PARASOFT®

Retail: Enabling Automated Continuous Testing of Complex Purchase Transactions

Transitioning to DevOps, the world's second largest retailer significantly accelerated much of their delivery pipeline, but testing transactions through their ecommerce site remained a bottleneck. Their ability to continuously execute automated tests involving core purchase functionality was impeded by unstable downstream components—some with significantly-delayed asynchronous responses—as well as third-party services that were difficult to configure for testing. Service virtualization eliminated these roadblocks, enabling them to execute a more expansive test suite faster, automatically, and continuously.

The Challenge: Test Environment Constraints Make Automated Continuous Testing Impractical for Continuous Integration

When a customer completes the checkout process on the retailer's ecommerce site, this triggers a number of system operations:

- Validate the address through a third-party service
- Validate credit cards and cash cards through a different third-party service
- Query internal systems to ensure that the items are still in stock
- Enter an order into the order fulfillment system

The transaction is not considered complete until a confirmation email is sent to the customer. However, before that can occur, the order fulfillment system must first receive the order, perform a number of operations to process the order, then send an asynchronous confirmation response once those operations are completed.

Most of the business-critical transactions that need to be continuously tested involve this checkout procedure. However, testing these transactions was complicated by a number of factors:

- The third-party validation services allowed only a limited number of test transactions before a fee was levied. Moreover, the team needed to continuously test how the AUT responded when these services were slow, unavailable, or returned errors—but they weren't able to configure this behavior within their test environment.
- Various downstream internal systems in the test environment were often unstable or down due to frequent updates—making them unavailable for testing. When tests failed as a result of these environment issues, a team member had to manually investigate the source of the failure. If it was related to system instability, they had to wait for the system to be brought back online before they could achieve the 100% test success rate required for the application to be promoted to the next stage of the software delivery pipeline.
- The order fulfillment system in the test environment typically took over 30 minutes to return the necessary response, which meant that any test involving the checkout transaction would take over a half hour to complete. This delay made it impractical to continuously execute the complete regression test suite as part of the team's Continuous Integration process.

The Solution: Service Virtualization Provides Continuous Access to a Stable and Complete Test Environment

The company was able to tackle all three of these issues by applying Parasoft's Continuous Testing platform. Parasoft Service Virtualization gave them control over dependencies that were difficult to access for testing. Moreover, Parasoft Environment Manager automatically detected test environment stability issues so that unstable components could be instantly replaced with "virtual assets" before they had a chance to compromise test results.

The challenge of accessing third-party validation services with the necessary frequency and response conditions was addressed with service virtualization. The company recorded how the AUT interacted with the actual services, then captured this behavior in flexible virtual assets. They then parameterized these virtual assets with a broad array of response conditions, including negative responses, performance delays, and the various error conditions that they needed to test against.

The complex downstream component stability issue was resolved through a combination of test environment monitoring and service virtualization. Parasoft Environment Manager was configured to continuously monitor whether the necessary downstream dependencies were truly functioning as expected (e.g., they were not only responsive, but successfully passing a battery of functional test scenarios). If a problem was detected, the team was immediately alerted, and the environment was reconfigured to use virtual assets instead of the actual systems. In order for this strategy to work, they used service virtualization to create virtual assets representing the behavior of each of these systems. When the systems were deemed "healthy," the AUT's interactions with them were recorded, then converted to virtual assets that were then parameterized via synthetic test data generation.

Finally, the problem of the delayed asynchronous responses bringing test execution to a crawl was addressed by replacing this system's behavior with a virtual asset—again, parameterized via synthetic test data generation—and configuring its response times to suit their testing needs. For testing during Continuous Integration, they configured the virtual asset to respond instantly so that test execution could complete as quickly as possible. Additionally, they added a second "performance profile" to the virtual asset so that the nightly tests could run against that same virtual asset using more realistic response times. This enabled them to rapidly validate each change as it was introduced while still testing the AUT versus a range of real-world response times.

PARASOFT®

USA PARASOFT HEADQUARTERS / 101 E. Huntington Drive, Monrovia, CA 91016
Phone: (888) 305-0041 / Email: info@parasoft.com