

ISTQB Agile Tester Practice Test (Sample)

Study Guide



Everything you need from our exam experts!

This is a sample study guide. To access the full version with hundreds of questions,

Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain from reliable sources accurate, complete, and timely information about this product.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	6
Answers	9
Explanations	11
Next Steps	17

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Don't worry about getting everything right, your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 - 45 minutes). Review a handful of questions, reflect on the explanations, and take breaks to retain information better.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning.

7. Use Other Tools

Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly — adapt the tips above to fit your pace and learning style. You've got this!

SAMPLE

Questions

- 1. What is the test planning activity focused on?**
 - A. Evaluating test outcomes**
 - B. Creating a final test report**
 - C. Establishing or updating a test plan**
 - D. Determining test execution tools**
- 2. What does 'user experience' focus on in software development?**
 - A. Code quality**
 - B. User interactions and satisfaction**
 - C. Testing methodologies**
 - D. Project management practices**
- 3. What is necessary to evaluate the exit criteria in the test process?**
 - A. Calculating resource requirements**
 - B. Reporting on test progress and outcomes**
 - C. Establishing test execution tools**
 - D. Planning test environments**
- 4. Which term refers to a statement of test objectives and ideas for testing, particularly in exploratory scenarios?**
 - A. Test charter**
 - B. Test design technique**
 - C. Test condition**
 - D. Test execution**
- 5. What role does 'validation' play in the software development lifecycle?**
 - A. Documenting the software requirements**
 - B. Ensuring the software conforms to its intended use**
 - C. Designing user interfaces**
 - D. Conducting testing in the development phase**

- 6. What type of testing determines whether a system satisfies the acceptance criteria?**
- A. Exploratory testing**
 - B. Unit testing**
 - C. Acceptance testing**
 - D. Integration testing**
- 7. What is the overall purpose of a test scope?**
- A. To summarize all test activities**
 - B. To define the features to be tested within a component**
 - C. To outline testing tools to be used**
 - D. To identify test failures**
- 8. What does the term 'test design technique' refer to in testing?**
- A. A method for deriving test cases**
 - B. A set of test execution outcomes**
 - C. A collection of test environment tools**
 - D. Pre-conditions needed for testing**
- 9. What type of testing involves executing the software of a component or system?**
- A. Dynamic testing**
 - B. Static testing**
 - C. Unit testing**
 - D. Integration testing**
- 10. In the context of testing quadrants, what aspects does one dimension address?**
- A. Technology-facing versus business-facing**
 - B. Static versus dynamic testing**
 - C. Automated versus manual testing**
 - D. Planned testing versus exploratory testing**

Answers

SAMPLE

1. C
2. B
3. B
4. A
5. B
6. C
7. B
8. A
9. A
10. A

SAMPLE

Explanations

SAMPLE

1. What is the test planning activity focused on?

- A. Evaluating test outcomes
- B. Creating a final test report
- C. Establishing or updating a test plan**
- D. Determining test execution tools

The focus of the test planning activity is primarily on establishing or updating a test plan. This involves defining the scope, objectives, resources, schedule, and overall strategy for the testing effort. During this phase, the team considers what needs to be tested, what risks may exist, and how to address those risks effectively. It also includes identifying the roles and responsibilities of team members, selecting testing techniques, and determining the deliverables for the testing phase. This foundational step is crucial because a well-defined test plan helps ensure that all aspects of testing are systematically considered and organized, leading to a more efficient and effective testing process. By having a clear test plan in place, teams can manage expectations and align their testing approach with the overall goals of the project. The other activities mentioned, such as evaluating test outcomes and creating final test reports, occur after the planning phase and are focused on monitoring and assessing the results of the executed tests rather than establishing the groundwork for them.

2. What does 'user experience' focus on in software development?

- A. Code quality
- B. User interactions and satisfaction**
- C. Testing methodologies
- D. Project management practices

The focus of 'user experience' in software development is primarily on user interactions and satisfaction. This area emphasizes how users perceive and interact with software applications. In user experience design, considerations include usability, accessibility, and overall satisfaction with the interface and functionality of the software. This holistic approach ensures that the end product not only meets functional requirements but also resonates positively with users, ultimately enhancing their engagement and facilitating a seamless interaction. In contrast, the other options concentrate on specific aspects of software development that do not directly address user experience. For instance, code quality pertains to how well the software is constructed from a technical perspective, while testing methodologies focus on verifying functionalities and identifying defects. Project management practices deal with planning, executing, and overseeing project developments and timelines, rather than the end-user experience. Thus, option B stands out as it encapsulates the core aim of user experience, prioritizing the user's perspective in software development.

