

DUMPSBOSS.

SAFe DevOps Practitioner Exam (SDP 6.0)

Scrum SAFe-DevOps

Version Demo

Total Demo Questions: 9

Total Premium Questions: 96

Buy Premium PDF

<https://dumpsboss.co>

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, Introducing DevOps	14
Topic 2, Building a Culture of Shared Responsibility	8
Topic 3, Improving the Flow of Work	28
Topic 4, Delivering Continuously	26
Topic 5, Providing Fast Feedback	11
Topic 6, Recovering Quickly	9
Total	96

QUESTION NO: 1

Which two technical practices focus on Built-in Quality? (Choose two.)

Choose the correct option from below list

- A. Environment configuration
- B. Pair work
- C. Test-driven development
- D. Feature toggles
- E. Canary releases

ANSWER: B C

Explanation:

Pair work and **Test-driven development** are technical practices that directly support Built-in Quality in SAFe. Pair work brings two people together to develop and review work in real time. This continuous collaboration improves shared understanding, spreads technical knowledge, and helps teams identify quality concerns while the work is being created rather than after defects have escaped downstream. It also supports collective responsibility for code and solution quality.

Test-driven development is a disciplined approach in which developers express expected behavior as automated tests before or alongside implementation, then build and refactor the solution while keeping those tests passing. The resulting automated regression coverage gives rapid feedback on whether changes preserve intended behavior. This enables frequent integration and safe evolution of the codebase, both of which are central to maintaining quality throughout development. SAFe describes Built-in Quality as an essential practice requiring quality to be addressed continuously, rather than relying on inspection at the end of delivery. Its Built-in Quality guidance specifically includes technical practices such as pair work and test-first development. See the [SAFe Built-in Quality](#) article and the [Scrum.org overview of test-driven development](#).

QUESTION NO: 2

Gemba walks are an important competency in support of which activity of the DevOps Health Radar?

- A. Hypothesize
- B. Continuous Exploration
- C. Collaborate and Research
- D. Learn

ANSWER: C

Explanation:

Collaborate and Research is correct because Gemba walks bring product and delivery teams directly to the place where operational work occurs so they can observe real workflows, understand stakeholder needs, and identify improvement opportunities from firsthand evidence. This direct observation supports meaningful collaboration with the people who perform the work and provides research-based insight into customer problems, operational constraints, and desired outcomes. Rather than relying solely on assumptions or reports, teams use the learning gained during a Gemba walk to validate understanding, clarify needs, and shape hypotheses about how to improve the flow of value. In the DevOps Health Radar, Collaborate and Research emphasizes alignment across stakeholders and the discovery of what should be improved or built. Gemba walks are therefore a practical technique for connecting development, operations, product, and business perspectives around actual value-stream conditions. The practice also reinforces SAFe's focus on customer centricity, continuous learning, and relentless improvement by ensuring decisions are informed by the real context in which value is created and delivered. See the [SAFe DevOps Health Radar](#) and SAFe guidance on [Design Thinking](#).

QUESTION NO: 3

Why is hypothesis evaluation important when analyzing data from monitoring systems in the Release on Demand aspect?

- A. It helps define the minimum viable product (MVP)
- B. It helps operations teams know where to apply emergency fixes
- C. It helps quickly create balanced scorecards for stakeholder review
- D. It helps link objective production data to the hypothesis being tested

ANSWER: D

Explanation:

It helps link objective production data to the hypothesis being tested. In SAFe's Release on Demand competency, teams use telemetry, monitoring, and other production data to determine whether a released solution is producing the intended customer and business outcomes. A hypothesis expresses an expected, measurable result—for example, that a capability will improve adoption, reduce task-completion time, or increase conversion. Evaluation compares that expected result with objective evidence collected from the live system. This creates a feedback loop based on real user behavior and solution performance rather than assumptions or opinions. Teams can then learn whether the hypothesis is supported, decide what to adjust, and use those insights to guide subsequent releases and investment decisions. This practice aligns with SAFe's emphasis on measuring outcomes and continuously improving the value delivered to customers. Monitoring data is especially valuable because it provides timely, observable evidence from the production environment, enabling teams to validate learning after releasing functionality on demand. See [SAFe Release on Demand](#) and [SAFe Innovation and Planning Iteration](#).

QUESTION NO: 4

A team can deploy an application successfully, but each environment requires operations specialists to manually create infrastructure and adjust configuration values. Which two improvements would best support Continuous Deployment? (Choose two.)

- A. Infrastructure as code
- B. Automated environment provisioning
- C. Larger batches of Features for each deployment
- D. A scheduled operations window for every deployment
- E. Environment validation after customer release

ANSWER: A B

Explanation:

Infrastructure as code and **automated environment provisioning** are correct because they make the infrastructure and configuration needed for deployment repeatable, versioned, and less dependent on manual handoffs. This supports the ability to deploy a validated solution reliably across environments and reduces configuration drift.

A larger release batch increases deployment risk rather than improving deployment flow. Waiting for a scheduled operations window preserves the manual dependency. Deferring environment validation until after release makes failures more likely to be discovered late.

QUESTION NO: 5

Which statement is true about DevOps?

- A. It enables low-risk releases and fast recovery with fast fix-forward
- B. It enables low-risk releases and fast recovery with no room for errors
- C. It enables high-risk releases and fast recovery with fast fix-forward
- D. It enables a tolerance for low-risk, low-failure, and rapid recovery

ANSWER: A

Explanation:

DevOps enables low-risk releases and fast recovery with fast fix-forward because it combines a collaborative culture with practices that make delivery and operations more reliable, repeatable, and responsive. Frequent delivery of small, tested changes reduces the scope and uncertainty of each release, helping teams manage risk instead of accumulating large, difficult-to-deploy batches of work. Automation in build, test, deployment, monitoring, and operational workflows provides rapid feedback and supports dependable releases.

