**Soft Skills**

QA Question

### General Questions:

1. Tell me about yourself?

My name is Adina, and I'm excited to be here today to talk about my background and experience in software testing with 5 years of experience in the field. Throughout my career, I've honed my skills in test planning, test case development, execution, and reporting. I'm well-versed in various testing methodologies, including Agile and Waterfall, and have experience working with a range of tools and technologies. My expertise includes Java, Selenium, JIRA,

MySQL , CI/CD ,

1. Tell me about your project?

In my last project, I was part of a team responsible for developing and automating the testing framework for a web application designed to streamline the customer onboarding process. The main objective was to improve the efficiency and accuracy of our testing efforts while supporting rapid development cycles.

As a QA Automation Tester, my key responsibilities included designing automated test scripts using Selenium and Java, developing a testing framework, and seamlessly integrating it with our (CI/CD) pipeline using Jenkins. I collaborated closely with the development team to understand the application’s functionality and create comprehensive test cases that covered both functional and regression testing.

One of the challenges we faced was the dynamic nature of the application, as it underwent frequent changes based on user feedback and stakeholder requirements. To tackle this, I implemented Page Object Model (POM) design principles, which helped make the test scripts more maintainable and scalable. As a result of our automation efforts, we were able to significantly reduce the testing time by about 30%, allowing for quicker releases and more frequent iterations of the application. This not only improved our team’s productivity but also enhanced the overall quality of the application, leading to positive feedback from both stakeholders and end-users.

This project taught me valuable lessons about the importance of collaboration between QA and development teams, as well as the need for flexibility in adapting our testing processes to align with agile methodologies. I’m looking forward to bringing these experiences and skills into future projects!

1. Tell me about framework?

My automation framework is designed to streamline the testing process and ensure that automation is efficient, maintainable, and scalable.

I primarily utilize the \*\*Page Object Model (POM)\*\*, which helps keep the code organized and enhances maintainability. In this approach, each web page in the application is represented by a separate class, which contains all the elements and methods related to that page. This means that if the UI changes, we only need to update the page class rather than modifying multiple test scripts.

---------------------

I'm using \*\*Selenium WebDriver\*\* for browser automation along with \*\*TestNG\*\* for managing the test execution. This combination allows for effective test case management, asynchronous test execution, and assertions, which are crucial for validation.

To handle dependencies and streamline builds, I integrate \*\*Maven\*\* in the project. This ensures that our environment is consistent across different machines and simplifies the process of adding new libraries and managing versions.

One of the standout features of my framework is its integration with \*\*Jenkins\*\* for Continuous Integration. With Jenkins, the automated tests run every time there’s a code commit, providing immediate feedback to the developers. This helps us catch any issues early in the development process.

Another key aspect is the \*\*data-driven testing\*\* capability. I use \*\*Excel\*\* and \*\*JSON\*\* files to feed different data sets into the same test cases, allowing us to test a variety of scenarios without duplicating the script. This enhances our test coverage significantly.

Lastly, my framework is built with modularity in mind. Each test case is designed to be reusable, which makes it easy to create new tests and reduces the effort when updating existing ones.

1. **How did you start your career?**

I have to admit, starting my career was a bit challenging for me. There were times when I felt lost and uncertain about what path to take, especially when I saw many of my friends landing jobs at top companies like Google and Chase. I often found myself questioning my direction and abilities.

However, reflecting on my background, it's clear that my passion for computers has always been there. Ever since I was in school, I was drawn to technology and fascinated by how things worked. This interest sparked my desire to pursue a career in the tech industry.

Fortunately, my friends have been incredibly supportive, encouraging me to try my hand at opportunities even when I felt uncertain. Their belief in me motivated me to take action and seek internships and projects that aligned with my interests. I started to explore roles that allowed me to combine my passion for computers with practical experience, which helped me build my confidence.

