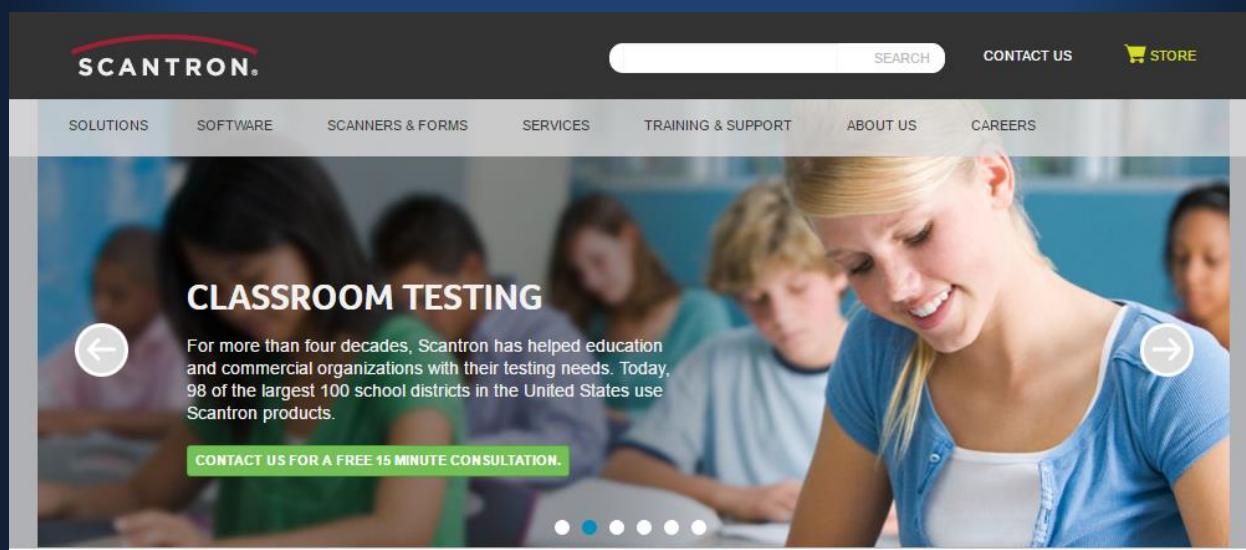




Case Study: Automation Testing for K-12 Districtwide Assessment Solution

Scenario: Automation testing of an educational System, Software as a service (SaaS) web application. The requirement was to automate approximately 8000+ test scenarios on the application for multiple OS/Browser combination using SilkTest as the automation tool.

Tool used: SilkTest (Automation tool), MS Excel (Data Handling), MS SQL Server (Automation results storing), XAMPP (Automation results web page hosting)

Duration: 4+ years

Overview of the solution provided: We used SilkTest to automate the scenarios for the Application Under Test (AUT). Two years ago, we started automation with emphasis on generic framework and scope for expansion. The AUT in itself is a group of 3 applications linked by functionality.

With the evolution of SilkTest, we have upgraded our framework to the new concept and approach of handling the objects which helped us in reducing the total execution time by 4 times of actual. Different OS & Browser combinations were covered which include Windows XP, Windows Vista and Windows 7 with Firefox 3.4+, IE 8.0+ and Chrome to latest version. Initially we started off with IE, and later Firefox browser & Chrome was added. The effort involved in supporting the Firefox browser was approximately 5% of the effort involved in developing the script for IE browser. We have the framework designed in such a way that same code runs on multiple environments like QA1, QA2, STAGING and PROD. Automation Scripts were divided into two sections

Automation of Critical Path Scenarios (ACPS):

Smoke test the new build provided for testing, which is part of build acceptance testing.

Scripts were run on different environments like Dev, INT, QA and Staging.

Modified version of automation scripts were run for Production environment.

Automation of Regression Scenarios (ARS):

ARS followed the smoke tests to cover the detail scenarios.

70% to 80% of functional test cases were covered.

Up to 50% of feature release test cases were automated in parallel.

Automation Results Publishing Tool: Framework was designed to view the results online using a browser while automation was running on the agent boxes. This gave the opportunity for all stakeholders to view the build status in real time even before the results were published to larger group in email.

Automation Workflow Diagram

