



InterviewBit

Agile Testing Interview Questions



To view the live version of the page, [click here.](#)

© Copyright by Interviewbit

Contents

Agile Testing Interview Questions for Freshers

1. What is Agile testing?
2. What are the principles of Agile testing?
3. Describe the Agile testing life cycle.
4. Explain the role and responsibilities of Agile testers.
5. What are the characteristics of a good Agile tester?
6. What is agile methodology in software development? Write its types.
7. What is an Agile test plan?
8. When should we use Agile Scrum Methodology?
9. What is Sprint in Agile Methodology?
10. What is the story point in Agile?
11. Explain Velocity in Agile Methodology.
12. Explain Epic, User Stories, and Tasks.
13. Why is it important to conduct daily stand-up meetings?
14. Explain Continuous testing.
15. What does the term Taskboard mean in Agile?

Agile Testing Interview Questions for Experienced

16. What is Scrumban in Agile?
17. How do you handle continuous changes in requirements as an Agile Tester?
18. What sets Agile Testing apart from other traditional Software Testing Models?

Agile Testing Interview Questions for Experienced

(.....Continued)

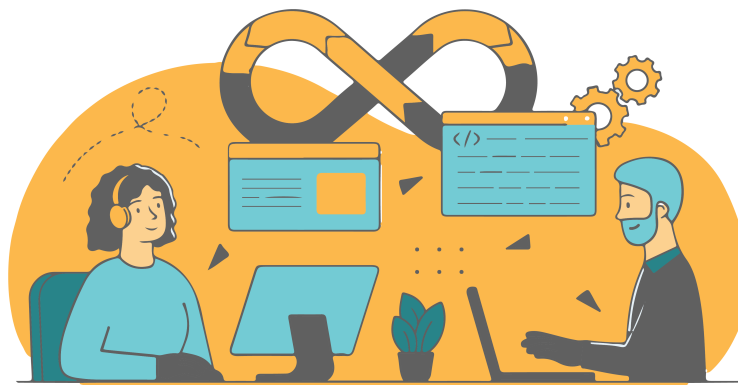
19. What is Acceptance Testing in Agile?
20. Explain refactoring.
21. What are the benefits of refactoring?
22. What do you mean by Spike and Zero sprint in Agile?
23. What is Exploratory Testing and Scripted Testing?
24. Explain Tracer Bullet.
25. What is Test stub?
26. What is Application Binary Interface?
27. Can you explain why Continuous Integration is important for Agile?
28. In what ways can QA benefit an agile team?
29. What are some impediments that a scrum master should remove?
30. How can keeping an iteration length consistent throughout benefit the project?
31. Explain build-breaker.

Let's get Started

In today's rapidly evolving software development industry, it is imperative to update testing methodologies regularly in order to remain abreast of technological advances. Agile testing is a new approach based on smarter testing rather than putting in a lot of effort to ensure high-quality products. Rather than being sequential, [Agile testing](#) is a continuous process, in which every step of the project is tested continuously while it is being developed and compiled.

The Agile testing methodology calls for a high degree of collaboration between testers and developers. Cross-functional Agile testers work closely with business experts to maintain a sustainable pace while meeting optimum business standards. Considering the myriad benefits of Agile testing and development, it is understandable why Agile testers are in such high demand in the development and testing industries.

Agile Testing Interview Questions and Answers



 InterviewBit

In this article, aspiring Agile testers are provided with a comprehensive list of top **Agile Testing interview questions and answers** that can serve as a good foundation as they begin or advance their careers. Having said that, we will discuss what is Agile Testing and some common Agile interview questions for Testing to help you prepare for your interview. Regularly practising these questions will increase your chances of acing the interview.

Agile Testing Interview Questions for Freshers

1. What is Agile testing?

Agile software development and provides fast feedback to ensure better and faster project outcomes. The basis for Agile Testing is continuous integration between development and testing from the beginning of the project. As opposed to being sequential (e.g. executed only after coding is complete), Agile testing is continuous. The product is enhanced with every new release, resulting in seamless management and higher satisfaction for customers. Agile testing focuses on finding and fixing errors as soon as possible, rather than waiting until the end of the project, and reducing the costs associated with fixing errors. By delivering high-quality products on time, this approach also results in a customer-centric approach.

2. What are the principles of Agile testing?

Eight main principles underlie agile testing. They are as follows:

- **Practice of continuous testing:** Agile teams continuously test their products to ensure that they are improving. It is carried out alongside development.
- **Feedback on an ongoing basis:** Agile testing provides team members with ongoing feedback and ensures that the product meets its business objectives. Feedback is primarily focused on quality rather than requirements.
- **Participation of the entire team:** In agile methodologies, developers, business analysts, and testers all take part in the testing process.
- **Speeding up feedback:** Agile involves the business team during each iteration, resulting in faster feedback and a lower cost associated with repairing errors.
- **Quality comes from clean code:** Testing the software regularly allows issues and vulnerabilities to be detected and fixed during the same iteration, maintaining code integrity.
- **Fewer Documentations:** Agile testers use reusable checklists rather than long, bloated documents and concentrate on the essence of the test rather than on the incidental details.
- **Test-Driven:** Unlike conventional methodologies, agile testing involves testing during the actual implementation process.
- **Client satisfaction:** The client can see their product being developed and make changes and updates to it as it evolves. The test procedures can be updated to fit the current requirements.

