New Relic Integration

Test Analytics and Monitoring Platform

PUBLIC DOCUMENTATION

MQE Unified OAO Test Automation Framework

October 22, 2025

Table of Contents

1. Executive Summary

The MQE Unified OAO Test Automation Framework integrates with New Relic Insights to provide comprehensive test analytics and monitoring capabilities. This integration enables real-time tracking of test execution metrics, code coverage analysis, and performance monitoring across multiple platforms and brands.

Key Benefits

- **Real-time Test Analytics**: Monitor test execution results instantly
- **Code Coverage Tracking**: Track code coverage across different platforms
- **Performance Insights**: Identify bottlenecks and optimize test execution
- **Historical Trends**: Analyze test trends over time
- **Multi-Platform Support**: Unified view across Roku, Apple TV, Android TV, Fire TV, and more

2. Architecture Overview

2.1 High-Level Architecture

The New Relic integration follows a modular architecture:

Integration Flow:
Test Execution → Data Collection → Data Transformation → New Relic Insights API → Dashboard Visualization

2.2 Components

1. **Data Logger**: Responsible for collecting and sending test data
2. **Analytics Service**: Queries and retrieves data from New Relic
3. **Event Models**: Structured data models for different event types
4. **Configuration**: Centralized configuration management

3. Event Types and Data Models

3.1 Test Execution Events

Event Type: `Unified_QE_TestExecution_v1`

Captures comprehensive test execution data including:

- Test case ID and name
- Execution result (PASS/FAIL/SKIP)
- Platform and brand information
- Application version
- Execution timestamp
- Failure reasons and categorization
- Sprint and environment details

3.2 Code Coverage Events

Event Types:

- `Unified_QE_TestCaseMapping_v2`: Maps test cases to code files
- `Unified_QE_TotalTestCoverage_v1`: Overall coverage metrics
- `Unified_QE_UncoveredFiles_v1`: Files with no coverage
- `Unified_QE_CoverageFileLinks_v1`: Links to coverage reports

3.3 Configuration Events

Event Types:

- `Unified_QE_AppConfig_Summary_v2` : Application configuration summary
- `Unified_QE_AppConfig_Value_v2` : Detailed configuration values

3.4 Integration Events

Event Types:

- `Unified_QE_TestRail_v1` : TestRail test case information
- `Unified_QE_TestRun_v1` : TestRail execution data
- `Unified_QE_Jira_v1` : Jira bug tracking data

4. Visualization and Dashboards

4.1 Dashboard Types

Test Execution Dashboard

Provides real-time visibility into test execution:

- Pass/Fail/Skip distribution (Pie Chart)
- Execution trends over time (Line Chart)
- Platform-wise breakdown (Bar Chart)
- Brand-wise analysis (Stacked Bar Chart)

Code Coverage Dashboard

Tracks code coverage metrics:

- Overall coverage percentage (Gauge Chart)
- Coverage by platform (Bar Chart)
- Test case to code mapping
- Uncovered files list (Table)

Test Analytics Dashboard

Advanced analytics and insights:

- Test execution time trends
- Flakiness detection
- Failure pattern analysis
- Sprint-wise progress tracking

4.2 Query Language (NRQL)

New Relic uses NRQL (New Relic Query Language) for data queries. Example:

```
 SELECT count(*) FROM Unified_QE_TestExecution_v1
WHERE platform = 'Roku' AND testResult = 'SUCCESS'
SINCE 1 week ago
```

5. Supported Platforms

The integration supports the following platforms:

Platform	Description
Roku	Streaming platform
Apple TV / tvOS	Apple's TV platform

iOS	Apple mobile devices
Android TV	Google's TV platform
Fire TV	Amazon's Fire TV devices
Android	Android mobile devices
Tizen TV	Samsung Smart TVs
LG WebOS	LG Smart TVs
Vizio	Vizio Smart TVs
Hisense TV	Hisense Smart TVs
Xbox	Microsoft Xbox platform
Comcast	Comcast Xfinity platform

6. Use Cases

6.1 Quality Metrics Tracking

Monitor quality metrics across releases:

- Track test pass rates over time
- Identify quality trends
- Compare quality across platforms
- Monitor regression detection effectiveness

6.2 Test Optimization

Optimize test execution:

- Identify slow-running tests
- Detect flaky tests
- Optimize test suite composition
- Balance test coverage vs execution time

6.3 Release Readiness

Assess release readiness:

- Sprint-wise quality metrics
- Code coverage requirements
- Critical path testing
- Platform-specific readiness

6.4 Root Cause Analysis

Investigate failures:

- Failure categorization and trends
- Link to known issues (Jira)
- Test case execution history
- Environment-specific issues

7. Best Practices

7.1 Dashboard Configuration

1. Create role-specific dashboards (Developer, QA, Manager)
2. Use filters for focused analysis

3. Set up alerts for critical metrics
4. Schedule regular dashboard reviews

7.2 Data Analysis

1. Analyze trends, not just snapshots
2. Compare across time periods
3. Correlate with release cycles
4. Share insights with stakeholders

7.3 Performance

1. Monitor dashboard load times
2. Optimize NRQL queries
3. Use appropriate time windows
4. Archive historical data appropriately

8. Integration Benefits

8.1 For QA Teams

- Real-time test execution visibility
- Quick identification of test failures
- Historical trend analysis
- Code coverage tracking

8.2 For Development Teams

- Code coverage by feature

- Test impact analysis
- Quality metrics per sprint
- Regression detection

8.3 For Management

- Executive dashboards
- Quality KPIs and metrics
- Release readiness assessment
- Resource allocation insights

9. Getting Started

9.1 Prerequisites

- Access to New Relic account
- Understanding of NRQL basics
- Familiarity with test automation framework

9.2 Accessing Dashboards

1. Log in to New Relic platform
2. Navigate to Insights section
3. Select the appropriate account
4. Choose or create dashboards
5. Apply filters as needed

9.3 Creating Custom Dashboards

1. Identify key metrics to track
2. Write NRQL queries
3. Select appropriate visualization types
4. Arrange widgets on dashboard
5. Save and share with team

10. Troubleshooting

10.1 Common Issues

No Data Appearing

- Verify test execution is sending data
- Check time range in dashboard
- Validate filter conditions
- Confirm event types are correct

Slow Dashboard Load

- Optimize NRQL queries
- Reduce time window
- Use appropriate aggregations
- Consider data sampling

10.2 Support Resources

- New Relic documentation: https://docs.newrelic.com
- NRQL reference guide
- Internal team documentation
- Support channels

11. Conclusion

The New Relic integration provides a powerful platform for test analytics and monitoring. By leveraging real-time data collection, comprehensive event tracking, and flexible visualization capabilities, teams can gain deep insights into test quality, coverage, and performance across all supported platforms.

Key Takeaways

- Centralized test analytics across all platforms
- Real-time visibility into test execution
- Comprehensive code coverage tracking
- Data-driven quality decisions
- Integration with TestRail and Jira
