HYPERTEST

CONTRACT TESTING

A Quick Overview:

Checklist for performing Integration Testing

Start Early and Test Often

Incorporating integration testing early in the development cycle is vital. It enables the early detection of defects and reduces the cost of fixing them. Regular testing as new modules are integrated ensures continuous oversight.

Choose the Right Tools

- Selecting appropriate tools is critical for effective integration testing. Tools should be compatible with the project's technology stack and support automation to streamline the testing process.

Create a Test Plan

- A comprehensive test plan outlines the scope, approach, resources, and schedule of the testing activities. It should include specific test cases that cover all pathways and interactions between modules.

Test Environment Configuration

- Setting up a test environment that closely mirrors the production environment is crucial for accurate results. This includes configuring hardware, software, network settings, and other system components.

Continuous Integration

- Implementing continuous integration practices allows for regular merging of code changes into a central repository, followed by automated builds and tests. This approach helps in identifying and addressing integration issues promptly.

✓ Error Handling

Effective integration testing should also focus on how the system handles errors. Testing should include scenarios where modules fail to communicate or return unexpected results.

✓ Performance Testing

- Besides checking functionality, it’s important to test the performance of integrated units under various load conditions.

✓ Documentation and Reporting

- Maintaining detailed documentation of the testing process, including test cases, results, and issues, is essential for future reference and accountability. Regular reporting keeps all stakeholders informed about the testing progress and outcomes.

HYPERTEST

95 Third Street
2nd Floor, 94103 San Francisco,
California, USA

Follow us on