3. Describe the Agile testing life cycle.

Unlike traditional development methodologies, agile development gives users flexibility through its iterative approach. An agile testing life cycle consists of the following five phases:



- **Impact Assessment:** The impact assessment process is probably the most important phase of an agile testing life cycle, as during this process, the agile testers are expected to gather all the necessary information from all stakeholders involved in the project.
- **Agile Testing Planning:** Stakeholders and testers collaborate at this stage to schedule the testing process and deliverables.
- **Release Readiness:** Agile testers must test the feature in this phase to see if it meets our clients' expectations and is ready to go live.
- **Daily Scrums:** This will probably include a morning meeting to check the progress of testing and set goals for the day ahead.
- **Test Agility Review:** Test Agility Review is the final phase of the project in which a meeting is held with the stakeholders to assess the progress of the project and to determine whether the milestones have been reached or not.

4. Explain the role and responsibilities of Agile testers.

Agile testers are tasked with the following responsibilities:

- Understand, implement, and update Agile test strategies.
- Measure and report test coverage.
- Assure proper testing tool usage.
- Set up, manage, and use testing environments.
- Write and execute automated tests and report results to the team.
- Identify defects and resolve them with the team.
- Provide coaching and training to other team members.
- Schedule appropriate testing activities during release and iteration planning.
- Develop clear requirements for testability, consistency, and completeness through collaboration with developers and stakeholders.
- Active participation in daily stand-up meetings, story grooming sessions, and retrospective meetings of the teams.
- Propose and implement improvements.
- Assessing new agile metrics such as velocity, sprint burndowns, release burndowns, etc.

5. What are the characteristics of a good Agile tester?

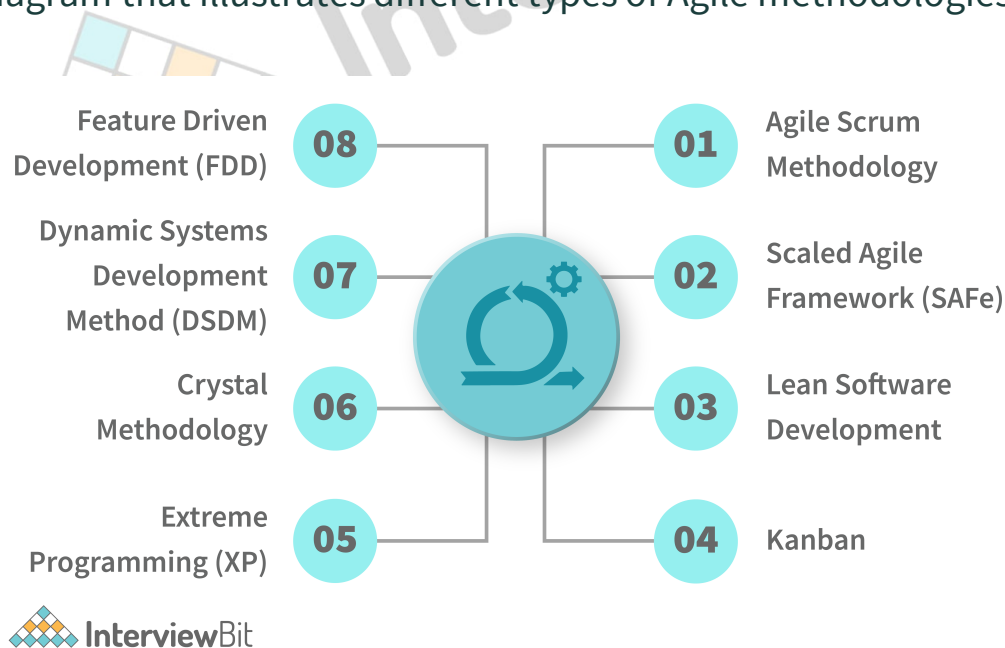
Good agile test leads lay a strong foundation for fostering an agile working culture in any organization through the application of good agile practices. An Agile tester should possess the following qualities:

- Focus on company goals and mission.
- Effectively sharing project ideas.
- A better understanding of requirements and risks associated with changing requirements.
- Ability to prioritize work based on requirements.
- Communicate effectively with business associates and developers.
- Understand the results of testing and examine them for specific product or application behaviour.
- Testers must have coding skills.
- An analytical mindset for improving testing methods and varying scenarios.
- Demonstrating information clearly and concisely.

