Architecture Specialization

Sample Exam

Before Starting

This sample exam has 10 questions that will help you get ready for the Architecture Specialization exam.

We recommend that you prepare a real exam environment, as much as possible.

- Book a quiet room just for you.
- Print this document, apart from the last page.
- Get a stopwatch or set a timer for the (recommended) duration of 40 minutes.

The last page of this document has the correct answers. Don't peek! Use it only after completing your exam, to check how well you did.

During the Sample Exam

To accurately simulate the real exam environment, we suggest that you:

- Read each question and its answers carefully.
- Take your time! Questions may be revisited and your choices can be changed.
- Mark the questions that you want to review at the end.
- Pick only one answer per question, as only one is correct.
- Answer all questions, as there's no benefit in not doing so.
- Try turning off all electronic devices during the exam.
- Refrain from using or reading any external materials during the exam.

After Completing the Sample Exam

After completing the exam, validate the answers you selected by checking the ones provided on the last page of this document, and count the total number of correct answers. Since the passing score is 70% or higher, you should get at least 7 questions right. In case you chose any wrong answers, we suggest you review the study materials where that specific topic is covered.

Sample Exam Questionsfindi

1. Which of the following options describes a benefit of adopting the Architecture Canvas?

- ☐ **A**. The Canvas provides a systematic approach to architecture design, supported by a validation tool.
- ☐ **B.** The Canvas provides an automatic way to follow and fix architecture principles.
- ☐ **C.** The Canvas provides a way to promote the collaboration and understanding of the business users easily.
- ☐ **D.** The Canvas provides faster architecture design without validation needs.

---

2. In OutSystems, is it recommended to join unrelated concepts in the same module?

- ☐ **A.** Yes, because it reduces the number of references and simplifies deployments.
- ☐ **B.** No, because it prevents the lifecycle independence of concepts and adds unnecessary impacts to consumers.
- ☐ **C.** Yes, to avoid cyclic references and reduce the footprint of generated code.
- ☐ **D.** No, because it adds unnecessary complexity to consumers requiring both concepts.

---

3. Consider the following statement: *"In OutSystems 11, it is recommended to use business-related Screens in Foundation layer modules, since it does not break any Architecture validation rule."* Which of the following options is **correct**?

- ☐ **A.** The sentence is true for all business-related Screens. All references to Screens are weak, so they can and should be used in Foundation Layer modules, without causing upward references.
- ☐ **B.** The sentence is false. Although all references to Screens are weak, only non-business-related Screens should be used in Foundation layer modules.
- ☐ **C.** The sentence is true for business-related Web Screens. References to Mobile Screens are still considered strong references, so they should not be used in Foundation Layer modules.
- ☐ **D.** The sentence is false. Although all references to screens are weak, only business-related Screens that are not being consumed by other modules should be used in Foundation layer modules.

---

4. Consider that we have a weak side reference between End-user modules. What is the best way to remove that reference to avoid an architecture finding that goes against the recommended best practices?

- ☐ **A.** Replace Screen destinations with external URLs and consume Actions through REST APIs.
- ☐ **B.** Weak side references between End-user modules are allowed, so we do not need to remove that reference.
- ☐ **C.** In these scenarios, always move the consumed elements from the producer module to the consumer module to avoid the reference.
- ☐ **D.** Identify the consumed elements, excluding destinations, and move them to reusable Core/Foundation modules.

---

5. Which option describes a valid reason to create a Calculation Engine *(_Eng)* module?

- ☐ **A.** To abstract different complex logic implemented in several Business Logic modules.
- ☐ **B.** To support a concept with entities and CRUD edition.
- ☐ **C.** To support complex calculations (e.g. an insurance simulator).
- ☐ **D.** To support reusable Actions from Core Widget modules.

---

6. Why is it so important to adopt a naming convention in our applications?

- ☐ **A.** To reveal the nature of each module.
- ☐ **B.** To enforce the reference architecture.
- ☐ **C.** To normalize patterns.
- ☐ **D.** All the remaining options are correct!

