Diligence

User Interface Testing

We transform our user stories in Jira into a test plan. The test plan is then scripted to provide an automated, system-level testing of the user interface and overall customer experience. This can be run headlessly in an automated pipeline.

As with the code-level tests above, system-level tests are written to prove the existence of bugs before they are fixed. In this way we build a library of tests that grow over time and provide ever more reassurance.

User Acceptance Testing

We test the system in real-world scenarios with real-world users, to validate that it meets their requirements and expectations.

Non-functional Testing

Some or all of the below may be necessary

- Stress Testing
  Assess the system's performance and stability under extreme workloads.
- Penetration Testing
  Simulate a cyber-attack against the system to identify vulnerabilities.
- Failover Testing
  Deliberately cause components of a system to fail, ensuring that backup components automatically take over (with minimal disruption to usage).
- Chaos Engineering
  Introduce random failures into the system to assess its resilience and fault tolerance.

Bug reporting

Bug reports should include the following attributes to assist in understanding and speedy resolution:

- Title.

- Description

  Including what the testing objective was, what the tester expected to happen and what happened
- Steps to Reproduce

  A step-by-step guide to reproduce the issue.
- Environment

  Where the bug was encountered.
- Severity and Priority
- Any relevant attachments

  Screenshots, error messages, logs, etc

Revision #2
Created 21 September 2023 10:58:23 by James Hall
Updated 21 September 2023 12:32:37 by James Hall