iSQI

Exam Questions CTFL-AT

Certified Tester Foundation Level Agile Tester

NEW QUESTION 1
Which of the following activities are done in release planning?
1) Identifying testable user stories with acceptance criteria.
2) Elaborating the user stories into tasks.
3) Prioritizing the user stories.
4) Creating acceptance tests for the user stories.
5) Analyzing risks for each of the user stories.
6) Performing high level estimation for the release.

A. Activities 1, 4 and 6
B. Activities 2 and 4
C. Activities 2, 3 and 5
D. Activities 1, 3 and 6

Answer: D

Explanation:
Release planning is a process of defining the scope and timeline for an iterative or incremental product development project. It is used in agile or hybrid projects where a mid- to long-term planning of the product or system development or integration is required12. Release planning involves the following activities:
? Identifying testable user stories with acceptance criteria. User stories are short descriptions of the features or functionalities that the customer or user wants from the product. Acceptance criteria are the conditions that must be met for the user story to be considered done and acceptable. Identifying testable user stories with acceptance criteria helps to define the scope and quality of the release13.
? Prioritizing the user stories. User stories are prioritized based on the value they deliver to the customer or user, as well as the dependencies, risks, and costs associated with them. Prioritizing the user stories helps to determine the order and frequency of the releases13.
? Performing high level estimation for the release. High level estimation is a technique to estimate the effort, time, and resources needed to complete the user stories in the release. High level estimation can be done using various methods, such as analogy, expert judgment, planning poker, etc. Performing high level estimation for the release helps to set realistic and achievable goals and deadlines13.
Therefore, activities 1, 3 and 6 are done in release planning. Activities 2, 4 and 5 are done in iteration planning, which is a more detailed and short-term planning of the work to be done in each iteration or sprint13. References: 1: ISTQB® Foundation Level Agile Tester Syllabus, Section 2.2, Fundamental Agile Testing Principles, Practices and Processes1; 2: Agile Release Planning in Hybrid and Agile Projects4; 3: How to Create an Agile Release Plan5

NEW QUESTION 2
Which of the following statements would you expect to be the MOST direct advantage of the whole-team approach?

A. Having at least once a day an automated build and test process that detects integration errors early and quickly.
B. Avoiding requirements misunderstandings which may not have been detected until later in the development cycle when they are more expensive to fix.
C. Capitalizing on the combined skills of business representatives, testers and developers working together to contribute to project success.
D. Reducing the involvement of business representatives because of the increased communication and collaboration between testers and developers.

Answer: C

Explanation:
The whole-team approach is a principle of agile testing that involves everyone with different knowledge and skills to ensure project success. The whole-team approach means that the business representatives, testers, and developers work together in every step of the development process, from planning to delivery. The whole-team approach aims to enhance communication and collaboration within the team, leverage the various skill sets of the team members, and make quality everyone's responsibility12. Therefore, the statement C is the most direct advantage of the whole-team approach, as it captures the essence of the principle and its benefits. The other statements are not directly related to the whole-team approach, or are incorrect. Statement A is about continuous integration, which is a practice of agile development that involves having at least once a day an automated build and test process that detects integration errors early and quickly. Continuous integration is not a direct consequence of the whole-team approach, although it may be facilitated by it13. Statement B is about avoiding requirements misunderstandings, which may be a benefit of the whole-team approach, but not the most direct one. The whole-team approach does not only focus on requirements, but also on design, implementation, testing, and delivery. Moreover, avoiding requirements misunderstandings may also depend on other factors, such as the quality of the user stories, the use of acceptance criteria, and the feedback from the customers and users14. Statement D is incorrect, as it contradicts the whole-team approach. The whole-team approach does not reduce the involvement of business representatives, but rather increases it. Business representatives are an integral part of the whole-team approach, as they provide the vision, the value, and the validation of the product. They collaborate with the testers and developers to define the features, prioritize the backlog, and verify the outcomes12. References: ISTQB Foundation Level Agile Tester Syllabus1, Section 1.2.1, page 9; What is Whole Team Approach in Agile Testing?2, Section What is Whole Team Approach?; Continuous Integration3, Section What is Continuous Integration?; Effective User Stories - 3C's and INVEST Guide4, Section The 3 C's (Card, Conversation, Confirmation) of User Stories.