6. What is agile methodology in software development? Write its types.

Known as Agile development methodologies, Agile is an iterative approach to software development projects. It focuses on delivering value to customers faster and with fewer headaches than traditional ways. In contrast to risking everything on one big launch, agile teams deliver work in incremental, but consumable, stages over time. Continuous evaluation of requirements, plans, and results allows teams to implement changes quickly. This approach aligns with the principles and values outlined in the Agile Manifesto.

Below is a diagram that illustrates different types of Agile methodologies:



7. What is an Agile test plan?

Agile test plans are very important documents that cover a variety of testing types carried out within an iteration, such as test data requirements, test environments, test infrastructure, and test results. In an agile model, test plans are written and updated for each new release. A typical agile test plan consists of the following:

- Scope of Testing.
- Test requirements, sprint objectives, and the scope of the task.
- New functionalities that are being tested.
- Testing levels or types based on feature complexity.
- Performance testing and load testing.
- Considerations related to infrastructure.
- Plan for mitigating or reducing risks.
- Resource allocation.
- Defining milestones and deliverables.

8. When should we use Agile Scrum Methodology?

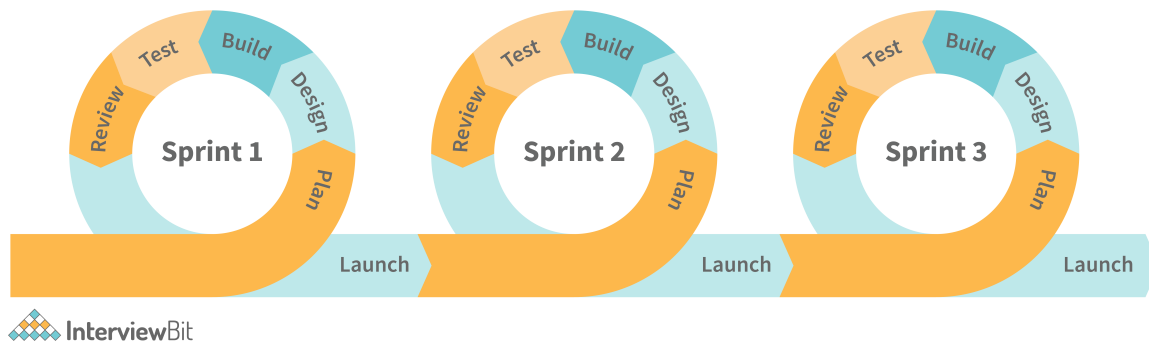
Agile Scrum Methodology can be used in the following situations:

- When the client isn't clear on what exactly they need.
- When the client expects a quick delivery of the product.
- When the client does not provide all the requirements at once.
- For projects where requirements fluctuate frequently.

9. What is Sprint in Agile Methodology?

A sprint is defined as a short, time-limited period in which an agile team completes specific amounts of tasks or deliverables. Getting sprints right will help your agile team deliver better software with fewer headaches, since sprints are the basis of the scrum and agile methodologies. Typically, sprints don't last longer than a month. While some teams may prefer to work in two-week sprints, others may prefer working in weekly sprints.

Agile Methodology

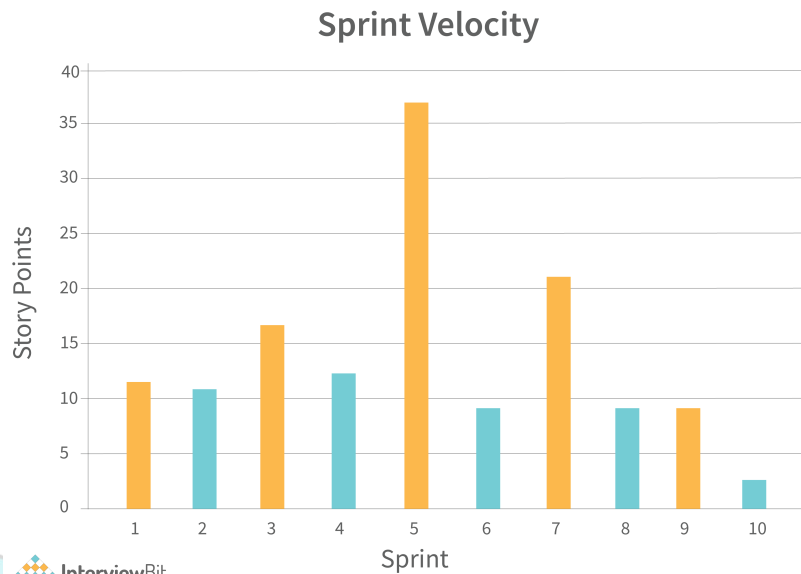


10. What is the story point in Agile?

Agile story points are a metric for measuring the difficulty of implementing a user story, and therefore reflect the level of effort involved in implementing it. Story points are measures of the difficulty level of a story. It can be related to the risks, complexities, and efforts involved. An increasing number of story points will result in a greater effort. One popular method for estimating agile story points is to use Fibonacci sequences. In this series, each number represents the sum of the two preceding ones. The Fibonacci sequence begins with 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89... and so forth.

