Up-to-date Practice Test with Latest Questions and Answers covering latest syllabus and topics of the exam. Makes you ready to face actual exam.

S2000-017 Practice Questions
S2000-017 Practice Test
S2000-017 Practice Exam
S2000-017 Exam Questions
S2000-017 Study Guide

killexams.com

IBM

S2000-017

IBM Engineering Test Management v7.x Specialist

ORDER FULL VERSION

https://killexams.com/pass4sure/exam-detail/S2000-017

Question: 870

A program has multiple streams for different releases. A single test case library is shared across streams. A test case is updated in Stream B (for Release 2) but should not be used in Stream A (Release 1). When generating a test execution report for Release 1, how do you ensure the report uses the correct version of the test case?

A. The reportable for Release 1 must be associated with the global configuration of Release 1, which specifies the correct version of all artifacts.
B. You must generate the report from within the Stream A workspace, and it will automatically use the version in that stream.
C. Manually select the historical version of each test case in the report parameters before running it.
D. The report always uses the latest version of the test case from the library, regardless of the stream.

Answer: A
Explanation: In a multi-stream development environment using Global Configuration Management (GCM), the definitive way to lock the version of artifacts for reporting is through a Global Configuration. For Release 1, a global configuration baseline is created that includes specific versions of requirements, test cases, etc. When you create a reportable for Release 1 reporting and set its "Configuration Context" to this Release 1 global configuration, all reports and widgets based on that reportable will resolve artifacts to the versions captured in that baseline. This ensures the Release 1 report uses the test case version valid for that release, not the latest version from Stream B. Generating from a workspace is a client-side action, not a report server configuration. Manually selecting versions is impractical.

Question: 871

In IBM ETM v7.x, a "Stream" is a development line where artifacts can be modified. A "Baseline" is a read-only snapshot. When a test team finishes a "Sprint 1" and moves to "Sprint 2," what is the IBM-recommended workflow for their ETM component? (Select two)

A. Create a baseline of the stream at the end of Sprint 1 to capture the final state of all test artifacts.
B. Delete all Sprint 1 artifacts to keep the stream clean for Sprint 2.
C. Continue working in the same stream for Sprint 2; the history tab will handle all versioning needs.
D. Create a new stream for Sprint 2 from the Sprint 1 baseline if parallel work on Sprint 1 bug fixes is required.

Answer: A,D

Explanation: Creating a baseline at the end of a sprint is a critical best practice for maintaining a historical record of the "Definition of Done" for that period. If the team needs to perform maintenance on Sprint 1 while simultaneously developing Sprint 2, creating a new stream from that baseline allows for isolated, parallel work. This ensures that Sprint 2 activities do not accidentally destabilize the validated Sprint 1 artifacts, while the baseline remains an immutable reference for future audits or regressions.

Question: 872

During the administration of a Project Area, you realize that several "Team Areas" need different "Custom Attributes" for their Test Results, but these should not be visible to other teams. What is the IBM-recommended approach for this scenario in ETM v7.x? (Select Two)

A. Use "Process Provider" relationships to share attributes while maintaining local "Team Area" overrides for specific attribute values.
B. Place each team in a separate "Project Area" because Custom Attributes are always global to a Project Area and cannot be hidden per Team Area.
C. Define the Custom Attributes at the Project Area level but use "Process Customization" to restrict visibility based on the Team Area context.
D. Implement "Categorization" for Test Results and use the "Category Viewer" role to filter attributes by Team.

Answer: B,C
Explanation: ETM architecture dictates that Custom Attributes are defined at the Project Area level; however, visibility and mandatory requirements can be controlled via Process Customization (Operation Behavior) based on the user's role in a specific Team Area. If absolute data isolation is required where teams must not even see the existence of certain attributes, the recommended approach is to move those teams into separate Project Areas, as attribute definitions are scoped to the Project Area.

