

ARCHITECT JOURNEY – DEVELOPMENT LIFECYCLE AND DEPLOYMENT PRACTICE EXAM (SAMPLE)

STUDY GUIDE



SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

VISIT US ONLINE FOR
THE FULL VERSION STUDY GUIDE

<https://architectdevlifecycledeployment.examzify.com>



Copyright © 2026 by Examzify - A Kaluba Technologies Inc. product.

ALL RIGHTS RESERVED.

No part of this book may be reproduced or transferred in any form or by any means, graphic, electronic, or mechanical, including photocopying, recording, web distribution, taping, or by any information storage retrieval system, without the written permission of the author.

Notice: Examzify makes every reasonable effort to obtain accurate, complete, and timely information about this product from reliable sources.

SAMPLE

Table of Contents

Copyright	1
Table of Contents	2
Introduction	3
How to Use This Guide	4
Questions	5
Answers	8
Explanations	10
Next Steps	15

SAMPLE

Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

Questions

SAMPLE

- 1. High-performing agile teams are built on what foundation?**
 - A. Trust and sound engineering fundamentals
 - B. Highly specialized team members
 - C. Strong project managers
 - D. Ongoing shifting of developers between teams
- 2. What are the three key stakeholder groups in a governance model?**
 - A. Information technology
 - B. End users
 - C. Information technology; Business units; End users
 - D. Business units - own project vision and strategy
- 3. Which statement describes a local repository?**
 - A. A repository stored on the user's computer
 - B. A copy of the repository hosted on GitHub
 - C. A temporary workspace in the cloud
 - D. A remote mirror of another repo
- 4. When using the package development model, which metadata changes do you need to track manually?**
 - A. Changes To Components Not Yet Supported By Source Tracking
 - B. Only Changes Made Via The Setup UI
 - C. All Metadata Changes For The Release
 - D. None. All The Changes Are Tracked Automatically
- 5. What is the first step in the Salesforce DX app creation workflow?**
 - A. Set up your project.
 - B. Authorize the Developer Hub org for the project.
 - C. Configure your local project.
 - D. Create a scratch org.
- 6. Which API capabilities does the Salesforce CLI provide?**
 - A. Tooling API
 - B. Metadata API
 - C. Data (SOAP) API
 - D. ANT migration tool scripting

- 7. To adopt unlocked packages, which development approach must be adopted?**
- A. Modular development
 - B. Monolithic development
 - C. Test-driven development
 - D. Waterfall development
- 8. Which trio forms the core Scrum team and what is each role primarily responsible for?**
- A. The team does not coordinate with the ScrumMaster
 - B. The product owner handles implementation
 - C. The team does the work, the ScrumMaster helps facilitate, and the product owner defines the direction for the team
 - D. The ScrumMaster approves backlog items
- 9. Which sequence correctly lists the ALM steps in order?**
- A. Plan Release - requirements, design, environments; Develop; Test; Build Release - single release artifact; Test Release - test in staging environment; Release
 - B. Plan Release; Develop; Build Release; Test Release; Release
 - C. Plan Release; Develop; Test; Build Release; Test Release; Release
 - D. Plan; Release; Develop; Test; Release
- 10. Which of the following are responsibilities of business teams/units?**
- A. Gathering end-user feedback and onboarding users
 - B. Owning and managing the budget
 - C. All of the above
 - D. Designating a product owner

Answers

SAMPLE

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

SAMPLE

Explanations

SAMPLE

1. High-performing agile teams are built on what foundation?

A. Trust and sound engineering fundamentals

B. Highly specialized team members

C. Strong project managers

D. Ongoing shifting of developers between teams

High-performing agile teams rest on trust and solid engineering fundamentals. Trust creates psychological safety, which lets team members speak up, share ideas, and admit mistakes without fear. When that open, collaborative atmosphere exists, communication patterns improve, decisions happen faster, and the team can adapt more readily to change. Pair that with strong engineering practices—continuous integration, automated testing, clean architectures, maintainable code, and shared standards—and the technical work stays aligned, quality stays high, and changes can be delivered reliably and safely. The other options don't fit as well because they either promote silos or reliance on a single leader. Highly specialized team members can hinder cross-functionality, which Agile teams rely on to deliver end-to-end value. A strong project manager fits some environments, but Agile emphasizes collaborative, self-organizing teams rather than relying on a single manager. Frequently shifting developers between teams erodes the shared mental model and trust needed for smooth collaboration, reducing velocity and cohesion.

