



STAR QUALITY

**Enabling Packaged Tests** Organizations often develop applications with similar features for different market segments. Using a traditional approach to automation, the effort to test the group of applications is largely linear. One of the strengths of StarTest is its ability to encapsulate tests that can be reused across applications in a vertical application space. For example, we have demonstrated through our consulting practice how tests for one search engine can be developed in such a manner that they are reusable for virtually any other. Our clients have enjoyed huge productivity gains by reusing common tests across multiple applications. Star Quality offers packaged test kits for Search, Retail and eRetail.

# StarTest

## Don't Write Tests When You Could Build Them

### The trouble with automated testing tools:

Today's automated testing tools fall short of their promise to provide the user with a good return on investment. Recorders are easy to use but produce tests that are difficult to maintain when the application under test changes. The common recommendation is to record the tests over again. But doing so forfeits the investment and, worse yet, discards the baseline. On the other side of the spectrum are tools that offer programming languages. They provide a more robust solution but have a steep learning

curve and are hard to use. Many organizations conclude that testing tools don't work and continue to test their applications manually. Automated testing tools take up more than their share of shelf-ware space.

### What makes StarTest

**different:** Star Quality marries the ease of use from recording with the robustness of hand-coded scripts. With StarTest, users build automated tests by grouping together predefined methods and data. The tool generates the test cases at runtime each time a test plan is executed. Tests can be de-

veloped faster than with a recorder and are easy to maintain. StarTest provides methods for driving and verifying the application under test and is extendible to allow users to add custom methods. Since many of the methods used to test an application are universal, there is very little test code to maintain. We've proven through nearly ten years of consulting with StarTest in the field that many more members of test organizations can contribute to the automation effort using our approach and the automated tests are typically used for many years across several application releases.

## Better Coverage and Lower Maintenance Costs

**Test Plan** - StarTest users focus on designing test plans and test data. Using an outline editor, they develop objectives-based test plans that describe the requirements for testing the application under test. The test plans drill down from the general to the specific allowing the user to express test coverage in a way that is easy to review and refine. Our users have found that they discover many more important test case permutations using this approach. The test plan also provides the user with a means for associating each test with the set of steps and data required to execute it.

**Test Case Generator** – StarTest takes the data provided by the user in the test plan and automatically generates test scripts at runtime. Since the generated test cases are never saved, there is very little test code to maintain when the application under test changes. Our approach largely eliminates the classic problem with test case maintenance.

**Universal methods** – Many of the methods used to run a test case are predefined and are therefore unaffected by application changes. StarTest includes several methods for driving and verifying the application

under test. The following are a few examples:

**SO\_InputData** is a method for entering data into one or more fields on a form or page. This method drives the application under test by setting values to controls, which may be members of standard or custom classes. Without any customization, **SO\_InputData** can set text to a TextField, select an item from a ListBox and communicate with virtually any application control.

**SO\_VerifyProperties** and **SO\_VerifyText** are examples of functions used to perform verification against a form or

page and its controls. These methods capture the state of the application at runtime and compare to a set of expected results. Without requiring any additional code, properties can be verified such as the value of a TextField, the state of a CheckBox, the selected value of a ListBox or the enabled state of a PushButton, as well as any other property that is available for standard or custom controls.

*“Use StarTest to make sure you're getting the best return on investment from your automated testing dollars”*