Question: 873

Your organization is scaling agile and wants to integrate Jira for sprint management with ETM for rigorous, audited test management. The process requires that when a Jira story is moved to "Done," it automatically triggers a validation check in ETM to ensure all linked test cases are in an "Approved" state and have at least one "Passed" execution record in the current sprint's context. What is the primary enabling mechanism within the IBM ecosystem for implementing this kind of cross-tool, state-based policy enforcement?

A. The IBM Engineering Workflow Management (EWM) to serve as an intermediary between Jira and ETM.
B. A custom event listener in Jira that calls the ETM REST API to run a query.
C. The out-of-the-box synchronization of the "Status" field between Jira and ETM work items.

D. Lifecycle Project Process Specifications and Linked Data Provider (LDP) relationships within a shared Global Configuration, enabling calculated attributes and state validation rules.

Answer: D
Explanation: While connectors can sync data, complex policy enforcement across tools requires deeper integration. When Jira (or IBM Engineering Requirements Management DOORS Next) and ETM are used within the same Lifecycle Project and share a Global Configuration via Linked Data Provider (LDP) relationships, their artifacts are intimately linked. In this setup, you can define a *calculated attribute* on the Story work item type (in the requirements management side). This attribute's calculation can query the state and execution results of linked test cases in ETM. Furthermore, a *state validation rule* can be placed on the Jira story's "Done" state that checks this calculated attribute. If the test coverage condition is not met (e.g., a linked test is not "Approved" or hasn't passed), the validation rule prevents the transition to "Done," enforcing the policy directly within the lifecycle workflow.

Question: 874

An automation engineer is integrating a third-party testing tool using the OSLC (Open Services for Lifecycle Collaboration) Automation specification. The engineer needs to ensure that the external tool can update the 'Execution State' of a TCER in IBM ETM. Which two approaches are valid for managing these execution records via the API? (Select two)

A. Perform an HTTP PUT request to the TCER URL with an updated 'state' property in the XML representation.
B. Use the 'qm:executionResult' OSLC resource to post a new result that references the existing TCER.
C. Utilize the RQM URL Utility to POST an execution result XML to the 'workitem' feed.
D. Create a 'Small XML' payload that utilizes the 'com.ibm.rqm.planning.Version' namespace to update the result verdict.

Answer: A,B
Explanation: Updating the TCER state directly via an HTTP PUT request is the standard REST-based method for modifying execution record metadata. Using the OSLC 'executionResult' resource to post new results is the architecturally preferred method for third-party integrations, as it maintains the link between the external execution and the ETM execution record.

Question: 875

Firmware testing in an agile-hybrid automotive project reveals configuration drift in test schedules between iteration streams in ETM v7.x (Sprint 17: ECU Calibration Bench, CAN Bus Integration, OBD Diagnostics). To realign schedules precisely without extensive rework across environments, what configuration feature provides the IBM recommended approach?

A. Recreate schedule sections entirely within the master test plan template for redistribution.
B. Bulk-edit schedules through work item assignments targeting schedule owners.
C. Re-sort test plans using iteration-specific categories to reflect updated timelines.
D. Compare configurations across streams and deliver schedule changes selectively to target local streams.

Answer: D
Explanation: The Compare Configurations feature combined with targeted stream delivery in ETM v7.x identifies and propagates schedule realignments precisely across iteration environments, minimizing rework while supporting iterative hybrid development cadences.

Question: 876

When analyzing "Test Progress" in a multi-stream environment using Global Configurations, a user reports that the "Test Execution Status" widget is aggregating data from all streams, rather than the one currently selected in the Global Configuration picker. How can this be corrected? (Select two)

A. Update the LQE index to "Scoped" mode for the specific Project Area
B. Restart the "Reportable REST API" service to clear the global context cache
C. Ensure the "Use Global Configuration" checkbox is enabled in the Dashboard settings
D. Change the widget preference to "Current Configuration" instead of "All Data"

Answer: C,D
Explanation: Dashboard widgets must be told to respect the user's current configuration context. This is done by enabling the Global Configuration support in the dashboard and ensuring the widget itself is set to use the "Current Configuration" . If these are not set, the widget defaults to a project-wide query. LQE indexing and API restarts do not control the UI-level context filtering.

