PSD-I Training Course

Professional Scrum Developer (PSD)

Structured Learning & Certification Preparation

Table of Contents

Introduction

The Professional Scrum Developer I (PSD-I) certification reflects a developer's ability to apply modern software development practices within the Scrum framework. It represents an understanding of how professional engineering practices support iterative and incremental product delivery. In modern software environments where agility, collaboration, and rapid feedback are essential, this certification validates that developers can contribute effectively to building, integrating, and delivering high-quality product increments within Scrum Teams.

About This Training / Certification

The Professional Scrum Developer certification evaluates a developer's understanding of how technical practices, teamwork, and Agile principles combine to support professional product development. It focuses on the developer's role in creating high-quality increments, maintaining sustainable development practices, and collaborating effectively within Scrum Teams.

The certification is generally positioned at a foundational to intermediate level for software developers who work in or are transitioning to Scrum-based environments. It emphasizes not only knowledge of the Scrum framework but also the engineering discipline required to continuously deliver valuable product increments. Within a broader professional learning journey, this certification supports developers seeking to strengthen their Agile development mindset and deepen their understanding of collaborative product development practices.

What We Offer (AAAdemy)

AAAdemy provides structured training resources designed to support certification preparation and skill development across a wide range of IT domains. Our learning materials are built around clear knowledge structures, practical study guidance, and exam-oriented practice to help learners progress with confidence.

We offer well-organized knowledge explanations that break down complex topics into clear, understandable sections aligned with official exam objectives and real-world skill requirements. Each topic is designed to support both conceptual understanding and practical application.

Our study plans and learning guidance help learners follow a logical progression, focusing on key concepts, common pitfalls, and effective preparation strategies. This approach enables learners to study efficiently while maintaining a clear view of their learning goals.

To reinforce understanding, AAAdemy also provides practice questions and exam-focused insights that reflect typical certification scenarios. These resources are intended to help learners evaluate their readiness and strengthen their confidence before taking an exam.

All content is designed for flexible, self-paced learning, allowing individuals to study independently or alongside their existing professional or academic commitments.

Knowledge Overview

Domain: Developing & Delivering Products Professionally
This area focuses on the professional responsibilities of developers when creating product increments. Candidates are expected to understand how disciplined engineering practices support sustainable development and reliable delivery. This includes concepts related to maintaining code quality, managing technical complexity, supporting maintainability, and ensuring that each increment meets agreed standards of completeness and quality.

Domain: Understanding and Applying the Scrum Framework
Candidates should understand the structure, purpose, and principles of the Scrum framework and how developers participate in Scrum activities. This includes understanding Scrum roles, events, and artifacts, as well as the collaborative behaviors required to support transparency, inspection, and adaptation. Developers should also understand how their daily work contributes to the creation of valuable product increments within iterative development cycles.

Domain: Developing People and Teams
This domain focuses on collaboration, communication, and teamwork within Scrum environments. Developers are expected to understand how effective teamwork contributes to product delivery, how cross-functional collaboration improves outcomes, and how shared ownership of work supports continuous improvement. The area also emphasizes the importance of self-management, knowledge sharing, and collective responsibility within Scrum Teams.

Domain: Managing Products with Agility
Candidates should understand how product development is guided by adaptability and continuous feedback. This includes awareness of how product backlogs support prioritization, how iterative delivery helps validate product assumptions, and how developers contribute to refining and delivering backlog items. The focus is on understanding how Agile product management practices align development activities with evolving product goals and stakeholder needs.

Detailed Knowledge Explanation

1. PSD-I Developing & Delivering Products Professionally

In the modern landscape of software engineering, technical excellence is the primary strategic driver of business agility. Integrating professional development practices directly into the Scrum framework is not an optional enhancement; it is a fundamental requirement for reducing technical debt and accelerating the delivery of "Done" increments. By embedding rigor into the daily workflow, Scrum teams transform technical debt from a silent productivity killer into a manageable variable, ensuring that the product remains in a potentially releasable state at the conclusion of every Sprint. This holistic approach ensures that the "Increment" is not just a collection of features, but a stable, high-quality asset capable of withstanding the pressures of a volatile market.

1.1. Continuous Integration (CI)

Continuous Integration (CI) serves as a critical risk-mitigation strategy by shifting the integration of code from a late-stage "event" to a frequent, automated process. The mechanics of CI follow a disciplined loop: a developer writes and updates code, then commits those changes to a shared central repository. This commit triggers an automated toolchain that executes a build and runs a battery of predefined tests. If any failures occur, the developer is notified immediately, allowing for a rapid fix. By catching integration conflicts and bugs within hours—rather than weeks—CI prevents the "integration hell" typical of traditional waterfall projects and maintains a stable baseline for the entire team.

1.1.1. Comparison of CI Tools

Selecting an appropriate CI tool requires balancing organizational flexibility with infrastructure requirements. Jenkins remains a stalwart for enterprises requiring deep customizability and complex pipeline orchestration through its open-source ecosystem. In contrast, teams seeking low-overhead, cloud-native solutions often turn to Travis CI for its simplicity in open-source projects or GitHub Actions for its seamless integration and YAML-based workflows within the GitHub environment. For containerized applications and Kubernetes-heavy environments, CircleCI offers superior parallel execution capabilities. Meanwhile, teams already utilizing GitLab for source control management often leverage GitLab CI/CD for its integrated container registry and built-in testing features, ensuring a unified development experience.

1.1.2. Role of CI in Scrum Teams

CI provides the technical foundation for the Scrum values of transparency and inspection. Through automated linting, static analysis, and accelerated feedback loops, the Development Team gains an objective view of the product’s health at all times. This constant validation is essential for the team to uphold their commitment to the Definition of Done, ensuring that a working, potentially releasable product increment is available by the end of every Sprint, thereby facilitating continuous delivery and empirical decision-making.

