

Boost up Your Certification Score

SAP C_S4CPR_2608

**SAP Certified - Implementation Consultant - SAP S/4HANA
Cloud Public Edition**



For More Information – Visit link below:

<https://www.examsboost.com/>

Product Version

- ✓ Up to Date products, reliable and verified.
- ✓ Questions and Answers in PDF Format.

Latest Version: 4.0

1. Unified Scenario Simulation
2. Micro Skill Drill Exam

Topic: 1

Unified Scenario Simulation

Hollandsworth Industrial Procurement Extensibility Build

Business Context

Hollandsworth Machine Works is a global manufacturer of heavy compaction and material-handling equipment, with assembly sites across three continents and a centralized procurement organization serving all plants. The company is midway through its adoption of SAP S/4HANA Cloud Public Edition for sourcing and procurement, having already completed fit-to-standard workshops and locked most of its business configuration. The procurement leadership sponsored the program on the promise that the new platform would stay current with continuous delivery and would not accumulate the modification debt that made their legacy environment expensive to upgrade.

System Landscape

The program runs across a multi-system cloud landscape: a development and configuration tier where business configuration and extensions are built, a quality tier used for integration testing and user acceptance, and the production tier that will go live at the next release window. Business configuration is authored centrally and moved forward through the tiers rather than changed directly downstream. The extensibility workstream has just begun, and the team is expected to deliver a set of procurement enhancements before the quality tier is frozen for user acceptance testing.

Requirements Intake

The procurement process owners have raised several enhancement requests during requisition and purchase order handling. Buyers want a field on the purchase order that records an internal equipment-programme code so spend can be traced back to a specific machine build. A category manager asked for a validation that blocks a requisition from being ordered when a supplier-provided compliance attribute is missing. A separate group in engineering wants a nightly reconciliation that pushes purchase order data into a supplier collaboration application the company already runs on SAP Business Technology Platform, and enriches it with data from a non-SAP logistics system.

Build Approach

An external partner joined the extensibility workstream and, eager to show progress, proposed handling all three requests the same way — as custom developments built against the digital core. The functional lead pushed back, noting that the programme code and the requisition validation appear to be things a business user could model without developer involvement, and that only the cross-application reconciliation clearly belongs somewhere else. The partner also suggested making one of the changes directly in the quality tier so the UAT group could see it immediately, since the change window in the development tier was tight.

Transport Observation

The team completed the custom field for the equipment-programme code in the development and configuration tier and confirmed it appeared on the purchase order there. When the UAT group opened the same purchase order app in the quality tier, the field was not present. A junior consultant assumed the field would replicate automatically because standard configuration had behaved that way earlier in the project, and had not collected the key-user extension into a transportable change before expecting it downstream.

Governance Context

The program office has a standing rule that nothing reaches production except by moving forward through the landscape with a recorded change, and that the digital core stays standard so the twice-yearly upgrades remain low-effort. The procurement director, under pressure to show buyers a working programme-code field before the next steering review, asked whether the team could simply keep the field maintained separately in each tier to save time. The clean-core commitment and the transport discipline are now in direct tension with the pressure to demonstrate quick wins.

Integration Dependency

The nightly reconciliation raises its own dependency. The supplier collaboration application on BTP needs purchase order changes as they happen, and the logistics enrichment comes from a system outside the SAP estate. The team debated whether this should be attempted as logic inside the procurement solution or as a separate application that consumes released procurement data through public interfaces. Whichever path is chosen has to survive the continuous-delivery upgrades without manual rework at each release.

Validation Expectation

Before any of these enhancements can move to user acceptance testing, the program office wants evidence that each was built at the correct extensibility tier, that the digital core was not modified, and that every change is captured in a transportable form and demonstrably present in the quality tier. The extensibility build phase is expected to close with a clean set of changes queued for the production release window, not a collection of tier-local edits that cannot be traced or promoted.

Decision Pressure

With the quality-tier freeze approaching, the team must decide how each request is realized, how the completed extensions travel through the landscape, and how to keep the reconciliation from becoming a core dependency that breaks at the next upgrade. Each decision carries a different exposure: the wrong tier choice erodes upgrade stability, and a mishandled transport leaves an enhancement stranded in one system while the program office believes it is ready to promote.