Fast recovery is a central DevOps outcome. Teams use observability, alerting, incident response, and deployment practices to identify production issues quickly and restore service. Fast fix-forward means correcting a defect by safely delivering an updated change rather than depending solely on reverting to an earlier version. This preserves learning, shortens feedback loops, and allows the team to address the underlying issue while continuing to deliver value. SAE describes DevOps as the mindset, culture, and technical practices that support the flow of value from development through operations, with a focus on continuous delivery and release on demand. See [SAFe DevOps](#) and [SAFe Continuous Delivery Pipeline](#).

QUESTION NO: 6

When does the Continuous Integration aspect of the Continuous Delivery Pipeline begin?

- A. When code is checked in to version control
- B. When a User Story has satisfied its definition of done
- C. When a Feature has been pulled for work
- D. When the Team Backlog has been refined

ANSWER: C

Explanation:

When a Feature has been pulled for work is correct because SAE frames Continuous Integration as the pipeline activity that turns a feature hypothesis into a validated, integrated solution increment. The work starts when the Agile Release Train commits to and pulls a Feature from the Program Backlog for implementation. Teams then decompose the Feature into stories, develop and test the work, integrate code frequently, and validate the resulting behavior in an integrated environment. The intent is not merely to compile individual changes, but to continuously build system-level confidence that the Feature works with the broader solution.

Within SAE's Continuous Delivery Pipeline, Continuous Integration provides the technical and process practices needed to build quality into the solution as Feature work proceeds. It includes ongoing development, automated testing, integration, and system-level validation, culminating in a working, integrated increment that can support objective evaluation in the System Demo. Beginning at Feature pull also connects the technical CI flow to the ART's value-delivery flow: the Feature supplies the business context and acceptance intent that guide the stories and integration work. SAE describes the pipeline and its CI activities in the [Continuous Delivery Pipeline](#) guidance and further explains the role of [Continuous Integration](#) in producing integrated, high-quality increments.

QUESTION NO: 7

A production incident requires developers, operations specialists, and security personnel to coordinate a recovery. The organization wants to strengthen the Culture element of CALMR so that future incidents are resolved with less delay. Which two changes would best support this goal? (Choose two.)

- A. Establish shared responsibility for production outcomes
- B. Create common objectives across development, operations, and security
- C. Separate incident ownership by functional department
- D. Measure each group only against local targets
- E. Require a single management function to approve every response decision

ANSWER: A B

Explanation:

Establish shared responsibility for production outcomes and **create common objectives across development, operations, and security** are correct. CALMR culture emphasizes collaboration across the value stream, shared ownership, and alignment around customer value and operational outcomes. These changes reduce functional handoffs and make it easier for people to swarm around a production issue.

Separating incident ownership by department reinforces silos. Measuring groups only against local targets can create conflicting behavior, such as prioritizing feature throughput over service health. Requiring every production decision to pass through a single management function adds delay rather than enabling collaboration.

QUESTION NO: 8

Which two areas should be monitored in the Release on Demand aspect to support DevOps and Continuous Delivery? (Choose two.)

- A. Full-stack system behavior
- B. Build status
- C. Agile Release Train velocity
- D. Deployment cycle time
- E. Business Metrics

ANSWER: A E

Explanation:

Full-stack system behavior and **Business Metrics** are the key monitoring areas for Release on Demand. Release on Demand concerns making functionality available to customers in a way that aligns with business timing and demand, while ensuring that the live solution remains healthy. Monitoring full-stack system behavior provides operational feedback across the end-to-end solution, including application behavior, infrastructure, integrations, reliability, performance, and user-facing service health. This feedback enables teams to identify production conditions quickly and make informed decisions about operating, stabilizing, and evolving the solution.

Business Metrics provide the complementary outcome-focused feedback. They help the enterprise assess whether released capabilities are creating the intended customer and business value, such as improved adoption, conversion, revenue, customer satisfaction, or reduced operational cost. Together, these measures connect technical operational health with real-world value realization. This supports the SAFe principle of measuring outcomes, learning from production usage, and adapting release decisions based on evidence rather than merely confirming that a deployment occurred. SAFe describes Release on Demand as the ability to release value to customers all at once or incrementally, based on market and business needs. See [SAFe Continuous Delivery Pipeline](#) and [SAFe Measure and Grow](#).

QUESTION NO: 9

In which activity are specific improvements to the Continuous Delivery Pipeline identified?

- A. Impact mapping
- B. Value Stream identification
- C. Problem-Solving Workshop
- D. Future-state Value Stream Mapping

ANSWER: D

Explanation:

Future-state Value Stream Mapping is the activity used to identify specific improvements to the Continuous Delivery Pipeline. In SAFe, teams first make the flow of work visible by examining how an idea moves from concept through development, deployment, release, and operation. A future-state map then describes the desired flow: it highlights improvements that will reduce delays, handoffs, bottlenecks, excessive work in process, and other sources of waste. This makes improvement opportunities concrete and actionable rather than leaving them as general aspirations to “improve delivery.” Teams can use the mapped future state to agree on prioritized changes, assign ownership, and measure whether pipeline flow and value delivery improve over time. The activity supports SAFe’s DevOps mindset of shared responsibility across the whole value stream, including Development, Operations, Security, and other relevant stakeholders. By defining a better target state for the end-to-end system, Future-state Value Stream Mapping guides continuous improvement investments in the pipeline and enables faster, more reliable delivery of customer value. SAFe describes value stream mapping as a Lean method for visualizing the steps required to deliver value and identifying opportunities to improve flow. See [SAFe: Value Stream Mapping](#) and [SAFe: Continuous Delivery Pipeline](#).