Software Asset Management Professional Practice Exam (Sample)

Study Guide

Everything you need from our exam experts!

Questions

1. How many phases are there in the software model lifecycle?

A. 4

B. 5

C. 6

D. 7

2. What is a key responsibility of a Software Asset Manager?

A. Overseeing hardware assets

B. Ensuring compliance with software licenses

C. Managing marketing strategies

D. Developing new software products

3. What is one of the primary objectives of Practical Management in SAM?

A. Managing software lifecycle

B. Driving out quick-wins and making them visible

C. Enhancing customer satisfaction with software

D. Reducing overall software costs

4. What do Removal Candidates include?

A. Software licenses available for purchase

B. Software installs planned to be uninstalled

C. Recently updated software applications

D. Financial reports on software usage

5. Who is allowed to export the License Position Report?

A. sam_user only

B. sam_user and sam_admin

C. Any user with software permissions

D. Only sam_admin

6. Which state indicates that user notification is checked but no user is assigned?

A. Ready

B. Attention Required

C. Awaiting User

D. Closed Skipped

7. Which type of software license typically has a one-time purchase fee and lasts indefinitely?

A. Subscription license

B. Open-source license

C. Proprietary license

D. Perpetual license

8. Which of the following is NOT a type of software product as outlined in ServiceNow?

A. Licensable

B. Driver

C. Patch

D. Hidden

9. What type of software would be involved in enhancing the performance of an application?

A. Unknown

B. Licensable

C. Patch

D. Non-licensable

10. Which remediation status indicates that the process is currently being worked on?

A. New

B. In Progress

C. Completed

D. Void

Answers

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

Explanations

1. How many phases are there in the software model lifecycle?

A. 4

B. 5

C. 6

D. 7

The correct answer identifies that the software model lifecycle typically encompasses six phases. Understanding these phases is crucial for effective software asset management, as they provide a structured approach to managing software throughout its entire life. 1. **Requirement Analysis: This phase involves gathering and specifying what the software needs to achieve, ensuring that stakeholders' needs are clearly understood and documented. 2. **Design**: In this phase, architects and designers create an outline of the software structure and design how different components will interact, focusing on usability and aesthetics. 3. **Development**: This is where the actual coding occurs, transforming design documents into functional software. It includes writing code and unit testing that ensure code correctness. 4. **Testing**: Critical for quality assurance, this phase involves systematically checking the software for defects, bugs, and ensuring that it meets the specified requirements. 5. **Deployment**: In this phase, the software is made available for use. This can involve installation in a production environment and, in some cases, user training. 6. **Maintenance**: After deployment, the software often requires updates and patches to fix bugs or respond to user needs. This phase ensures the software remains reliable and functional over time. By recognizing that the lifecycle spans these six phases**

2. What is a key responsibility of a Software Asset Manager?

A. Overseeing hardware assets

B. Ensuring compliance with software licenses

C. Managing marketing strategies

D. Developing new software products

Ensuring compliance with software licenses is a central responsibility of a Software Asset Manager because it directly pertains to the effective governance of software assets within an organization. This role involves monitoring and managing software licenses to ensure that all software in use is authorized and compliant with the terms set forth by the software vendors. This includes understanding licensing agreements, conducting audits, and maintaining accurate records of software usage. Software Asset Managers play a critical role in mitigating risks associated with non-compliance, which can lead to legal issues or financial penalties. Additionally, they help optimize software spending by ensuring that organizations only pay for what they actually use, avoiding unnecessary costs associated with over-licensing or purchasing redundant software. While overseeing hardware assets, managing marketing strategies, and developing new software products are important functions within an organization, they fall outside the primary focus of software asset management. Therefore, ensuring compliance with software licenses is the key responsibility that aligns with the objectives of a Software Asset Manager.

3. What is one of the primary objectives of Practical Management in SAM?

A. Managing software lifecycle

B. Driving out quick-wins and making them visible

C. Enhancing customer satisfaction with software

D. Reducing overall software costs

One of the primary objectives of Practical Management in Software Asset Management (SAM) involves driving out quick-wins and making them visible. This approach emphasizes the importance of identifying and implementing immediate benefits that demonstrate the value of SAM initiatives. By spotlighting these quick wins, organizations can build momentum and gain support for broader, long-term SAM strategies. Highlighting quick-wins not only showcases the effectiveness of SAM practices but also boosts stakeholder confidence and buy-in. When tangible, short-term successes are evident, it encourages teams to commit to more comprehensive asset management strategies that ultimately lead to sustained improvements, cost savings, and enhanced software utilization efficiencies. While managing the software lifecycle, enhancing customer satisfaction, and reducing overall software costs are all critical aspects of SAM, the focus on quick-wins is fundamental as it serves as a catalyst for encouraging further investment and engagement in SAM processes. It reinforces a culture of continued improvement and adaptability in managing software assets effectively.

4. What do Removal Candidates include?

A. Software licenses available for purchase

B. Software installs planned to be uninstalled

C. Recently updated software applications

D. Financial reports on software usage

Removal Candidates refer to installations of software that are identified as being uninstalled in the near future. These are typically programs or applications that are no longer needed, have become obsolete, or are being replaced by newer software. By identifying such removal candidates, organizations can optimize their software inventory, reduce maintenance and support costs, and free up resources for more critical applications. Understanding this concept is crucial for effective Software Asset Management, as it allows organizations to manage their software assets proactively, ensuring they only keep what is necessary for their operations. Analyzing the software landscape and making informed decisions about what to remove can lead to significant improvements in efficiency and cost savings.