Your assignment

You are working as an implementation consultant on the Hollandsworth Machine Works extensibility workstream in SAP S/4HANA Cloud Public Edition. You are operating in a multi-system cloud landscape consisting of a development and configuration tier, a quality tier for integration testing and user acceptance, and a production tier. Business configuration is authored centrally and moved forward through the tiers. Your working context is the extensibility build phase, ahead of the quality-tier freeze for user acceptance testing. The digital core must remain standard so the platform stays upgrade-stable under continuous delivery

1. CHALLENGE 1 — Selecting the Correct Extensibility Tier for Each Procurement Request

- For the request to add an internal equipment-programme code field to the purchase order, determine the lowest sufficient extensibility tier that a business user can model without developer involvement.
- For the request to block a requisition from being ordered when a supplier compliance attribute is missing, determine whether custom business logic can be added through the same business-user extensibility capability.
- For each request, confirm whether standard configuration alone would satisfy it before selecting any extension.
- Distinguish the one request that genuinely requires an application outside the digital core from those that do not.

Checkpoints

- The equipment-programme field and the requisition validation are each realized at the lowest tier that meets the requirement, keeping the digital core standard.
- No request is escalated to developer-level or side-by-side extensibility when key-user in-app extensibility is sufficient.

- Any request satisfied by standard configuration is not built as an extension.

2. CHALLENGE 2 — Transporting Key-User Extensions Through the System Landscape

- After building the equipment-programme field in the development and configuration tier, collect the key-user extension into a transportable change.
- Move the change forward to the quality tier through the landscape rather than re-creating the field there.
- Verify the field is present and behaves identically in the quality tier before user acceptance testing begins.

Checkpoints

- The custom field appears in the quality tier as a result of a recorded promotion, not manual re-entry.
- The extension is confirmed present and functioning downstream, evidencing a completed transport.
- No extension intended for production remains stranded in the development and configuration tier.

3. CHALLENGE 3 — Positioning the Cross-Application Reconciliation Without a Core Dependency

- Determine where the nightly purchase order reconciliation with the BTP supplier collaboration application and the non-SAP logistics enrichment should run.
- Ensure the chosen approach consumes released procurement data through public interfaces rather than modifying the digital core.
- Confirm the approach will not require manual rework at each continuous-delivery upgrade.

Checkpoints

- The reconciliation runs as a decoupled application on the platform, not as logic embedded in the digital core.
- Integration uses released extension points and public interfaces set up through the appropriate communication configuration.
- The design is upgrade-stable and introduces no core modification.

4. CHALLENGE 4 — Keeping Configuration and Extension Changes Landscape-Governed

- Ensure every configuration and extension change originates in the configuration or development tier.
- Reject any approach that maintains a field or setting separately in each tier or edits the quality or production tier directly.
- Promote each change forward through the landscape as a single recorded change.

Checkpoints

- No tier-local edits exist that diverge from the promoted change set.
- Each change is traceable and demonstrably promoted rather than duplicated.
- The completed change set is queued for the production release window.

Question: 1

CHALLENGE 1 — Selecting the Correct Extensibility Tier for Each Procurement Request

Buyers want an internal equipment-programme code recorded on the purchase order so spend traces to a specific machine build. A business user should be able to model this without developer involvement, and the digital core must stay standard.

Which approach realizes this request at the lowest sufficient extensibility tier?

A. Commission a side-by-side application on the platform that stores the programme code and surfaces it on each purchase order.

B. Add the equipment-programme code as a custom field through key-user in-app extensibility, keeping the core standard.

- C. Have the partner add the programme-code field through a developer in-app change built against the digital core objects.
- D. Raise a modification request to extend the standard purchase order structure directly in the digital core objects.

Answer: B

Explanation:

A custom field a business user can model on a standard app is exactly what key-user in-app extensibility provides, and it keeps the digital core standard so upgrades stay low-effort. It is the lowest tier that satisfies a simple field-capture requirement, which is what the request describes.

Question: 2

CHALLENGE 1 — Selecting the Correct Extensibility Tier for Each Procurement Request

A category manager wants a requisition blocked from being ordered when a supplier-provided compliance attribute is missing. The team wants to add this rule without developer involvement and without touching the core.

