

Professional Scrum Master (PSM) III Practice Test (Sample)

Study Guide



Everything you need from our exam experts!

Copyright © 2025 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

Questions

SAMPLE

- 1. What does Code Coverage measure in software testing?**
 - A. The number of lines of code written**
 - B. Code being exercised by tests**
 - C. The number of bugs reported**
 - D. The performance of the application**
- 2. Which of the following is a benefit of self-organization within a Development Team?**
 - A. Increased supervision**
 - B. Reduced reliance on processes**
 - C. Increased self-accountability**
 - D. Lower costs**
- 3. What aspect of the Sprint Backlog allows it to be adaptable?**
 - A. The ability to change tasks based on feedback**
 - B. The requirement for detailed specifications**
 - C. The fixed set of tasks defined upfront**
 - D. The reliance on historical data**
- 4. Which of the following consequences may arise if the definition of done is inconsistent?**
 - A. The Development Team will never meet sprint goals**
 - B. The Product Owner may misjudge progress toward goals**
 - C. The team will always know their Sprint capacity**
 - D. The backlog items will be completed quicker**
- 5. What is the key focus of a Scrum Master in supporting the Development Team?**
 - A. To manage team members directly**
 - B. To ensure the team meets deadlines regardless of quality**
 - C. To facilitate teamwork and improve processes**
 - D. To define the team's goals without input**

- 6. Which activities are involved in refining Product Backlog items?**
- A. Implementation and testing**
 - B. Analysis, design, and decomposition**
 - C. Approval and validation**
 - D. Documentation and reporting**
- 7. How often does the Product Owner interact with the stakeholders?**
- A. Rarely**
 - B. Only during Sprint Reviews**
 - C. On a regular basis throughout the Sprint**
 - D. At the end of the project only**
- 8. How are architecture and infrastructure treated in Scrum?**
- A. By ignoring them until the product is fully developed**
 - B. They are solely handled in the last Sprint**
 - C. They are added to the Product Backlog and addressed in early Sprints**
 - D. Only implemented during functional development phases**
- 9. How is ownership of a Sprint Backlog Item viewed within the Development Team?**
- A. Each item is owned by the individual who works on it**
 - B. All items are owned collectively by the entire Development Team**
 - C. Only the Scrum Master owns the Sprint Backlog items**
 - D. No one owns the items**
- 10. Which type of test is NOT typically automated?**
- A. Unit**
 - B. Exploratory**
 - C. Performance**
 - D. Functional**

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. What does Code Coverage measure in software testing?

- A. The number of lines of code written
- B. Code being exercised by tests**
- C. The number of bugs reported
- D. The performance of the application

Code coverage is a software testing metric that assesses how much of the codebase is tested through automated tests. Specifically, it measures the proportion of the code that is executed during testing, indicating which parts of the codebase are exercised by the tests. A higher code coverage percentage implies that more of the code is being validated by the tests, which can lead to greater confidence in the software's quality. This metric often helps to identify untested parts of the application, guiding developers and testers to areas that may need more attention to ensure comprehensive test coverage. It is particularly valuable in unit testing, integration testing, and other automated testing practices, allowing teams to understand the effectiveness of their testing efforts.

2. Which of the following is a benefit of self-organization within a Development Team?

- A. Increased supervision
- B. Reduced reliance on processes
- C. Increased self-accountability**
- D. Lower costs

Self-organization within a Development Team fosters increased self-accountability among team members. When a team is self-organizing, individuals take greater ownership of their tasks and responsibilities. This sense of accountability enhances commitment to the team's goals and motivates members to work collaboratively towards delivering high-quality results. In a self-organized team, members are empowered to make decisions about how to approach their work, which encourages initiative and innovation. This empowerment leads to individuals actively monitoring their performance and the team's overall progress, holding themselves and their peers accountable for the outcomes. As a result, the team can adapt and respond to challenges more effectively, ensuring that they remain focused on delivering value to the customer. The other options do not align with the core principles of self-organization. Increased supervision contradicts the idea of a self-managing team, while reducing reliance on processes may not necessarily lead to effective outcomes without a proper balance. Lower costs may or may not result from self-organization and depends on various other factors. Hence, the primary benefit of self-organization distinctly highlights the enhancement of self-accountability within the Development Team.