Ultimately, I’ve learned that while it’s normal to feel unsure at times, surrounding myself with supportive friends and reflecting on my interests can guide me toward a fulfilling career. Now, I’m eager to take the next step and apply what I’ve learned in a new role

1. **What is the difference between QA, QC, and testing?**
2. **What is the role of a QA Automation Tester in the software development lifecycle?**

**### Technical Questions:**

1. **Can you explain the difference between functional and non-functional testing?**
2. **What is the Page Object Model (POM) and how have you implemented it?**
3. **How do you decide which test cases to automate?**
4. **What programming languages are you comfortable with for writing test scripts?**
5. **What automation tools have you used, and which do you prefer? Why?**
6. **How much time you test regression take ?**
7. **What is the Code Covarege ?**
8. **What domains do you have experience?**

I have experience primarily in two key domains: \*\*e-commerce\*\* and \*\*educational bootcamps\*\*.

1. In the e-commerce domain, I've worked on various online shopping platforms where I contributed significantly to the testing of website functionality, payment gateways, and user experience. My role involved automating regression tests to ensure critical features such as product search, cart functionality, and order processing operated smoothly. I used Selenium WebDriver in conjunction with TestNG to create robust test scripts. Additionally, I was involved in validating the integration with third-party services, such as payment and shipping APIs, to ensure that transactions were seamless and reliable. This experience helped me develop a keen eye for both functionality and user experience, as I focused on ensuring that the platform not only worked correctly but also provided an intuitive shopping experience for users.

2. Alongside my e-commerce experience, I've also worked in the context of educational bootcamps. Here, I focused on testing learning management systems and platforms designed to facilitate online education. My role included validating features such as user registration, course enrollment, content delivery, and progress tracking. I helped implement automated tests to ensure that new content and features did not disrupt the existing functionality. This experience was particularly fulfilling, as it allowed me to contribute to a platform that helps individuals learn new skills and advance their careers. I appreciated the iterative nature of development in this environment, which reinforced the importance of continuous testing and feedback.

Overall, my experience in both e-commerce and educational bootcamps has provided me with unique insights into different user interactions and requirements, enabling me to adapt my testing strategies effectively and ensure high-quality software in each context."

**### Company Questions:**

1. **How do you ensure the reliability and maintainability of your test scripts?**
2. **What is a CI/CD pipeline, and how does automation testing fit into it?**
3. **How do you handle synchronization issues in automation?**
4. **Describe the automation testing process you follow**
5. **Can you tell me about company?**

E-commerce Experience:

In the e-commerce domain, I contributed to various online shopping platforms, playing a crucial role in testing website functionality, payment gateways, and overall user experience. My responsibilities included automating regression tests to ensure that critical features—such as product search, cart functionality, and order processing—operated smoothly. I utilized Selenium WebDriver along with TestNG to develop robust test scripts, which greatly enhanced our testing efficiency. Additionally, I was involved in validating the integration with third-party services, including payment and shipping APIs, to ensure seamless and reliable transactions.

This role not only allowed me to enhance my technical skills but also developed my attention to detail, as I had to ensure that the platform functioned correctly while providing an intuitive shopping experience for users. By critically assessing both functionality and user experience, I contributed to the overall success of the platform, which ultimately aimed to deliver a positive and efficient online shopping journey.

Educational Bootcamps Experience:

Alongside my e-commerce experience, I also worked within the context of educational bootcamps, focusing on testing learning management systems designed for online education. My role involved validating key features such as user registration, course enrollment, content delivery, and progress tracking. I implemented automated tests to ensure that the introduction of new content and features would not disrupt existing functionality, which was vital for maintaining a smooth user experience for learners.

This experience was particularly fulfilling for me, as I was contributing to a platform that empowers individuals to learn new skills and advance their careers. I appreciated the iterative nature of development in this environment, which emphasized the importance of continuous testing and feedback. Working closely with development teams, I learned how pivotal it is to adapt quickly and respond to user needs in a fast-paced educational landscape.

1. **Team structure in your company?**

Our team was structured into two main branches: Functional Testing and Automation Testing. The Functional Testing team was responsible for manual testing, while the Automation Testing team focused on developing automated tests using tools like Selenium and Appium.

We worked closely with cross-functional teams, including Development, Product Management, and Design, to ensure that our testing efforts aligned with the project goals and requirements. Our team was also involved in Agile ceremonies like daily stand-ups, sprint planning, and retrospectives to ensure seamless communication and collaboration.

