

ISTQB Agile Tester Practice Test (Sample)

Study Guide



Everything you need from our exam experts!

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 accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

- Copyright** 1
- Table of Contents** 2
- Introduction** 3
- How to Use This Guide** 4
- Questions** 5
- Answers** 8
- Explanations** 10
- Next Steps** 16

SAMPLE

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. 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.

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. 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!

Questions

SAMPLE

- 1. Which of the following best describes the term 'compatibility' in software testing?**
 - A. The ability of software to function across various platforms**
 - B. The ease with which users can interact with the software**
 - C. The capacity of a system to handle multiple tasks simultaneously**
 - D. The duration for which software can operate before failure**

- 2. Which process involves designing tests based on the experience and knowledge of the tester?**
 - A. Performing exploratory tests**
 - B. Conducting automated tests**
 - C. Engaging in experience-based testing**
 - D. Implementing integration testing**

- 3. What term refers to the user's ability to find the software pleasant and straightforward to use?**
 - A. Understandability**
 - B. Accessibility**
 - C. Usability**
 - D. Effectiveness**

- 4. Which testing method is based on an analysis of the internal structure of the component or system?**
 - A. Black-box testing**
 - B. White-box testing**
 - C. Functional testing**
 - D. Regression testing**

- 5. Which type of evaluation assesses software products against established standards and guidelines?**
 - A. Auditing**
 - B. Compliance testing**
 - C. Certification**
 - D. Inspection**

- 6. What is the primary objective of performance testing?**
- A. To determine usability of the software**
 - B. To assess the performance of the software product**
 - C. To verify the software's security features**
 - D. To evaluate customer satisfaction**
- 7. What concept indicates the effectiveness and efficiency of an organization's processes?**
- A. Quality assurance**
 - B. Maturity**
 - C. Capability**
 - D. Compliance**
- 8. What defines a test level?**
- A. A collection of test resources**
 - B. A group of organized test activities**
 - C. Strategies for test execution**
 - D. The procedure for following up on test results**
- 9. What type of testing occurs at an external site to assess system satisfaction regarding user needs?**
- A. Alpha testing**
 - B. Integration testing**
 - C. Beta testing**
 - D. System testing**
- 10. What is the primary focus of quality assurance?**
- A. Providing confidence that quality requirements will be fulfilled**
 - B. Fixing defects in the software**
 - C. Managing project timelines**
 - D. Conducting user acceptance testing**

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. Which of the following best describes the term 'compatibility' in software testing?

A. The ability of software to function across various platforms

B. The ease with which users can interact with the software

C. The capacity of a system to handle multiple tasks simultaneously

D. The duration for which software can operate before failure

The term 'compatibility' in software testing primarily refers to the ability of software to function across various platforms, devices, operating systems, and environments. This means that the software should perform as expected regardless of where it is run, which could include different operating systems (like Windows, macOS, or Linux), multiple browser versions, or various hardware configurations. Compatibility testing ensures that users will have a consistent and reliable experience, regardless of the different settings from which they access the software. In contrast, the other options focus on different aspects of software functionality. The interaction ease with the software pertains more to usability rather than compatibility, emphasizing the user's experience and interface more than cross-platform functionality. The capacity to handle multiple tasks relates to performance testing and concurrency capabilities, reflecting how well the software manages resources under load rather than its ability to operate in different environments. Finally, the operational duration before failure pertains to reliability and maintainability, which measure the stability and durability of the software rather than its compatibility with various platforms.

2. Which process involves designing tests based on the experience and knowledge of the tester?

A. Performing exploratory tests

B. Conducting automated tests

C. Engaging in experience-based testing

D. Implementing integration testing

Experience-based testing is a process where the tester utilizes their knowledge, skills, and experiences to design tests. This approach allows testers to leverage their familiarity with the application domain, past testing experiences, and understanding of user behavior to create tests that might uncover defects that other structured testing methods may overlook. In this type of testing, the tester draws upon their mental models, intuition, and creativity to identify scenarios that are important to validate. This process recognizes that knowledge gained from previous testing efforts can be a powerful tool in shaping effective test cases and scenarios, making it an efficient approach in environments where flexibility and context-specific insights are valuable. The other processes mentioned involve different methodologies and do not focus primarily on the tester's personal experience and knowledge in the same way. For instance, exploratory testing emphasizes learning about the application during testing, while automated testing involves creating test scripts usually based on predefined requirements rather than the tester's experience. Integration testing is a specific type of testing aimed at verifying the interactions between integrated components, which does not inherently leverage the experience-based design of tests.

