

Doing More with the Same—How to Make the Most of Testing Resources

By David Gehringer, Vice President of Marketing, Fanfare

In today's economic downturn, even successful companies are scaling back. Some are reducing growth, others are slashing their workforce. As companies contemplate strategies to tighten their belts, testing organizations will likely be in the crosshairs. Because testing has so little visibility at the top of the food chain, many view it as just another cost center. Company belt-tightening then really means that testing organizations will have to do more with the same (or less). Yet most QA teams and others already feel understaffed and unable to keep up with business demands. So, how are they now supposed to get the job done?

The answer is simple. To do more (and better) with what we have, we need to think about ways we can do things more efficiently. Then we need to look beyond our four walls to see whether new technology and tool advancements can make these changes possible.

Until recently, testing in the communications market was very much like the land that time forgot. Little was being done to provide more advanced testing tools or help testing teams keep pace with development. They had no choice but to rely on manual testing techniques, build testing tools in-house, and use outdated scripting languages. The drawbacks to these approaches have been costly—from wasted resources and unhappy workers to inefficient processes.

Manual testing, for example, is time-consuming, error prone, and not scalable. And with more complex products on the horizon, meeting the same quality objectives will require more people or longer testing cycles—neither of which is really an option in this economy. But building tools in-house is not the answer, either. Years ago, organizations thought that this would help them move toward automation. Instead it proved unsustainable and a drain on talent. Many of the brightest testers are now chained to tool maintenance duties and have very little time to test. As for scripting, talent is in short supply, and scripting languages can be brittle, making maintenance time-intensive and a burden on testers who must spend hours creating libraries. Not to mention, many testers have trouble understanding each other's scripts, and advanced tasks such as multithreading cannot be done in Tcl, Perl, or Python.

With the availability of new technologies and testing tools, testing organizations have more choices. Here are some ways that they can start doing more with the same (or less).

First, give tools to manual testers to increase their own productivity. You can expect productivity increases of 100 to 200 percent. Manual testers now can leverage a mature stable of testing tools to accelerate day-to-day testing processes. The key is to find tools that are somewhat automated and easy to use. They can then run a test once and, when the next build comes, simply push a button to repeat the test. If the tool can also run the test and generate a large share of the documentation, including pass/fail data, manual testers will save additional time. Until recently, we did not have good commercial tools for creating documentation. As a result, those who tried to share documentation and tests often found that the amount of work required to understand them cancelled out any time savings. Today's new tools can make semi-automated testing a reality, while providing a robust work product and clear, detailed documentation that makes test cases and reports valuable down the workflow.

Second, give automaters and scripters a tool that creates work products that can flow in an assembly-line fashion. This will make the construction of an automated test case a more efficient process for all involved, and improve communication about tests and defects. You can expect a three-to four-fold

productivity gain from your existing team, across the quality process. In the past, automaters and scripters had to take written test plans and reports, and build a script from scratch. In addition, they had to build and maintain countless libraries to protect their work as the product changed over time. Now automaters can use a tool to build sophisticated tests that are more quickly and easily maintained than Tcl. Being able to create a pass/fail analysis and rule with a single mouse click, without a complex equation, is a dramatic advancement. Such tools also can help them create and maintain looping, branching, and abstraction. Thus, they can more quickly create automated test cases and finally be free of arduous library maintenance. Better yet, they can take the work product of a manual tester—a semi-automated test—and simply modify it to get a completely automated test case. And it makes global updates easy should software or hardware change, so maintenance takes a fraction of the time to create scripts. In addition, this tool allows those without scripting experience to create sophisticated test cases.

Third, get a system that makes maintaining tests and running regression much less time-intensive, so you can free up resources. Up to 10 percent of the team that was stuck doing maintenance will be able to return to testing. Tools are now available to help regression teams run nightly tests and ensure reports are sent to the right person or system. More importantly, such tools are free of scripting weaknesses, so engineers can run tests that are written in a consistent format and that can be easily understood. Using these tools, they can figure out what the feature is supposed to do and quickly perform triage if the failure is due to regression system issues or an actual bug. If the output is configurable to existing reporting or bug systems, they can greatly reduce their time running regression and triaging test failures. And they can greatly simplify communication with everyone who writes test cases. This should increase test coverage, making time to market short and more predictable, while delivering a higher quality product. It also should allow talented testers to return to testing.

Although companies expect testing organizations to do more with the same (or less), this is not an impossible challenge given the new technologies and testing tools now available. More importantly, by leveraging these innovations, testing organizations have a chance to ensure people are doing the right job, resulting in new processes based on efficiencies. For many, this has resulted in increased code coverage, faster testing cycles, improved on-time delivery, and a surprise—happier engineers.