PassQueen
a.k.a dumpsqueen.com

Certified Tester Foundation Level Agile Tester

iSQI CTFL-AT

Version Demo

Total Demo Questions: 10

Total Premium Questions: 104

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Agile Software Development	23
Topic 2, Fundamental Agile Testing Principles, Practices and Processes	48
Topic 3, Agile Testing Methods, Techniques and Tools	33
Total	104

QUESTION NO: 1

Which of the following statements about Agile retrospectives is true?

A. Agile retrospectives should be focused mainly on obstructions and hindrances that are outside the control of the team because these issues are more challenging

B. Traditional projects require feedback from retrospectives; however, Agile retrospectives do not require follow-up activities

C. Agile retrospectives should be focused on the team members and business representatives, and other participants should not attend or contribute

D. Agile retrospectives should also provide feedback to the current iteration planning regarding planning accuracy, contents, and the efficiency of the work in the team

ANSWER: D

Explanation:

"Agile retrospectives should also provide feedback to the current iteration planning regarding planning accuracy, contents, and the efficiency of the work in the team" is the correct statement because a retrospective is the Agile team's structured opportunity to inspect how it worked during an iteration and determine practical improvements. Its scope includes not only technical work, but also the effectiveness of collaboration, processes, tools, working practices, and how accurately the team planned and delivered the iteration's selected work. The insights obtained are used to adapt the team's approach, including improving the quality and realism of subsequent iteration planning.

The CTFL-AT syllabus describes retrospectives as a way for the team to review how it worked during the iteration, including planning accuracy and team collaboration, and to identify actionable improvements. This supports continuous improvement: the team turns observations into specific actions that can improve efficiency, delivery, and teamwork in future iterations. Retrospectives therefore provide an important feedback loop between the experience of completing an iteration and the planning of upcoming work. See the ISTQB Agile Tester certification page and the Scrum Guide, which identifies the Sprint Retrospective as the event for planning ways to increase quality and effectiveness.

QUESTION NO: 2

You are working in a software development company which, for many years, used a sequential development model and was organized into separate departments for each functional group (e.g. business analysts, developers, testers) located within their own office space. Your organization has recently changed to a SCRUM agile framework. Which of the following is an important organizational and behavioral best practice for a tester in the SCRUM team that should have also been practiced when using the sequential model?

A. Resilient testing means that the testing process is capable of dealing with rapid changes throughout the development process with test plans being updated during each iteration.

B. Credibility means that the tester must share information with the stakeholders about the test process so that they find the selected test strategy and testing activities trustworthy.

C. Cross-functional teamwork means that all team members contribute to testing in various ways. For example, involving people with the test strategy, test planning and execution as well as test reporting.

D. Co-located teamwork means that all team members, including developers and testers, must sit together in the same office, so they can quickly communicate face-to-face.

ANSWER: C

Explanation:

Cross-functional teamwork means that all team members contribute to testing in various ways. For example, involving people with the test strategy, test planning and execution as well as test reporting. This describes the whole-team approach: quality and testing are shared responsibilities, rather than work owned exclusively by a separate test department. A tester remains an essential testing specialist, but collaborates closely with developers, business representatives, and other team members to make risks, quality criteria, test ideas, and test results visible early. Team

members can contribute by reviewing requirements and acceptance criteria, identifying test conditions, preparing test data or automation, executing appropriate tests, and communicating the resulting quality information.

In Scrum, this cross-functional collaboration enables the team to deliver a potentially releasable increment each Sprint. It is also a sound practice in a sequential lifecycle. Even where analysis, development, and test activities occur in more distinct phases, early tester involvement and shared participation in quality-related work improve mutual understanding and help identify defects or misunderstandings before formal test execution. Thus, the organizational move away from isolated functional silos supports a practice that is valuable independently of the lifecycle model. The Agile Manifesto similarly emphasizes individuals and interactions, while ISTQB's Agile Tester certification covers collaboration and the whole-team approach. See the Agile Manifesto and ISTQB Certified Tester Foundation Level Agile Tester.