NEW QUESTION 3
Consider an online application that allows registered users to pay the annual car tax based on the vehicle's engine power in kW. Given the following user story:
"As a customer I need the online application to calculate the annual car tax amount that I need to pay for my car:
* If the power of the vehicle is less than 20 kW, then the annual car tax is free
* If the power of the vehicle is more or equal than 20 kW but less or equal than 150 kW, then the annual car tax is 250 Euros
* If the power of the vehicle is more than 150 kW, then the annual car tax is 750 Euros" What is the MOST suitable use of a black-box test design technique for this user story?

A. Decision table testin
B. Test the following conditions:Conditions=registered user logged in; inserted power of the vehicle=20kW; Action=Car tax paid
C. State transition testin
D. Test the transitions between the following states: logging in, inserting the power of the vehicle, making payment, logging ou
E. Equivalence partitionin
F. Test the annual car tax value for the following partitions: [power of the vehicle<20 kW ; 20 kW power of the vehicles150 kW; power of the vehicle>150 kW]
G. Use case testing Test the following use case (Actor=registered user): Pre-condition=registered user logged in Scenario=registered user inserts the power of the vehicle, making payment and logs out Post-condition=car tax paid and registered user logged out

Answer: C

Explanation:
Equivalence partitioning is a black-box test design technique that divides the input domain of a system into classes of data from which test cases can be derived.

The idea is that if a system works correctly for a representative value from an equivalence class, it will work correctly for all values from that class, and vice versa. Equivalence partitioning reduces the number of test cases by eliminating redundant ones. For the given user story, equivalence partitioning is the most suitable technique because it can test the different outcomes of the annual car tax calculation based on the power of the vehicle, which is the main input for the system. By testing one value from each partition, the tester can verify the functionality of the system and detect any errors in the calculation logic. The other techniques are not as suitable because they do not focus on the inputdomain of the system, but rather on the conditions, transitions, or scenarios that are not directly related to the user story. References:
? : ISTQB® Foundation Level Agile Tester Syllabus, Version 2014, Section 2.2.2
? : ASTQB Agile Tester Certification Resources, Agile Testing Foundations, Chapter 3, Section 3.2.2
? : 3

NEW QUESTION 4
Which of the following statements about a test charter are CORRECT?
1) It is used mainly in exploratory tests.
2) It is used to monitor a test process.
3) It may make reference to user stories.
4) It contains notes taken during a test session.
5) It is used to outline the company test policy.

A. 1, 2, 5
B. 2, 3, 4
C. 2, 4, 5
D. 1, 3, 4

Answer: D

Explanation:
A test charter is a document that describes the scope, objective, and approach of an exploratory testing session. It is used mainly in exploratory tests to guide the tester's actions and record the findings. A test charter may make reference to user stories, requirements, risks, or other sources of information that are relevant to the testing mission. A test charter also contains notes taken during a test session, such as test ideas, test results, bugs, issues, and observations. A test charter is not used to monitor a test process, as it is not a formal metric or report. It is also not used to outline the company test policy, as it is specific to a particular test session and context. References: ISTQB® Foundation Level Agile Tester Syllabus, Section 2.2.3, page 18; ASTQB Agile Tester Certification Resources, Section 2.2.3, page 18; How to Write an Exploratory Test Charter, Creating an Exploratory Testing Charter, What is Exploratory Testing?.

NEW QUESTION 5
What is the main benefit of the Test Pyramid?

A. It means testing is involved early in the development cycle.
B. It helps in evaluating the amount of test cases needed.
C. It shows complexity of testing activities.
D. It acts as a metric for testing progress.