1. **Which methodology work in your work?**

I've had the opportunity to work with various methodologies throughout my career, but my primary experience is with Agile. In my previous role at previos Company, we followed a Scrum framework for our development and testing processes.

In Scrum, our QA team was responsible for creating test plans, test cases, and test data in collaboration with the Development team. We also participated in sprint planning to prioritize testing tasks and ensured that our testing efforts aligned with the sprint goals.

I've also worked with Waterfall methodology in the past, where we would receive detailed requirements documentation upfront and then plan and execute our testing efforts accordingly.

However, I've found that Agile provides a more flexible and iterative approach to testing, allowing us to adapt quickly to changing requirements and prioritize testing tasks based on business needs.

1. **Why did you apply for this job?**

I'm excited to share that I'm passionate about testing and quality assurance, and I'm always on the lookout for new challenges and opportunities to grow. When I saw the job posting for [Job Title] at [Company], I was immediately drawn to the company's mission and values. I was impressed by the company's commitment to innovation, customer satisfaction, and employee development.

As a tester, I'm always eager to work with new technologies, methodologies, and teams. The job description highlighted the opportunity to work on cutting-edge projects, collaborate with cross-functional teams, and contribute to the development of high-quality software products. I was also intrigued by the company's emphasis on continuous learning, professional growth, and employee empowerment.

I applied for this job because I believe it would be an incredible opportunity to leverage my skills, experience, and passion for testing to make a meaningful impact at [Company]. I'm excited about the prospect of working with a talented team of professionals who share my enthusiasm for quality and innovation.

1. **Why are you leaving from your work?**

I've enjoyed my time at [Previous Company], and I appreciate the opportunities I've had to grow as a tester. However, after careful consideration, I've decided it's time for me to move on to a new challenge that aligns better with my long-term career goals.

In my current role, I've been working primarily on manual testing, which is a vital part of our team's responsibilities. While I've enjoyed the opportunity to work on various projects and collaborate with colleagues, I feel that my skills and expertise are not being fully utilized. The company has undergone significant changes in recent months, and unfortunately, the testing team has been impacted by these changes.

I've come to realize that I'm looking for a more dynamic environment that allows me to work on automation testing, leverage my skills in Agile methodologies, and contribute to the development of innovative software products. I'm excited about the prospect of joining [Company] and being part of a team that shares my passion for quality assurance and innovation.

**### Tools and Technologies:**

1. **What issues have you faced when introducing automation into a project?**
2. **Have you worked with any test management tools? If so, which ones?**
3. **What is the biggest (last) bug you find?**

Ah, yes! One of the most significant bugs I found recently was during a regression testing phase for a mobile app. The issue was that when users attempted to make a payment using their credit card, the app would crash immediately after submitting the payment information. The error was intermittent, making it challenging to reproduce consistently.

After conducting thorough investigation and analysis, I discovered that the issue was caused by a subtle interaction between the app's backend API and the third-party payment gateway. Specifically, the API was sending an invalid request parameter to the gateway, which was resulting in an unexpected response that crashed the app.

To resolve the issue, I worked closely with the development team to modify the API request to send the correct parameter values. We also implemented additional logging and error handling mechanisms to detect similar issues in the future.

This bug fix had a significant impact on user experience, as it ensured that users could successfully complete transactions without encountering errors. It was a great example of how attention to detail and collaboration between teams can lead to tangible improvements in product quality.

1. **Most impact bug you found?**

One of the most impactful bugs I discovered was during a critical testing phase of an e-commerce application. The bug involved the checkout process, where users could input an invalid payment method, and the system would not return any error messages. This led to users being unable to complete their purchases without any feedback, which resulted in confusion and frustration.

I identified this issue through automated functional tests that I had set up to simulate real user scenarios. When I noticed that certain invalid payment inputs were accepted without any validation, I escalated the issue immediately. Given the importance of the checkout flow for user experience and revenue, I collaborated closely with the development team to ensure the bug was prioritized in the fix schedule.

