

ASTQB Foundation Level Practice Exam (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	15

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. In defect tracking, what does 'severity' indicate?**
 - A. The speed of defect resolution
 - B. The impact of a defect on application functionality
 - C. The number of defects identified
 - D. The category of the testing phase
- 2. What role do stakeholders play in the testing process?**
 - A. They execute all test cases
 - B. They provide requirements and feedback
 - C. They are only involved at the end of development
 - D. They create detailed test scripts
- 3. What does stress testing determine about an application?**
 - A. User-friendliness
 - B. Stability and reliability under extreme load
 - C. Error rates in normal operations
 - D. Code quality and readability
- 4. Which of the following best describes stakeholder communication during testing?**
 - A. Limited to project updates
 - B. Regular feedback and requirements alignment
 - C. Only at the project's start and end
 - D. Focused solely on budget constraints
- 5. Which of the following is NOT a type of software review?**
 - A. Informal reviews
 - B. Walkthroughs
 - C. Quality Assurance Reviews
 - D. Inspections
- 6. What is the significance of impact analysis severity?**
 - A. It refers to the efficiency of the testing process
 - B. It evaluates the potential negative consequences of defects
 - C. It determines how incidents should be logged
 - D. It is related to test case execution times

- 7. Which aspect of software is determined through software quality assessment?**
- A. The technical specifications of the software
 - B. The degree to which it meets specified requirements
 - C. The number of developers involved
 - D. The overall cost of development
- 8. Dynamic testing is characterized by what process?**
- A. Reviewing code and static analysis
 - B. Executing code and checking actual outputs
 - C. Reviewing documentation only
 - D. Creating test plans without code execution
- 9. In boundary value analysis (3 Point Method), what are the conditions tested for the lower boundary?**
- A. Just below valid, on the boundary, and just above
 - B. Just below invalid, on the boundary, and above
 - C. Two invalid and one valid
 - D. All possible values including invalid
- 10. What is software testing primarily concerned with?**
- A. Planning, execution, and evaluation of software products
 - B. Managing team dynamics
 - C. Building new software features
 - D. Cost reduction in software development

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. In defect tracking, what does 'severity' indicate?

- A. The speed of defect resolution
- B. The impact of a defect on application functionality**
- C. The number of defects identified
- D. The category of the testing phase

In defect tracking, 'severity' specifically refers to the impact a defect has on the application's functionality. This assessment focuses on how significantly the defect affects the user experience, operational capabilities, or overall performance of the application. For instance, a defect that causes a critical function of the application to fail would be categorized with high severity because it can prevent users from completing essential tasks. On the other hand, a cosmetic issue that does not hinder functionality might be considered low severity, even though it still needs to be addressed. Understanding severity is crucial for prioritizing defects and determining which issues should be addressed first, ensuring that the most critical problems are resolved promptly to maintain application quality and user satisfaction. This helps teams allocate resources effectively for testing and defect resolution.

2. What role do stakeholders play in the testing process?

- A. They execute all test cases
- B. They provide requirements and feedback**
- C. They are only involved at the end of development
- D. They create detailed test scripts

Stakeholders play a crucial role in the testing process by providing requirements and feedback. Their input is vital in the early stages of a project as they help define what the system or product should accomplish, which guides the testing team in developing relevant test cases. Throughout the testing process, stakeholders can also offer valuable feedback on the system's performance or usability, helping ensure that the final product meets user expectations and business needs. By actively engaging with stakeholders, the testing process becomes more aligned with the goals of the project, ultimately leading to better quality outcomes. This involvement is not limited to just summary input; it encompasses ongoing communication that helps address issues early, ensuring that necessary adjustments can be made before the product reaches the final stages of development.

3. What does stress testing determine about an application?

- A. User-friendliness
- B. Stability and reliability under extreme load**
- C. Error rates in normal operations
- D. Code quality and readability

Stress testing is a type of non-functional testing that evaluates how an application behaves under heavy load conditions. The primary goal is to determine the application's stability and reliability when subjected to extreme or excessive demands, which can simulate scenarios such as high user traffic, increased data processing, or maximum system capacity. Through this testing, teams can uncover how the system performs under stress, identifying limitations, breaking points, and potential failure modes that could occur under such conditions. This information is crucial for ensuring that the application can handle anticipated peak loads, maintain performance levels, and provide a reliable user experience when traffic or processing demands surge. The other choices focus on aspects that do not align with the core purpose of stress testing: user-friendliness pertains more to usability testing; error rates during normal operations relate to functional testing; and code quality and readability are critical in code reviews and static analysis, not stress testing. Thus, the primary objective of stress testing firmly ties it to assessing stability and reliability under extreme load conditions.

4. Which of the following best describes stakeholder communication during testing?

- A. Limited to project updates
- B. Regular feedback and requirements alignment**
- C. Only at the project's start and end
- D. Focused solely on budget constraints

Stakeholder communication during testing is best described by the concept of regular feedback and requirements alignment. This entails not just providing updates on the project's progress but also actively engaging stakeholders throughout the testing process to ensure their needs and expectations are understood and integrated into the quality assurance efforts. Regular communication allows for ongoing feedback that helps in identifying any misunderstandings or shifting requirements early on, which can significantly enhance the quality of the final product. By aligning testing activities with stakeholders' requirements, teams can ensure that the testing process is effectively contributing to the overall success of the project and ultimately delivering what stakeholders expect in terms of functionality, performance, and user satisfaction. This continuous engagement fosters a collaborative environment, where stakeholders feel involved and informed, which is vital for making necessary adjustments and decisions that may arise as the testing progresses.