Question: 877

A geographically distributed team executes tests under shared configurations. Time zone differences cause progress snapshots to appear inconsistent when viewed from different regions. Which dashboard setting standardizes visibility?

A. Enable “Use server time zone for all progress calculations and snapshots” in global preferences
B. Rely on live status instead of saved snapshots
C. Schedule progress snapshots at a fixed UTC time daily
D. Allow per-user time zone adjustment in dashboard preferences

Answer: A
Explanation: Setting the global preference to use server time zone consistently for progress calculations,

snapshots, and dashboard rendering eliminates time-zone-induced discrepancies across distributed teams. This ensures all stakeholders view identical progress metrics per configuration regardless of location, supporting reliable release decisions and status communication.

Question: 878

A project uses multiple team areas for different application modules. A global "Test Data Version" enumeration (e.g., V1.0, V1.1, V2.0) must be applied to Test Cases and Test Plans across all team areas. The allowed values for this enumeration must be controlled centrally but the attribute itself needs to be editable by Testers in their respective team areas. How should this be implemented?

A. Create the enumeration and custom attribute in the project area's process, then share the process configuration to all team areas.
B. Create a global enumeration in the Jazz application, then create a linked attribute in each team area's Test Case and Test Plan artifact types.
C. Create the enumeration and custom attribute in the master process template, then deploy it to all project and team areas.
D. Create the enumeration in the project area, then create a team-area-specific custom attribute in each team area that references the project-level enumeration.

Answer: A
Explanation: In ETM, enumerations and custom attributes are defined at the process level. To have a centrally controlled set of values (enumeration) used by an attribute across team areas, you must define both the enumeration and the custom attribute within the project area's process. When you create a team area, it inherits the process configuration from its parent project area by default. By sharing the process configuration (or using a common process template), all child team areas will have access to the same enumeration definition and attribute. Permissions for editing the attribute's value on individual work items are then managed via role-based permissions within each team area, independent of the centralized value list.

Question: 879

During regression testing in a scaled agile framework for enterprise software, ETM v7.x employs multiple local configurations for feature branch test plans, resulting in fragmented test coverage reporting within the master test plan dashboard across environments (On-prem VMs, Public Cloud Instances, Container Orchestrators). To integrate end-to-end quality metrics from child plans' execution environments into unified master reporting, what configuration practice does IBM recommend?

A. Define shared test environment baselines accessible across all local configurations for consistent metric collection.
B. Create work items tracking coverage gaps specifically between disparate configuration environments.

C. Manually aggregate execution status metrics from reports generated within each individual local configuration stream.
D. Contribute child configuration streams to the global master configuration for seamless unified coverage reporting.

Answer: D
Explanation: Contributing local streams from child test plans to the global master configuration in ETM v7.x integrates versioning and reporting of execution status, quality objectives, and coverage metrics across diverse environments, enabling comprehensive agile end-to-end traceability.

Question: 880

In a multi-stream project using ETM v7.x, a test case is linked to a requirement. The requirement is modified in its stream. The test case exists in three different stream configurations. According to configuration-aware traceability principles, in how many streams will the link to the test case be marked as suspect?

A. In none, because suspect marking does not propagate across configuration contexts automatically
B. Only in the stream that is designated as the project's main development line
C. In all three streams containing the test case
D. Only in the stream where the requirement was modified

Answer: C
Explanation: Configuration-aware traceability in IBM ETM v7.x manages links between specific versions of artifacts in specific configurations. When a requirement in a given stream is modified, the link itself becomes suspect. This suspect status is a property of the link relationship. If that same link is referenced from the test case artifact in multiple stream configurations (i.e., the test case is present in multiple streams, and the link to the requirement is valid in each of those contexts), then the link will appear as suspect from the perspective of the test case in *all* those streams. The suspect marking is tied to the linked pair, not isolated to a single configuration context of the source artifact.