After implementing the fix, I verified the changes through regression testing and ensured that the application now correctly handled invalid inputs by displaying appropriate error messages. The resolution of this bug not only improved the user experience significantly but also helped to prevent potential loss in sales due to abandoned carts.

This experience taught me the importance of thorough testing around critical application functions and reinforced my commitment to identifying issues that could have a significant impact on both users and the business.

1. **Have a been in situation where you had delay release?**

Yes, I have been in situations where I had to delay a release due to critical bugs or issues. One notable example was during my time at [Previous Company], where we were working on a new mobile app for a major retailer.

We were working towards a tight deadline, and the release date was just around the corner. However, during our final testing phase, we discovered some critical bugs that affected the app's performance and stability. These issues included crashes, slow loading times, and incorrect data display.

I worked closely with the development team to identify the root causes of these issues and developed a plan to fix them. Unfortunately, given the complexity of the issues, we were unable to resolve them within the original timeline.

After discussing with stakeholders, we decided to delay the release by two weeks to ensure that we could thoroughly test and validate the fixes before launching the app. This decision was not taken lightly, but it was essential to ensure that we delivered a high-quality product that met our standards.

In hindsight, delaying the release proved to be the right decision, as it allowed us to deliver an app that received positive reviews from users and met our internal quality standards. It was a valuable lesson in prioritizing quality over meeting deadlines and understanding that sometimes delays are necessary to ensure long-term success.

1. The best STLC practices?

To be honest, I'm a big fan of the Test-Driven Development (TDD) approach, which involves writing automated tests before writing the actual code. This approach ensures that the test cases are comprehensive and robust, and it helps to catch bugs early on in the development cycle.

I also believe that a solid test plan is crucial for success. A test plan should outline the scope of testing, test objectives, test environments, and the schedule for testing. It's essential to ensure that everyone on the team is aligned and working towards the same goals.

Another important aspect of STLC is test data management. Test data can make or break a test case, so it's essential to ensure that the data is accurate, relevant, and well-managed.

In terms of tools, I'm a big fan of JIRA for defect tracking and TestRail for test case management. Both tools have been incredibly useful in my previous roles.

1. Where did you see after 5 years?

To be honest, I'm a big fan of the Test-Driven Development (TDD) approach, which involves writing automated tests before writing the actual code. This approach ensures that the test cases are comprehensive and robust, and it helps to catch bugs early on in the development cycle.

I also believe that a solid test plan is crucial for success. A test plan should outline the scope of testing, test objectives, test environments, and the schedule for testing. It's essential to ensure that everyone on the team is aligned and working towards the same goals.

Another important aspect of STLC is test data management. Test data can make or break a test case, so it's essential to ensure that the data is accurate, relevant, and well-managed.

In terms of tools, I'm a big fan of JIRA for defect tracking and TestRail for test case management. Both tools have been incredibly useful in my previous roles.

1. What can you bring in company?

Honestly, I think I can bring a solid foundation in software testing and a willingness to learn and adapt to the company's specific needs. My strengths include:

I'm meticulous and thorough in my testing approach, which helps ensure that defects are caught early on and resolved promptly.

I'm effective in communicating technical issues to non-technical stakeholders and providing clear, concise reports on test results.

I'm a team player and enjoy working closely with developers, product managers, and other stakeholders to ensure that testing is aligned with project goals and objectives.

I'm adaptable and able to adjust to changing project requirements, priorities, and timelines.

I have a basic understanding of programming concepts, which helps me to understand the code and identify issues more effectively.

I'm not looking to bring any revolutionary ideas or a silver bullet solution, but rather a solid work ethic, a willingness to learn, and a commitment to delivering high-quality results.

1. What is your accomplishment?

One of my accomplishments was when I successfully identified and reported a bug in a web application's login functionality during my internship. The issue was causing users to be unable to log in, which was affecting the entire development team's workflow.

I worked with the development team to reproduce the issue, provided clear steps to reproduce the bug, and collaborated with them to develop a fix. The fix was implemented, and subsequent testing confirmed that the issue was resolved.

The impact of my discovery was significant:

\* It prevented user frustration and increased productivity for the development team.

\* It demonstrated the importance of thorough testing and attention to detail.

