PassQueen
a.k.a dumpsqueen.com

PeopleCert DevOps Engineer Exam

PEOPLECERT DevOps-Engineer

Version Demo

Total Demo Questions: 7

Total Premium Questions: 76

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, DevOps Engineering Introduction	13
Topic 2, DevOps Engineering Practices	27
Topic 3, DevOps Automation	11
Topic 4, DevOps Metrics	10
Topic 5, DevOps Engineering Architecture	11
Topic 6, DevOps Engineering Ecosystem	4
Total	76

QUESTION NO: 1

Which of the following situations DO NOT represent unplanned work eroding capacity and decreasing productivity?

A. A new piece or work was added to a protect after it was approved which required extra resources and time to complete resulting in other work being uncompleted.

B. A scheduled and approved change to update a series of servers was completed within the scheduled time with no reported incidents reported as a result.

C. A major incident was reported that required specific subject matter experts to focus on the incident and a project. They were working on was delayed as a result.

D. A recently implemented system suffered capacity-related performance and response issues which were not experienced in the pre-release testing phase.

ANSWER: B

Explanation:

“A scheduled and approved change to update a series of servers was completed within the scheduled time with no reported incidents reported as a result.” is the situation that does not represent unplanned work eroding capacity. The activity was known in advance, authorized through change control, and assigned a defined implementation window. It could therefore be incorporated into team planning, staffing, and capacity commitments before work began. Completing it within that window also means the change did not consume additional, unexpected engineering effort or displace planned delivery work.

In DevOps, productive flow depends on making work visible and deliberately allocating capacity among delivery, operations, maintenance, and improvement activities. Planned operational work is not automatically a productivity loss: it is an expected investment that can be scheduled, automated, measured, and improved. Unplanned work, by contrast, interrupts planned flow because people must react to unforeseen demand. The described server update has neither an unexpected interruption nor a resulting incident, so it is an example of controlled, planned operational work rather than capacity erosion. This distinction supports the DevOps emphasis on reliable flow and continual improvement described by PeopleCert’s DevOps Engineer certification and the change-enablement practice outlined in AXELOS guidance on change enablement.

QUESTION NO: 2

Which of the following is BEST described by "ensuring data is accessed only by an authorized person"?

A. integrity

B. Reliability

C. Availability

D. Confidentiality

ANSWER: D

Explanation:

Confidentiality is the security property concerned with preventing unauthorized disclosure of information. Ensuring that data can be accessed only by authorized people, services, or processes directly protects confidentiality: the organization establishes who is permitted to view or use information and applies controls to enforce that decision. Typical confidentiality controls include identity verification, authorization rules, least-privilege access assignments, role-based access control, encryption, and secure handling procedures. These measures limit exposure of sensitive information whether it is stored, transmitted, or processed.

This meaning is consistent with the widely used CIA security triad, in which confidentiality addresses protection from unauthorized access or disclosure. NIST defines confidentiality as preserving authorized restrictions on information access and disclosure, including protecting personal privacy and proprietary information. In a DevOps environment, confidentiality is supported by practices such as managing secrets securely, restricting pipeline and repository permissions, and ensuring production-data access is approved and auditable. See the NIST definition of confidentiality and the OWASP Access Control Cheat Sheet for practical access-control guidance.

QUESTION NO: 3

There are seven main areas of comparison between open source andcommercial software:

1. Infrastructure

2. Availability and reliability

3. Enhancement

4. Performance and scalability

5. Person-hours of the it team

6. Maintenance and support

7. _____

When of the following BEST describes the seventh missing item?

A. Security

B. Observability

C. Alerting

D. Flexibility

ANSWER: A

Explanation:

Security completes this comparison framework because the security model, exposure, assurance practices, and responsibility boundaries can differ materially between open-source and commercial software. A meaningful software-selection decision needs to assess how vulnerabilities are identified, reported, corrected, validated, and deployed, as well as the organization's ability to review code and manage dependencies. For open-source components, transparency can enable broad review and rapid community-led remediation, but the consuming organization must also maintain governance for component inventory, patching, provenance, and support arrangements. Commercial software may instead provide a vendor-managed security process, contractual commitments, advisory services, and defined update channels. In either delivery model, security remains a core operational and risk consideration alongside reliability, performance, enhancement, and support. This aligns with the DevOps focus on building security into the software delivery lifecycle rather than treating it as a separate late-stage activity. NIST's Secure Software Development Framework describes practices for preparing, protecting, producing, and responding throughout software development and maintenance: NIST SP 800-218. PeopleCert's DevOps certifications likewise emphasize practices that improve collaboration and delivery across the software lifecycle: PeopleCert DevOps certifications.

