REGRESSION TESTING AUTOMATION WITH ORACLE APPLICATION TESTING SUITE

BENEFITS

- Reduced development time for script creation
- Improved security of data during testing with built-in encryption
- Simplified debugging of script development issues through user-friendly error messages
- Increased availability of Java-based customization, which allows for improved script logic complexities
- Proactive and efficient support
- Streamlined hardware requirements and script support maintenance when compared to prior test-tool usage

What Oracle Has Done

Oracle has successfully initiated the deployment of Oracle Application Testing Suite to transform its functional and performance regression testing using the following components:

- Oracle Functional Testing - OpenScript platform
- Oracle Load Testing

A key part of Oracle's test automation strategy is to implement Oracle Application Testing Suite to provide test managers and test developers with a synchronized, standardized, and automated regression testing process that utilizes best practices and application functionality.

Before Business Transformation

Prior to our deployment, application group testers either performed regression tests using a third-party tool or tested all processes manually. This was time-intensive and required technical knowledge to develop the scripts. The tests were recorded and then edited using VBScript with a user interface for managing application UI object attributes. The development timescale for scripts and application upgrades required regular updating between execution runs. The manual testing was resource intensive and prone to inaccuracies. Test results were distributed to a set distribution list and any additional requests for the results would be done via e-mail. Manual test results were added within the test-case documents and either uploaded to a shared hard disk; the link was distributed by e-mail or the file was distributed to all stakeholders.

This process was time-intensive and kept test developers and test managers resourced on script development or script execution. This impacted efforts toward introducing efficiency measures and the appropriate transfer of information to all stakeholders involved.

Best Practices—After Business Transformation

After the implementation of Oracle Functional Testing – OpenScript platform and Oracle Load Testing, the following tasks can be performed.

Test developers can:

- Develop functional test scripts with limited technical knowledge
- Develop performance test scripts using the built-in application performance modules
- Execute tests using multiple and varied data sources such as databank, database, variables, or other Java objects and leverage data encryption
- Utilize available user interface components to leverage automated test capabilities
- Use standard Java programming to customize and enhance script capabilities
- Create shared functions to reduce development effort and leverage script legibility by all users, regardless of technical knowledge

Test managers can:

- Understand script activity using the centralized script repository across all lines of business
- Receive and act on reports distributed either by performance analysts or functional test results
- Distribute responsibility of testing to more team members and concentrate on application functionality rather than simplifying test complexity to ease flow development

Performance analysts can:

- Execute performance tests using Oracle Load Testing for a given time period and with a set number of use
- View and create numerous reports on results from the performance test
- Create and distribute reports to project stakeholders

Application groups can:

- Use reports from performance and functional tests to make strategic decisions with reference to application performance and areas for application enhancement

Business groups can:

- Delegate regression testing duties to business group members by introducing the user interface functionality in the test tool

Best Practices

- Use the password entry shared function instead of hardcoding passwords, whether it's Single Sign-On (SSO) passwords or database credentials
- Avoid launching more than one browser and repeating the log in process—this will help to reduce regression test runtime
- Consider using a shared function instead of repeating a block of steps three or more times—this will help improve maintenance and script readability
- Move completed sanity and functional scripts to a relevant workspace in the shared repository
- Choose the correct module in the OpenScript platform for script development, as they are application specific
- Use OpenScript Eclipse debugging tools where relevant to resolve errors in script development
- Ensure all application test data is present before commencing script development
- Ensure relevant permissions and privileges are granted before commencing script development, especially recording

Lessons Learned

- Ensure there is key stakeholder buy-in at the beginning of the implementation process
- Ensure there is sufficient training for those new to the product
- Identify key scenarios and ensure they are updated and documented
- Develop base templates for each testing category(Sanity/Functional/Regression Testing)
- Define a standard set of object identification criteria prior to script development
- Be aware application performance is not static and not impervious to external factors
- Aim to achieve the highest level of automation by avoiding multiple stops and starts through a testing cycle necessitating manual intervention
- Ensure all advanced team members have sufficient Java training to make full use of the product

Oracle@Oracle

Oracle@Oracle has been established to leverage Oracle's internal experience and expertise with our products and processes to help customers and partners maximize their investment with Oracle. It essentially uses Oracle as a Reference Customer sharing Oracle's approach, challenges, benefits achieved, and lessons learned to help with customer initiatives and to enhance the credibility of the sales cycle by sharing real experiences that bring Oracle's staff face-to-face with the customer.

CONTACT US

To learn more, visit oracle.com or call **+1.800.ORACLE1** to speak to an Oracle repsentative.

ORACLE®