Answer: B

Explanation:
The Test Pyramid is a model for organizing tests in a way to make the process of testing faster, efficient and cost-effective. This model focusses on getting maximum functional testing getting covered by faster and less brittle tests like Unit and API tests1. The main benefit of the Test Pyramid is that it helps in evaluating the amount of test cases needed for each level of testing. The Test Pyramid suggests that the number of test cases should decrease as we move up the pyramid, from unit tests to integration tests to end-to-end tests. This is because unit tests are more granular, isolated, and easy to write and maintain, while end-to-end tests are more complex, dependent, and brittle. The Test Pyramid also helps in balancing the test coverage and the test execution time, as unit tests provide high coverage and low execution time, while end-to-end tests provide low coverage and high execution time. By following the Test Pyramid, teams can optimize their testing efforts and resources, and ensure that they have a sufficient and effective test suite for their software. References: ISTQB® Foundation Level Agile Tester Syllabus, Section 2.2.1, page 16; ASTQB Agile Tester Certification Resources, Section 2.2.1, page 16; What is Test Pyramid : Getting started with Test Automation Pyramid, The Practical Test Pyramid - Martin Fowler, Testing Pyramid: What Is It and How To Use It | Solvd.

NEW QUESTION 6
Which ONE of the following is an example of a typical "Business-oriented work product"?

A. The released product.
B. Acceptance testing entry criteria.
C. A user manual.
D. Usability testing test results.

Answer: C

Explanation:
Business-oriented work products are those that describe what is needed (e.g., requirements specifications) and how to use it (e.g., user documentation). A user manual is an example of a business-oriented work product, as it provides instructions and guidance on how to use the product from the user's perspective. A user manual may also contain information about the product's features, benefits, and limitations. A user manual is typically written by technical writers, who may collaborate with developers, testers, and business analysts to ensure the accuracy and clarity of the content. A user manual may be delivered in various formats, such as printed, online, or interactive. References: ISTQB® Foundation Level Agile Tester Syllabus1, Section 1.2.1, page 10; ASTQB Agile Tester Certification Resources2, Section 1.2.1, page 10.

NEW QUESTION 7
Your agile team is using the Testing Quadrants to ensure that all important test levels and test types are covered in the test plan.
In relation to Quadrant 3 - business facing and product critique, what should be considered for the plan?

A. Exploratory Testing
B. Prototype Testing
C. Performance Testing

D. Functional Testing

Answer: A

Explanation:
Exploratory testing is a type of testing that involves simultaneous learning, test design, and test execution. It is suitable for Quadrant 3 because it is business facing
and product critique, meaning that it focuses on the user's perspective and the quality attributes of the product. Exploratory testing can help discover new risks, requirements, and defects that may not be covered by other test levels and test types. It can also provide feedback on the usability, functionality, and reliability of the product. References: ISTQB® Foundation Level Agile Tester Syllabus1, Section 2.3.2, page 17; ISTQB® Glossary of Testing Terms2, version 4.0, page 23.

NEW QUESTION 8
You are a tester in an agile team. The user story you are due to test is still under development so your tests are blocked. The main issue holding progress on this user story is that the developer's unit tests are constantly failing.
As an agile tester, which of the following actions should you take?

A. Review the design of the problematic user story and improve it where possible.
B. Create a bug report for each of your blocked tests.
C. Work together with the developer, suggesting reasons why the tests are failing.
D. Use the time to improve and automate existing test cases of other user stories.

Answer: C

Explanation:
As an agile tester, you should work together with the developer, suggesting reasons why the tests are failing. This is an example of the agile principle of collaboration and communication within the team, as well as the agile testing practice of early and frequent feedback. By working together with the developer, you can help to identify and resolve the root causes of the test failures, as well as share your testing knowledge and perspective. This can lead to faster and better quality delivery of the user story, as well as improved team relationships and trust.
Option A is not a good action, because reviewing and improving the design of the user story is not the tester's responsibility, and it may not address the test failures. Option B is also not a good action, because creating bug reports for blocked tests is not an agile way of handling issues, and it may create unnecessary overhead and waste. Option D is not a good action, because it does not help to unblock the current user story, and it may distract you from the sprint goal and the team's focus.
References: ISTQB Foundation Level Agile Tester Syllabus, Section 2.3.1, page 171; ISTQB Foundation Level Agile Tester Sample Exam Questions, Question 2.3.1-2, page 82

NEW QUESTION 9
What is the definition of agile software development?