Question: 881

A user needs to track the progress of a 'Test Plan' across multiple 'Sub-test Plans'. Which two reporting configurations will provide a consolidated view of execution progress? (Select two)

A. Enable 'Roll up results from child plans' in the Master Test Plan's 'Quality Objectives' section.
B. Manually export the results of all sub-plans to Excel and merge them using a pivot table.
C. Create a 'Dashboard' with a 'Test Execution' widget filtered by the Master Test Plan.
D. Use the 'Execution Status using Master Test Plan' report.

Answer: C,D
Explanation: The 'Execution Status using Master Test Plan' report is specifically designed to aggregate TCER data from all child plans linked to a master. Using a Dashboard widget with a filter for the Master Test Plan also provides a real-time, consolidated visualization of the entire hierarchy's execution status.

Question: 882

A team needs to perform 'Cross-Project Traceability' where a Test Plan in 'Project A' covers Requirements in 'Project B'. Which configuration is mandatory for this to work in ELM? (Select three)

A. Both Project Areas must have 'Friend' relationships established in the 'Server Administration' console.
B. 'Cross-Project Links' must be enabled in the 'Lifecycle Project Administration' (LPA) tool.
C. The user must have 'Read' permissions in both Project Area A and Project Area B.
D. Both Project Areas must be associated with the same 'Global Configuration' (if GC is enabled).

Answer: A,C,D
Explanation: Cross-project traceability relies on 'Friend' relationships between servers and the proper 'Link Index' configuration. If GC is used, both projects must contribute to the same Global Configuration to resolve links correctly. Finally, security is paramount; a user cannot see or follow a link to a requirement in Project B unless they have 'Read' access to that specific Project Area.

Question: 883

A test manager is configuring automated test execution for a complex test plan containing 450 test cases across multiple configurations representing different browser-OS combinations. The team uses Selenium Grid for execution and needs to ensure that execution records maintain traceability to specific configuration contexts while allowing parallel execution across 15 agent machines. The execution adapter must handle dynamic result posting as tests complete. Which approach should the test manager implement to optimize execution record management while maintaining configuration-specific traceability?

A. Create separate execution records for each configuration upfront, then configure the adapter to use ETM's execution record feed mechanism with configuration identifiers in the feed URL to route results to pre-created execution records
B. Implement a custom result parser that batches all execution results and posts them to a master execution record, then manually split results into configuration-specific child execution records using ETM's bulk operation API
C. Use the adapter's configuration matrix feature to generate execution records dynamically during runtime, allowing the adapter to create new execution records for each configuration as tests execute
D. Configure a single test suite execution record with configuration variables mapped to Selenium capabilities, enabling the adapter to post results with configuration context tags that ETM automatically

associates with the appropriate test case execution records

Answer: A
Explanation: Creating separate execution records for each configuration upfront and configuring the adapter to use ETM's execution record feed mechanism with configuration identifiers ensures proper traceability and parallel execution management. The execution record feed mechanism is the IBM-recommended approach for automated test execution scenarios involving multiple configurations because it allows adapters to route results to pre-created execution records using feed URLs that contain configuration context. This approach maintains the integrity of configuration-specific execution records, enables proper progress tracking per configuration, supports parallel execution without record conflicts, and ensures that execution history is properly segmented by configuration context. Pre-creating execution records also allows test managers to set up proper scheduling, resource allocation, and progress monitoring dashboards before execution begins, which is critical for large-scale test execution scenarios with multiple configurations.

Question: 884

A team needs to enforce that only users with the Project Administrator role can create new Test Case Categories in a project area, while allowing Test Managers to modify existing categories. The current process template's permission scheme is being customized. Which specific action within the "Test Case Category" artifact type's permissions must be modified to achieve this granular control?

A. Modify the "Create New" permission under the "Contributed Actions" section for the Test Case Category.
B. Modify the "Create" permission, setting it to Project Administrators only.
C. Modify the "Modify" permission for the "Work Item" under the Test Case Category artifact.
D. Modify the "Save" permission, restricting it to Project Administrators for new categories only.