1.2. Test-Driven Development (TDD)

Test-Driven Development (TDD) is a "test-first" methodology that fundamentally redefines the coding process. Instead of writing implementation code and testing it later, a developer first writes a small, failing test that describes a specific desired behavior. Only then is the minimal amount of code written to make that test pass, followed by a refactoring phase to clean the implementation. This approach shifts the focus from writing lines of code to validating requirements and design upfront, ensuring that every function serves a verified purpose.

1.2.1. TDD vs. Traditional Development

The TDD workflow contrasts sharply with the traditional "code-first" approach, where testing is often an afterthought and bugs are discovered late in the cycle. By utilizing the "Red-Green-Refactor" cycle, TDD provides early error detection and prevents the accumulation of technical debt. While TDD requires a higher initial investment in development time, it offers a critical strategic trade-off: it significantly increases "internal quality" (code health and maintainability), which eventually drives higher "external quality" by reducing the frequency of bugs encountered by the end-user over the product's lifecycle.

1.2.2. Behavior-Driven Development (BDD)

Behavior-Driven Development (BDD) extends the principles of TDD into the realm of stakeholder collaboration. By using Gherkin syntax—a natural language format using "Given, When, Then" scenarios—BDD creates a shared understanding between the business, developers, and testers. This strategic alignment ensures that technical execution is always rooted in intended business outcomes, preventing the development of features that are technically sound but fail to meet user needs.

1.2.3. Testing Pyramid

To optimize resource allocation, teams utilize the Testing Pyramid model. The foundational layer consists of Unit Tests (using tools like JUnit, PyTest, or Mocha), which are fast, numerous, and focus on isolated functions. The middle layer, Integration Tests (leveraging Postman or RestAssured), validates the interactions between modules. The apex contains End-to-End (E2E) Tests (using Selenium or Cypress), which verify the full system functionality from the user's perspective. By maintaining a broad base of unit tests and a narrow top of E2E tests, teams achieve high reliability without the bottleneck of slow-running full-system suites.

1.3. Automated Testing

Automated testing replaces manual, error-prone verification with software-driven suites that provide a safety net for continuous delivery. By executing these tests with every commit, teams eliminate human error from the verification process and gain the confidence required to make rapid changes to the codebase.

1.3.1. Types of Automated Testing

System reliability is ensured through a multi-faceted testing approach. UI Testing (Selenium, Cypress) validates that interface elements behave correctly for the user. API Testing (Postman, RestAssured) ensures that backend services and endpoints communicate reliably. Finally, Performance Testing, utilizing tools like JMeter or Gatling, evaluates the system’s response times and scalability under load. Each type serves a distinct role in ensuring the product is robust and production-ready.

1.3.2. Integration of Automated Testing into CI/CD

The power of automated testing is maximized within the sequential flow of a CI/CD pipeline. Upon a code commit, the CI tool first runs unit tests; if successful, it triggers integration tests. The pipeline then deploys the code to a staging environment where automated UI and API tests validate the full system functionality. This rigorous, automated sequence acts as a quality gate, ensuring that only code that has passed every level of validation reaches production.

1.4. Lean and Agile Development Principles

Lean and Agile principles focus on maximizing value delivery while ruthlessly eliminating waste. Waste is defined as any activity—such as unnecessary documentation, over-processing, or waiting for feedback—that does not contribute to the final product. These principles allow teams to remain adaptable and responsive to the volatile demands of modern markets.

1.4.1. Agile Methodologies

Different methodologies offer unique frameworks for engineering excellence. Scrum emphasizes iterative delivery through Sprints and empirical process control. Kanban focuses on visual workflow management and Work in Progress (WIP) limits to optimize flow. Extreme Programming (XP) specifically targets technical health through engineering practices such as pair programming and TDD.

1.4.2. Lean Software Development: The 7 Principles

Lean development is anchored by seven principles that impact value delivery and team empowerment:

1. **Eliminate Waste**: Removing non-value-added steps, like excessive meetings or documentation, to streamline the path to "Done."
2. **Amplify Learning**: Using frequent iterations and experimentation to build knowledge quickly.
3. **Decide as Late as Possible**: Delaying irreversible decisions until maximum information is available to reduce the risk of costly rework.
4. **Deliver as Fast as Possible**: Utilizing incremental delivery to get value into users' hands early.
5. **Empower the Team**: Trusting the people doing the work to make technical decisions, which increases engagement.
6. **Build Integrity In**: Using CI/CD and TDD to ensure that quality is a built-in feature, not an afterthought.
7. **Optimize the Whole**: Focusing on the end-to-end process improvement rather than perfecting isolated components at the expense of the system.

While these technical practices provide the machinery for delivery, their success is ultimately dependent on the social and structural health of the team.

1.5 Developing & Delivering Products Professionally Practice Question

Q1: Which of the following best describes Continuous Integration (CI)?
A) A practice where developers integrate code at the end of a project to ensure stability
B) A development approach where code is integrated into a shared repository frequently, with automated testing to detect issues early
C) A process where developers manually test their code before integration
D) A testing method used only in Test-Driven Development (TDD)

Q2: What is the primary benefit of Test-Driven Development (TDD)?
A) Developers can write code faster without worrying about testing
B) It eliminates the need for integration testing
C) It helps ensure that code meets requirements before it is written
D) It allows for manual testing to replace automated testing

Q3: Which of the following statements about Automated Testing is true?
A) Automated testing is only useful for small projects
B) Automated testing reduces the need for Continuous Integration
C) Automated testing helps identify errors quickly and efficiently
D) Automated testing completely eliminates the need for manual testing

Q4: Which of the following are key principles of Lean Development? (Select two)
A) Maximizing waste in the development process
B) Delivering small, incremental releases quickly
C) Making decisions as early as possible
D) Continuous improvement of both the product and the process

Q5: In the context of Agile and Lean development, what is the purpose of prioritizing work based on customer value?
A) To reduce the workload of the development team
B) To ensure the most valuable features are developed first, maximizing business impact
C) To complete all low-priority tasks before starting high-priority ones
D) To follow a strict project timeline without considering changes in customer needs

