International Software Testing Qualifications Board (ISTQB) Foundation Level Practice Test (Sample)

Study Guide

BY EXAMZIFY

Everything you need from our exam experts!

Table of Contents

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

1. **What is a common defect type associated with contradictions in acceptance criteria?**
    - **A. Performance issues**
    - **B. Missing functionality**
    - **C. Contradiction**
    - **D. Duplicated requirements**

2. **What is a valid objective for testing?**
    - **A. The test should start as late as possible to allow good product development.**
    - **B. To find as many failures as possible so that defects can be corrected.**
    - **C. To prove that all possible defects are identified.**
    - **D. To demonstrate that any remaining defects will not cause any failures.**

3. **In which situation is exploratory testing most suitable?**
    - **A. When there is advanced documentation for testing.**
    - **B. When there is significant time pressure on testing.**
    - **C. When all requirements are clearly specified.**
    - **D. When testers lack knowledge of similar applications.**

4. **Which of the following best describes test design techniques?**
    - **A. Procedures to document test results**
    - **B. Strategies for developing test cases based on specifications**
    - **C. Methods to automate testing processes**
    - **D. Standards for reporting defects**

5. **What process is missing if developers release code for testing that is not version controlled?**
    - **A. Configuration management**
    - **B. Debugging**
    - **C. Test and defect management**
    - **D. Risk analysis**

6. What potential outcome might result from poor communication among project stakeholders?

A. Enhanced collaboration and teamwork

B. Increased uncertainty regarding project objectives

C. Higher levels of test coverage

D. Improved stakeholder satisfaction with products

7. In what scenario would the priority of a defect be rated low while the severity is rated high?

A. When the defect is common and affects all users

B. When the defect occurs rarely and has limited impact

C. When the defect causes data loss

D. When the defect leads to a system crash

8. Which component is NOT typically a function of a test execution tool?

A. Writing test cases

B. Running automated tests

C. Recording test results

D. Logging defects

9. Which categorization best describes the technique of evaluating a mobile app against UI best practices?

A. Specification-based

B. Exploratory

C. Checklist-based

D. Error guessing

10. What is the primary focus of acceptance testing?

A. To verify that all requirements have been met

B. To ensure performance under load

C. To check for security vulnerabilities

D. To validate integration with other systems

Answers

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

Explanations

1. What is a common defect type associated with contradictions in acceptance criteria?

A. Performance issues

B. Missing functionality

<u>C. Contradiction</u>

D. Duplicated requirements

The common defect type associated with contradictions in acceptance criteria is indeed contradictions. When acceptance criteria are contradictory, it creates confusion about what the software is expected to do. Testers rely on clear, consistent acceptance criteria to guide their testing efforts, ensuring that the software meets the specified requirements. When these criteria conflict with one another, it can lead to misunderstandings and ultimately result in defects because the development team may implement features or functionalities that do not align with what was agreed upon. Identifying contradictions is crucial in the testing process because they can lead to a variety of issues, such as incomplete or incorrect implementations. Therefore, ensuring that acceptance criteria are free from contradictions is a fundamental aspect of clear communication between stakeholders, benefiting the entire development lifecycle.

2. What is a valid objective for testing?

A. The test should start as late as possible to allow good product development.

<u>B. To find as many failures as possible so that defects can be corrected.</u>

C. To prove that all possible defects are identified.

D. To demonstrate that any remaining defects will not cause any failures.

The objective of testing is fundamentally about ensuring the quality and performance of a software product before it is released. Among the options presented, the goal of finding as many failures as possible so that defects can be corrected directly aligns with the primary purpose of testing. This approach focuses on identifying issues that might affect the functionality, usability, and overall user experience of the software, allowing the development team to address these flaws prior to deployment. By finding failures, testers provide critical feedback to developers, facilitating the refinement of the product and enhancing its reliability. The more issues discovered during testing, the more chances there are to fix these problems before the software reaches the end users, thereby improving customer satisfaction and reducing potential post-release issues. In contrast, starting testing late in the development cycle limits the time available for fixing defects, while proving that all possible defects are identified or demonstrating that remaining defects will not cause failures sets unrealistic expectations. Testing is inherently about exploration and reduction of risk, not absolute assurance. Therefore, option B encapsulates the valid objective of testing by emphasizing the importance of defect identification and rectification.