2. What are the three key stakeholder groups in a governance model?

A. Information technology

B. End users

C. Information technology; Business units; End users

D. Business units - own project vision and strategy

In governance models, three groups must be represented to balance technology, business goals, and real-world use: information technology, the business units that own the processes and requirements, and the end users who will interact with the system. IT brings feasibility, security, and technical standards; business units ensure alignment with strategy, regulations, and operational needs; end users provide practical input on usability, workflows, and adoption. Together, these perspectives ensure decisions consider technical feasibility, business value, and user impact. Focusing only on IT misses the business goals and user experience. Focusing only on end users misses governance, risk, and strategic direction. Focusing only on business units misses technical governance and frontline feedback.

3. Which statement describes a local repository?

A. A repository stored on the user's computer

B. A copy of the repository hosted on GitHub

C. A temporary workspace in the cloud

D. A remote mirror of another repo

A local repository is the version-control repository that resides on your own computer, containing the project files plus the version history in the hidden .git directory. This is where you can commit, branch, and inspect history offline, without needing a network connection. It becomes local when you clone a repository or initialize one on your machine. In contrast, a repository hosted on GitHub is a remote repository stored on a server, not on your computer; a temporary cloud workspace isn't the repository itself; and a remote mirror is another copy hosted elsewhere, not the one you have on your machine. So the statement that describes a local repository is that it is stored on the user's computer.

4. When using the package development model, which metadata changes do you need to track manually?

- A. Changes To Components Not Yet Supported By Source Tracking**
- B. Only Changes Made Via The Setup UI
- C. All Metadata Changes For The Release
- D. None. All The Changes Are Tracked Automatically

Not all metadata is captured by source tracking in the package development model. You must manually track changes to components that aren't yet supported by source tracking to ensure they're included in the release and can be reproduced in other environments. Source tracking handles many types automatically, but gaps remain for certain metadata types, so manual tracking fills those gaps. Changes made only via the Setup UI aren't the whole story, and you can't assume every metadata change is tracked automatically since some types aren't supported yet. Manual tracking is required specifically for those unsupported components.

5. What is the first step in the Salesforce DX app creation workflow?

- A. Set up your project.**
- B. Authorize the Developer Hub org for the project.
- C. Configure your local project.
- D. Create a scratch org.

Starting with a Salesforce DX app workflow, the very first step is to set up your DX project. This creates the project skeleton—the folder structure, the sfdx-project.json configuration, and the places where metadata and scratch org definitions will live. Without this workspace in place, there's no centralized context to organize source, track changes, or manage scratch orgs and packages. Once the project exists, you can authorize the Dev Hub to enable scratch org management, then configure the local project settings (like paths and package directories), and finally create a scratch org to begin development. Each later step relies on having the project scaffold, so skipping the setup would leave you with no coherent workspace to operate in.

6. Which API capabilities does the Salesforce CLI provide?

- A. Tooling API
- B. Metadata API**
- C. Data (SOAP) API
- D. ANT migration tool scripting

The Salesforce CLI is designed around managing metadata for deployment and source-driven development, and it uses the Metadata API to perform those operations. When you push or pull metadata with the CLI, you're working with components like custom objects, fields, layouts, Apex classes, validation rules, and other metadata types, all handled through the Metadata API under the hood. The Tooling API exists for certain development tooling and real-time metadata interactions, but it isn't the primary mechanism the CLI relies on for source-based workflows. The Data (SOAP) API deals with runtime data, not metadata movement, so it isn't what the CLI uses for managing deployments. ANT migration tool scripting is an older, separate approach that used the Metadata API via Ant; the CLI provides a newer, command-based interface focused on metadata management rather than Ant scripting.