Answer: A
Explanation: The correct approach is to modify the "Create New" permission under the "Contributed Actions" section for the Test Case Category artifact type. In IBM ETM v7.x, the ability to create a new instance of a category (like a Test Case Category) is controlled by a specific contributed action permission, not the general "Create" permission on the artifact. This allows for precise control, separating the creation of the category artifact itself from the modification of its attributes or the work items classified under it.

Question: 885

In the context of 'Test Governance', a Specialist is asked to set up an 'Audit Trail' that shows who changed the 'Test Script' and what the specific code change was. Where is this information found? (Select two)

A. In the 'Change Sets' associated with the ETM Stream (if Configuration Management is enabled).
B. In the 'Revision' history of the manual test steps editor.
C. In the 'History' tab of the Test Script artifact in ETM.
D. In the 'Audit Report' generated by the 'Engineering Insights' tool.

Answer: A,C
Explanation: ETM provides a basic 'History' tab for every artifact that logs who changed it and when. However, when 'Configuration Management' is enabled, ETM provides much deeper auditing through 'Change Sets'. A change set groups changes together and provides a granular view of what was modified, serving as a robust audit trail for versioned artifacts.

Question: 886

In distributed testing, execution records in IBM Engineering Test Management v7.x are executed across time zones. Suspension at end-of-day must preserve partial verdicts and linked defects for next-day resumption. Which setting ensures continuity?

A. Force daily record closure, create new for next day\
B. Configure partial verdict persistence during suspension, retain all linked defects and partial test log data for seamless next-session resumption\
C. Discard partial verdicts on suspension, require full restart
D. Save only completed steps, lose partial context and links\

Answer: B
Explanation: Suspension preserves partial verdicts, test logs, and all linked defects, allowing direct resumption with complete continuity across sessions and time zones in distributed teams.

Question: 887

After a major test cycle, a test lead needs to analyze the root causes of failures. They want to query ETM to find all test cases that failed due to a specific common cause, like "Network Timeout." How can testers be guided to provide this structured failure reason during execution?

A. Create a separate defect for every failure and categorize the defect.
B. Rely on testers to type "Network Timeout" into the failure comment field.
C. Use a script to parse free-text comments for keywords after the execution is complete.
D. Define a custom "Failure Category" list (e.g., Defect, Test Data Issue, Environment Issue, Network Timeout) as an enumeration attribute on the Test Case Execution Record, and require testers to select one when marking a step as "Failed."

Answer: D

Explanation: To enable structured analysis of failure reasons, the IBM-recommended approach is to extend the data model of the "Test Case Execution Record" artifact type. A project administrator can add a custom attribute (e.g., `Failure Category`) of type "Enumeration" with a predefined list of common causes. This attribute can be set as "Required" when the verdict is "Failed." During manual execution, when a tester marks a step as failed, ETM will prompt them to select a value from this dropdown before saving. For automated execution, adapters can be configured to populate this attribute via the API (e.g., mapping certain error logs to a specific category). This ensures consistent, queryable data from the moment of failure recording, making root cause analysis reports efficient and accurate. Relying on free-text comments is unreliable for aggregation.

Question: 888

When creating an automated test script in ETM that will trigger a Rational Functional Tester (RFT) script located on a shared network drive, which components must be configured to ensure a successful handshake? (Select 3)

A. A Functional Test script type selection during the Test Script creation in ETM.
B. The correct URI or path to the script file provided in the ETM Test Script's "Path" field.
C. A properly configured ETM Adapter installed and running on the machine where RFT is located.
D. The "External Tool Path" property in the ETM Project Area under 'Global Script Settings'.

Answer: A,B,C
Explanation: For ETM to execute external automated scripts, the Test Script artifact must be set to the appropriate type (e.g., Functional Test). An adapter (such as the RFT adapter) must be active and registered with the ETM server to act as a bridge, and the specific location of the script must be identified within the ETM Script's configuration.

Question: 889

A performance test suite produces a large output file (several GB) . The team wants this file attached to the test execution record for analysis but understands that storing it directly in the ETM database is inefficient. What is the IBM-recommended approach for handling such large artifacts in relation to test execution records?