QUESTION NO: 3

You are planning testing for a user story that introduces a new payment provider. The existing payment process is business-critical and changes to it could prevent customers from completing orders.

Which of the following is the MOST appropriate risk-based testing action?

A. Allocate additional testing effort to payment processing, including regression testing of existing payment methods.

B. Test only the new payment provider because previously delivered payment methods were already accepted.

C. Delay all payment testing until the end of the release, when the complete system is available.

D. Reduce testing of the payment process because the story has a clearly defined acceptance criterion.

ANSWER: A

Explanation:

Allocate additional testing effort to payment processing, including regression testing of existing payment methods. Risk-based testing uses product-risk information to determine the scope, priority, and intensity of testing. Payment processing has high potential business impact because a failure can directly affect revenue and customer confidence. The introduction of a new provider also increases the likelihood of unintended effects on existing payment behavior.

Testing should therefore include focused coverage of the new provider and suitable regression tests of established payment methods. The purpose is not to test every feature with equal depth, but to direct effort toward the areas with the greatest risk. See the ISTQB Glossary definition of risk-based testing.

QUESTION NO: 4

A calculator application is being developed. The third sprint has been planned to add functionality to the calculator to allow scientific calculations.

Which TWO examples below represent activities that would likely be managed on an agile task board for the third sprint?

1) A task to design the features planned for the next sprint.

2) A task to run an acceptance test for a user story.

3) A task to automate regression tests.

4) A task to participate in training in preparation for the fourth sprint.

5) A task to produce a daily progress report for the agile team members.

A. 2, 3

B. 1, 4

C. 4, 5

D. 1, 5

ANSWER: A

Explanation:

2, 3 is correct because an agile task board visualizes the work that the team needs to perform during the current sprint to meet its Sprint Goal and complete the selected user stories. Running an acceptance test for a user story is a direct activity for establishing that the implemented scientific-calculation functionality satisfies its defined acceptance criteria. Acceptance testing is therefore part of determining whether the story is complete and potentially releasable within the sprint.

Automating regression tests is also appropriate work for the sprint task board. As the calculator gains scientific functions, automated regression coverage helps the team repeatedly check that existing calculator behavior still works after changes are integrated. Test automation is a practical way for an Agile team to maintain rapid, repeatable feedback and support the frequent regression testing needed as the product increment evolves. Such testing work can be planned, tracked, and completed alongside development work as part of delivering a quality increment. The ISTQB Agile Tester syllabus describes Agile testing as continuous and collaborative, with automation supporting rapid feedback. Scrum likewise defines the Sprint Backlog as the plan and actionable work for achieving the Sprint Goal. See the ISTQB Certified Tester Foundation Level Agile Tester certification page and the Scrum Guide.

QUESTION NO: 5

Consider the estimation of a given user story performed separately by two different teams (Team A and Team B) using the Planning Poker technique. Both teams consist of five members and use a deck of cards (0, 1, 2, 3, 5, 8, 13, 20).

Team A quickly decides on an estimate of **8 story points** after four rounds, with **no discussions** between rounds.

Team B arrives at **3 story points** after three rounds, with **discussions** between the lowest and highest estimators after each round.

Which of the following statements would you expect most likely to be true in this scenario?

A. Team B estimates that the user story can be "done" with significantly less effort in absolute terms than what was estimated by Team A

B. Team A has a better estimation process than Team B because at the end of the last round, all five team members' estimates are an exact match

C. In the estimation process, Team B, unlike Team A, can take advantage of the discussions that represent an important benefit of using Planning Poker

D. Both Team A and B have an ineffective estimation process because the best estimate is the average of the results of the first round of estimation

ANSWER: C

Explanation:

In the estimation process, Team B, unlike Team A, can take advantage of the discussions that represent an important benefit of using Planning Poker. Planning Poker is intended to be a collaborative, relative-estimation technique rather than merely a mechanism for selecting a number. After team members reveal their estimates, differences—particularly between the lowest and highest estimates—provide a useful prompt to discuss assumptions, complexity, risks, dependencies, implementation approaches, and differing interpretations of the user story. This exchange helps expose information that may not have been shared by every participant before the estimate was made.