11. Explain Velocity in Agile Methodology.

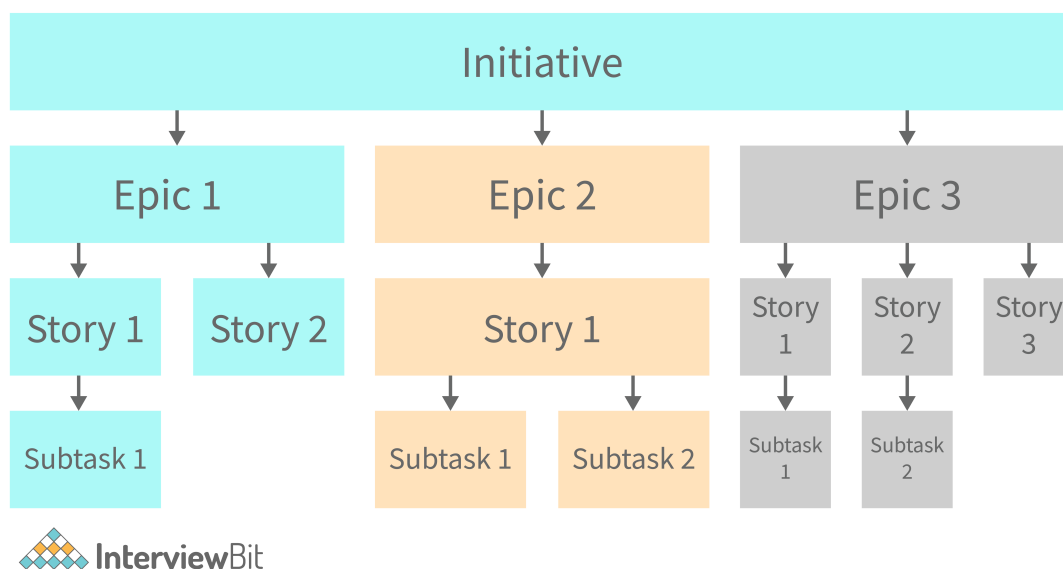
In Agile Methodology, Velocity is a measure of the amount of work a team can accomplish during an iteration, and is frequently used by development teams to build accurate and effective timeframes. It is a simple calculation that measures how many units of work have been completed in a given period of time. There are several ways to measure units of work, such as story points, user stories, and engineer hours.



From the chart above, it's clear that Sprint 5 was exceptionally productive - almost 40 story points were produced. Their average velocity is approximately 10 story points.

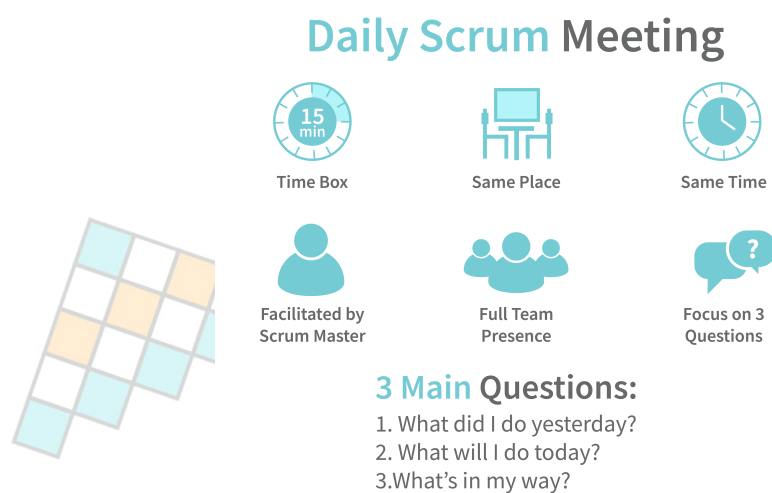
12. Explain Epic, User Stories, and Tasks.

- **Epic:** An epic is usually a large story that is difficult to accomplish in a single sprint. An epic is usually accomplished over the course of months. Epics are usually very broad in scope, lack details, and must be broken into smaller user stories before the story can be developed. Epic is often viewed as being the pinnacle of the work hierarchy. In cases where multiple epics share a common goal, they can be grouped into a still-broader business objective, called a theme.
- **User Story:** Basically, a User Story is a list of what needs to be done within a project. It's similar to a to-do list and is owned by the Scrum Product Owner. It provides the Scrum team with enough to get a sense of what the finished product should be like and to calculate an estimate for completion.
- **Tasks:** Each epic is followed by a set of user stories, and for these stories to become workable components, the Scrum team needs to identify and sort tasks. Depending on the complexity of the task, it can take a few hours to several hours (typically up to 12 hours). The task will be assigned to team members with the relevant skills and expertise. A story can only be considered complete when all tasks associated with it have been completed.