\* It gave me confidence in my abilities as a software tester.

This experience reinforced the value of software testing in ensuring the quality of software applications, and it motivated me to continue developing my skills and knowledge in this field.

1. Why testing?

I chose to pursue a career in software testing because I'm fascinated by technology and enjoy solving puzzles. Testing allows me to combine my analytical skills with my passion for problem-solving to ensure that software applications meet the required standards.

I also appreciate the importance of testing in the software development lifecycle. It's essential to catch bugs and defects early on, as it saves time and resources in the long run. As a tester, I can make a tangible impact on the quality of software and contribute to delivering a better user experience.

Furthermore, I find that testing is a dynamic field that requires continuous learning and adaptation. New technologies and methodologies emerge regularly, and staying up-to-date with these advancements keeps my job exciting and challenging.

1. What did you do of I hire you?

If you hire me, I would:

1. \*\*Familiarize myself with your company's products and processes:\*\* I would take the time to learn about your company's software products, services, and testing processes to understand your specific needs and challenges.

2. \*\*Develop a deep understanding of your testing requirements:\*\* I would work closely with your team to understand the testing requirements, goals, and objectives, and ensure that my approach aligns with your expectations.

3. \*\*Collaborate with the development team:\*\* I would work closely with the development team to understand the code, identify defects, and provide feedback to improve the overall quality of the software.

4. \*\*Develop and execute comprehensive test plans:\*\* I would design and execute thorough test plans to ensure that the software meets the required standards, including functional, usability, performance, and security testing.

5. \*\*Report defects and track progress:\*\* I would maintain accurate records of defects found during testing, report them to the development team, and track progress on their resolution.

6. \*\*Continuously improve my skills:\*\* I would stay up-to-date with industry trends, best practices, and new technologies to improve my skills and contribute to the company's growth.

By working together, I'm confident that we can deliver high-quality software products that meet your customers' expectations!

1. What is white/black block testing ?

In the context of software testing, white-box testing and black-box testing are two distinct methodologies that serve different purposes.

\*\*White-box testing\*\* involves testing the internal workings of the application. As a QA Automation Tester, I utilize my knowledge of the underlying code and system architecture to create test cases. This allows me to focus on the application’s logic, control flow, and data flow. For instance, I might write unit tests to validate individual functions or integration tests to verify how different modules work together. White-box testing is crucial for finding issues such as logical errors, or performance bottlenecks that could be hidden from an outside perspective.

On the other hand, \*\*black-box testing\*\* is about evaluating the functionality of the application from an external viewpoint, without any knowledge of its internal code structure. In my automation work, I often write automated functional tests that simulate user interactions based on requirements and user stories. This could involve testing APIs or UI elements to ensure that they respond correctly to various inputs. Black-box testing is essential for validating that the software meets its intended purpose and provides a good user experience.

Together, these testing strategies allow me to ensure both the robustness of the code and the overall functionality of the application."

1. Can you work under pressure?

Yes, I can work effectively under pressure. In my previous roles as a QA Automation Tester, I have encountered several situations with tight deadlines and high stakes, such as approaching release dates or critical bug fixes before a major deployment.

When faced with pressure, I adopt a structured approach to prioritize tasks. I break down larger deliverables into smaller, manageable parts and focus on the most critical areas first. For example, during a recent project, we had a significant number of automated tests to execute before our product launch. I prioritized high-risk areas that had the greatest impact on functionality, allowing us to ensure that the most important features were thoroughly tested and ready for release.

Additionally, I maintain open communication with my team and stakeholders, which helps ensure alignment on expectations and timelines. I believe that collaboration can alleviate pressure by sharing the workload and leveraging the strengths of the team.

Lastly, I approach challenges with a positive attitude and view them as opportunities for growth. Working under pressure has taught me to stay focused and adaptable, ultimately contributing to both my personal development and the success of the projects I work on."