3. What term refers to the user's ability to find the software pleasant and straightforward to use?

- A. Understandability**
- B. Accessibility**
- C. Usability**
- D. Effectiveness**

The term that refers to the user's ability to find the software pleasant and straightforward to use is usability. Usability encompasses how easy and satisfying a system is for users to interact with and accomplish their tasks. It involves elements such as the intuitiveness of the interface, the efficiency of the interactions, and the overall satisfaction experienced by the users while using the software. When evaluating usability, factors like learnability, memorability, error frequency, and user satisfaction come into play, making it a central concept in user-centered design. This focus on the end-user experience ensures that the software is not only functional but also enjoyable to work with, directly impacting user engagement and productivity. Other options such as understandability relate more to how well a user can grasp the system's functionality, accessibility refers to how easily all users, including those with disabilities, can use the software, and effectiveness generally measures how successful a user is at completing tasks. While these are important aspects of software quality, they do not encompass the full scope of user experience defined by usability.

4. Which testing method is based on an analysis of the internal structure of the component or system?

- A. Black-box testing**
- B. White-box testing**
- C. Functional testing**
- D. Regression testing**

White-box testing is based on an analysis of the internal structure of the component or system. This method involves a detailed examination of the code, algorithms, and logic that dictate how the system operates. Testers use their knowledge of the internal workings to create test cases that can validate the correct implementation of various pathways, conditions, and loops within the application. This testing approach allows for a more thorough examination of the software by verifying the flow of inputs through the internal code structure. It can help identify logical errors or deviations from expected behavior that may not be obvious through external observation alone. White-box testing is particularly useful for unit testing and integration testing, where understanding the internal logic is crucial to assessing the performance and correctness of the code. In contrast, other testing methods, such as black-box testing, focus solely on the inputs and outputs without considering the internal structure, which is why they do not fit the definition provided in the question. Functional testing is centered on verifying that the software performs its intended functions, and regression testing is aimed at ensuring that recent changes haven't adversely affected existing features. Each of these approaches serves distinct purposes that do not involve analysis of the internal structure, which is the hallmark of white-box testing.

5. Which type of evaluation assesses software products against established standards and guidelines?

- A. Auditing**
- B. Compliance testing**
- C. Certification**
- D. Inspection**

The correct answer pertains to auditing, which is a systematic evaluation process that involves reviewing and assessing software products to determine their compliance with established standards and guidelines. This process helps ensure that the software meets specific quality criteria and adheres to regulatory standards that may be applicable in various industries. Auditing may involve checking documentation, examining the software's design and implementation, and verifying that all necessary processes are followed correctly. It plays a crucial role in risk management, quality assurance, and regulatory compliance, providing stakeholders with confidence in the software's reliability and integrity. Other types of evaluations mentioned, such as compliance testing and certification, although related, focus on different aspects. Compliance testing verifies that the software behaves in accordance with external specifications and requirements rather than internal standards. Certification generally entails a formal recognition that a product meets specific criteria and might involve a more comprehensive evaluation process, potentially including auditing as part of the overall assessment. Inspections are typically focused on assessing work products like code or documentation rather than the broader compliance with established standards or guidelines.

6. What is the primary objective of performance testing?

- A. To determine usability of the software**
- B. To assess the performance of the software product**
- C. To verify the software's security features**
- D. To evaluate customer satisfaction**

The primary objective of performance testing is to assess the performance of the software product. This type of testing focuses on evaluating how the software behaves under various conditions, such as different loads and user traffic, to ensure it meets performance criteria. This assessment can include metrics such as response time, throughput, and resource utilization, which are crucial for understanding how the application performs in a real-world scenario. While usability, security, and customer satisfaction are important aspects of software quality, they fall outside the scope of what performance testing specifically aims to address. Usability relates to how easy and intuitive the software is for users, security testing focuses on identifying vulnerabilities and ensuring data protection, and customer satisfaction is a broader measure that encompasses various factors beyond just performance. Hence, the objective of performance testing is central to ensuring that the software can handle expected workloads efficiently and effectively.