13. Why is it important to conduct daily stand-up meetings?

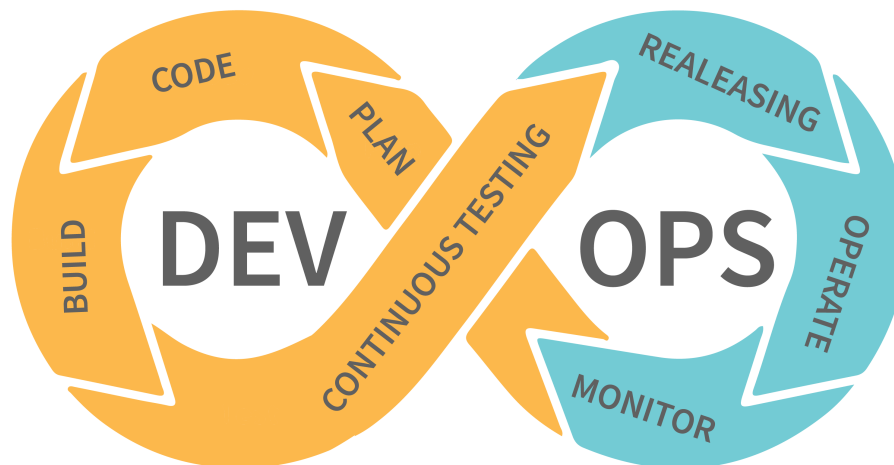
As a team's huddle, daily stand-ups are the basis for Agile project management. Often referred to as the daily scrum, it keeps everyone on top of the team's progress and landscape by focusing on "we". Agile teams have daily standup meetings as part of their daily activities. They set a mutually convenient time for team meetings and meet every day to stay on top of all the pertinent information for improving project coordination.



- This facilitates effective communication between team members. Integrating effective communication concepts with face-to-face interaction into the team's culture fosters greater agility.
- Moreover, it aids in ascertaining roles and responsibilities in a concise and clear manner and prevents silos throughout the organization. Each team member is aware of the other's work and progress continuously, allowing transparency about project's progress.
- By holding daily stand-ups, companies can adapt to changes and address issues that arise during the day-to-day operations of an organization or during the course of a project.
- It leads to a sense of shared goals, which is a key element to the team since, without shared objectives, there is only a group of individuals working together.
- Furthermore, regular team meetings facilitate knowledge transfer within the organization.

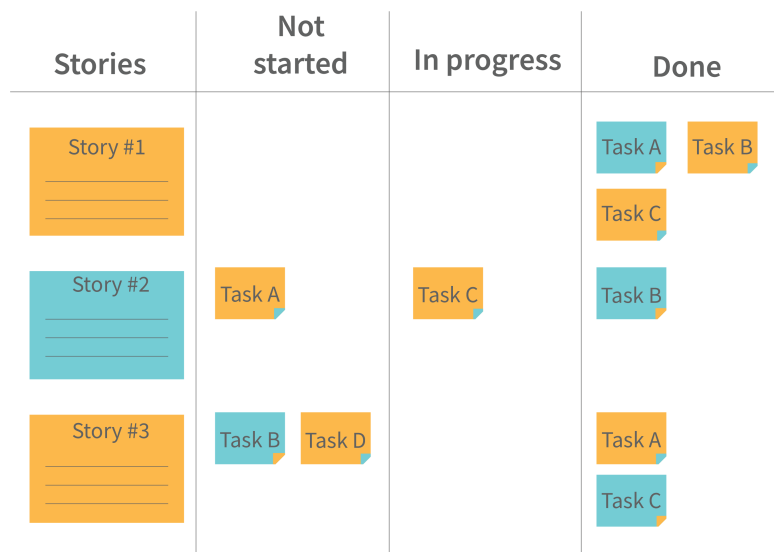
14. Explain Continuous testing.

Continuous testing (CT) is a methodology used in software development, where applications are continuously tested during the entire software development life cycle (SDLC) to determine whether a new release entails risks. CT is designed to provide critical feedback earlier on in the software development life cycle and to enable high-quality and faster deliveries. It extends and develops test automation strategies to address the increasing complexity and pace of development and delivery of modern applications.