A. Testing carried out informally where no formal test preparation or execution takes place, no recognized test design technique is used and there are no expectations for results.
B. A group of software development methodologies based on iterative incremental development with self-organizing cross-functional teams who cooperate to define requirements and to implement the solution.
C. A framework to describe the software development lifecycle activities from requirements specification to maintenance where test planning of the various test levels is done as soon as the test basis is ready
D. A way of developing software where the test cases are developed, and often automated, before the software under test is developed.

Answer: B

Explanation:
Agile software development is a term that encompasses a group of software development methodologies that are based on iterative incremental development, where requirements and solutions evolve through collaboration between self-organizing cross- functional teams. Agile methods promote adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourage rapid and flexible response to change. Some examples of agile methods are Scrum, Extreme Programming (XP),
Kanban, and Lean Software Development. References:
? : ISTQB® Foundation Level Agile Tester Syllabus, Version 2014, Section 1.1.1
? : ASTQB Agile Tester Certification Resources, Agile Testing Foundations, Chapter 1, Section 1.1.1

NEW QUESTION 10
Which of the following statements about the Planning poker test estimate technique are CORRECT?
1) Planning poker is a consensus based technique using a deck of cards.
2) A low test estimate usually means the story should be broken down into multiple smaller stories.
3) A high test estimate usually means the story should be broken down into multiple smaller stories.
4) One poker round is played and then consensus has to be reached.
5) The risk level of each backlog item should be decided before the poker session.

A. 1, 3, 5
B. 1, 2, 3
C. 2, 3, 4
D. 1, 2, 4

Answer: A

Explanation:
Planning poker is a consensus-based technique for agile estimation, using a deck of cards with predefined numerical values, usually based on the Fibonacci sequence or a modified version12. Therefore, statement 1 is correct. A high test estimate usually means that the user story or task is too complex, ambiguous, or risky, and should be broken down into multiple smaller stories that are easier to understand and estimate13. Therefore, statement 3 is correct. The risk level of each backlog item should be decided before the poker session, as it can affect the estimation process and the prioritization of the work14. Therefore, statement 5 is correct. Statement 2 is incorrect, as a low test estimate usually means that the user story or task is simple, clear, and well-defined, and does not need to be broken down further13. Statement 4 is incorrect, as planning poker can involve multiple rounds of estimation, reveal, and discussion, until the team reaches a consensus or agrees to defer the item12. References: 1: ISTQB® Foundation Level Agile Tester Syllabus, Section 3.3.1, Test Automation1; 2: ASTQB Agile Tester

Certification Resources, Section 3.3.1, Test Automation2; 3: Planning Poker: An Agile Estimating and Planning Technique3; 4: Planning poker: The all-in strategy for Agile estimation - Asana4

NEW QUESTION 10
Which of the following describes the main purpose of a task management tool in agile projects?

A. A task management tool is used by team members to share ideas and collaborate on assigned tasks.
B. A task management tool is used to manage and track user stories, tests and other tasks.
C. A task management tool is used to store source code and automated tests.
D. A task management tool allows developers to continuously integrate their code.

Answer: B

Explanation:
A task management tool is a software application that helps agile teams plan, organize, and monitor their work. A task management tool typically allows the team to create, assign, prioritize, update, and track user stories, tests, and other tasks that are part of the agile project. A task management tool can also provide various views and reports to visualize the progress and status of the project, such as Kanban boards, burndown charts, velocity charts, etc. Some examples of task management tools are Jira, Trello, Asana, and Monday.com12345. References: ISTQB® Foundation Level Agile Tester Syllabus, Section 2.1.1, page 13; ASTQB Agile Tester Certification Resources, Section 2.1.1, page 13.