3. What aspect of the Sprint Backlog allows it to be adaptable?

- A. The ability to change tasks based on feedback**
- B. The requirement for detailed specifications**
- C. The fixed set of tasks defined upfront**
- D. The reliance on historical data**

The adaptability of the Sprint Backlog primarily stems from the ability to change tasks based on feedback. This characteristic is crucial in Scrum because it aligns with the framework's emphasis on iterative development and responding to changes in requirements. During the Sprint, the team may receive feedback from stakeholders or discover new insights while working on the product. This ongoing feedback loop allows the team to adjust the tasks within the Sprint Backlog, ensuring that it remains relevant and aligned with the project goals. In Scrum, the Sprint Backlog is intentionally designed to be a living artifact. It evolves throughout the Sprint, reflecting both the team's progress and any new information that may necessitate adjustments. This flexibility enables the team to prioritize the most valuable work and facilitate better outcomes for the product being developed. In contrast, the other options do not support adaptability. Detailed specifications can lead to a rigid approach, fixed sets of tasks limit flexibility, and reliance on historical data may not account for current realities. Hence, the ability to adjust tasks based on real-time feedback is what empowers teams to remain nimble and responsive to change within their development process.

4. Which of the following consequences may arise if the definition of done is inconsistent?

- A. The Development Team will never meet sprint goals**
- B. The Product Owner may misjudge progress toward goals**
- C. The team will always know their Sprint capacity**
- D. The backlog items will be completed quicker**

The correct answer highlights a significant implication of having an inconsistent definition of done. When the definition of done varies, it can lead to confusion about what constitutes a completed item. Consequently, the Product Owner may misjudge the team's progress toward the goals because they may not have a clear and unified understanding of what "done" actually means. If some items are considered complete while others are not due to differing interpretations of the definition, it creates ambiguity in measuring the team's velocity and overall progress. This inconsistency can significantly impair the Product Owner's ability to make informed decisions based on the team's output, as they may base their assessments on incomplete or misrepresented work. This can lead to ineffective planning and decision-making regarding future sprints, ultimately impacting the product's trajectory and stakeholder satisfaction. Other choices involve misconceptions about team dynamics or operational capacity that don't directly address the misalignment caused by an inconsistent definition of done. Thus, option B stands out as the most relevant consequence in the context of an inconsistent definition of done.

5. What is the key focus of a Scrum Master in supporting the Development Team?

- A. To manage team members directly**
- B. To ensure the team meets deadlines regardless of quality**
- C. To facilitate teamwork and improve processes**
- D. To define the team's goals without input**

The key focus of a Scrum Master in supporting the Development Team is to facilitate teamwork and improve processes. This role is centered on ensuring that the team is able to work collaboratively and effectively, fostering an environment where team members can communicate openly and are encouraged to participate in problem-solving. By facilitating teamwork, the Scrum Master helps remove obstacles that may hinder the team's progress, thus enabling them to focus on delivering high-quality work. Additionally, the Scrum Master plays a crucial role in guiding the team in their use of Agile practices and Scrum framework, which involves promoting continuous process improvement. This can include organizing and leading retrospectives, coaching the team on best practices, and encouraging a culture of inspection and adaptation. In this context, the responsibilities of the Scrum Master extend to ensuring that the Development Team is self-organizing and empowered to make decisions regarding how to best achieve their objectives. This collaborative approach not only enhances the team's performance but also contributes to their growth and maturity as a unit within the Scrum framework.

6. Which activities are involved in refining Product Backlog items?

- A. Implementation and testing**
- B. Analysis, design, and decomposition**
- C. Approval and validation**
- D. Documentation and reporting**

Refining Product Backlog items is a critical activity in Scrum that enables the team to clarify and enhance the backlog items before they are worked on in a Sprint. The selected answer highlights the processes of analysis, design, and decomposition, which are essential for ensuring that backlog items are well-understood, properly sized, and actionable. When refining backlog items, analysis involves examining the requirements and understanding the business value or user needs associated with them. This step is crucial as it helps the team grasp what is truly needed and aligns the work with stakeholder expectations. Design refers to the initial considerations for how features or enhancements will work. This might include discussing potential user interfaces, architecture, or technologies that may be used to implement the backlog items. By engaging in design discussions during refinement, the team can anticipate potential challenges and focus on solutions early in the process. Decomposition is the process of breaking down larger or complex backlog items into smaller, more manageable parts that can be completed within a single Sprint. This activity helps in estimating effort accurately and enables more efficient planning. Overall, these activities ensure that the Product Backlog remains a dynamic and well-defined tool that optimally supports the team's ability to deliver value by aligning the work with organizational goals and user needs.