15. What does the term Taskboard mean in Agile?

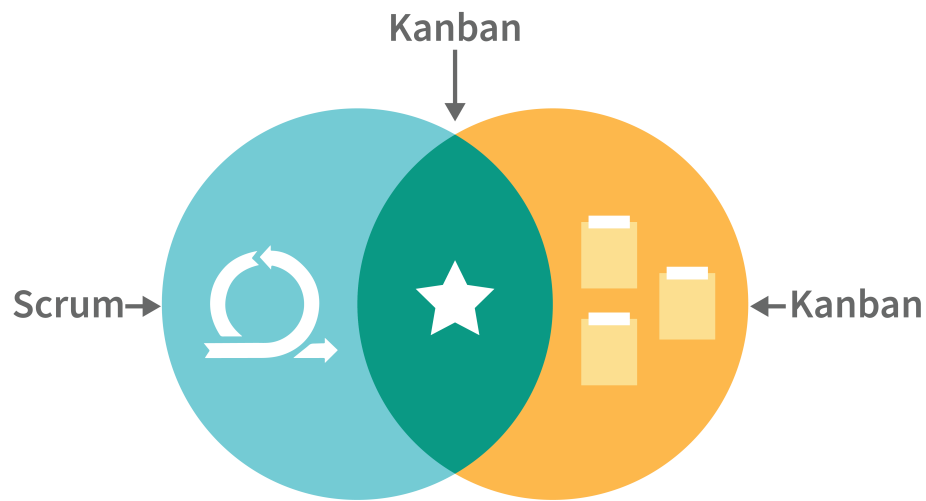
Basically, task boards are physical displays in which the user stories that make up the sprint backlog are displayed, along with their constituent tasks. Task boards are used by individuals, teams, and organizations to represent the work being done and to track its progress towards completion. Since they are so flexible, task boards are becoming a classic tool for project management. As illustrated below, it includes tasks in progress, completed tasks, as well as new upcoming tasks that may have been added to the backlog.



Agile Testing Interview Questions for Experienced

16. What is Scrumban in Agile?

Scrumban, as its name implies, is a hybrid Agile methodology that combines Scrum and Kanban (Scrum+Kanban = Scrumban). Hybrid project methodologies like Scrumban allow teams to be flexible in adapting to stakeholders and production requirements without being overburdened by their project methodologies. Scrumban combines the structure of Scrum with flow-based methods and the flexibility of Kanban, making it an extremely versatile way of managing workflows. It increases team agility, efficiency, and productivity.



InterviewBit

17. How do you handle continuous changes in requirements as an Agile Tester?

In cases where requirements are constantly changing, agile testers are advised to follow the following approach:

- Prepare generic test cases and test plans focused on the intent of the requirement rather than its details.
- Product Owners and teams must work together to understand and minimize the impact of requirements changes.
- Develop contingency plans to address the risks of changing requirements at the end of the iteration.
- It is best to automate a feature only after the feature has been stabilized and the requirements are finalized.
- Use Agile Task Boards for better project management as they help arrange development tasks into various categories like To-Do, In Progress, and Done. With this categorization, team members can see where they stand in their progress, as well as which parts of the project are in need of attention.
- A daily stand-up meeting with clients is highly beneficial because any changes that clients suggest can be discussed right away. Changes in requirements, timeframe, effort, and overall project schedule can all be discussed at a higher level and agreed upon during these meetings.
- Let your customer participate in all stages of the project, from the initial stage of identifying requirements to the final phase of implementation. The customer may change requirements during each interaction.

18. What sets Agile Testing apart from other traditional Software Testing Models?

There are various factors that set Agile testing apart from other traditional Software testing methodologies.

Agile Testing V/S Traditional Testing

 InterviewBit

Agile Testing	Traditional Software Testing
Agile testing involves performing tests alongside software development in order to ensure greater transparency of the development process.	Traditional testing takes place after the entire phase of software development has been completed. The testing process comes after the development cycle is complete, and it could take a long time.
In contrast, agile testing enables the team to "move forward (release) with working components while addressing defects in subsequent iterations".	It emphasizes on "Fix Defect Then Release".
It requires regular interaction with customers.	Customer interaction is not mandatory on a daily or weekly basis.
This process, however, is based on continuous improvement; changes are made in the next iteration of the testing cycle.	The traditional approach allows for any necessary modifications to be made only during the next release.
It is more suitable for projects where the requirements fluctuate frequently.	Specifically, it is best suited for situations where the requirements are little or never likely to change.

19. What is Acceptance Testing in Agile?

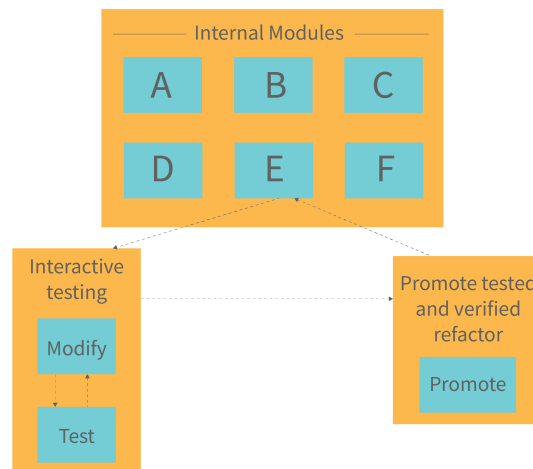
An acceptance test is a test conducted to evaluate whether or not the requirements of a software specification have been met. Testing involves comparing new features, functionality, or systems against predefined acceptance criteria. Acceptance testing is an integral part of agile development and is not merely an afterthought. The intent, however, remains the same: ensuring that software meets customer and end-user expectations.

