iSQI

Exam Questions CTFL-AT

Certified Tester Foundation Level Agile Tester

About Exambible

Your Partner of IT Exam

Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

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
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 3
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 4
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 5
Which of the following is the BEST way for a test team to keep its independence when working in an Agile development environment?

A. Share the Test Strategy with the Agile development team, but not the details of the Test Cases.
B. Locate the team that develops the test automation framework in a different location to the Agile development team.
C. Assign testers to be members of the Agile team, but ensure the testers report to a different manager than the developers.
D. Co-locate only some of the testers with the Agile development team, while the rest of the testers are in a different location.

Answer: C

Explanation:
According to the ISTQB Tester Foundation Level Agile Tester syllabus, one of the key principles of agile testing is that testers are integrated into the agile team and work closely with developers and other stakeholders. However, this does not mean that testers lose their independence or objectivity. Testers should still be able to provide an unbiased view of the quality of the software and challenge the assumptions and decisions made by the team. Therefore, option C is the best way for a test team to keep its independence when working in an agile development environment, as it allows testers to be part of the agile team, but also report to a different manager than the developers, who can support their professional development and ensure their independence. Option A is not a good way to keep independence, as it limits the transparency and collaboration between testers and developers, which are essential for agile testing. Option B is also not a good way to keep independence, as it creates a physical and organizational barrier between the test automation team and the agile development team, which can hinder communication and feedback. Option D is also not a good way to keep independence, as it creates an inconsistency and imbalance between the testers who are co-located with the agile development team and those who are not, which can affect the quality and efficiency of the testing process. References: ISTQB Tester Foundation Level Agile Tester syllabus, section 1.2.1, page 91; ISTQB Tester Foundation Level Agile Tester syllabus, section 1.2.2, page 101; ISTQB Tester Foundation Level Agile Tester syllabus, section 2.1.1, page 141; ISTQB Tester Foundation Level Agile Tester syllabus, section 2.2.1, page 161; ISTQB Tester Foundation Level Agile Tester syllabus, section 2.2.2, page 171.

NEW QUESTION 6
Which agile development approach incorporates the following practices:
* a project is divided into iterations called sprints
* each sprint results in a potentially releasable/shippable product?

A. Kanban
B. Extreme Programming
C. Continuous Integration
D. Scrum

Answer: D

Explanation:
Scrum is an agile development approach that incorporates the following practices:
? a project is divided into iterations called sprints, which are typically 2-4 weeks long
? each sprint starts with a planning meeting, where the team selects a subset of user stories from the product backlog to work on
? each sprint ends with a review meeting, where the team demonstrates the potentially releasable/shippable product increment to the stakeholders and collects feedback
? each sprint also includes a retrospective meeting, where the team reflects on the process and identifies areas for improvement123 References: 1: ISTQB® Foundation Level Agile Tester Syllabus, Section 2.1, Agile Software Development1; 2: ASTQB Agile Tester Certification Resources, Section 2.1, Agile Software Development2; 3: What is Agile? | Atlassian3

NEW QUESTION 7
Which tasks are typically performed by a tester on an Agile project?
1) Implementing test strategy.
2) Documenting business requirements.
3) Measuring and reporting test coverage.
4) Coaching development team in relevant aspects of testing.

5) Executing test-driven development tests.

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

Answer: C

Explanation:
A tester on an Agile project typically performs the following tasks12:
? Implementing test strategy: A tester helps to define and implement the test strategy for the Agile project, which includes the test approach, test levels, test types, test techniques, test tools, test environment, test data, test metrics, and test documentation.
? Measuring and reporting test coverage: A tester measures and reports the test coverage of the product features and quality attributes, such as functionality, usability, performance, security, etc. Test coverage can be expressed in terms of test cases, test scenarios, test sessions, test conditions, test data, code, etc.
? Coaching development team in relevant aspects of testing: A tester coaches the development team in relevant aspects of testing, such as test design, test execution, test automation, test-driven development, behavior-driven development, exploratory testing, etc. A tester also helps the development team to improve their testing skills and practices.
The following tasks are not typically performed by a tester on an Agile project:
? Documenting business requirements: Business requirements are usually documented by the product owner or the business analyst, not by the tester. The tester may review and provide feedback on the business requirements, but the tester is not responsible for documenting them.
? Executing test-driven development tests: Test-driven development tests are usually executed by the developers, not by the tester. The tester may assist the developers in creating and reviewing the test-driven development tests, but the tester is not responsible for executing them.
Therefore, the correct answer is C, as it contains the tasks that are typically performed by a tester on an Agile project. References: ISTQB Foundation Level Agile Tester Extension Syllabus1, pages 14-15, 18-19, 22-23; ISTQB Agile Tester Sample Exam2, question 17.