Q6: A Scrum team is practicing Continuous Integration. Which of the following is a key practice they should follow?
A) Developers should integrate their changes into the shared repository at the end of each Sprint
B) Every integration should trigger an automated build and test suite
C) Manual testing should be performed before each integration
D) Code integration should only happen after a feature is fully complete

Q7: Which of the following best describes the "Testing Pyramid" concept?
A) A method where integration tests are the primary focus, with little need for unit tests
B) A model that emphasizes a strong foundation of unit tests, with fewer integration and UI tests
C) A strategy that eliminates the need for UI testing in Agile projects
D) A testing approach where manual tests are prioritized over automated tests

Q8: A development team is using TDD (Test-Driven Development). What should they do **before** writing a new feature's implementation?
A) Write a test that describes the expected behavior of the feature
B) Implement the feature and then write tests to verify it
C) Perform manual testing to check if the feature works as expected
D) Merge the feature into the main codebase before running any tests

Q9: Which Agile methodology emphasizes engineering practices like Test-Driven Development (TDD), Pair Programming, and Continuous Integration?
A) Scrum
B) Kanban
C) Extreme Programming (XP)
D) Lean

2. PSD-I Developing People and Teams

The effectiveness of any technical practice is constrained by the social and structural organization of the team. In professional Scrum, success is driven by the transition from managed groups to autonomous, high-performing units. This shift acknowledges that the individuals closest to the technical challenges are best equipped to solve them, provided they are supported by a framework that fosters collaboration, cross-functionality, and continuous improvement.

2.1. Cross-Functional Teams

Cross-functional teams possess all the expertise—including coding, testing, design, and infrastructure—necessary to deliver a "Done" increment without external help. By consolidating these skills into a single unit, the team minimizes dependencies on other departments, which significantly reduces delays and handoff-related errors. This structure promotes a high degree of independence and accelerates knowledge sharing across traditional boundaries.

2.1.1. T-Shaped Skills: The Foundation of Cross-Functionality

The "T" metaphor describes a professional with "depth" (the vertical bar) in a specific area, such as database security, and "breadth" (the horizontal bar) across other domains like frontend development or testing. T-shaped skills are the bedrock of cross-functionality because they enhance team adaptability. When a specialist is unavailable, others possess enough breadth to assist, preventing the bottlenecks that plague single-specialist teams.

2.1.2. Cross-Functional vs. Multi-Functional Teams

Scrum mandates a cross-functional approach where members switch tasks across disciplines to ensure the increment is completed. This differs from multi-functional teams, where individuals with different roles are grouped together but still work in silos, maintaining separate responsibilities and handoff-based workflows. Scrum breaks these silos to ensure collective ownership of the Sprint Goal.

2.2. Self-Organizing Teams

Self-organization grants the team the authority to decide how to plan and execute their work. When teams operate with autonomy, they demonstrate higher engagement and faster decision-making. Accountability for the outcome is higher because the team is responsible for both the "how" and the execution of the work.

2.2.1. Management in a Self-Organizing Context

Self-organization does not mean "no management"; rather, the team manages its own internal workflow and task distribution. The Product Owner manages the product's direction, and the Scrum Master manages the process health. This replaces traditional command-and-control with a framework where management is a shared responsibility focused on facilitation and vision.

2.2.2. The Scrum Master's Role in Self-Organizing Teams

The Scrum Master facilitates autonomy by coaching the team toward self-sufficiency. This involves shielding the team from external interruptions, such as non-Sprint work or shifting priorities, and helping the team resolve internal conflicts collaboratively. By resisting the urge to dictate solutions, the Scrum Master empowers the team to evolve their own processes.

2.2.3. Anti-Patterns in Self-Organizing Teams

Autonomy is often threatened by "Fake Self-Organization," where a manager outside the team still dictates task assignments or technical choices. Another risk is the lack of structure where no one takes ownership, or a Scrum Master who lapses into micro-management by assigning tickets. These patterns erode trust and stall the development of a high-performing culture.

2.3. Feedback and Continuous Improvement

Feedback loops are the lifeblood of Scrum, ensuring that the team and the product remain sustainable and relevant. Frequent evaluation of the work and the process allows the team to pivot based on real-world evidence.

2.3.1. Sources of Feedback Beyond Sprint Events

Beyond the formal Sprint events, teams leverage 360-degree feedback from users (via interviews or analytics), internal team feedback (via peer code reviews and pair programming), and automated feedback (via CI tools). This varied input ensures that issues are caught as they arise, not just at the end of a cycle.

2.3.2. Key Characteristics of Effective Feedback

To drive meaningful improvement, feedback must be timely (provided as soon as the issue is spotted), specific (using concrete examples rather than vague critiques), and constructive (focused on the solution rather than blame). Feedback that is delayed or vague fails to offer the clarity needed for the team to improve.

2.3.3. Kaizen: The Scrum Approach to Continuous Improvement

Scrum adopts the Lean principle of Kaizen—continuous, small, incremental improvements. By using the Sprint Retrospective to identify one or two small process optimizations every Sprint, teams achieve significant long-term growth. This approach makes every team member responsible for improvement, ensuring the workflow stays efficient and optimized.

While team dynamics and engineering practices provide the capacity for work, that work must be strategically aligned to provide business value.