20. Explain refactoring.

The word 'refactoring' describes the process of improving the internal structure, the operation, or the aesthetics of a code or component without changing/affecting its external behavior. Teams working in Agile are responsible for maintaining and enhancing their code on an incremental basis, Sprint by Sprint. Without refactoring, an Agile project will end up with poor code quality, such as unhealthy dependencies between classes, too many responsibilities per method, improper design of class responsibilities, and multiple forms of confusion and clutter. Code refactoring helps to simplify ambiguous and complex code by removing chaos.

In the following example, it can be seen that the 5th module is a refactor. A refactor (5th) is usually broken down into several sequential micro-refactors, and each of those micro-refactors must be tested to ensure they are free from errors.

Unchanged External System Behavior



21. What are the benefits of refactoring?

Refactoring offers the following benefits:

- Refactoring code to make it less complex and easier to understand.
- Streamline the source code design to speed up the creation of new features.
- Reduce messy and unrefined code, which may otherwise require additional rework.
- Reducing the cost of code maintenance and making it more modular and readable.
- Enhancing the architecture of software without affecting its behavior.
- Improved modularity to increase reusability.

22. What do you mean by Spike and Zero sprint in Agile?

- **Sprint Zero:** The Sprint Zero is the initial sprint of a team, which occurs before the formal start of the project, before official documentation, and/or before the team's inception. Sprints are intended for the Development Team to develop a minimum number of User Stories, story mapping, a project skeleton, and develop a full-fledged product. As part of sprint zero, the team answers the following questions:
 - What is the duration of sprints?
 - When is a sprint considered completed?
 - What tools do we use, and how should they be used?
 - How should we work? Would it be best to formalize our agreement in writing?
- **Spike:** This shorthand term refers to "spike solutions". It represents the initial investigation or research of a software solution (which could be a feature, a bug fix, a migration task, etc.). Instead of documenting what needs to be done, try to construct a quick, throwaway solution to get a better sense of the scope of work (by validating the technical assumptions). Ideally, spikes should be scheduled along with all other features (bugs, fixes, etc.) during sprints. If a team is unable to agree on an estimate, it may realize the need for a spike.

23. What is Exploratory Testing and Scripted Testing?

There are two different testing methods that can be employed to test software release schedules, namely exploratory testing and scripted testing.

Exploratory Testing	Scripted Testing
An exploratory testing approach allows testers to rely on their instincts, skills, and experience to perform individual tests in-depth.	The scripted testing process requires testers to follow a set of guidelines specifically designed to make sure that the actual results are as intended.
In-depth domain knowledge is required for this type of testing. Without proper domain expertise, tests cannot be conducted.	It is possible to overcome the lack of domain knowledge during the test design phase. They can analyze the document along with learning from the subject matter experts.
Often, documentation is lacking in exploratory environments.	Documentation is essential in Scripted Testing.
No preparation is required, and no overhead is associated with reading the documentation.	Prior to executing tests, it is very important to prepare test scripts and create documentation.
Testing involves comparing the application to the tester's expectations and understanding how the application should behave.	An application is verified against the specifications of the document.
Learning and adaptability are key components of this	Prediction and decision-making are important aspects of this

24. Explain Tracer Bullet.

This spike is designed to integrate with the current architecture, current technology, and a set of best practices that will result in high-quality production code. This might seem like a very narrow implementation of the functionality, but it is not junk code. In fact, this code is of production quality, and as such, it will probably be utilized in future iterations of the project. Therefore, the next iterations can build upon this base code.

25. What is Test stub?

In Agile, a test stub is a small piece of code (modules) that simulates an component/element in the system being tested and can replace it. Top-down integration testing often involves the use of stubs to simulate or mimic the behavior of lower-level modules that have not been integrated. Stubs are temporary replacements for a called module, and produce the same output as the actual product.

26. What is Application Binary Interface?

In the field of computer software, ABI stands for Application Binary Interface. It's primarily an interface between two binary program modules, one of which is a library or system facility, and the other is a user program. In essence, ABI deals with the implementation details of programs, i.e., what the code turns into after it has been compiled. API covers calling convention, data types, alignment, size, as well as how an application should handle system calls to the operating system.