A. Use the ETM's integrated artifact management system to store the file as a separate large artifact and create a formal "Attachment" link from the execution record to this artifact.
B. Store the file on a shared network drive and create a URL link type artifact in ETM that points to the file, then link this URL artifact to the execution record.
C. Configure ETM to use an external object store (like S3 or a filesystem) for large attachments, and attach the file normally; the link will reference the external store.
D. Compress the file before attaching it to the execution record; the database can handle compressed large

objects.

Answer: C
Explanation: IBM Engineering Test Management v7.x supports integration with external object stores (such as a file system or cloud storage like S3) for handling large attachments. This is the recommended and built-in approach. An administrator can configure the ETM server to use an external store. Once configured, when a user attaches a large file to an execution record (or any work item), the file's content is automatically stored in the external repository, while a lightweight reference (metadata, pointer) is stored in the ETM database. This maintains the user experience of a seamless attachment while ensuring database performance and scalability are not impacted by large binary data. The attachment remains accessible through the standard ETM UI.

Question: 890

You need to integrate a third-party test data management (TDM) tool with ETM. The TDM tool provides a REST API to reserve a data set, returning a unique I**D.** This ID must be automatically attached to a Test Execution Record before the test runs. Which ETM integration point should be used?

A. Develop a custom build definition in EWM that reserves data and passes the ID as a build property to the ETM execution.
B. Create a custom operation on the Test Execution work item type that, when set to 'In Progress', calls the TDM API and populates a custom attribute.
C. Train testers to manually reserve data and enter the ID in the execution record's comment field.
D. Configure a pre-test action in the automation adapter that calls the TDM API and appends the data set ID to the test script as a variable.

Answer: B
Explanation: A custom operation tied to a state change in the ETM process provides a clean integration point. You can add a custom operation (e.g., "Reserve Test Data") to the Test Execution work item that triggers when moving to an 'In Progress' state. The operation's server-side behavior can be implemented via a custom Java class (process behavior) that calls the external TDM REST API, retrieves the data set ID, and sets it into a custom attribute on the Test Execution record. This automates the process within the ETM workflow without relying on the adapter or manual steps.

Question: 891

A test engineer is working in a "Stream" and makes several changes to a "Test Script." They realize they made a mistake and want to revert the script to the way it looked two days ago, but they don't want to revert the entire stream. How can this be done in ETM v7.x? (Select two)

A. Open the "History" tab of the Test Script, select the version from two days ago, and use the "Restore"

or "Set as Current" action.
B. Delete the script and hope the system has an "Undo" button in the trash can.
C. Use the "Compare" feature to see the differences between the current version and a previous version, then manually undo the changes.
D. Create a new baseline and then merge it into the current stream.

Answer: A,C
Explanation: ETM v7.x maintains a detailed version history for every artifact within a stream. The History tab allows users to see every "Save" event and provides the ability to promote an older version to be the current one, effectively reverting the changes for that specific artifact without affecting anything else in the stream. Alternatively, the Compare tool provides a side-by-side view that helps the user manually identify and roll back specific lines of code or steps in a test script, providing granular control over the recovery process.

Question: 892

In a nuclear power plant control system project under strict regulatory audit requirements, the team must produce immutable, versioned test artifacts for compliance evidence. Using ETM v7.x configuration management, what is the IBM-recommended way to version test artifacts and link them to DOORS Next requirements baselines for full lifecycle traceability?

A. Duplicate artifacts periodically into archive projects
B. Use active streams without baselining and rely on change history logs
C. Create baselines of the ETM component artifacts and include them in global configurations alongside DOORS Next baselines for immutable, auditable traceability
D. Export test artifacts to external document management systems

Answer: C
Explanation: Baselining ETM artifacts creates immutable versions linkable via global configurations with DOORS Next baselines, ensuring end-to-end lifecycle traceability suitable for regulatory audits.

Question: 893

An organization wants to track test progress by milestone with burn-up charts showing completed vs. planned test cases per environment combination. What is the IBM-recommended reporting method?