2.4 Developing People and Teams Practice Question

Q1: Which of the following statements best describes a cross-functional team in Scrum?
A) A team where every member specializes in one area and does not work on tasks outside their expertise
B) A team that includes individuals with different skills needed to deliver a working product increment
C) A team that consists of developers only, while testing and design are handled by separate teams
D) A team that follows strict role assignments where each member is responsible for a specific task

Q2: Which of the following is a key characteristic of a self-organizing Scrum team?
A) The Scrum Master assigns tasks to the team members
B) The team decides how to complete its work without being directed by external management
C) The Product Owner controls how tasks should be implemented
D) The Scrum Master monitors and evaluates individual team member performance

Q3: What is one of the main benefits of a self-organizing team?
A) It eliminates the need for a Product Owner
B) It allows management to closely supervise every aspect of the team's work
C) It increases team ownership and accountability, leading to higher engagement
D) It ensures that the Scrum Master makes all key decisions for the team

Q4: Which of the following are TRUE about cross-functional teams? (Select two)
A) They include all the skills needed to deliver a product increment
B) They require external teams to handle testing and deployment
C) They are essential for reducing dependencies and improving efficiency
D) They do not require collaboration between different disciplines

Q5: What is the primary goal of the Sprint Retrospective?
A) To inspect and adapt the way the Scrum team works to improve future Sprints
B) To demonstrate the product increment to stakeholders
C) To assign new tasks for the next Sprint
D) To update the Product Backlog

Q6: In a Sprint Review, what type of feedback is typically gathered?
A) Feedback from the Scrum Master about the team's performance
B) Feedback from stakeholders about the product increment
C) Feedback from the Development Team about process improvements
D) Feedback about individual team member contributions

Q7: What is a key characteristic of effective feedback in Scrum?
A) It should be personal and focus on individual performance
B) It should be immediate, specific, and constructive
C) It should be given only by senior management
D) It should be vague to avoid conflict

Q8: A Scrum team is struggling to work effectively together. What action should the Scrum Master take?
A) Assign tasks to each team member to ensure accountability
B) Organize a Retrospective to help the team identify collaboration improvements
C) Request management to intervene and enforce a new working process
D) Replace underperforming team members with new hires

Q9: How does the Scrum Master support continuous improvement?
A) By enforcing strict rules and monitoring individual team members
B) By facilitating Sprint Retrospectives and encouraging experimentation
C) By ensuring the Product Owner dictates the development process
D) By assigning work to developers based on their skills

Q10: Which of the following are feedback mechanisms in Scrum? (Select two)
A) Sprint Retrospective
B) Sprint Review
C) Daily Scrum
D) Product Release

3. PSD-I Managing Products with Agility

Product management in Agile is an empirical discipline that prioritizes flexibility and value capture over rigid planning. By shifting from fixed milestones to iterative delivery, organizations can respond to market feedback and technological changes in real-time. This ensures that the product evolves based on what users actually need, rather than what was assumed at the start of a project.

3.1. Product Goals and Roadmaps

Product Goals provide the long-term "North Star," while Roadmaps offer a high-level, visual plan for achieving that vision. These tools align the team's efforts with the business strategy while remaining adaptable enough to pivot when new insights emerge.

3.1.1. Scrum vs. Traditional Product Management

Traditional management often relies on fixed, 12-month waterfall plans that are difficult to change. Scrum uses "rolling wave planning," where the roadmap is continuously updated after every Sprint based on stakeholder feedback. this allows for rapid adjustments and ensures the team is always working on the most relevant features.

3.1.2. Breaking Down Product Goals in Scrum

The vision is structured as a hierarchy: the long-term Product Goal is broken into themes on the Roadmap, which are then decomposed into actionable Product Backlog Items (PBIs). This ensures that even the smallest task performed by a developer is directly connected to a strategic objective.

3.1.3. Using OKRs (Objectives and Key Results) in Scrum

Objectives and Key Results (OKRs) bridge the gap between high-level goals and measurable impact. For example, if the Objective is to "Improve user-friendliness," the Key Result might be "Reduce checkout time by 20%." This prevents the team from focusing on mere output (number of features) and shifts the focus to outcomes (measurable value).

3.2. Product Backlog Management

The Product Backlog is a dynamic, living document representing all work required for the product. The Product Owner maintains its transparency and priority to ensure the team is consistently delivering maximum value.

3.2.1. Prioritization Techniques for Product Backlog

Product Owners use several frameworks to balance competing needs. The MoSCoW method categorizes items as Must-have, Should-have, Could-have, or Won't-have to define the MVP. The Kano Model prioritizes based on customer satisfaction categories like Basic, Performance, and Excitement needs. Finally, Weighted Shortest Job First (WSJF) calculates priority by considering the "Cost of Delay" (value, urgency, risk) divided by the "Job Size" (effort), effectively highlighting high-value "quick wins."

3.2.2. Backlog Refinement: A Continuous Process

Refinement is an ongoing activity where the Product Owner and Development Team clarify items, resolve dependencies, and ensure PBIs are small enough to be completed within a single Sprint. Effective refinement reduces ambiguity and ensures that the most important work is always ready for Sprint Planning.

3.2.3. Scrum Anti-Patterns in Product Backlog

Agility is hindered by "Static Backlogs" that never change, "PO Dictatorships" that ignore the team's technical input, or "Oversized PBIs" that take multiple Sprints to complete. Avoiding these patterns is essential for maintaining a transparent and responsive development flow.

3.3. Incremental Delivery

Incremental delivery is the practice of producing a working, potentially releasable product increment every Sprint. This provides stakeholders with tangible evidence of progress and significantly reduces the risk of market misalignment.

3.3.1. Ensuring Increments Are “Potentially Releasable”

Every increment must meet the "Definition of Done" (DoD) to be considered complete. Typical DoD criteria include code being reviewed and merged, all automated tests passing, the increment being deployed to a staging environment, and documentation being updated. A rigorous DoD prevents the accumulation of unfinished work and ensures quality is never compromised for speed.

3.3.2. Minimum Viable Product (MVP) in Scrum

The MVP is the smallest version of a product that allows for early market validation. This approach enables teams to test hypotheses and gather user feedback quickly, contrasting with traditional methods that attempt to build every feature before a single release.

3.3.3. Incremental Delivery vs. Waterfall Delivery

Incremental delivery offers a lower risk profile by creating frequent feedback loops. While a waterfall project might take a year to discover a fundamental flaw or market shift, a Scrum team identifies these issues within weeks, allowing for course correction before significant resources are wasted.

The strategic management of products and teams is ultimately governed by the formal rules of the Scrum framework.