### Behavioral Questions:

1. If given a legacy application with no existing tests, how would you approach automation?
2. What would you do if a test case fails? How would you investigate?
3. How do you prioritize your tasks when working on multiple projects?
4. How do you keep your skills updated in the ever-evolving field of QA automation?
5. Can you describe a time you collaborated with developers to resolve a testing issue?
6. **Who create product backlog?**

Product backlog -->A product backlog is a list of all the tasks, features, and requirements that need to be completed for a product or project. It is like a to-do list or roadmap for the development team, showing them what needs to be done and in what order. The items on the product backlog are typically prioritized based on their importance and value to the overall project.

1. **What is hot fix?**

A defect that has high priority and need to be fixed asap.

A hotfix is a specific type of software update that is designed to address a critical issue or defect that requires immediate attention. This could include security vulnerabilities, major bugs, or other problems that significantly impact the functionality or performance of the software. Hotfixes are typically released quickly and are often applied without waiting for a scheduled update or release cycle. They are intended to minimize downtime and ensure that users can continue to operate the software effectively.

In summary, a hotfix is a high-priority patch meant to resolve urgent issues in a software application.

1. What is the different between Waterfall and Agile?

answer like i answering in this question in interview as QA Automation Test

1. Elevator speech

An elevator speech (or elevator pitch) is a concise and persuasive summary of an idea, product, or personal brand, typically designed to capture the listener's interest in a short amount of time—often the length of an elevator ride, hence the name. Here are the key elements of an effective elevator speech:

Introduction: Start with your name and a brief introduction about yourself or your organization.

Purpose: Clearly and briefly state what you do or what your product/service is about.

Value Proposition: Highlight the benefits or value that you can provide to your audience or potential clients. Explain why your idea or offering is important or what problem it solves.

Engagement: Conclude with a call to action or an invitation for further discussion. This could be a request for a meeting, to exchange contact information, or simply to ask if they have any questions.

1. Fun fact about me

I have a knack for turning complex automation problems into simple solutions. For fun, I often challenge myself to automate everyday tasks in my personal life too—like creating scripts to manage my smart home devices. It’s amazing how much more efficient your life can become when you apply automation everywhere, not just in work!

1. **CRUD**

CRUD stands for \*\*Create, Read, Update, and Delete\*\*, which are the four fundamental operations used in database management and data manipulation. They form the backbone of the applications I test

1. \*\*Create\*\*: This operation involves adding new records to the database. In terms of testing, I ensure that when a new entity, like a user or product, is created through the application interface, the data is correctly inserted into the database. I would automate tests to verify that all necessary fields are populated correctly and that appropriate validation messages appear if any required fields are omitted.

2. \*\*Read\*\*: The Read operation is used to retrieve existing data. During automation testing, I focus on making sure that the application can effectively fetch and display the data as expected. This includes checking that the correct records are returned based on specified criteria and that the data is displayed accurately in the user interface. Automated tests often involve comparing the displayed data with the underlying database records to confirm consistency.

3. \*\*Update\*\*: This operation modifies existing records in the database. When testing update functionality, I ensure that changes made to a record—like editing user information or product details—are correctly processed and reflected in the database. I would write automated tests to verify that updated data matches expected outcomes and that the application handles update scenarios gracefully, including cases where invalid data is provided.

4. \*\*Delete\*\*: The Delete operation removes records from the database. In my testing, I assess whether the application correctly deletes records when requested and ensures that deleted data cannot be retrieved thereafter. I also check for confirmation prompts and error handling to avoid accidental deletions, validating that the implementation adheres to user expectations regarding data loss.

Overall, my experience with CRUD operations informs my testing strategy, allowing me to create comprehensive automated test cases that ensure each of these operations works seamlessly within the application. This understanding helps me identify potential issues and enhance the overall quality of the s

1. **What is the difference between authorization vs authentication in apis?**

Authentication and authorization are two critical concepts in API security, and while they work closely together, they serve distinct purposes.

Authentication is the process of verifying the identity of a user or system. It answers the question, 'Who are you?' This typically involves validating credentials like usernames, passwords, API keys, or tokens. For example, when a user logs into an application, the system checks their provided credentials against stored records to confirm their identity.

On the other hand, authorization determines what an authenticated user is allowed to do. It answers the question, 'What can you do?' Once a user is authenticated, the system checks the permissions associated with that user to control their access to resources or functionalities. For instance, an admin user may have access to perform administrative tasks, while a regular user may only be allowed to view content.

