

# APC FORUM: SOFTWARE TEST AUTOMATION AT COMPUTER AID, INC.<sup>1</sup>

**Madeline Weiss**  
Program Director,  
SIM Advanced  
Practices Council

<p><b>Steve Heilenman</b> Chief Information Officer</p> <ul style="list-style-type: none"> <li>• Graduate of Widener University</li> <li>• Previous employers include Universal Trading Tech Corp, Vanguard Investment Co., Price Waterhouse, and Electronic Data Systems</li> </ul>	<p><b>Computer Aid, Inc.</b></p> <ul style="list-style-type: none"> <li>• Global IT services firm currently managing active engagements with over 100 Fortune 1000 companies and government agencies</li> <li>• Headquarters: Allentown, PA</li> <li>• Offices in Toronto, Suffolk, Shanghai, Manila, Sydney and Kuwait</li> </ul>
--	--

## DRIVING SCALABILITY WITH TEST AUTOMATION AT COMPUTER AID

Software testing is a perennial challenge for companies eager to launch bug-free software quickly. Like many CIOs, Steve Heilenman faced multiple challenges in improving this critical IT function. At Computer Aid, software testing was manual, heavily dependent on people who used inconsistent testing techniques, and didn't always resolve defects early in the development cycle. The cost and time spent on manual testing was also limiting the amount of new features being deployed in a given release. Testing cycles for each product took a minimum of six weeks. According to CIO Heilenman, it took about 4,000 hours to regression test one of their legacy products every time they enhanced them. In sum, it was a slow, error-prone, and expensive process.

The goal of reducing time and cost while improving software test quality was obvious, but the solution wasn't. The CIO and his team had tried multiple automated testing solutions, but with only limited success. These prior attempts were just trading time and cost from one area, to another area, with no additional value or cost reduction gains.

A better solution began to emerge at one of the CIO's weekly Innovation Friday meetings, in which a cross-section of thinkers and doers were invited to attend a one-hour meeting to explore new ideas. When the subject of testing software came up, one participant suggested that an open source testing product designed to test web-based applications (Selenium) could be adapted for more comprehensive test automation. With the CIO's encouragement, a customized interface was then built with this third party product, which was able to drastically simplify the test automation process. In addition to introducing the new tool, they re-engineered the testing process itself, creating or converting test scripts, integrating automated testing into the development

<sup>1</sup> This is part of a series of columns from the SIM Advanced Practices Council, an exclusive forum for senior IT executives who value directing and applying pragmatic research; exploring emerging IT issues in depth; and learning different, global perspectives from colleagues in other industries.

*MISQE is  
Sponsored by*



methodology, tracking results, and incorporating ongoing enhancements and upgrades.

The tool and simplified process, which took only a month to develop and implement, significantly reduced the testing cycle and dramatically improved testing consistency. Defects are now found much earlier in the development process. Developers are able to execute the same automated test scripts and fix/repair defects before getting to the test team. As a result, they have experienced a 300% reduction of defects found in the testing phase. For example, in one case a product that previously required six weeks (827 hrs) of testing only required two weeks (67 hrs) with the new system and processes, all while tripling the size of the system with new functionality. According to Heilenman, the testing time has “continued to shrink as we become more adept.” Overall testing costs have been reduced by 75%.

The new test automation system is based on some freeware and a custom-developed user interface which drastically reduces the complexity. The new process has added no additional time or cost to the development cycle. Testers are involved earlier in the process and the “cloud” is used for performance testing. The tool currently only works for the web development environment, but Computer Aid hopes to adapt it for other environments.

This success story demonstrates the value of persistence in seeking new solutions, utilizing brainstorming sessions with a variety of participants, innovation with open source products, and rethinking processes.