3. In which situation is exploratory testing most suitable?

A. When there is advanced documentation for testing.

B. When there is significant time pressure on testing.

C. When all requirements are clearly specified.

D. When testers lack knowledge of similar applications.

Exploratory testing is particularly suitable in scenarios where there is significant time pressure on testing. In cases where time is limited, exploratory testing allows testers to use their skills, experience, and intuition to investigate the application without being constrained by extensive documentation or predefined test scripts. This approach enables quick adaptation and focus on areas that might be more susceptible to defects, facilitating rapid feedback and higher coverage in a shorter time frame. While advanced documentation and clearly specified requirements are beneficial for more structured testing approaches like scripted testing, exploratory testing thrives in situations where flexibility and creativity can help address time constraints. Likewise, when testers lack knowledge of similar applications, exploratory testing can still be effective as it encourages exploration and learning about the application on the spot. However, time pressure is a primary driver for choosing exploratory testing, as it promotes an efficient and effective way to uncover issues swiftly.

4. Which of the following best describes test design techniques?

A. Procedures to document test results

B. Strategies for developing test cases based on specifications

C. Methods to automate testing processes

D. Standards for reporting defects

Test design techniques are fundamentally methods or strategies that help testers create effective test cases based on the requirements and specifications of the software being tested. These techniques include approaches such as equivalence partitioning, boundary value analysis, decision table testing, and state transition testing, among others. Each of these methods provides a structured way to derive tests by focusing on specific aspects of the software's behavior, ensuring thorough coverage and enhancing the likelihood of discovering defects. In contrast, other options focus on different aspects of the software testing process. Documenting test results relates to capturing outcomes and findings after tests have been executed rather than how to create the test cases. Automating testing processes pertains to the use of tools and scripts to run tests without manual intervention, which is a separate concern from how tests are designed. Lastly, standards for reporting defects emphasize the guidelines for communicating issues found during testing rather than the techniques used to construct the tests themselves. Thus, the choice that best aligns with the definition of test design techniques is the strategy of developing test cases based on specifications.

5. What process is missing if developers release code for testing that is not version controlled?

A. Configuration management

B. Debugging

C. Test and defect management

D. Risk analysis

The process that is missing when developers release code for testing that is not version controlled is configuration management. Configuration management involves systematic handling of changes to a system, including the management of files, tools, and data throughout the software development lifecycle. It ensures that the correct versions of software and related documents are available, which is crucial for a successful testing phase. Without version control, there is a high risk of confusion and errors, as testers may work with outdated or incorrect code versions, leading to inaccurate testing results. In the context of development and testing, effective configuration management enables better collaboration, traceability of changes, and consistency across different environments. It allows teams to understand what changes have been made, by whom, and why, thereby aligning development efforts with overall project objectives and requirements. This foundational aspect is essential for maintaining the integrity and quality of the software being tested.

6. What potential outcome might result from poor communication among project stakeholders?

A. Enhanced collaboration and teamwork

B. Increased uncertainty regarding project objectives

C. Higher levels of test coverage

D. Improved stakeholder satisfaction with products

Poor communication among project stakeholders can lead to increased uncertainty regarding project objectives. When different parties involved in a project do not effectively share information, misunderstandings can arise about the goals, requirements, and expectations. This uncertainty can hinder decision-making and cause inconsistencies in how teams understand their roles and deliverables. As a result, team members may become misaligned in their work, leading to wasted efforts, missed deadlines, and a lack of clarity on what the project is truly aiming to achieve. This can create additional challenges such as scope creep, delays, and ultimately, a product that does not meet the stakeholders' needs or expectations. Meanwhile, enhanced collaboration and teamwork, higher levels of test coverage, or improved stakeholder satisfaction with products are unlikely outcomes in scenarios where communication is poor. Effective communication is fundamental for achieving positive results in these areas, making the link between communication and the clarity of project objectives critical to successful project outcomes.

