include:

Defining Product Vision: Clearly communicating the product vision and goals to the team and stakeholders.

Managing the Product Backlog: Creating, refining, and prioritizing the product backlog to ensure it is up-to-date and reflects the current needs of the project.

Stakeholder Management: Acting as the primary point of contact for stakeholders, gathering requirements, and ensuring their needs are met.

Defining Acceptance Criteria: Providing clear acceptance criteria for backlog items to ensure that the team understands what is needed for each task.

Making Decisions: Being available to make quick decisions about the direction of the product and clarifying requirements.

C) Development Team :

The Development Team is a group of professionals who do the work of delivering a potentially releasable increment of "Done" product at the end of each Sprint. They are self-organizing and cross-functional, meaning they collectively have all the skills necessary to create the product increment. Key responsibilities include:

Delivering Increments: Building and delivering product increments that meet the definition of "Done" at the end of each Sprint.

Self-Organizing: Managing their own work and collaborating without needing to be directed by others.

Continuous Improvement: Continuously improving their processes, skills, and quality of work through regular retrospectives and feedback.

Sprint Planning and Execution: Participating in Sprint Planning to commit to the work for the Sprint and then executing that work during the Sprint.

Collaborating: Working closely with the Product Owner and Scrum Master to ensure that the product backlog is understood and that obstacles are removed.

Each role is essential to the success of the Scrum framework. The Scrum Master ensures the process runs smoothly, the Product Owner ensures the team is building the right product, and the

Development Team ensures the product is built correctly and incrementally. Together, they help deliver valuable and high-quality products.

********** LINUX **********

Q) 1) Write some 30 commands in Linux

1) ls: The ls command is used to list files and directories in the current working directory.

2) pwd: The pwd command allows you to print the current working directory on your terminal.

3) cd: The command is used to Change the current directory.

Syntax: $ cd <directory path>

4) mkdir: The mkdir command allows you to create directories from within the terminal.

Syntax: $ mkdir <folder name>

5) cp: Copy files or directories.

Syntax: $ cp <source> <destination>

6) mv: Move or rename files or directories.

Syntax: $ mv <source> <destination

7) rm: The rm command is used to delete files and folders.

Syntax: $ rm <file name>

8) touch: The touch command in Linux creates an empty file or updates the timestamp of an existing file.

Syntax: $ touch <file name>

9) cat: Concatenate and display the contents of a file.

SYntax: $ cat <file name>

10) ln: To create a link to another file, we use the ln command.

Syntax: $ ln -s <source path> <link name>

11) clear: The clear command in Linux clears the terminal screen. It removes all the text and output currently displayed on the terminal.

Syntax: $ clear

12) echo: when we use echo command, it simply prints whatever follows after the command.

Syntax: $ echo

13) less: View file contents one screen at a time, with backward navigation.

Syntax: $ less <file name>

14) whoami: The whoami command in Linux returns the current user's username.

Syntax: $ whoami

15) head: Display the first few lines of a file.

Syntax: $ head file_name

16) tail: Display the last few lines of a file.

Syntax: $ tail file_name

17) find: Search for files in a directory hierarchy.

Syntax: $ find /path -name file_name

18) grep: The grep command which reflects its ability to search for regular expressions across multiple lines and files.

Syntax: $ grep

19) du: Estimate file space usage.

Syntax: $ du -h

20) df: Report file system disk space usage.

Syntax: $ df -h

21) chmod: Change file modes or Access Control Lists.

Syntax: $ chmod 755 <file name>

22) chown: Change file owner and group.

Syntax: $ chown user:group file_name

23) kill: Terminate a process.

Syntax: $ kill process_id

24) history: Show command history.

Syntax: $ history

25) sort: The sort command is used to sort lines in a text file or standard input in Linux and Unix-based operating systems.

Syntax: $ sort <filename>

26) export: The export command in Linux and Unix-based operating systems is used to set environment variables.

Syntax: $ export <variable name>=<value>

27) ssh: The ssh command in Linux and Unix-based operating systems establishes a secure shell connection to a remote server. The command provides a secure encrypted connection between the local and remote servers.

Syntax: $ ssh username@remote-server

28) ifconfig: The ifconfig command will give you the list of all the network interfaces along with the IP addresses, MAC addresses and other information about the interface.

Syntax: $ ifconfig

29) wget: If you want to download a file from within the terminal, the wget command is used.

Syntax: $ wget <link to file>

30) sudo: This command is equivalent to logging in as root.

Syntax: $ sudo

Q) 3) Explain absolute and relative path?

Absolute Path: An absolute path is a complete path from the root directory (/) to the target file or directory. It always starts with a forward slash (/), which signifies the root directory.

Example: /home/user/documents/report.txt

/home/user/documents/report.txt is an absolute path.

It starts from the root directory (/) and specifies each directory in the hierarchy until it reaches the target file.

Characteristics:

a)Absolute paths are unambiguous and unique.

b)They can be used from anywhere in the filesystem to refer to a specific file or directory.

Relative Path: A relative path is a path that starts from the current working directory and does not start with a forward slash (/). It specifies a location relative to the current directory.

Example: Assume the current directory is /home/user.

documents/report.txt is a relative path.

It starts from the current directory (/home/user) and specifies the documents subdirectory and the report.txt file within it.

Characteristics:

a)Relative paths are context-sensitive. They depend on the current working directory.

b)They are shorter and often more convenient to use when working within a specific part of the filesystem.

b)They are shorter and often more convenient to use when working within a specific part of the filesystem.