27. Can you explain why Continuous Integration is important for Agile?

Continuous Integration (CI) is considered one of the cornerstones of Agile development. This method of developing software requires small components of code to be integrated continuously into the main code following all the phases of the software development lifecycle (SDLC). The following are some of the benefits of CI in agile:

- New features can be easily enhanced after the initial version has been released. It reduces the time required to develop new features while ensuring product stability.
- Continuous integration ensures that code stays robust enough to be utilized by customers or other stakeholders at any time. Overall, this will speed up the development process.
- Moreover, it spurs more feedback between developers and customers, assisting the development team in getting things right before meeting iteration deadlines.
- By combining Agile and Continuous Integration, the SDLC can be transformed and many of its shortcomings eliminated. Ultimately, this leads to an efficient and faster deployment process.
- Effective CI assures that integration never becomes an issue as your code only reflects slight deviations from the codebase. As long as the team continuously deals with small divergences, it does not have to deal with scary divergences.

28. In what ways can QA benefit an agile team?

QA ([Quality Assurance](#)) analysts have myriad responsibilities in agile development, as follows:

- Since QA works closely with the software development team from the beginning, QA is capable of identifying potential risks. As QA analysts participate in daily scrums, they can update the team with known and critical issues, so they can stay focused on the bug and speed up their planning.
- QA estimates the testing effort required for each user story in the sprint planning session.
- QA acts as a proxy product owner in the absence of a Product Owner. Additionally, they can discuss the business requirements with the Product Owner.
- For the ongoing enhancement of user stories, QA analysts provide the product owner with feedback from their past testing experiences and the experiences of their team members during each sprint.
- Developers and QA analysts engage in quick handoff demonstrations for every new feature. QA will be able to see how the newly developed feature works, allowing them to raise questions to the development team.

29. What are some impediments that a scrum master should remove?

Agile development teams continuously strive to improve their production processes. However, certain impediments may reduce the value produced, thus slowing down the development process. Impediments are factors that prevent the Development Team from producing useful software in a Sprint or impede the team from achieving its intrinsic goals. The Scrum Master's main role is to remove all impediments that prevent the Development team from moving forward. Given the self-organizing nature of the team, the Scrum Master should assist the team in trying to solve their own problems. Or better still, prevent obstacles from becoming impediments. There are many types of impediments that may arise, as you can see below.

- Illness among the team members
- Unforeseen changes team composition
- Tool issues for the Development Team
- Lack of skilled personnel
- Supplier issues
- Product owner's absence
- Unnecessary meetings for the Development Team
- Restrictions on teamwork.
- Conflict among team members, etc.

30. How can keeping an iteration length consistent throughout benefit the project?

Here are some advantages of keeping iteration lengths consistent throughout the project:

- It can serve as a consistent way of measuring the velocity of the team.
- Establishing a consistent delivery pattern becomes more straightforward using it.
- It is inefficient to decide the iteration length at the beginning of each iteration. Test a few lengths, come to a decision, then stick with it until there is a reason to change.
- Having a consistent iteration length, be it one week, three weeks, or somewhere in between, helps teams settle into the pace that works best for them.

31. Explain build-breaker.

Software developers sometimes make inadvertent mistakes that halt the build process, produce undesirable warnings, or fail automated tests. This type of error is known as a 'Build Breaker'. As soon as the developer is aware of the problem, he/she must take immediate action to get the build back to its normal state. The tester must address these issues quickly and equitably to bring the software development back to a normal/standard state by resolving the bugs.

Conclusion

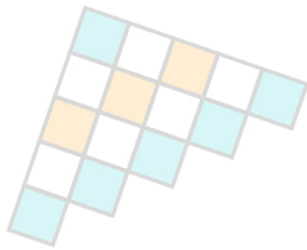
Agile testing offers many career prospects to people seeking employment with leading organizations; however, it is crucial that employees work hard and consistently in order to achieve employment.

Agile testing facilitates closer collaboration between developers and testers, leading to a higher level of quality. Agile tester has to list out the corrective feedback and provide it to the developer's team as part of the continuous update cycle. Essentially, this is a continuous integration of Agile development and testing methodologies to achieve the desired results. Those preparing for Agile scrum interviews should read in-depth about this topic.

For those who are preparing to attend an interview in this niche, the above-mentioned Agile interview questions for experienced, as well as those who are new to the field, will send you straight to the top. Candidates should have a strong grasp of concepts but also have the ability to speak confidently and decisively. Practising mock interviews can help you develop your self-confidence and prepare you for different interview scenarios. You can easily pass this interview with good preparation and a little bit of cleverness.

Additional Resources

- <https://www.interviewbit.com/blog/agile-model/>
- <https://www.interviewbit.com/blog/agile-principles/>
- <https://www.interviewbit.com/software-testing-interview-questions/>
- <https://www.interviewbit.com/performance-testing-interview-questions/>
- <https://www.interviewbit.com/scrum-master-interview-questions/>
- <https://www.interviewbit.com/automation-testing-interview-questions/>
- <https://www.interviewbit.com/sdlc-interview-questions/>
- <https://www.interviewbit.com/sdet-interview-questions/>
- <https://www.interviewbit.com/technical-interview-questions/>
- <https://www.interviewbit.com/software-testing-mcq/>



Links to More Interview Questions

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)