7. In what scenario would the priority of a defect be rated low while the severity is rated high?

A. When the defect is common and affects all users

B. When the defect occurs rarely and has limited impact

C. When the defect causes data loss

D. When the defect leads to a system crash

The reasoning behind rating the priority of a defect as low while the severity is high involves understanding the nuances of both terms within the context of software testing. Severity refers to the impact of the defect on the application's functionality, while priority indicates how urgently the defect needs to be addressed. In this scenario, where the defect occurs rarely and has limited impact, it highlights a situation where the defect can be significant in terms of functionality (high severity) but does not necessitate immediate attention or a quick fix (low priority). For instance, a software application might have a serious flaw that, when it appears, could prevent a specific feature from functioning correctly. However, if this issue is infrequent and does not affect the majority of users or critical operations, it would be deemed low priority for resolution given that most users are not affected by it. The other scenarios depict defects that invoke both high severity and high priority because they either affect the majority of users or lead to critical issues like data loss or system crashes, necessitating immediate attention.

8. Which component is NOT typically a function of a test execution tool?

A. Writing test cases

B. Running automated tests

C. Recording test results

D. Logging defects

In the context of test execution tools, writing test cases is generally not a primary function of these tools. Test execution tools are designed primarily for executing tests that have already been created and for managing the results of those executions. The primary functions of test execution tools include running automated tests, which allows for efficient and consistent testing of software; recording test results, which captures the outcomes of executed tests for analysis and reporting; and logging defects, which refers to the process of documenting issues found during testing for further investigation and resolution. Writing test cases, on the other hand, is typically a function associated with test management tools or test design tools. These tools focus on the initial creation and organization of test cases based on requirements, specifications, and various testing strategies. Thus, while test execution tools may interface with test cases, their core capabilities center around executing those tests and managing the results rather than creating them from scratch.

9. Which categorization best describes the technique of evaluating a mobile app against UI best practices?

A. Specification-based

B. Exploratory

C. Checklist-based

D. Error guessing

The technique of evaluating a mobile app against UI best practices is best described as checklist-based because it involves systematically checking each aspect of the user interface against a predefined list of best practices or standards. This method is structured, allowing for comprehensive coverage of the UI elements and ensuring that all important guidelines concerning usability and design are considered. Checklist-based testing helps testers ensure consistency and thoroughness in their evaluations. By using a checklist, testers can focus on key areas of user interaction, such as navigation flows, visual elements, accessibility concerns, and responsiveness, which are all critical for a mobile application's success. This approach contrasts with exploratory testing, which is more ad-hoc and relies on testers' creativity and intuition rather than a defined set of criteria. While specification-based testing might involve checking against requirements, it does not specifically focus on UI best practices. Error guessing is a technique based on the tester's experience and intuition of where defects might exist, rather than a structured evaluation of best practices. Thus, the checklist-based approach aligns most closely with the requirements of evaluating a mobile app against established UI criteria.

10. What is the primary focus of acceptance testing?

A. To verify that all requirements have been met

B. To ensure performance under load

C. To check for security vulnerabilities

D. To validate integration with other systems

Acceptance testing primarily focuses on verifying that the software meets the agreed-upon requirements and is ready for use by the end user or customer. This type of testing is often performed in the final phase of development and is critical in determining whether the software is fit for release. It emphasizes ensuring that the functionality aligns with the business needs and requirements specified at the project's outset. While the verification of requirements is significant, the other areas, such as performance under load, security vulnerabilities, and integration with other systems, may be evaluated during different testing phases like performance testing, security testing, or integration testing. Acceptance testing is specifically centered on end-user needs and satisfaction, ensuring that the final product complies with the specified criteria and can be confidently delivered to users.

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://istqbfoundationlevel.examzify.com

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