NEW QUESTION 8
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 9
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:
Regression of software is the risk that a change in one part of the software causes a defect in another part of the software that was previously working correctly. Regression of software is a high risk in agile projects because there is code churn due to change in business needs over several sprints. Code churn is the amount of code that is added, modified, or deleted in a software project over time. Code churn can indicate the volatility and complexity of the software, and the frequency and magnitude of the changes. In agile projects, code churn can be high because the business needs and the user requirements can change rapidly and frequently over several sprints, which are short iterations of development and testing. This means that the software is constantly evolving and adapting to the changing needs, and that the existing functionality and quality may be affected by the new or modified code. Therefore, regression of software is a high risk in agile projects, and it requires effective testing strategies and techniques to prevent, detect, and fix the regression defects. References: ISTQB® Foundation Level Agile Tester Syllabus1, Section 2.3.2, page 17; ISTQB® Glossary of Testing Terms2, version 4.0, pages 36 and 55

NEW QUESTION 10
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 10
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 13
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 15
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 19
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 way
D. For example, involving people with the test strategy, test planning and execution as well as test reporting.
E. 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 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. Cross-functional teamwork means that all team members, regardless of their functional roles, collaborate and share their skills and knowledge to achieve a common goal. In the context of testing, this means that testing is not seen as a separate activity or phase, but as an integral part of the development process. All team members contribute to testing in various ways, such as:
? Involving people with the test strategy, test planning and execution as well as test reporting. This can help ensure that the testing activities are aligned with the business objectives, the user needs, and the technical requirements. It can also help improve the test coverage, the test quality, and the test efficiency.
? Sharing the responsibility for testing among the team members. This can help reduce the workload and the dependency on a single tester or a testing team. It can also help increase the feedback and the communication among the team members, and foster a culture of quality and learning.
? Leveraging the diverse skills and perspectives of the team members. This can help enhance the test design and the test execution by applying different techniques, tools, and approaches. It can also help identify and address the risks, the issues, and the opportunities for improvement from various angles.
References: ISTQB® Foundation Level Agile Tester Syllabus1, Section 1.2.1, page 9; ISTQB® Glossary of Testing Terms2, version 4.0, page 16.

NEW QUESTION 20
Which of the following statements about Agile retrospectives is CORRECT?

A. During Agile retrospectives, testers should be encouraged to provide constructive suggestions only on non-testing activities.
B. In an Agile retrospective the moderator can encourage and make sure that good practices are kept by the team, by asking what the team is doing well.
C. Agile retrospectives should be focused mainly on impediments that are outside the control of the team because these issues are more challenging.
D. Unlike working sessions or meetings held in non-Agile projects, Agile retrospectives do not require follow-up activities.

Answer: B

Explanation:
An Agile retrospective is a regular meeting where the team reflects on their work process and identifies the areas for improvement12. The following statements about Agile retrospectives are correct12:
? During Agile retrospectives, testers should be encouraged to provide constructive
suggestions on both testing and non-testing activities, as testing is an integral part of the Agile team and testers can contribute to the overall quality of the product and the process.
? In an Agile retrospective, the moderator can encourage and make sure that good practices are kept by the team, by asking what the team is doing well. This helps to reinforce the positive aspects of the team's work and to appreciate the team members' efforts and achievements.
? Agile retrospectives should be focused mainly on impediments that are within the control of the team because these issues are more actionable and can be resolved by the team. Impediments that are outside the control of the team should also be discussed, but they may require the involvement of other stakeholders or external parties to be addressed.
The following statement about Agile retrospectives is incorrect12:
? Unlike working sessions or meetings held in non-Agile projects, Agile retrospectives do require follow-up activities. The team should agree on the action items that result from the retrospective and assign them to the responsible team members. The team should also monitor the progress and effectiveness of the action items in the next iteration and review them in the next retrospective.
Therefore, the correct answer is B, as it is the only statement that is correct about Agile retrospectives. References: ISTQB Foundation Level Agile Tester Extension Syllabus1, page 24; ISTQB Agile Tester Sample Exam2,

NEW QUESTION 24
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 27
You are working on an Agile project and have been asked to implement exploratory testing for the current sprint. Which one of the following is a correct approach to adopt?

A. Allocate independent testers to design exploratory tests using test charters in time boxed session
B. Plan to run all sessions in parallel with each session lasting more than 5hours.
C. Ask experienced testers to try and find new defects by using the system without the constraint of documentation and tools.
D. Use testers who have not been involved in the sprint to write new test cases from the user storie
E. These test cases are then executed in a time boxed session for the sprint.
F. Ask experienced testers to prepare test charters for time boxed sessions lasting no more than 2hour
G. Tests should be designed and executed within each session using heuristics, creativity and intuition.