QUESTION NO: 4

Which orthe following commonly adoptee DevOps team approaches is the BEST example of a DevOpsteaming model that does not fully support the overall goals of DevOps?

A. DevOps as a tools team

B. DevOps advocacy team

C. A stream-aligned team

D. A site reliability team

ANSWER: A

Explanation:

DevOps as a tools team is the best example because it can turn DevOps into a centralized service responsible primarily for selecting, building, and operating automation platforms. Although effective tooling is important, DevOps is fundamentally a way of working that emphasizes shared responsibility, close collaboration, rapid feedback, and continual improvement across development, operations, security, and other delivery stakeholders. When a separate tools team becomes the owner of "DevOps," product teams may simply submit requests to it rather than developing the capabilities and shared operational ownership needed to deliver and run services effectively. This recreates a handoff and queue between teams, reducing autonomy and slowing feedback—the opposite of the fast flow DevOps seeks to enable. Tooling should therefore be treated as an enabling capability that supports teams' delivery and operational work, not as the organizational definition of DevOps. This aligns with the Team Topologies principle that platform capabilities should reduce cognitive load while allowing stream-aligned teams to deliver value independently. See Team Topologies: key concepts and PeopleCert's DevOps certification portfolio.

QUESTION NO: 5

Which of the following BEST describes why Agile teams aim for a work pace that they would be awe to sustain indefinitely?

A. Teams are generally more creative and energized when working a sustainable pace of about 40 hours per week.

B. Overtime lends lo mash schedule management or quality deficiencies

C. An Agile mindset insists on principles and a manifesto with lofty ideals.

D. Committing to more work means less tone to spend playing games

ANSWER: A

Explanation:

Teams are generally more creative and energized when working a sustainable pace of about 40 hours per week. This reflects the Agile principle that work should proceed at a rate that can be maintained indefinitely, rather than relying on repeated extended hours to meet commitments. A sustainable pace protects the team's ability to think clearly, collaborate effectively, solve problems, and continue delivering valuable increments over time. It also recognizes that software development is knowledge work: consistent focus, learning, and sound technical decisions are more valuable than short bursts of unsustainable effort. The reference to roughly 40 hours is not a universal fixed limit; it expresses the normal expectation that teams should plan work within a healthy, repeatable working rhythm. When a team can maintain its pace from iteration to iteration, forecasts become more dependable and delivery remains resilient as priorities evolve. The Agile Manifesto explicitly states that sponsors, developers, and users should be able to maintain a constant pace indefinitely. See the Agile Manifesto principles and the Agile Alliance description of sustainable pace.

QUESTION NO: 6

Which of the following BEST describes the DevOps release strategy mat manes use of a launch in which a new feature is initially released to a small targeted group of users for testing under production conditions?

A. Canary release

B. Dark-horse release

C. Blue/green release

D. Trial release

ANSWER: A

Explanation:

Canary release is the correct term for a deployment strategy in which a new feature or version is exposed first to a limited, targeted subset of real users in the production environment. Teams monitor the canary population closely through telemetry such as error rates, latency, availability, resource consumption, and user-behaviour or business metrics. If the release

performs as expected, the team progressively expands exposure to more users. If monitoring identifies an issue, exposure can be stopped or rolled back while the impact remains limited to the initial group.

This approach supports DevOps objectives by enabling rapid delivery while managing operational risk with evidence gathered under genuine production conditions. It also allows validation of assumptions that cannot always be fully tested in pre-production environments, including real traffic patterns, integrations, scale characteristics, and user response. Progressive rollout and automated monitoring are therefore core practices commonly associated with canary deployments. Google's Site Reliability Engineering guidance describes canarying as pushing a change to a small fraction of production before wider rollout, while Microsoft documents progressive exposure and evaluation of a new version through canary deployment. See Google SRE: Canarying Releases and Microsoft Azure Architecture Center: Canary releases.

QUESTION NO: 7

Which of the following BEST describes the purpose of a service level objective (SLO)?

A. A target value for a measurable level of service reliability or performance

B. A contractual penalty applied when a service is unavailable

C. An operational event that automatically creates an incident

D. A report showing the number of changes deployed by a team

ANSWER: A

Explanation:

A service level objective (SLO) is a target level of reliability or performance for a service over a defined period. It gives teams a measurable objective, such as a percentage of successful requests, availability, or latency threshold, against which they can assess whether users are receiving the expected service. SLOs help teams make informed engineering and operational decisions by connecting technical measures to an agreed reliability target.

An SLO is not the same as a service level agreement, which is a formal commitment that may include consequences if its terms are not met. Nor is it simply an alert threshold or an internal operational metric with no defined target. Google SRE describes SLOs as target values or ranges for service-level indicators. See the Google SRE Workbook guidance on implementing SLOs.