3. What is necessary to evaluate the exit criteria in the test process?

- A. Calculating resource requirements**
- B. Reporting on test progress and outcomes**
- C. Establishing test execution tools**
- D. Planning test environments**

To evaluate the exit criteria in the test process, it is crucial to report on test progress and outcomes. The exit criteria essentially define the conditions that must be met before concluding the testing phase. These criteria typically include metrics related to test coverage, defect rates, and the completion of certain test cases. By effectively reporting on the progress and outcomes, testers can assess whether the established exit criteria have been satisfactorily met. This reporting provides insights into the overall quality of the software, the effectiveness of the testing process, and readiness for release. Therefore, actionable data garnered from test reports is vital in making informed decisions about whether to proceed or address further issues prior to deployment. In contrast, options related to calculating resource requirements, establishing test execution tools, and planning test environments, while important aspects of the testing process, do not directly impact the evaluation of whether exit criteria have been satisfied. They focus more on preparation and resources rather than the assessment of testing outcomes.

4. Which term refers to a statement of test objectives and ideas for testing, particularly in exploratory scenarios?

- A. Test charter**
- B. Test design technique**
- C. Test condition**
- D. Test execution**

The term that refers to a statement of test objectives and ideas for testing, especially in exploratory scenarios, is "test charter." A test charter serves as a guideline that outlines what the tester aims to explore and validate during testing. It typically includes high-level objectives, the scope of testing, and any constraints or specific areas of focus. This is particularly valuable in exploratory testing where testers need the flexibility to adapt their approach based on their discoveries and insights as they interact with the system. Test charters encourage creativity and critical thinking, allowing testers to deviate from predefined scripts and react to the application in real-time. This is essential in agile environments, where requirements can change rapidly, and testing needs to be equally agile and responsive. In contrast, a test design technique refers to systematic methods applied to derive test cases based on the requirements or specifications of the software, rather than an overarching objective as seen in a test charter. A test condition describes specific criteria under which tests can be executed but does not encapsulate the broader objectives or exploratory ideas. Test execution is the actual process of running the tests, which comes after developing a test charter or design.

5. What role does 'validation' play in the software development lifecycle?

- A. Documenting the software requirements**
- B. Ensuring the software conforms to its intended use**
- C. Designing user interfaces**
- D. Conducting testing in the development phase**

Validation in the software development lifecycle is crucial as it focuses on ensuring that the software product meets the needs and expectations of the users for whom it was designed. This process ensures that the system actually fulfills its intended purpose and provides the expected results when used in a real-world context. By performing validation, teams can verify that the software works according to user requirements and that it is fit for use. This involves checking that the software achieves the right functionality, user experience, and performance expectations. It goes beyond simply checking whether the product has been built correctly (which is typically addressed in verification) to affirming that it is suitable for its intended use in practice. This focus on intended use is particularly critical in Agile environments, where user feedback loops can impact both current and future development iterations. Validation helps to minimize misalignment between the software being built and the users' needs, thereby improving overall satisfaction and reducing the risk of costly rework later in the development process. The other options touch on various aspects of the software development lifecycle, but they do not encapsulate the specific essence of validation. Documenting requirements, designing interfaces, and conducting tests mid-development are all important but do not directly address the ultimate goal of confirming that the software does what the users truly

6. What type of testing determines whether a system satisfies the acceptance criteria?

- A. Exploratory testing**
- B. Unit testing**
- C. Acceptance testing**
- D. Integration testing**

Acceptance testing is a crucial phase in the software development lifecycle that specifically evaluates whether a system meets the defined acceptance criteria established by stakeholders. This type of testing focuses on verifying that the end product fulfills the functional requirements and quality standards necessary for release. The main goal is to ensure that the users' needs and expectations are validated against the developed system. Acceptance testing is typically performed in the final stages of development, often involving end-users or stakeholders. By successfully completing acceptance testing, teams can confirm that the software not only functions correctly but also delivers the business value intended. In contrast, exploratory testing involves simultaneous learning, test design, and execution, rather than directly determining compliance with acceptance criteria. Unit testing focuses on individual components or functions within the system to ensure they behave as expected, while integration testing checks interactions between components or systems rather than the overall acceptance of the product. Therefore, acceptance testing is definitively the type of testing that aligns with assessing whether a system fulfills its acceptance criteria.