By discussing these differing perspectives between rounds, Team B can build a more common understanding of the work and then re-estimate based on that shared understanding. The resulting story-point value is therefore supported by team communication and alignment, which is a key value of the technique. Story points are relative measures and are meaningful within a team's own estimation context; the numerical difference between the two teams' final values does not by itself establish an absolute difference in effort. The Agile Tester syllabus describes Planning Poker as an estimation approach in

which estimates are discussed, with attention to high and low estimates, before another estimation round is performed. See the ISTQB Agile Tester certification information and the Planning Poker overview by Mountain Goat Software.

QUESTION NO: 6

Which of the following sentences about the integration of development and testing activities in Agile projects is INCORRECT?

A. While developers develop automated unit test scripts, testers write automated system level tests.

B. Testers replace developers in writing unit test automation scripts.

C. Developers write acceptance criteria and test cases, together with testers.

D. Developers and testers may work as a pair to develop and test a feature.

ANSWER: B

Explanation:

“Testers replace developers in writing unit test automation scripts.” is the incorrect statement. In an Agile team, testing is a shared team responsibility, but sharing responsibility does not mean transferring ownership of all testing activities from developers to testers. Developers remain principally responsible for creating and maintaining automated unit tests because unit testing validates the smallest testable pieces of code and is closely integrated with implementation and refactoring. These tests give developers rapid technical feedback as they build the feature.

Agile testers contribute their testing expertise throughout the iteration: they collaborate on acceptance criteria and examples, help identify test conditions and risks, support automation at appropriate levels, and provide feedback on product quality. They may also pair with developers, including when developing or reviewing tests. This close collaboration improves the team’s collective quality practices, but it does not make testers substitutes for developers’ unit-test automation responsibilities. The Agile Tester syllabus emphasizes whole-team collaboration and the complementary testing contributions of developers and testers. See the ISTQB Certified Tester Foundation Level Agile Tester certification page and the Agile Alliance definition of unit testing.

QUESTION NO: 7

You are working in a team preparing a bank loan application. Your task is the preparation of acceptance tests for the following user story:

"IF a customer needs a loan for less than 50,000 Euros and they have made repayments regularly (without any delay) and the customer's monthly income is more than 3000 Euros for the last year, THEN the bank will accept the loan request; in other cases, the bank will not accept the request. A customer assistant is responsible for preparing data for approval but the approval is done by the bank manager."

Which of the following test cases can be treated as acceptance test criteria for the above user story?

1) As a customer assistant I can log in to the system and check the history of the customer account for the last year.

2) As a customer assistant I can log in to the system and check the history of the customer debts and repayments.

3) As a customer assistant I can log in to the system and change my password.

4) As a bank manager I can log in to the system and receive the information of all requests waiting for approval.

5) As a bank manager I can log in to the system and decide whether to approve a loan for a customer.

6) As a bank manager I can log in to the system within 10 seconds.

A. Acceptance test criteria are 1, 2, 3 and 6

B. Acceptance test criteria are 1, 3, 4 and 5

C. Acceptance test criteria are 2, 4, 5 and 6

D. Acceptance test criteria are 1, 2, 4 and 5

ANSWER: D

Explanation:

Acceptance test criteria are 1, 2, 4 and 5. Acceptance criteria express the conditions that make the user story acceptable to its stakeholders and provide a basis for collaborative acceptance testing. Here, the business outcome is a correctly handled loan decision: the customer assistant must be able to obtain the customer information needed to prepare the request, while the bank manager must be able to see pending requests and make the approval decision.

