



WHAT IS ACTION BASED TESTING?



Action Based Testing is a systematic testing method that embodies the two pillars of test design best practices: modular-design and action-driven tests. Action Based Testing is the secret sauce that makes TestArchitect a winning recipe for test teams.

MODULAR TEST DESIGN

Modular planning is a top-down approach that pushes logical test flow into a more efficient test creation, which omits unnecessary details and redundant checks—and at the same time provides a higher level of business and technical coverage. Thus modular planning pushes logical test flows into well-defined, non-overlapping areas of categorization that account for business objects and processes.

ACTION DRIVEN TESTS

Action-driven test authoring is action-based, separating automating tests from their design. Test modules developed like this optimize coverage, and minimize work in their development and maintenance. Test modules are containers for organizing the tests in regards to user stories or software requirements. Tests within the modules are interrelated, but are intended to be independent from tests in other test modules.



Modular Test Design

The modular design addresses the challenges of large-scale test planning, test manageability, and maintainability through thorough test organization from the point of initiation.



Action Driven Tests

The other dimension, action-driven test development, eliminates the burdens of extensive programming that are usually required to automate tests. The readability of action-driven tests also enables domain experts to contribute to the test design and test execution.

HOW TO IMPLEMENT ABT

To develop a design process, there are three levels to implement: modular planning, individual test module design, and action-driven test authoring.

Modular Planning

Modular planning is a top-down approach that pushes logical test flow into a more efficient test creation, which in turn, omits unnecessary details and redundant checks—the level of coverage is higher.

Thus modular planning pushes logical test flows into well-defined, non-overlapping areas of categorization that accounts for all business objects and processes.

Individual Test Module Design

The second fragment is test module design. Test modules are containers for organizing the tests in regards to user stories or software requirements. Tests within the modules are interrelated, but are intended to be independent.

The second fragment is test module design. Test modules are containers for organizing the tests in regards to user stories or software requirements. Tests within the modules are interrelated, but are intended to be independent.

Action Driven Test Authoring

Action-driven test authoring is action-based, separating automating tests from the test design. By using TestArchitect, the built-in actions are pre-programmed enabling a single test to run against numerous platforms.

Tests are a network of actions that specify an activity followed by a verification point. Again this is a great thing to break up. Make each level it's own box/layer.

When considering a test development plan, we evaluate the testing through numerous modules. Once established these modules are developed using actions, and via those action become automated. There are two different tests to assess: interaction tests (low level, more detailed) and business tests (high level, less intricate and sensitive).

Interaction Tests



Interaction tests focus on the user interface

Business Tests

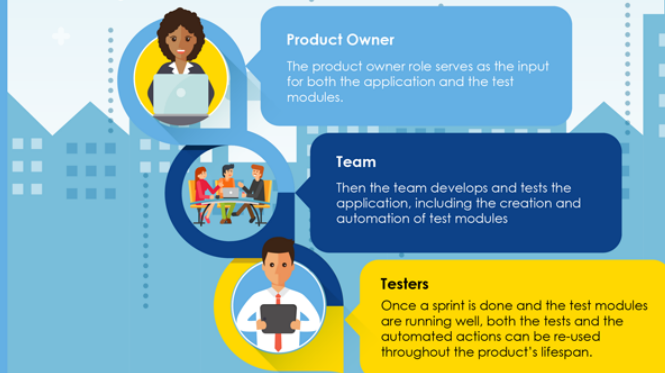


Business tests are broken into specified actions.



HOW DOES ABT FIT IN AGILE TEAMS?

By taking an Agile approach, ABT becomes more flexible and allows teams to self-manage. The agile lifecycle is as follows:



CONCLUSION

What problems does ABT solve?

By using TestArchitect and Action Based Testing, a user will profit from the many built-in pre-programmed actions to make them run against numerous platforms. One of the great features of TestArchitect is the ability to create your own actions, which makes it customizable for the users' needs. This solves problem

Maintenance and Scalability:

One of the biggest problems in Enterprise Test Automation is when changes to the AUT happen. In Action-Based Testing, the Actions are used to specify the test flows that exercise the business-logic of your application. Since the test flows are at the business operation level, the test cases seldom change, since the business logic of the application also seldom changes. This makes maintaining test over time easy. The ABT method of Test Organization enables the creation of multiple Test Modules, with Test Cases inside them, for maximum scalability to create a large volume of tests.

Perfectly Designed for Smaller Teams:

By using TestArchitect and ABT, a much smaller test team can support a larger organization. ABT allows tests to be written in a plain-text language. This allows tests to be designed as easy to read and maintain a sequences of actions, specifying activities followed by verification points. This means that those without a programming background can create scripts.