7. What is the overall purpose of a test scope?

- A. To summarize all test activities
- B. To define the features to be tested within a component**
- C. To outline testing tools to be used
- D. To identify test failures

The overall purpose of a test scope is to precisely define the features or aspects of the system that will be tested within a given component. This focus ensures that the testing efforts are aligned with the objectives and requirements of the project, allowing testers to concentrate on critical areas of the application. By establishing a clear test scope, teams can prioritize their testing efforts, allocate resources effectively, and identify what is outside the boundaries of testing, leading to a more organized and efficient testing process. This clarity helps manage stakeholder expectations and improves communication among team members by ensuring everyone understands what is included in the testing and what is not.

8. What does the term 'test design technique' refer to in testing?

- A. A method for deriving test cases**
- B. A set of test execution outcomes
- C. A collection of test environment tools
- D. Pre-conditions needed for testing

The term 'test design technique' refers to methods used to derive test cases based on various aspects of the software being tested, such as requirements, specifications, and user needs. These techniques help testers systematically create test cases that effectively cover the necessary scenarios and functionalities of the application, ensuring a thorough evaluation of the software's behavior. By utilizing these techniques, testers can enhance the quality and effectiveness of their testing efforts, allowing them to identify defects more efficiently and contribute to the overall success of the software testing process. Techniques can include boundary value analysis, equivalence partitioning, decision table testing, and state transition testing, among others, providing a structured approach to crafting tests that align with defined criteria. This definition clearly distinguishes test design techniques from other concepts in testing. For instance, outcomes of test executions relate to the results obtained after running tests, while test environment tools are related to the setup used in the testing process. Additionally, pre-conditions are the requirements or states that need to be satisfied before executing a test. These concepts, while relevant to the testing lifecycle, do not fit the specific definition of a technique for deriving test cases.

9. What type of testing involves executing the software of a component or system?

A. Dynamic testing

B. Static testing

C. Unit testing

D. Integration testing

Dynamic testing is characterized by executing the software to validate its behavior and performance. This type of testing allows testers to observe the system's reactions in real-time, which is essential for identifying issues related to run-time behaviors, such as performance, memory leaks, and incorrect responses to user inputs. Through dynamic testing, various execution paths of the application are exercised, enabling the detection of defects that may not be evident from the code alone. Static testing, in contrast, involves examining the code, requirements, or design documents without executing the software. This type of testing can uncover certain classes of defects early in the development process, but it doesn't provide insights into how the software behaves during execution. Unit testing and integration testing are indeed forms of dynamic testing, but they are more specific in their focus. Unit testing involves testing individual components or modules in isolation, while integration testing assesses the interaction between various components or systems after they have been combined. While both are crucial parts of the overall testing strategy, the term "dynamic testing" broadly encompasses any execution of the software, making it the more appropriate choice in this context.

10. In the context of testing quadrants, what aspects does one dimension address?

A. Technology-facing versus business-facing

B. Static versus dynamic testing

C. Automated versus manual testing

D. Planned testing versus exploratory testing

The correct answer addresses the distinction between technology-facing and business-facing aspects of testing, which is a fundamental consideration in the context of the testing quadrants. This dimension helps teams understand the purpose and target of their testing efforts. When testing is categorized as technology-facing, the focus is on the technical aspects of the application, such as code quality, performance, and system functionality. This can include activities like automated unit testing, integration testing, and performance testing, which are aimed at ensuring that the underlying technology works as intended. On the other hand, business-facing testing emphasizes the end-user and customer perspective, assessing how well the application meets business requirements and user expectations. It typically involves exploratory testing, user acceptance testing, and other forms of validation that consider usability and business value. Understanding this distinction allows testing teams to align their testing efforts more effectively with project goals, ensuring that both technology quality and business functionality are adequately addressed. This alignment is critical for delivering a product that meets both technical standards and user needs in an Agile environment, where feedback and flexibility are key. The other dimensions mentioned—static versus dynamic testing, automated versus manual testing, and planned versus exploratory testing—are also important but do not specifically address the fundamental distinction between technology and business perspectives that is central

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://istqbagiletester.examzify.com>

We wish you the very best on your exam journey. You've got this!