3.4 Managing Products with Agility Practice Question

Q1: What is the primary purpose of a Product Goal in Scrum?
A) To describe the specific work to be completed in the current Sprint
B) To provide a long-term objective that the Scrum Team works toward
C) To define the daily tasks for each Development Team member
D) To ensure the Product Owner assigns work to the Development Team

Q2: Which of the following best describes a Product Roadmap in Scrum?
A) A fixed, long-term plan that cannot change
B) A high-level, flexible plan that outlines how the Product Goal will be achieved
C) A detailed breakdown of every Sprint's tasks for the next six months
D) A list of user stories assigned to each developer

Q3: Who is responsible for managing the Product Backlog in Scrum?
A) The Scrum Master
B) The Product Owner
C) The Development Team
D) The Stakeholders

Q4: What are the key characteristics of an effective Product Backlog? (Select two)
A) It is dynamic and continuously evolving
B) It is a fixed list of requirements for the entire project
C) It is ordered based on value and priority
D) It only contains user stories, without technical tasks or bug fixes

Q5: What is the purpose of Backlog Refinement?
A) To ensure that all backlog items are fully detailed before the Sprint starts
B) To continuously update and improve backlog items, making them clearer and more actionable
C) To assign specific backlog items to individual developers
D) To finalize the Sprint Backlog for the next Sprint

Q6: In Scrum, what does "incremental delivery" mean?
A) The entire product is delivered only at the end of the project
B) Features are developed separately and delivered as large releases
C) The team delivers a working product increment at the end of each Sprint
D) Work is only released once all backlog items are completed

Q7: Which of the following are benefits of incremental delivery in Scrum? (Select two)
A) Faster feedback from users and stakeholders
B) The ability to adapt to changing market conditions
C) A fixed plan that does not change
D) Development is completed in large, infrequent releases

Q8: What is the Definition of Done (DoD) in Scrum?
A) A checklist of requirements that an item must meet before it is considered complete

B) A list of tasks assigned to each developer
C) A document created only by the Scrum Master
D) A fixed requirement that cannot change throughout the project

Q9: What is the role of the Sprint Review in managing a product with agility?
A) To inspect the product increment and adapt the Product Backlog based on feedback
B) To discuss team performance and assign tasks for the next Sprint
C) To document all completed work in a final project report
D) To finalize the product roadmap for the next year

Q10: How does Scrum ensure continuous value delivery?
A) By delivering working product increments in every Sprint
B) By planning all Sprints at the beginning of the project
C) By waiting until all product features are completed before release
D) By focusing only on technical improvements rather than business value

4. PSD-I Understanding and Applying the Scrum Framework

Scrum is a framework for tackling complex, unpredictable problems through empirical process control. By balancing roles, events, and artifacts, Scrum creates a system of transparency, inspection, and adaptation that allows teams to deliver high-quality products consistently.

4.1. Scrum Roles

The three Scrum roles—Scrum Master, Product Owner, and Development Team—create a balance of power between product vision, technical execution, and process health.

4.1.1. Scrum Master's Additional Responsibilities

The Scrum Master is an organizational change agent and the upholder of Scrum values: Commitment, Focus, Openness, Respect, and Courage. They ensure these values are practiced daily to foster trust and help the organization understand how to support the Scrum Team's success.

4.1.2. Common Pitfalls for Product Owners

Anti-patterns for Product Owners include overstepping into technical "how-to" decisions or neglecting stakeholder engagement. If a PO fails to act as the liaison for the customer, the team risks building a product that is technically perfect but commercially irrelevant.

4.1.3. Key Principles for the Development Team

The team must remain self-organizing and cross-functional. By developing T-shaped skills, they ensure they have all the expertise necessary to deliver a "Done" increment without external help, maintaining the speed and autonomy required for true agility.

4.2. Scrum Events

Scrum events are time-boxed opportunities for inspection and adaptation, ensuring that communication remains focused and progress is transparent.

4.2.1. Key Questions in Sprint Planning

Sprint Planning establishes the "Why" (the Sprint Goal), the "What" (items from the Product Backlog), and the "How" (the task breakdown). Answering these questions results in a clear Sprint Goal and a realistic Sprint Backlog that the team owns.

4.2.2. Anti-Patterns in Daily Scrum

The Daily Scrum is a 15-minute synchronization for the Development Team, not a status report for the PO or SM. A healthy meeting focuses on collaboration and identifying impediments. If technical discussions require more than 15 minutes, they must be taken "offline" immediately after the meeting to respect the timebox.

4.2.3. Key Components of Sprint Review

The Review is a collaborative session where stakeholders and the team inspect the increment and adapt the Product Backlog. It is more than a demo; it is a critical feedback loop where the roadmap is adjusted based on current market realities.

4.2.4. Sprint Retrospective Techniques

Teams use structured techniques like "Start-Stop-Continue" (identifying new practices to adopt, poor ones to drop, and good ones to keep) or the "4L" method (Liked, Learned, Lacked, Longed For). these frameworks ensure that process improvements are actionable and specific.

4.3. Scrum Artifacts

Artifacts provide the transparency needed for inspection and adaptation. Each artifact contains a commitment that reinforces its purpose and quality.

4.3.1. Key Characteristics of Product Backlog

The Product Backlog is dynamic and prioritized by the Product Owner based on business value, risk, complexity, and dependencies. It is the single source of work for the team and evolves as the product and market change.

4.3.2. Key Aspects of Sprint Backlog

The Sprint Backlog is the team's plan for the Sprint. It is owned exclusively by the Development Team and is updated throughout the Sprint as work is decomposed into small, manageable tasks to maintain visibility.

4.3.3. Definition of Done (DoD)

The DoD is the shared quality standard for the team. It ensures transparency by providing a common understanding of what "Done" means. For a professional Scrum team, "Done" typically means the code is

reviewed, merged, all tests (unit, integration, and UI) have passed, and the feature is deployed to a staging environment for validation.

The foundation of professional Scrum development is the seamless integration of technical excellence, team empowerment, and agile product management. When these elements are combined within the Scrum framework, teams are equipped to navigate complexity, reduce waste, and deliver meaningful value in an ever-changing world.