5. Who is allowed to export the License Position Report?

A. sam_user only

B. sam_user and sam_admin

C. Any user with software permissions

D. Only sam_admin

The License Position Report is a critical document in Software Asset Management as it provides insights into the software licenses owned, deployed, and required for compliance. The correct choice, which includes both sam_user and sam_admin, reflects a logical approach to managing software assets effectively. Having multiple user roles that can export this report allows for broader collaboration and oversight within a company. The sam_user likely has a fundamental understanding of software asset management processes and can generate reports that are essential for day-to-day operations. In contrast, sam_admin typically holds more authority and responsibility over software management, enabling them to act on the findings from such reports. This structure of accessibility ensures that essential reports can be generated by different team members, promoting efficiency and timely decision-making regarding licensing compliance and software utilization. The combination of these roles allows for better governance and shared responsibility, which is vital in maintaining accurate and up-to-date software asset management practices.

6. Which state indicates that user notification is checked but no user is assigned?

A. Ready

B. Attention Required

C. Awaiting User

D. Closed Skipped

The state that indicates user notification is checked but no user is assigned is "Attention Required." In this context, this state signals that while the system has attempted to notify users about a particular issue or task, there has been no assignment made to any user. This could suggest that action or intervention is needed from a manager or admin to either assign a user to the task or re-evaluate the notification process. Other states do not convey this specific meaning. For instance, "Ready" implies that an action or task is fully prepared for processing, while "Awaiting User" suggests a state where user assignment may be pending but does not confirm that notification has already occurred. "Closed Skipped" usually indicates a situation where a task has been bypassed rather than indicating any user assignment or notification checked status. Thus, "Attention Required" accurately reflects a scenario where notification has taken place, yet there is no user engagement in the process.

7. Which type of software license typically has a one-time purchase fee and lasts indefinitely?

A. Subscription license

B. Open-source license

C. Proprietary license

D. Perpetual license

A perpetual license is designed to allow a user to use a specific version of software indefinitely upon a one-time purchase. This means that once the software is bought, the user has the right to continue using that software version for as long as they wish, without any additional fees for ongoing usage. This type of licensing aligns well with organizations that prefer to make a single upfront investment rather than continual payments. In contrast, subscription licenses generally require periodic payments (monthly or annually) to maintain access to the software, effectively limiting use to the duration of the subscription. Open-source licenses grant users rights to use, modify, and distribute software freely, but their primary focus is on the software's availability and freedom rather than on ownership through a traditional purchase. Proprietary licenses, while they can encompass various models including both subscription and perpetual forms, do not inherently guarantee indefinite use upon one-time payment; their terms can vary widely. Thus, the perpetual license stands out as the option explicitly associated with a one-time fee and indefinite duration, which is why this is the correct choice.

8. Which of the following is NOT a type of software product as outlined in ServiceNow?

A. Licensable

B. Driver

C. Patch

D. Hidden

The classification of software products in ServiceNow includes various well-defined categories, such as licensable software, drivers, and patches. Each of these types plays a specific role in software management. Licensable software refers to products for which the organization holds a license and has the legal right to use within the terms specified by the vendor. Drivers are essential components that allow the operating system to communicate effectively with hardware devices. Patches are updates provided by software vendors to fix bugs, enhance features, or address security vulnerabilities in existing software. On the other hand, "hidden" is not typically recognized as a standard category of software products within the ServiceNow framework. While there may be instances of software that is not easily detectable or is intentionally concealed, these do not fit into the formal classifications used in software asset management as outlined by ServiceNow. Recognizing this helps ensure effective tracking, compliance, and management of assets.

9. What type of software would be involved in enhancing the performance of an application?

A. Unknown

B. Licensable

C. Patch

D. Non-licensable

The correct answer is a patch, which is a type of software enhancement that is specifically designed to improve the performance or functionality of an existing application. Patches can address various issues, including bugs, security vulnerabilities, and performance enhancements, thus helping to optimize the application's behavior and efficiency. When a patch is applied to software, it may also include updates that enhance the application's capability or add new features without requiring a complete overhaul or reinstallation of the software. This ability to improve performance through relatively minor updates makes patches a critical component in the ongoing maintenance and management of software applications. Other options like licensable and non-licensable may refer to the nature of the software in terms of its licensing agreement, but they do not directly address software specifically engineered to enhance performance. "Unknown" does not provide any relevant context or specifics in relation to improving application performance.

10. Which remediation status indicates that the process is currently being worked on?

A. New

B. In Progress

C. Completed

D. Void

The status indicating that the process is currently being worked on is "In Progress." This term is commonly used in various project management and remediation contexts to signify that actions are actively taking place to address an issue or fulfill a requirement. When a remediation is marked as "In Progress," it alerts stakeholders and team members that the necessary measures or fixes are not yet finalized but are underway, thus allowing for tracking and management of ongoing efforts effectively. In contrast, other statuses describe different phases of the process. For instance, "New" may indicate that an issue has been identified but not yet addressed, while "Completed" signifies that all necessary actions have been finalized, and the matter is resolved. "Void" typically suggests that the issue is no longer relevant or applicable, making it clear that it isn't currently being worked on. Understanding this distinction is essential for effective tracking and management within software asset management processes.