1. **If you need to start automation from scratch what would be your steps and process?**

Starting automation from scratch involves several key steps to ensure it is effective and sustainable. First, I would clarify the goals of automation. Are we looking to improve efficiency, reduce errors, or enhance overall quality? Setting specific, measurable objectives will guide the entire process.

Next, I would review the existing workflows to identify manual tasks that are repetitive, time-consuming, or error-prone. This analysis helps in selecting the best candidates for automation.

Based on the identified tasks, I would research and select suitable automation tools and frameworks that fit the project requirements. This includes considering factors like the technology stack, integration capabilities, and the skill set of the team.

I would create a clear automation framework, establishing coding standards, organizational structure for tests, and reusable components. This strategy serves as the foundation for all automation efforts.

Following that, I would start developing the initial test cases, focusing first on high-priority scenarios. It’s often helpful to begin with simpler cases before tackling more complex ones.

After building the test suite, I’d run the automated tests in the designated environment. Monitoring results is crucial to identify any failures quickly and address them accordingly.

I would regularly analyze the test outcomes, refactoring the tests as needed to enhance reliability and maintainability. Continuous improvement is vital as the application evolves.

Finally, as the framework matures, I would gradually expand the test suite, including different types of tests, and ensure that team members are trained to effectively utilize the automation strategy.

1. **How OAuth is used for both authentication and authorization in many APIs?**

Understanding OAuth is essential in the realm of APIs because it serves both authentication and authorization purposes, often leading to confusion.

Authentication vs. Authorization:

Authentication: This is the process of verifying who a user is. For example, when a user logs into a service using their credentials (like a username and password), that verifies their identity.

Authorization: This refers to granting permissions to a user after their identity has been confirmed. It dictates what resources a user can access and what actions they are permitted to perform.

How OAuth Works:

OAuth is an open standard for access delegation, commonly used to grant websites or applications limited access to user information without exposing passwords

Authentication: In systems utilizing OAuth, when a user logs in to a third-party application using their account (like Google or Facebook), they are effectively authenticated via OAuth. The third-party application does not handle the user's credentials directly; instead, the user is redirected to the authentication provider (the OAuth server), where they log in securely. Once authenticated, they grant permission to the application to access specific information.

Authorization: After successful authentication, OAuth handles authorization by issuing an access token. This token allows the application to access resources on behalf of the user. The application can perform actions based on the scope of the access granted (e.g., reading user profile information, posting updates). The user can typically see what permissions they are granting, and they retain control over their consent.

Example in Practice:

For instance, when you use a mobile app that allows you to log in with your Google account, OAuth comes into play. The app redirects you to Google for authentication, and once you verify your identity, it asks you for permission to access your email and calendar data. Upon granting permission, the app receives an access token, allowing it to interact with Google APIs on your behalf, such as adding events to your calendar.

1. **Velocity and Capacity**

Velocity refers to the amount of work a team can complete during a sprint, which is a fixed time period in Scrum (usually 1-4 weeks). Velocity is measured by looking at how many user stories or story points the team completes in each sprint.

Capacity, on the other hand, refers to the total amount of work that a team can realistically take on during a sprint. It takes into account factors such as team members' availability, skills, and any external dependencies that may impact their ability to deliver work.

**7)Planning poker** is a collaborative technique used in Agile and Scrum practices for estimating the effort or size of user stories or tasks during the sprint planning meeting.

In planning poker, each team member is given a set of cards with numbers representing the size or effort required to complete the task. The team discusses each task or user story, and then everyone secretly selects a card that represents their estimate of the effort involved. The team members then reveal their cards simultaneously, and discuss any significant differences in estimates. This process helps facilitate discussion, build consensus, and improve accuracy in estimating the work involved in each task.

7)**Waterfall?**

Waterfall:has to develop step by step like Requirement,Design,Execution,Testing,Release.This means that any phase in the development process begins only if the previous phase is complete.