4.4 Understanding and Applying the Scrum Framework Practice Question

Q1: Which of the following is NOT a responsibility of the Scrum Master?
A) Ensuring the Scrum team follows Scrum principles
B) Assigning tasks to the Development Team
C) Removing obstacles that hinder the team's progress
D) Coaching the team on Agile best practices

Q2: Which of the following is a key responsibility of the Product Owner?
A) Ensuring the team completes all tasks assigned in a Sprint
B) Maintaining and prioritizing the Product Backlog
C) Conducting daily stand-up meetings
D) Assigning tasks to individual developers

Q3: What is the main purpose of the Sprint Retrospective?
A) To review and demonstrate the product increment to stakeholders
B) To refine and update the Product Backlog
C) To inspect how the last Sprint went and identify improvements
D) To assign new tasks to the team for the next Sprint

Q4: Which of the following are characteristics of a well-formed Scrum team? (Select two)
A) Cross-functional, meaning they have all the skills needed to complete the work
B) Self-organizing, meaning they determine how to complete the work
C) Led by a Project Manager who assigns work
D) Composed only of developers, without any testers or designers

Q5: In Scrum, who is responsible for defining the Sprint Goal?
A) The Scrum Master
B) The Product Owner
C) The Development Team
D) The Scrum Team as a whole

Q6: Which of the following best describes the purpose of the Sprint Review?
A) To inspect the Increment and adapt the Product Backlog based on feedback
B) To identify process improvements for the next Sprint
C) To discuss personal performance of team members
D) To finalize the work for the upcoming Sprint

Q7: Which statement about the Sprint Backlog is TRUE?
A) It contains all items from the Product Backlog
B) It remains fixed throughout the Sprint and cannot be changed
C) It is owned by the Development Team
D) It is defined by the Scrum Master

Q8: What does the Definition of Done (DoD) ensure?
A) The product is ready for release after each Sprint
B) All work in the Product Backlog is completed
C) All incomplete work is added to the next Sprint
D) Each task has been assigned to a specific developer

Q9: Which of the following is TRUE about the Daily Scrum?
A) It is an optional meeting if the team prefers other forms of communication
B) It should be scheduled for at least 30 minutes to ensure detailed updates
C) It helps the Development Team synchronize and plan the next 24 hours
D) It is primarily used by the Scrum Master to check team progress

Q10: Which of the following statements about Product Backlog refinement is correct?
A) It is an official Scrum event that occurs after the Sprint Planning
B) It is the sole responsibility of the Scrum Master
C) It is an ongoing activity where backlog items are clarified and prioritized
D) It happens only when the Sprint Backlog is completed

Learning Path & Study Advice

A structured preparation approach begins with building a clear understanding of Agile values and the principles that guide iterative product development. Learners should first become familiar with the purpose and structure of the Scrum framework, including how Scrum Teams organize work and collaborate to deliver product increments.

After establishing a conceptual understanding of Scrum, candidates should focus on professional engineering practices that enable continuous integration, reliable delivery, and maintainable software. Understanding how these practices contribute to sustainable product development is essential.

Candidates should also reflect on team collaboration and communication patterns that support effective Scrum Teams. Observing or participating in iterative development environments can help reinforce how Scrum events, shared ownership, and incremental delivery function in practice. Studying the relationships between product goals, backlog refinement, and development activities further strengthens conceptual understanding of Agile product management.

Who This PDF Is For

This document is intended for software developers and technical professionals who work in Agile environments or plan to work within Scrum Teams. It is particularly relevant for individuals who contribute directly to building, integrating, testing, and delivering software increments.

Readers who already possess basic programming knowledge and an interest in Agile development practices will benefit most from this material. The content is also suitable for developers transitioning from traditional development models to iterative product development approaches, as well as professionals seeking a structured understanding of how modern engineering practices align with Scrum-based product delivery.

Call To Action

This document provides an overview of structured learning and certification preparation approaches. For learners seeking clear knowledge organization, guided study planning, and exam-focused practice resources, AAAdemy offers a comprehensive platform to support independent and effective learning.

Explore additional training materials, study guidance, and practice resources at:

https://www.aaademy.com/Professional-Scrum-Developer/PSD-I.html

Online Flashcards (Quizlet):

https://quizlet.com/user/AAAdemy/folders/psd-i-professional-scrum-developer-flashcards-aaademy?i=6zfa5t&x=1xqt

Attachment: Answers by Knowledge Point

Developing & Delivering Products Professionally Practice Question
A1: Answer: B
Explanation: Continuous Integration (CI) is a development practice where developers frequently integrate their code into a shared repository, allowing automated tools to check for errors and maintain code stability. Option A is incorrect because CI happens throughout the project, not just at the end. Option C is incorrect because automated testing is a key aspect of CI. Option D is incorrect because CI is not exclusive to TDD.

A2: Answer: C
Explanation: TDD ensures that code meets requirements before being written by enforcing a test-first approach. Developers write tests before writing the actual code, ensuring clarity and correctness. Option A is incorrect because TDD involves additional steps that may initially slow development but improve code quality. Option B is

incorrect because integration testing is still necessary. Option D is incorrect because TDD emphasizes automated testing, not manual testing.

A3: Answer: C
Explanation: Automated testing allows for faster and more efficient detection of errors, improving the reliability of software development. Option A is incorrect because automated testing benefits projects of all sizes. Option B is incorrect because automated testing supports Continuous Integration, rather than reducing its need. Option D is incorrect because some forms of manual testing, like exploratory testing, are still valuable.

A4: Answer: B, D
Explanation: Lean development focuses on eliminating waste, delivering value quickly, and continuously improving both the product and the development process. Option A is incorrect because Lean Development seeks to eliminate waste. Option C is incorrect because Lean Development promotes delayed decision-making to allow for more informed choices.

A5: Answer: B
Explanation: Prioritizing work based on customer value ensures that the development team focuses on delivering the most impactful features first, maximizing business and user satisfaction. Option A is incorrect because prioritization is about value, not reducing workload. Option C is incorrect because prioritization means addressing the highest-value tasks first, not necessarily completing all low-priority tasks first. Option D is incorrect because Agile methodologies emphasize adaptability and responding to customer needs.