Answer: D

Explanation:
Exploratory testing is a testing approach that emphasizes learning, creativity, and adaptability. It involves simultaneous test design and test execution, where the tester uses heuristics, intuition, and experience to explore the system under test and discover new information12. Exploratory testing can be performed in an Agile project to complement other testing activities, such as test-driven development, behavior-driven development, and acceptance test-driven development12.
The correct approach to adopt for exploratory testing in an Agile project is D, as it follows the best practices for exploratory testing1234:
? Ask experienced testers to prepare test charters for time boxed sessions lasting no more than 2 hours: A test charter is a brief document that describes the scope, objective, and strategy of an exploratory testing session. A test charter helps to guide the tester's exploration and to document the results. A time box is a fixed period of time allocated for an exploratory testing session. A time box helps to focus the tester's attention and to limit the scope of exploration. A time box should not be too long, as it may reduce the tester's concentration and creativity. A recommended duration for a time box is between 45 minutes and 2 hours.
? Tests should be designed and executed within each session using heuristics, creativity and intuition: Exploratory testing is an iterative and interactive process, where the tester designs and executes tests based on the observations and feedback from the system under test. The tester uses heuristics, which are rules of thumb or shortcuts that help to simplify the testing problem and to generate test ideas. The tester also uses creativity and intuition, which are mental abilities that help to generate novel and useful solutions and to make judgments based on incomplete or uncertain information.
The incorrect approaches to adopt for exploratory testing in an Agile project are A, B, and C, as they violate the principles and practices of exploratory testing1234:
? A: Allocate independent testers to design exploratory tests using test charters in time boxed sessions. Plan to run all sessions in parallel with each session lasting more than 5 hours: This approach is incorrect because it does not involve simultaneous test design and test execution, which is the essence of exploratory testing. It also uses too long time boxes, which may reduce the tester's concentration and creativity. It also does not leverage the collaboration and communication within the Agile team, as it isolates the testers from the developers and other stakeholders.
? B: Ask experienced testers to try and find new defects by using the system without the constraint of documentation and tools: This approach is incorrect because it does not use test charters, which are essential for guiding and documenting the exploratory testing sessions. It also does not use heuristics, creativity, and intuition, which are important for generating test ideas and making decisions. It also implies that exploratory testing is an unstructured and random activity, which is a common misconception. Exploratory testing is a disciplined and systematic approach that requires planning, analysis, and evaluation.
? C: Use testers who have not been involved in the sprint to write new test cases from the user stories. These test cases are then executed in a time boxed session for the sprint: This approach is incorrect because it does not involve simultaneous test design and test execution, which is the essence of exploratory testing. It also uses testers who have not been involved in the sprint, which may reduce their understanding of the system under test and the customer needs. It also does not use test charters, which are essential for guiding and documenting the exploratory testing sessions. It also does not use heuristics, creativity, and intuition, which are important for generating test ideas and making decisions.
References: ISTQB Foundation Level Agile Tester Extension Syllabus1, page 23; ISTQB Agile Tester Sample Exam2, question 19; Exploratory Testing; ISTQB Agile Tester #56 – What is Exploratory testing?

NEW QUESTION 32
Which of the following is NOT a statement of value from the Agile Manifesto?

A. Working software over comprehensive documentation
B. Processes and tools over individuals and interactions.
C. Responding to change over following a plan.
D. Customer collaboration over contract negotiation.

Answer: B

Explanation:
The Agile Manifesto is a declaration of four values and twelve principles that guide the Agile software development approach12. The four values of the Agile Manifesto are12:
? Individuals and interactions over processes and tools
? Working software over comprehensive documentation
? Customer collaboration over contract negotiation
? Responding to change over following a plan
These values emphasize the importance of human collaboration, working product, customer feedback, and adaptability over rigid processes, extensive documentation, fixedcontracts, and predefined plans. The values do not imply that the items on the right are not important, but rather that the items on the left are more important and should be prioritized.
Therefore, the statement that is NOT a value from the Agile Manifesto is B, as it contradicts the first value of the Agile Manifesto. The correct statement should be "Individuals and interactions over processes and tools". References: ISTQB Foundation Level Agile Tester Extension Syllabus1, page 10; ISTQB Agile Tester Sample Exam2, question 1.

NEW QUESTION 37
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 39
Which of the following allows a developer to define accurate unit tests focused on business needs?

A. Design-Driven Development
B. Behavior-Driven Development
C. Test-Driven Development
D. Acceptance Test-Driven Development

Answer: B

Explanation:
Behavior-Driven Development (BDD) is a software development approach that allows a developer to define accurate unit tests focused on business needs. BDD uses a common language that is understandable by both technical and non-technical stakeholders, such as Given-When-Then scenarios.BDD helps to align the development and testing activities with the customer expectations and business goals . 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

NEW QUESTION 44
.....

Relate Links

100% Pass Your CTFL-AT Exam with Exambible Prep Materials

https://www.exambible.com/CTFL-AT-exam/

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - https://www.exambible.com/