1. **Agile** methodologies are frameworks or approaches that help teams work in a more adaptive and flexible way to deliver software products or services.
2. **Scrum and Kanban** are Agile methodologies, they differ in their approach to project management. Scrum is more structured and prescriptive, with specific roles, events, and artifacts that teams must follow. Kanban, on the other hand, is more flexible and adaptable, allowing teams to continuously improve their processes without being constrained by a fixed set of rules.
3. What does MVP?

Minimum Viable Product

1. **SMART** --> specific, measurable , attainable, relevant , time frame

**11)SDLC stands for Software Development Life Cycle,** which is a process followed by software development teams to design, build, and test high-quality software typical phases of the SDLC:

1. Planning: In this phase, the project scope, requirements, timeline, and budget are defined. The feasibility of the project is also assessed.

2. Analysis: Requirements gathering is done in this phase. Business requirements are collected and analyzed to determine what the software should do.

3. Design: In this phase, the software architecture, system design, and detailed specifications are created based on the requirements identified in the analysis phase.

4. Implementation: The actual coding or development of the software takes place in this phase. Developers write the code according to the design specifications.

5. Testing: In this phase, the software is tested to identify any defects or issues. Different types of testing, such as unit testing, integration testing, system testing, and user acceptance testing, are performed.

6. Deployment: Once the software has been successfully tested and meets the requirements, it is deployed to the production environment. Users start using the software in their operational environment.

7. Maintenance: After the software is deployed, it enters into the maintenance phase. Updates, bug fixes, and enhancements are made to the software as necessary to ensure it continues to meet user needs.

1. Software Testing Life Cycle (**STLC**) is a systematic process for testing software applications in order to ensure they meet quality standards and requirements. It involves planning, designing, executing, and reporting on tests to identify and fix defects in the software before it is released to users.

**12)Agile Scrum ceremonies** are regular meetings held during the development of a project using the Agile Scrum methodology. These ceremonies help teams stay organized, communicate effectively, and collaborate towards achieving the project goals. Here is a simplified explanation of some common Agile Scrum ceremonies:

1. Sprint Planning: At the beginning of each sprint (a fixed time frame for completing work), the team plans what tasks will be worked on and how they will accomplish them.

2. Daily Stand-up: Also known as a daily stand-up meeting or a daily scrum, this is a short meeting where team members quickly share updates on what they have accomplished, what they are working on, and any obstacles they are facing.

3. Sprint Review: At the end of each sprint, the team demonstrates the work they have completed to stakeholders, gathers feedback, and discusses what worked well and what can be improved.

4. Sprint Retrospective: After the sprint review, the team holds a retrospective meeting to reflect on the sprint, discuss what went well, what didn't go well, and how they can improve in the next sprint.

5. Backlog Refinement: Periodically, the team meets to prioritize and refine the items in the product backlog, which is a list of all the work that needs to be done on the project.

**11)Bug Life Cycle**

New: The bug is identified either through testing or by a user reporting an issue.

Assigned: The bug is assigned to a developer to investigate and fix.

Open: The developer is working on fixing the bug.

Fixed: The bug has been fixed by the developer.

Ready for Testing: The fixed bug is ready to be tested to ensure that the issue has been resolved.

Closed: The bug is verified to be fixed and closed by the developer.

Reopened: If the bug reoccurs after being closed, it is reopened for further investigation and fixing.

Verified: The bug has been fixed and verified by the testing team.

1. **Testing**

Smoke Test:

- Smoke testing is conducted to verify that the essential functionalities of an application are working correctly. It is usually performed at the beginning of the testing process before more in-depth testing is conducted.

Regression Testing:

- Regression testing is a type of software testing that is performed to ensure that new changes or modifications to the software do not negatively impact existing functionalities. It involves retesting the areas of the software that have been changed and also testing other parts of the software that may be affected by those changes.

Integration Testing:

- Integration testing is the process of testing the interfaces and interactions between different components or modules of a software system. It ensures that the individual components work together as expected when integrated into a complete system.

User Acceptance Testing (UAT):

- User Acceptance Testing, also known as UAT, is the final phase of testing before the software is released to the end-users. It is conducted by the users or stakeholders of the software to validate that the system meets their requirements and is ready for production deployment.