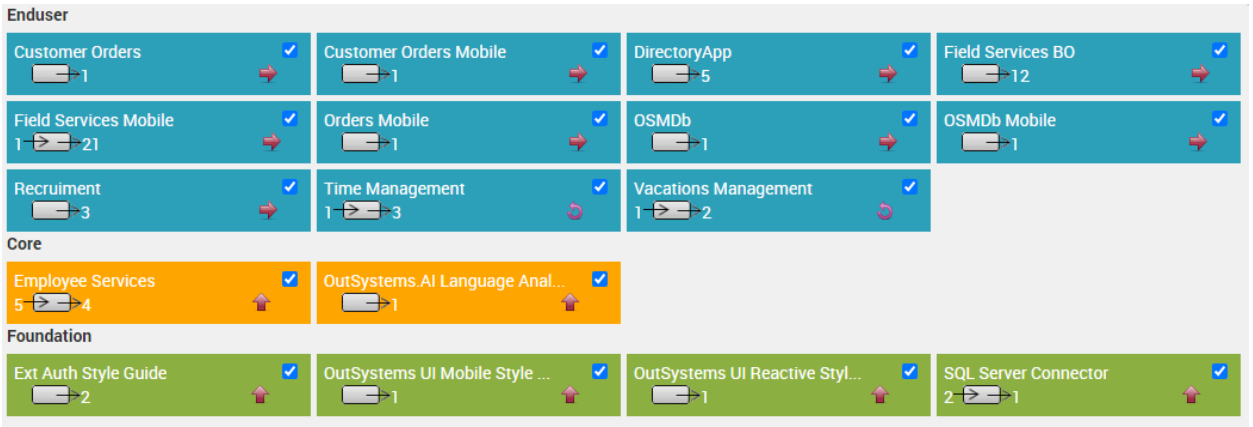
---

7. When moving elements between modules in OutSystems, to which elements should we pay extra attention and act more carefully?

- ☐ **A.** The elements with no data persistency, like Actions, Structures, or Blocks. Since these elements are not persistent, they can be deleted unintentionally.
- ☐ **B.** The elements with data persistency, like Entities. When moving an Entity, only its logical definition is moved. Physically, it is another database table.
- ☐ **C.** There is no element in particular. All elements moved between modules maintain their integrity.
- ☐ **D.** Only the configurable elements, like Timers, Site Properties, or Roles. These elements are the only ones that have data persistency that differs from Development to Production.

---

8. Consider the following Discovery screenshot and its findings. Which of the following findings, if any, should be solved first?



- ☐ **A.** The upward references between Core Services modules and End-user modules.
- ☐ **B.** The upward references between Foundation modules and Core Services modules.
- ☐ **C.** The references between End-user modules.
- ☐ **D.** The order to solve the architecture findings is not relevant.

---

9. Why is it so important to have a multi-layer framework to support Architecture Design?

- ☐ **A.** To promote a correct abstraction of reusable services.
- ☐ **B.** To optimize lifecycle independence.
- ☐ **C.** To minimize the impact of changes.
- ☐ **D.** All the remaining options are correct!

---

10. In which of the following scenarios should a custom Style Guide be implemented?

- ☐ **A.** Whenever applications cannot benefit from any existing theme. The Style Guide should be implemented by cloning the most complex template available, to leverage the Blocks and Actions from the original template and accelerate developments.
- ☐ **B.** Whenever applications cannot benefit from any existing theme. The Style Guide should be implemented by starting from the most basic template possible since it only contains the minimum required structure to start.
- ☐ **C.** Whenever an application needs to have independent deployments and low-level of interdependence among other applications.
- ☐ **D.** Always. Every project/application should have its custom Style Guide.

Answers

1. A

2. B

3. B

4. B

5. C

6. D

7. B

8. C

9. D

10. B