7. What concept indicates the effectiveness and efficiency of an organization's processes?

- A. Quality assurance**
- B. Maturity**
- C. Capability**
- D. Compliance**

The concept that indicates the effectiveness and efficiency of an organization's processes is maturity. Maturity models, such as the Capability Maturity Model (CMM), provide a framework for assessing the maturity levels of an organization's processes. These models help organizations understand where they stand in their process capabilities and identify areas for improvement. As an organization matures, its processes typically become more defined, standardized, and optimized, leading to increased effectiveness in achieving objectives and enhanced efficiency in resource utilization. This progression is critical to ensuring that the organization's practices evolve with the changing demands of projects and technologies. While related concepts like quality assurance, capability, and compliance pertain to certain aspects of process quality and alignment with standards or regulations, they do not specifically encapsulate the overall effectiveness and efficiency of processes as comprehensively as maturity does. Maturity focuses on process evolution and improvement over time, making it the correct choice in this context.

8. What defines a test level?

- A. A collection of test resources**
- B. A group of organized test activities**
- C. Strategies for test execution**
- D. The procedure for following up on test results**

A test level is defined by a group of organized test activities that correspond to a specific stage in the software development lifecycle. This encompasses the various testing phases, such as unit testing, integration testing, system testing, and acceptance testing. Each level focuses on different aspects of the software, such as individual components or the entire system, and involves distinct objectives, environments, and levels of detail regarding the testing performed. This organized structure helps to ensure that testing is systematic and comprehensive, allowing teams to better manage and execute their testing efforts throughout the development process. By clearly defining test levels, organizations can facilitate effective communication about testing progress, identify risks at various stages, and ultimately deliver higher quality software. In contrast to the other options, while test resources, strategies for test execution, and procedures for following up on test results are all important aspects of testing, they do not specifically define what a test level is. Test resources pertain to the tools and personnel involved, strategies focus on methodologies and techniques for execution, and procedures relate to the processes in place for managing test outcomes. Test levels, however, are distinctly about the categorization of testing efforts tied to different stages in development.

9. What type of testing occurs at an external site to assess system satisfaction regarding user needs?

- A. Alpha testing**
- B. Integration testing**
- C. Beta testing**
- D. System testing**

Beta testing is a form of testing that takes place at an external site where actual users evaluate the system to provide feedback based on their experiences. This phase is crucial for assessing how well the system meets user requirements and satisfaction. During beta testing, a select group of end users interacts with the software in a real-world environment, providing insights that are invaluable for identifying any issues, usability concerns, or areas needing improvement before the final release. This user-centered approach helps developers understand if the product aligns with user expectations and needs. The feedback gathered during beta testing helps refine the application further and ensures a greater chance of success upon full deployment. Other types of testing, such as alpha testing, integration testing, and system testing, primarily focus on finding defects within the software and ensuring components function correctly, rather than directly assessing user satisfaction in a real-world scenario. These phases often occur internally and do not typically involve real users in external environments.

10. What is the primary focus of quality assurance?

- A. Providing confidence that quality requirements will be fulfilled**
- B. Fixing defects in the software**
- C. Managing project timelines**
- D. Conducting user acceptance testing**

The primary focus of quality assurance is to provide confidence that quality requirements will be fulfilled. Quality assurance involves systematic processes that are designed to ensure that the final product meets the established standards and requirements of quality. It is proactive, seeking to prevent defects and ensure that quality control mechanisms are in place throughout the software development lifecycle. This focus on prevention and assurance differentiates it from other activities like fixing defects, which is more of a reactive approach, addressing issues after they have been identified. Quality assurance encompasses activities such as process definition and auditing, as well as training team members to ensure that everyone understands and adheres to quality practices. In an Agile context, quality assurance is critical because it supports the iterative nature of development and encourages ongoing improvement of processes, tools, and techniques across teams, thereby fostering a quality-first mindset. The emphasis on fulfilling quality requirements aligns with the objective of quality assurance to build a consistent framework that not only meets customer expectations but also enhances the reliability and efficiency of the entire development process.

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!

SAMPLE