7. To adopt unlocked packages, which development approach must be adopted?

- A. Modular development**
- B. Monolithic development
- C. Test-driven development
- D. Waterfall development

Unlocked packages rely on modular development. They're designed so the application is split into small, self-contained packages that can be versioned, maintained, and deployed independently. This modular approach lets teams package a feature or component on its own, manage dependencies between packages, and push updates without rewriting or redeploying the entire application. It also supports reuse of components across multiple orgs and enables more granular continuous delivery. Monolithic development bundles everything into one big package, which makes independent updates and reuse much harder. Test-driven development focuses on testing strategies rather than how the code is packaged, and waterfall development describes a linear process rather than a packaging or delivery approach. Thus, to adopt unlocked packages effectively, a modular development approach is the best fit.

8. Which trio forms the core Scrum team and what is each role primarily responsible for?

- A. The team does not coordinate with the ScrumMaster
- B. The product owner handles implementation
- C. The team does the work, the ScrumMaster helps facilitate, and the product owner defines the direction for the team**
- D. The ScrumMaster approves backlog items

In Scrum, the core trio that forms the Scrum Team is the group that actually builds the product, the facilitator who keeps the process healthy, and the person responsible for guiding what to build and in what order. The description in the answer matches this shape: the team does the work, the Scrum Master helps facilitate, and the Product Owner defines the direction for the team. Think of it this way: the Developers are the cross-functional group who turn ideas into increments every Sprint. The Product Owner owns the Product Backlog, clarifies requirements, and prioritizes work to maximize value, providing the direction the team follows. The Scrum Master acts as a servant-leader who helps the team adopt and follow Scrum, removes impediments, and shields the team from external distractions, ensuring the process runs smoothly. This arrangement is essential because it keeps clear responsibilities: backlog decisions and value come from the Product Owner, execution comes from the Developers, and process health comes from the Scrum Master. Other options imply means of control or activity that don't align with Scrum roles—e.g., the Scrum Master approving backlog items or the Product Owner doing the implementation—so they miss the intended balance of accountability and collaboration.

9. Which sequence correctly lists the ALM steps in order?

- A. Plan Release - requirements, design, environments; Develop; Test; Build Release - single release artifact; Test Release - test in staging environment; Release
- B. Plan Release; Develop; Build Release; Test Release; Release
- C. Plan Release; Develop; Test; Build Release; Test Release; Release**
- D. Plan; Release; Develop; Test; Release

Understanding ALM flow starts with planning the release, then turning plans into code, validating that work through testing, packaging a single release artifact, testing that packaged release in a staging-like environment, and finally releasing to users. The sequence here follows that path: Plan Release establishes what will be built and tested. Develop implements the planned features. Test during development checks that the new code works and meets requirements, catching defects early when they're cheaper to fix. Build Release then creates the single, packaged artifact that will be shipped, ensuring you're packaging something that has already been tested at the code level. Test Release validates the packaged artifact in an environment that simulates production, catching issues that only appear once packaging and deployment are involved. Release completes the process by delivering the validated product to users. This ordering emphasizes early quality checks during development and a final, environment-appropriate validation before deployment, which helps reduce risk and rework. Other sequences either delay testing until after packaging or mix steps in ways that don't align with how feedback and validation typically occur across development and release stages.

10. Which of the following are responsibilities of business teams/units?

- A. Gathering end-user feedback and onboarding users
- B. Owning and managing the budget
- C. All of the above**
- D. Designating a product owner

The main idea here is that business teams/units oversee activities that connect user needs to how the product is funded and governed. Gathering end-user feedback and onboarding users happens because real-world input and quick adoption drive a product's success; feedback informs what to build next and onboarding helps users start delivering value quickly. Owning and managing the budget is another clear responsibility, ensuring resources align with business goals, controlling costs, and prioritizing work based on ROI. Designating a product owner is also a business responsibility because this person represents business priorities to the development team, prioritizes the backlog, and ensures the team delivers what matters most to stakeholders. Taken together, these aspects cover the spectrum of business involvement in planning, funding, and guiding the product. That's why all of the above is the best answer.

Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

<https://architectdevlifecycledeployment.examzify.com>

We wish you the very best on your exam journey. You've got this!

SAMPLE