Which approach best fits this request?

- A. Ask the partner to code the check as a developer in-app enhancement against the requisition object inside the core.
- B. Build a side-by-side application on the platform that intercepts requisitions and applies the compliance rule before ordering.
- C. Add custom business logic through key-user in-app extensibility so a business user enforces the check without core changes.
- D. Configure the standard requisition process so the ordering block is achieved without any extension at all being built.

Answer: C

Explanation:

Key-user in-app extensibility includes custom business logic that a business user can add to a released extension point, which is the right fit for a validation rule that must not modify the core. It keeps the check inside the solution and upgrade-stable.

Question: 3

CHALLENGE 1 — Selecting the Correct Extensibility Tier for Each Procurement Request

The partner proposed building all three enhancement requests the same way, as custom developments against the digital core. The functional lead wants each request routed to its correct tier instead.

What is the strongest reason to reject the partner's uniform approach?

- A. A uniform approach is acceptable if the partner documents each development so the changes can be re-applied after upgrades.
- B. Core developments build more slowly than side-by-side applications, so the requests should instead all move to the platform.
- C. Custom developments cannot record a purchase order field, so the programme-code request would fail on technical grounds.
- D. Most requests are satisfiable at a lower tier, so uniform core development over-scopes them and erodes upgrade stability.

Answer: D

Explanation:

The requests differ in what they need, and treating them uniformly as core developments both over-scopes the simple ones and modifies the core where released extensibility would keep it standard. Routing each to its lowest sufficient tier is what preserves upgrade stability.

Question: 4

CHALLENGE 1 — Selecting the Correct Extensibility Tier for Each Procurement Request

Of the three enhancement requests, the team must identify which one genuinely requires an application outside the digital core and which belong to business-user extensibility. Which classification is correct?

- A. All three requests belong outside the core as side-by-side applications, since each of them touches procurement data.
- B. The field and the validation are key-user in-app extensions; only the reconciliation is a side-by-side application.
- C. The programme-code field is standard configuration, and the other two are developer in-app extensions inside the core.
- D. The validation and the reconciliation are key-user in-app extensions, and only the programme-code field is side-by-side.

Answer: B

Explanation:

The field and the validation are point extensions a business user can model with key-user in-app extensibility, while the cross-application reconciliation spans a BTP application and a non-SAP system and belongs in a decoupled side-by-side application. This routes each request to its lowest sufficient tier and keeps the core standard.

Question: 5

CHALLENGE 1 — Selecting the Correct Extensibility Tier for Each Procurement Request

An external partner joined the extensibility workstream and wants to move quickly. The functional lead insists each of the three requests is routed to the lowest sufficient tier before any build begins. Which sequence reflects the correct decision order for classifying a request?

- A. Check standard configuration first, then key-user in-app, then developer in-app, and only then a side-by-side application.
- B. Default every request to a side-by-side application first, then step down to key-user or configuration only if needed.
- C. Begin with developer in-app extensions because they are the most flexible, then simplify to key-user only where possible.
- D. Route each request to whichever tier the partner can build fastest, then adjust the tier after the freeze if problems arise.

Answer: A

Explanation:

Classifying by the lowest sufficient tier means testing standard configuration first, then key-user in-app extensibility, then developer in-app, and only then side-by-side. This ordering keeps the core standard and avoids over-engineering, which is what preserves upgrade stability.

Question: 6

CHALLENGE 2 — Transporting Key-User Extensions Through the System Landscape

The equipment-programme field was built and confirmed on the purchase order in the development and configuration tier, but it does not appear in the quality tier. A junior consultant assumed it would replicate automatically.

What is the correct next action?

- A. Collect the key-user extension into a transportable change, move it forward to the quality tier, and confirm it there.
- B. Re-create the same custom field manually in the quality tier so user acceptance testing can start without any delay.
- C. Rebuild the field as a side-by-side application so it becomes visible across every tier without needing a transport.
- D. Wait for the next continuous-delivery upgrade, which will carry the field into the quality tier automatically for you.

Answer: A

Explanation:

Key-user extensions do not replicate on their own; they must be collected into a transportable change and promoted through the landscape. Doing so, then verifying presence in the quality tier, is what actually moves the field forward and evidences a completed transport.

