High-Level Logic: Octane Test Result Integration

# 1. Objective

To automate the mapping and upload of regression test results from internal builds to Octane, and associate them with related JIRA stories for better traceability and analytics.

# 2. Flow Summary

## Step 1: Identify the Build and Regression Stories

The process starts by selecting a specific build (typically the latest from a pipeline). A set of JIRA IDs related to regression tests are identified.

## Step 2: Generate Test Results

The system generates a formatted dataset containing test case outcomes (pass/fail), metadata, and execution timestamps based on the build and selected stories.

## Step 3: Identify the Target Workspace in Octane

For each JIRA story, the system dynamically identifies the corresponding Octane space and workspace by querying Octane using the JIRA key. This avoids hardcoding and supports multiple workspace mappings.

## Step 4: Prepare and Structure the Test Data

A unique test suite path is generated based on the release train and time (e.g., Q1, Q2). The system ensures required application modules and test suites exist in Octane, creating them if necessary.

## Step 5: Upload Test Results to Octane

The test results are converted into Octane’s accepted format and uploaded. Each result is associated with a specific run and test suite under the relevant workspace.

## Step 6: Update Coverage Links

After the test runs are uploaded, the system links them to the related Octane backlog items (e.g., stories). This ensures that Octane dashboards and traceability reports reflect the correct relationship between test coverage and business requirements.

# 3. Key Highlights

• Dynamic Workspace Detection: Automatically resolves Octane workspace for a given JIRA.

• Modular and Reusable: Works for any release train or project without manual reconfiguration.

• Traceability: Links test execution data with JIRA stories in Octane for full traceability.

• Failsafe Uploading: Handles retries, partial uploads, and batch processing.

# 4. Outcome

• Manual test reporting and linking are eliminated.  
• QA dashboards in Octane show real-time test execution data.  
• Product owners and QA leads get traceability from requirements to test outcomes.