A. Use custom dashboard widgets without environment breakdown
B. Export data weekly and build charts in Excel
C. Rely on built-in test plan progress bars only
D. Use Jazz Reporting Service (LQE) with SPARQL queries joining milestones, execution records, and environment values

Answer: D
Explanation: Advanced burn-up and progress tracking by milestone and environment requires LQE-based reporting with SPARQL queries that correlate test plan milestones, execution records, results, and scoped test environments — the IBM-recommended approach for detailed metrics.

Question: 894

You are the Test Architect for a multinational financial services project using a hybrid V-model for regulatory modules and agile for customer-facing features. The program uses a single ETM project area. You discover that a change to a core "Transaction Logging" test case, made in the context of the agile stream to support a new UI feature, has inadvertently altered the version of that test case used in a regulatory V-model test cycle, potentially breaking audit trails. According to IBM's best practices for ETM v7.x configuration management in complex lifecycles, what is the most critical configuration element that was likely missing or misapplied to prevent this cross-context contamination?

A. Failure to use distinct Work Item Categories for "Regulatory" and "Agile" test cases, allowing them to be modified in the wrong context.
B. Lack of separate Project Areas for the V-model and agile workstreams, causing process and artifact overlap.
C. Not utilizing separate Change Sets within the same repository workspace for the two types of work.
D. Absence of unique Global Configurations for the regulatory baseline and the agile development stream, leading to artifact version mismatch.

Answer: D
Explanation: The core issue is the unintended sharing of a mutable artifact version across two divergent development contexts. In ETM, Global Configurations define the exact set of component versions (including test artifacts, requirements, and builds) that constitute a consistent configuration. For a hybrid model, the regulatory V-model cycle should use a Global Configuration pointing to a stable baseline of all components. The agile development should use a separate Global Configuration pointing to the active development stream. By associating the Test Plan for the regulatory cycle with its own Global Configuration (locked to a baseline), and the agile Test Plan with a configuration for the stream, the test case artifact would be resolved as two different version IDs in each context. A change in the agile stream creates a new version of the test case in that stream, but the version referenced by the regulatory baseline configuration remains untouched, preserving audit integrity.

KILLEXAMS.COM

Killexams.com is a leading online platform specializing in high-quality certification exam preparation. Offering a robust suite of tools, including Exam Questions, practice tests, and advanced test engines, Killexams.com empowers candidates to excel in their certification exams. Discover the key features that make Killexams.com the go-to choice for exam success.

Practice Exam Questions Based on Current Exam Objectives

Killexams.com provides practice exam questions aligned with the latest official exam objectives and latest syllabus. Our content is reviewed and updated regularly to reflect recent changes announced by certification vendors. By studying these practice questions, candidates will cover the structure, difficulty level, and topics of the actual exam, helping them prepare more effectively and efficiently.

Comprehensive Practice Exams (PDF Format)

Killexams.com offers multiple-choice questions (MCQs) in easy-to-read PDF format, covering all major domains of the exam. Each PDF contains a structured collection of practice questions and verified answers designed to support focused study. These MCQs help candidates reinforce key concepts, identify knowledge gaps, and improve exam readiness through consistent practice.

Realistic Practice Tests (Online Test Engine & Desktop Test Engine)

To support hands-on preparation, Killexams.com provides practice tests through both an Online Test Engine and a Desktop Test Engine. These tools are designed to simulate a real exam environment, allowing candidates to practice under exam-like conditions, with latest syllbus and topics of the exam. Performance tracking, test history, and result analysis help users evaluate their progress and focus on areas that need improvement.

Risk-Free Purchase Policy

Killexams.com follows a transparent and customer-friendly purchase policy. If users are not satisfied with the study materials, they may request assistance or a refund in accordance with our published terms and conditions. This policy reflects our commitment to customer satisfaction, fairness, and confidence in our preparation resources.

Regularly Updated Content

Our practice question bank is reviewed and updated on an ongoing basis to stay aligned with the latest exam outlines and vendor updates. This ensures candidates are studying up-to-date, relevant material, and preparing with content that reflects current exam expectations, helping them stay confident and well-prepared.