Question: 7

CHALLENGE 2 — Transporting Key-User Extensions Through the System Landscape

The program office wants evidence that the equipment-programme field is ready to promote toward production. The team performed a transport of the key-user extension to the quality tier. Which observation confirms the transport was successful?

- A. The custom field still displays correctly in the development and configuration tier where it was originally built and confirmed.
- B. The custom field is present on the purchase order in the quality tier from the recorded promotion and behaves as upstream.
- C. A consultant re-entered the field definition in the quality tier by hand and it now matches the upstream tier exactly.
- D. The next upgrade window has been scheduled, after which the field is expected to appear in the downstream tier by itself.

Answer: B

Explanation:

A completed transport is evidenced by the extension being present and functioning in the target tier as a result of the recorded promotion. Confirming the field on the purchase order in the quality tier, matching upstream behavior, is the observable completion evidence the program office needs.

Question: 8

CHALLENGE 2 — Transporting Key-User Extensions Through the System Landscape

Two enhancements are complete in the development and configuration tier: the equipment-programme field and the requisition compliance logic. The quality-tier freeze for user acceptance testing is approaching.

What must happen before these extensions can enter user acceptance testing?

- A. The extensions can be tested in the development and configuration tier itself, since that is where they were built and confirmed.
- B. The extensions are left in the development tier and promoted straight to production once user acceptance testing signs off.
- C. Each completed extension is collected and promoted forward so it is verifiably present in the quality tier before the freeze.
- D. Only the field needs promoting; the requisition logic can be added directly in the quality tier during the testing window.

Answer: C

Explanation:

User acceptance testing runs in the quality tier, so every completed extension must be promoted forward and confirmed present there before the freeze. Promoting each as a recorded change is what makes them available and traceable for testing.

Question: 9

CHALLENGE 2 — Transporting Key-User Extensions Through the System Landscape

A completed key-user extension in the development and configuration tier must reach the quality tier before the freeze. A consultant is deciding how to make it available downstream.

Which action correctly moves the extension and preserves traceability?

- A. Export the extension definition and hand it to the quality-tier administrator to rebuild it identically by hand downstream.
- B. Collect the extension into a recorded transportable change, promote it forward, and verify it functions in the quality tier.
- C. Grant the user acceptance testing group access to the development and configuration tier so no move through the landscape is needed.
- D. Leave the extension in place and note it for direct manual entry in production at the coming release window instead.

Answer: B

Explanation:

Collecting the extension into a recorded transportable change and promoting it forward is the governed path that both moves the extension and keeps it traceable. Verifying it functions downstream confirms the promotion completed.

Question: 10

CHALLENGE 3 — Positioning the Cross-Application Reconciliation Without a Core Dependency

The nightly reconciliation must push purchase order changes into a supplier collaboration application already running on SAP Business Technology Platform and enrich them with data from a non-SAP logistics system.

Where should this reconciliation be positioned to stay upgrade-stable?

- A. As custom business logic added inside the procurement solution through key-user in-app extensibility so it runs near the data.
- B. As a developer in-app extension embedded in the digital core so the reconciliation reads the purchase orders directly there.
- C. As a modification to the standard purchase order process so the reconciliation is triggered automatically on each change.
- D. As a decoupled side-by-side application on the platform that consumes released procurement data and adds the non-SAP source.

Answer: D

Explanation:

A cross-application reconciliation that spans a BTP application and a non-SAP system belongs in a decoupled side-by-side application consuming released procurement data through public interfaces. That keeps the digital core standard and survives continuous-delivery upgrades without rework.

Thank You for Trying Our Product

For More Information – **Visit link below:**

<https://www.examsboost.com/>

15 USD Discount Coupon Code:

G74JA8UF

FEATURES

- ✓ **90 Days Free Updates**
- ✓ **Money Back Pass Guarantee**
- ✓ **Instant Download or Email Attachment**
- ✓ **24/7 Live Chat Support**
- ✓ **PDF file could be used at any Platform**
- ✓ **50,000 Happy Customer**



Visit us at: <https://www.examsboost.com/test/c-s4cpr-2608>