A6: Answer: B
Explanation: In Continuous Integration, every integration should trigger an automated build and test suite to detect issues early. Option A is incorrect because integrations should happen frequently, not just at the end of a Sprint. Option C is incorrect because automated testing is a key aspect of CI, though manual testing may still be used. Option D is incorrect because CI encourages early and frequent integration, not waiting until a feature is fully complete.

A7: Answer: B
Explanation: The "Testing Pyramid" suggests having a strong foundation of unit tests, fewer integration tests, and even fewer UI tests, to create a cost-effective and efficient testing strategy. Option A is incorrect because unit tests, not integration tests, form the foundation. Option C is incorrect because UI testing is still needed, but in a limited scope. Option D is incorrect because automated tests are preferred for efficiency.

A8: Answer: A
Explanation: In TDD, developers first write a test that defines the expected behavior, then write the minimal code needed to pass the test, and finally refactor the code while ensuring all tests continue to pass. Option B is incorrect because tests should be written before the implementation. Option C is incorrect because automated tests are prioritized in TDD. Option D is incorrect because merging untested code can introduce instability.

A9: Answer: C
Explanation: Extreme Programming (XP) is an Agile methodology that emphasizes engineering practices like TDD, Pair Programming, and Continuous Integration to improve software quality. Option A (Scrum) is incorrect because Scrum focuses more on process management than engineering practices. Option B (Kanban) is incorrect because it focuses on workflow visualization and limiting work in progress. Option D (Lean) is incorrect because it is a broader philosophy focused on eliminating waste.

Understanding and Applying the Scrum Framework Practice Question

A1: Answer: B

Explanation: The Scrum Master does not assign tasks to the Development Team because Scrum promotes self-organization. The Development Team is responsible for selecting tasks from the Sprint Backlog. The Scrum Master facilitates the Scrum process (A), removes obstacles (C), and helps the team improve their Agile practices (D).

A2: Answer: B

Explanation: The Product Owner's primary responsibility is to manage the Product Backlog, ensuring that the team works on the most valuable features. They prioritize backlog items based on business needs. Option A is incorrect because the Development Team, not the PO, ensures Sprint tasks are completed. Option C is incorrect because the Scrum Master facilitates the Daily Scrum. Option D is incorrect because Scrum teams are self-organizing.

A3: Answer: C

Explanation: The Sprint Retrospective focuses on continuous improvement by allowing the team to reflect on what went well, what didn't, and how to improve in the next Sprint. Option A describes the Sprint Review. Option B is incorrect because backlog refinement is an ongoing activity, not the focus of the Retrospective. Option D is incorrect because tasks are selected in Sprint Planning, not the Retrospective.

A4: Answer: A, B

Explanation: A Scrum team is cross-functional (A), meaning it includes developers, testers, and other necessary roles. They are self-organizing (B), meaning they decide how to complete their work without a manager assigning tasks. Option C is incorrect because Scrum does not have a Project Manager role. Option D is incorrect because a Scrum team needs all necessary skills, not just developers.

A5: Answer: D

Explanation: The Scrum Team as a whole collaborates to define the Sprint Goal during Sprint Planning. The Product Owner provides business priorities, the Development Team determines technical feasibility, and the Scrum Master ensures the team follows Scrum principles. No single role defines the goal alone.

A6: Answer: A

Explanation: The Sprint Review focuses on inspecting the product Increment and receiving feedback from stakeholders, which may lead to updates in the Product Backlog. Option B describes the Sprint Retrospective. Option C is incorrect because Scrum does not evaluate individual performance. Option D is incorrect because Sprint Planning, not the Review, defines the next Sprint's work.

A7: Answer: C

Explanation: The Development Team owns the Sprint Backlog and is responsible for updating it. Option A is incorrect because the Sprint Backlog is a subset of the Product Backlog. Option B is incorrect because the Sprint Backlog can be adjusted as new learnings emerge. Option D is incorrect because the Scrum Master does not define the Sprint Backlog.

A8: Answer: A

Explanation: The Definition of Done (DoD) ensures that a product Increment is ready for potential release. It establishes quality standards, such as passing tests, code reviews, and documentation updates. Option B is incorrect because the Product Backlog is never fully completed. Option C is incorrect because incomplete work

goes back into the Product Backlog, not necessarily the next Sprint. Option D is incorrect because Scrum does not assign tasks to individuals; teams self-organize.

A9: Answer: C
Explanation: The Daily Scrum helps the Development Team synchronize their work and plan for the next 24 hours. Option A is incorrect because the Daily Scrum is mandatory in Scrum. Option B is incorrect because it should be limited to 15 minutes. Option D is incorrect because it is for the Development Team, not a status update for the Scrum Master.

A10: Answer: C
Explanation: Product Backlog refinement is an ongoing process where backlog items are clarified, estimated, and reprioritized. Option A is incorrect because refinement is not an official Scrum event. Option B is incorrect because the Product Owner leads refinement with team input, not the Scrum Master alone. Option D is incorrect because backlog refinement is continuous, not dependent on Sprint Backlog completion.

Developing People and Teams Practice Question
A1: Answer: B
Explanation: A cross-functional team in Scrum consists of members with various skills (such as development, testing, UX design) who work together to deliver a working product increment. Option A is incorrect because team members should have T-shaped skills (deep expertise + broad general knowledge). Option C is incorrect because Scrum teams should not rely on external testing or design teams. Option D is incorrect because roles in a Scrum team are flexible rather than strictly assigned.

A2: Answer: B
Explanation: A self-organizing team in Scrum decides how to complete its work without being directed by external managers. Option A is incorrect because the Scrum Master does not assign tasks. Option C is incorrect because the Product Owner defines the "what" (product requirements), but the team decides the "how" (implementation approach). Option D is incorrect because Scrum does not include individual performance evaluation.