NEW QUESTION 13
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:
According to the ISTQB Tester Foundation Level Agile Tester syllabus, an agile task board is a visual tool that displays the status of the work items in an agile sprint. The task board typically shows the user stories, tasks, and their progress from “to do” to “done”. The task board helps the agile team to monitor and coordinate their work, and to communicate with stakeholders. Therefore, the examples that represent activities that would likely be managed on an agile task board for the third sprint are those that are related to the user stories, tasks, and their progress in the current sprint. Option A is the correct answer, as it contains two examples of such activities: running an acceptance test for a user story, and automating regression tests. These are both tasks that are part of the testing process in the current sprint, and their status can betracked on the task board. Option B is not a correct answer, as it contains two examples of activities that are not related to the current sprint: designing the features planned for the next sprint, and participating in training in preparation for the fourth sprint. These are both activities that are part of the planning or learning process for the future sprints, and they are not managed on the task board. Option C is also not a correct answer, as it contains two examples of activities that are not related to the current sprint: participating in training in preparation for the fourth sprint, and producing a daily progress report for the agile team members. These are both activities that are part of the learning or reporting process, and they are not managed on the task board. Option D is also not a correct answer, as it contains two examples of activities that are not related to the current sprint: designing the features planned for the next sprint, and producing a daily progress report for the agile team members. These are both activities that are part of the planning or reporting process, and they are not managed on the task board. References: ISTQB Tester Foundation Level Agile Tester syllabus, section 2.1.1, page 14; ISTQB Tester Foundation Level Agile Tester syllabus, section 2.1.2, page 15; ISTQB Tester Foundation Level Agile Tester syllabus, section 2.2.1, page 16; ISTQB Tester Foundation Level Agile Tester syllabus, section 2.2.2, page 17.

NEW QUESTION 16
Which of the following statements is FALSE regarding early and frequent feedback?

A. Early feedback decreases the amount of time needed for system testing.
B. Early feedback promotes early discovery and resolution of quality problems.
C. Early feedback provides the Agile team with information on its productivity.
D. Early feedback helps to deliver a product that better reflects what the customer wants.

Answer: A

Explanation:
Early and frequent feedback is one of the core values of Agile development. It helps the Agile team to deliver features with the highest business value first, to discover and resolve quality problems as soon as possible, to provide information on the team’s productivity and progress, and to ensure that the product meets the customer’s expectations and needs. However, early feedback does not necessarily decrease the amount of time needed for system testing, as system testing is still an important activity in Agile projects to verify the integration and functionality of the whole system. Early feedback may reduce the number of defects found in system testing, but it does not eliminate the need for system testing. References: ISTQB Foundation Level Agile Tester Extension Syllabus1, page 10; ISTQB Agile Tester Sample Exam2, question 11.

NEW QUESTION 21
You have been asked to explain to your client how to define acceptance criteria that are fully testable. Which of the following is the BEST EXAMPLE for testable acceptance criteria?

A. The “ID” field must accept input value of a length between 2 and 10 characters.
B. The interface to External System shall be specified.
C. Action “Reopen” must be available only for a user with a specific authorization level.
D. The program's icon should be clear and attractive.

Answer: A

Explanation:
According to the ISTQB Tester Foundation Level Agile Tester syllabus, acceptance criteria are a set of conditions that a user story must satisfy to be accepted by the customer or stakeholder. Acceptance criteria should be testable, meaning that they can be verified by objective measurements or observations. Testable acceptance criteria should be clear, unambiguous, complete, and consistent. Therefore, option A is the best example for testable acceptance criteria, as it specifies a clear and measurable condition for the input value of the ID field. Option B is not a good example for testable acceptance criteria, as it is vague and does not define any specific condition or expectation for the interface to External System. Option C is not a good example for testable acceptance criteria, as it is incomplete and does not specify what the specific authorization level is or how it is determined. Option D is not a good example for testable acceptance criteria, as it is subjective and not measurable. What is clear and attractive for one user may not be for another. References: ISTQB Tester Foundation Level Agile Tester syllabus, section 1.1.1, page 7; ISTQB Tester Foundation Level Agile Tester syllabus, section 1.1.2, page 8; ISTQB Tester Foundation Level Agile Tester syllabus, section 3.1.1, page 23; ISTQB Tester Foundation Level Agile Tester syllabus, section 3.1.2, page 24. 3of30

NEW QUESTION 26
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

CTFL-AT Practice Exam Features:

* CTFL-AT Questions and Answers Updated Frequently

* CTFL-AT Practice Questions Verified by Expert Senior Certified Staff

* CTFL-AT Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* CTFL-AT Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year