Answering Your Questions about Negative Testing

What is Negative Testing?

Negative testing is aimed at discovering system flaws. Skilled, experienced testers with an eye for details generate test scenarios to find those hidden errors. Without the skill, proactive attitude, and intelligence of the tester, negative testing would not be as powerful in detecting system bugs.

The Two (2) Major Negative Testing Categories:

1) Tests that are designed to make the system fail:

- Discovery of faults that result in significant failures; crashes, corruption, and security breaches
  Example: The backend database is made unavailable while testing.

- Exposure of software weakness or potential for exploitation
  Example: An automated test attempts to launch web pages within the web application.

2) Tests that are designed to determine if code is written to handle exceptions:

- Input validation
  Example: Enter a future date for a "Date of Death" field to ensure that the system is able to handle the exception and provides an appropriate error message.

- Coping with absent, slow, or broken external resources
  Example: Enter a broken web site link or an unavailable web service to determine how the application responds.

- Error-handling functionality
  Example: Enter invalid data to verify how the application handles messaging, logging, and monitoring.

- Recovery functionality
  Example: Shut down the primary system to verify functionality of fail-over, rollback, and restoration.

Negative testing is an open-ended activity, most often during System and Integration testing, and is generally not used by the user community. It is unrealistic to perform all possible negative test cases, so teams should exercise limits regarding the number of negative tests to be performed.

How can you learn more?

Contact us at info@valytics.com to learn more about Negative Testing or how Valytics' full service independent test capabilities can help your organization.