5. Which of the following is NOT a type of software review?

- A. Informal reviews
- B. Walkthroughs
- C. Quality Assurance Reviews**
- D. Inspections

Quality Assurance Reviews are typically broader in scope and encompass various activities that ensure compliance with quality standards but are not classified as software reviews in the same manner as the other types listed. Informal reviews, walkthroughs, and inspections are specific techniques used to examine software artifacts, such as code or design documents, with the goal of identifying defects or improvements. Informal reviews involve casual discussions around the software work products, while walkthroughs represent a guided tour through the software to gather feedback from participants. Inspections are formalized reviews that follow a structured process, including defined roles and criteria for evaluating the material. In contrast, Quality Assurance Reviews focus on the overall quality processes and compliance with established standards rather than being centered specifically on examining software artifacts for defects. This distinction helps clarify why Quality Assurance Reviews do not fit within the traditional categories of software reviews.

6. What is the significance of impact analysis severity?

- A. It refers to the efficiency of the testing process
- B. It evaluates the potential negative consequences of defects**
- C. It determines how incidents should be logged
- D. It is related to test case execution times

The significance of impact analysis severity lies in its ability to evaluate the potential negative consequences of defects in a software product. By assessing the severity, teams can prioritize which defects need immediate attention based on their impact on the functionality, stability, or user experience of the software. This helps in making informed decisions about resource allocation during testing and development. Understanding the severity of defects ensures that high-impact issues are addressed promptly, thereby enhancing the overall quality of the software and reducing the risk of failures in critical areas. This approach enables development and testing teams to focus their efforts where they are most needed, ensuring that the software meets quality standards and user expectations effectively.

7. Which aspect of software is determined through software quality assessment?

- A. The technical specifications of the software
- B. The degree to which it meets specified requirements**
- C. The number of developers involved
- D. The overall cost of development

The degree to which software meets specified requirements is a fundamental aspect of software quality assessment. This process involves evaluating whether the software fulfills the intended needs and expectations that were laid out at the beginning of the project. Essentially, it checks the alignment between the actual delivered software and the defined requirements, including functionality, performance, reliability, and other criteria. Quality assessments typically involve reviewing documentation, conducting tests, and gathering user feedback to ensure that the software not only performs as expected but also meets the user's requirements and industry standards. Achieving high quality means that the software is reliable, useful, and of value to the end-user, which directly correlates with meeting specified requirements. This focus on conformance to requirements is what drives successful software development and satisfaction among stakeholders.

8. Dynamic testing is characterized by what process?

- A. Reviewing code and static analysis
- B. Executing code and checking actual outputs**
- C. Reviewing documentation only
- D. Creating test plans without code execution

Dynamic testing is characterized by the execution of code and checking actual outputs against expected outcomes. This process involves running the software application to identify any discrepancies between what the program should do and its current functionality when actively in operation. Through this form of testing, testers can observe software behavior in real-time, which helps in uncovering issues that may not be evident through non-executive approaches. Unlike static testing, which is focused on inspecting the code, documentation, or other project artifacts without executing the program (as seen in the other options), dynamic testing reveals actual runtime behavior. This gives testers the ability to evaluate aspects like performance, usability, and response to various inputs, offering a comprehensive understanding of the system's effectiveness. By executing the code, testers can also facilitate error detection and validation of requirements, which are critical elements in ensuring software quality.

9. In boundary value analysis (3 Point Method), what are the conditions tested for the lower boundary?

- A. Just below valid, on the boundary, and just above
- B. Just below invalid, on the boundary, and above**
- C. Two invalid and one valid
- D. All possible values including invalid

In boundary value analysis, particularly the 3 Point Method, the focus is on identifying the critical points at the edges of input ranges, as errors are often found at these boundaries. When considering the lower boundary of a valid input range, the conditions tested involve assessing values that are immediately below the valid lower limit, the exact value of the lower boundary itself, and a value that is just above the lower boundary. Testing just below the valid lower boundary helps identify how the system behaves when faced with inputs that fall outside the acceptable range. By also including the boundary itself, the method ensures that this critical value is processed correctly. Finally, testing just above the lower boundary checks if the system successfully accepts the first valid input above that limit. This approach effectively helps in validating the correct range of inputs and ensuring robust handling of edge cases. Other options do not accurately capture the principles of boundary value analysis, particularly in defining the conditions for the lower boundary. Only one valid input scenario, alongside the invalid lower and the immediate valid above it, accurately reflects the intent of boundary value testing.

10. What is software testing primarily concerned with?

- A. Planning, execution, and evaluation of software products**
- B. Managing team dynamics
- C. Building new software features
- D. Cost reduction in software development

Software testing is primarily focused on the planning, execution, and evaluation of software products to ensure that they meet the required standards of quality and functionality. This process involves creating test plans, designing test cases, executing those tests, and analyzing the results to identify defects or issues in the software. The goal is to verify that the software works as intended and to ensure that any potential problems are identified before the software is released to users. This comprehensive process helps to enhance the reliability of software, improve user satisfaction, and ultimately contribute to the overall success of the software development project. The other choices, while related to the broader context of software development, do not capture the essence of what software testing is primarily about. Managing team dynamics, building new software features, and cost reduction are important aspects of software development as a whole, but they do not specifically define the focus and responsibilities within the software testing domain.

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://astqbfoundationlevel.examzify.com>

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

SAMPLE