7. How often does the Product Owner interact with the stakeholders?

- A. Rarely**
- B. Only during Sprint Reviews**
- C. On a regular basis throughout the Sprint**
- D. At the end of the project only**

The Product Owner's role is to maximize the value of the product by managing the Product Backlog effectively and ensuring that the right features are prioritized. Regular interaction with stakeholders is essential for gathering feedback, understanding their needs, and aligning the product development with market demands and expectations. By engaging with stakeholders on a regular basis throughout the Sprint, the Product Owner can gather insights that inform the prioritization of the backlog, ensuring that the team is building the most valuable features first. This ongoing communication helps to clarify requirements and allows for adjustments based on stakeholder feedback, thereby enhancing the final product's quality and market fit. In contrast to the other options, regular interaction fosters a collaborative environment where stakeholders feel involved and invested in the product's development, which ultimately leads to better outcomes and stronger stakeholder relationships.

8. How are architecture and infrastructure treated in Scrum?

- A. By ignoring them until the product is fully developed**
- B. They are solely handled in the last Sprint**
- C. They are added to the Product Backlog and addressed in early Sprints**
- D. Only implemented during functional development phases**

In Scrum, architecture and infrastructure are considered essential components of the product being developed. Treating them by adding them to the Product Backlog and addressing them in early Sprints aligns with the principles of iterative and incremental development inherent to Scrum. This approach allows the team to identify and build foundational elements of the product that support future development efforts. It ensures that architectural decisions are made in a timely manner, reducing the risk of encountering significant issues later in the development process that could hinder progress or lead to technical debt. Addressing architecture and infrastructure early in the development cycle allows for continuous refinement and adaptation of both as requirements evolve. This is crucial in Scrum, where the focus is on delivering value incrementally and fostering collaboration among team members. By prioritizing these elements in the Product Backlog, the Scrum Team can ensure that they are integrated with feature development, enhancing the overall quality and performance of the product.

9. How is ownership of a Sprint Backlog Item viewed within the Development Team?

- A. Each item is owned by the individual who works on it
- B. All items are owned collectively by the entire Development Team**
- C. Only the Scrum Master owns the Sprint Backlog items
- D. No one owns the items

The correct perspective is that all items in the Sprint Backlog are owned collectively by the entire Development Team. In Scrum, the Sprint Backlog is a commitment by the Development Team to deliver a specific set of outcomes during the Sprint. This collective ownership fosters collaboration, accountability, and shared responsibility among team members. When the Development Team operates as a unit, it encourages members to support each other, share knowledge, and work together towards a common goal. This approach not only enhances commitment to the Sprint goals but also allows team members to step in and assist each other as needed, which can lead to faster problem resolution and more effective progress toward delivering the increment. The idea of individual ownership, where each item is owned by solely the individual working on it, contrasts with this collaborative spirit by potentially leading to siloed thinking and a lack of team cohesion. Additionally, the notion that only the Scrum Master or no one owns the items does not align with the core principles of Scrum, which emphasize collaboration and collective responsibility within the Development Team.

10. Which type of test is NOT typically automated?

- A. Unit
- B. Exploratory**
- C. Performance
- D. Functional

Exploratory testing is primarily human-driven and involves manual exploration of the application to discover issues, usability concerns, and unforeseen behavior. Testers engage in this process using their creativity and intuition, allowing them to adapt their testing strategy based on immediate feedback and observations. Unlike automated testing, where tests are scripted and run without human intervention, exploratory testing relies significantly on a tester's experience, skill, and adaptability to identify potential problems in the software. In contrast, unit, performance, and functional tests are often automated due to their repetitive nature and the need for consistent execution across multiple iterations of the software. Automated tests are more efficient for these types because they can be run frequently as part of continuous integration and continuous deployment (CI/CD) processes and can provide quick feedback on specific aspects of the software's functionality and performance.