A3: Answer: C
Explanation: Self-organizing teams take ownership of their work, leading to higher engagement, accountability, and motivation. Option A is incorrect because the Product Owner is still required to define priorities. Option B is incorrect because Scrum teams are not micromanaged by management. Option D is incorrect because the Scrum Master facilitates but does not make key decisions for the team.

A4: Answer: A, C
Explanation: A cross-functional team includes all necessary skills (A) and helps reduce dependencies on external teams, improving efficiency (C). Option B is incorrect because a truly cross-functional team should be able to test and deploy on its own. Option D is incorrect because Scrum teams require collaboration across disciplines.

A5: Answer: A
Explanation: The Sprint Retrospective is focused on inspecting and improving the team’s processes and collaboration. Option B describes the Sprint Review, not the Retrospective. Option C is incorrect because task assignment happens in Sprint Planning. Option D is incorrect because Product Backlog refinement is an ongoing process, not the purpose of the Retrospective.

A6: Answer: B
Explanation: The Sprint Review focuses on stakeholder feedback regarding the product increment to ensure that the product is evolving in the right direction. Option A is incorrect because the Scrum Master does not evaluate team performance. Option C describes the Sprint Retrospective, not the Review. Option D is incorrect because Scrum focuses on team performance, not individual evaluation.

A7: Answer: B
Explanation: Effective feedback in Scrum is timely, specific, and constructive, helping teams continuously improve. Option A is incorrect because feedback should focus on team processes and product quality, not personal criticism. Option C is incorrect because feedback comes from multiple sources, not just management. Option D is incorrect because vague feedback does not lead to actionable improvements.

A8: Answer: B
Explanation: A Retrospective helps the team reflect on challenges and identify ways to improve collaboration. Option A is incorrect because the Scrum Master does not assign tasks. Option C is incorrect because Scrum teams are self-organizing, and management interference undermines this. Option D is incorrect because Scrum emphasizes continuous improvement rather than replacing team members.

A9: Answer: B
Explanation: The Scrum Master facilitates Sprint Retrospectives and encourages the team to experiment with new ways to improve. Option A is incorrect because the Scrum Master does not enforce strict rules or monitor individuals. Option C is incorrect because the Product Owner does not dictate development processes. Option D is incorrect because teams self-organize and choose their own work.

A10: Answer: A, B
Explanation: The Sprint Retrospective provides internal team feedback to improve collaboration and efficiency, while the Sprint Review gathers stakeholder feedback on the product. Option C is incorrect because the Daily Scrum is for team synchronization, not feedback collection. Option D is incorrect because feedback should be continuous, not only after a product release.

Managing Products with Agility Practice Question
A1: Answer: B
Explanation: The Product Goal is a high-level objective that describes the overall direction of the product, guiding the Scrum Team's efforts. Option A is incorrect because the Sprint Goal defines the work for a Sprint. Option C is incorrect because Scrum Teams are self-organizing. Option D is incorrect because the Product Owner does not assign work to the team.

A2: Answer: B
Explanation: A Product Roadmap in Scrum is a high-level, evolving plan that outlines how the Product Goal will be achieved over time. Option A is incorrect because Scrum roadmaps are flexible and can change. Option C is incorrect because detailed Sprint planning happens per Sprint, not months in advance. Option D is incorrect because Scrum does not assign stories to individual developers.

A3: Answer: B
Explanation: The Product Owner is responsible for managing and prioritizing the Product Backlog to ensure the team is working on the most valuable items. Option A is incorrect because the Scrum Master facilitates Scrum but does not manage the backlog. Option C is incorrect because the Development Team selects work from the

backlog but does not own its prioritization. Option D is incorrect because stakeholders provide input but do not manage the backlog.

A4: Answer: A, C
Explanation: A well-maintained Product Backlog is dynamic (A), meaning it evolves based on new information, and it is prioritized (C) to ensure the most valuable work is done first. Option B is incorrect because a backlog is not fixed; it adapts to changes. Option D is incorrect because a backlog includes features, technical tasks, bug fixes, and research work.

A5: Answer: B
Explanation: Backlog Refinement is an ongoing activity where the Product Owner and team improve backlog items by clarifying requirements, breaking down work, and ensuring prioritization. Option A is incorrect because not all backlog items need to be fully detailed in advance. Option C is incorrect because Scrum teams self-organize and select work collaboratively. Option D is incorrect because backlog refinement is not the same as Sprint Planning.

A6: Answer: C
Explanation: In Scrum, incremental delivery means that each Sprint results in a usable and potentially releasable product increment. Option A is incorrect because Scrum is not waterfall; delivery happens incrementally. Option B is incorrect because Scrum encourages small, frequent releases, not large ones. Option D is incorrect because work can be released at any time as long as it meets the Definition of Done.

A7: Answer: A, B
Explanation: Incremental delivery allows for faster feedback (A) and makes it easier to adapt (B) to changes. Option C is incorrect because Scrum embraces change rather than following a fixed plan. Option D is incorrect because Scrum delivers in small, frequent increments, not large releases.

A8: Answer: A
Explanation: The Definition of Done (DoD) is a shared agreement on what conditions must be met before an increment is considered complete and potentially releasable. Option B is incorrect because Scrum does not assign tasks to individuals. Option C is incorrect because the Scrum Team collectively defines DoD. Option D is incorrect because DoD can evolve as the team improves.

A9: Answer: A
Explanation: The Sprint Review allows stakeholders to inspect the latest increment and provide feedback, which helps adapt the Product Backlog. Option B is incorrect because team performance is discussed in the Sprint Retrospective, not the Review. Option C is incorrect because Scrum is not focused on extensive documentation. Option D is incorrect because roadmaps in Scrum are flexible and evolve continuously.

A10: Answer: A
Explanation: Scrum ensures continuous value delivery by producing a potentially releasable product increment at the end of each Sprint. Option B is incorrect because Scrum plans iteratively, not all at once. Option C is incorrect because Scrum delivers incrementally, rather than waiting for all features to be completed. Option D is incorrect because Scrum balances technical improvements with business value.