Checking the customer account history for the previous year supports verification of the stated income condition. Checking debt and repayment history supports verification that repayments have been regular and without delay. Receiving the information for requests awaiting approval establishes that the bank manager can access the work requiring a decision. Deciding whether to approve a customer loan directly supports the central business capability and permits testing of the specified decision rule: approval when all stated conditions are met and rejection otherwise.

These criteria are testable from the perspectives of the roles explicitly identified in the story and connect the required data preparation with the manager's approval workflow. The Agile Tester syllabus describes acceptance criteria as conditions that a user story must satisfy and as a basis for acceptance tests; see the ISTQB Agile Tester certification overview and the ISTQB glossary definition of acceptance criteria.

QUESTION NO: 8

Why is regression of software a high risk in agile projects?

A. Test-driven development means that existing functionality is not considered.

B. Test automation can cause regression of software in the test environment.

C. Regression is built into software as a safeguard against unexpected failures.

D. There is code churn due to change in business needs over several sprints.

ANSWER: D

Explanation:

There is code churn due to change in business needs over several sprints. Agile development deliberately embraces changing requirements, with functionality being added, refined, and sometimes reworked in short iterations. Each change to shared code, interfaces, data handling, or integrations can unintentionally affect behaviour that was implemented and tested in earlier sprints. As the product increment grows, the number of existing features that could be affected also grows, making regression risk significant.

The Agile Tester syllabus identifies regression testing as important because frequent change can introduce unintended side effects into previously working parts of the system. Agile teams address this risk by maintaining suitable automated regression tests, running them frequently—commonly as part of continuous integration—and ensuring that tests cover relevant acceptance criteria and critical business workflows. Fast, reliable feedback from this regression-test suite enables the team to detect a change-induced defect close to the time it is introduced, when investigation and correction are less costly. This supports the agile objective of delivering a potentially releasable increment every iteration while preserving confidence in existing product behaviour. See the ISTQB Agile Tester certification overview and the Agile Manifesto principles.

QUESTION NO: 9

Which of the following is a benefit of using a shared vocabulary between business representatives, developers and testers?

A. It removes the need to define acceptance criteria.

B. It reduces the risk that the same requirement is interpreted differently by different team members.

C. It allows developers to work without feedback from business representatives.

D. It ensures that every test can be automated at unit-test level.

ANSWER: B

Explanation:

It reduces the risk that the same requirement is interpreted differently by different team members. is correct. A shared vocabulary gives the whole team consistent names for business concepts, rules, events, and product behavior. When discussing user stories and acceptance criteria, this helps business representatives, developers, and testers develop the same understanding of what is to be built and how it should behave.

Shared vocabulary is particularly valuable when writing examples for acceptance tests or behavior-driven development, because ambiguous terms can otherwise lead to incorrect implementation and ineffective tests. It does not eliminate the need for conversation, but it makes those conversations more precise and productive. See the Cucumber guide to Behavior-Driven Development and the Manifesto for Agile Software Development.

QUESTION NO: 10

Which of the following is the MOST important contribution a tester can make when a user story is being split into smaller stories?

A. Help ensure that each resulting story remains independently testable and delivers observable value.

B. Ensure that each resulting story is assigned to the same developer.

C. Separate all testing activities into a final story for the release.

D. Remove acceptance criteria until all smaller stories are implemented.

ANSWER: A

Explanation:

Help ensure that each resulting story remains independently testable and delivers observable value. is correct. Splitting a large user story can make the work easier to estimate, implement, and complete within an iteration. The tester can help the team identify meaningful slices, clarify acceptance criteria for each slice, and ensure that each resulting story has testable behavior and a demonstrable outcome.

Splitting work only by technical layers, such as creating a database story followed by a user-interface story, may create dependencies and deliver no observable customer value until all layers are completed. The tester's quality perspective helps the team identify vertical slices that can be tested and accepted independently. This supports the Agile objective of delivering working, potentially releasable increments frequently. See the Agile Manifesto principles and the ISTQB Certified Tester Foundation Level Agile Tester certification page.