

# Learning New Tools And Implementing Them In Process And Practice

**Mike Kelly**

**First Published on SQAtester.com**

It is easy to fall into habits. When a deadline looms near and the work needs to get done tunnel-vision sets in and there is often little time for anything not mission-critical. It is important that we try not to fall into this trap. If your profession is software testing (and it must be, else you would not be reading this) then you already know how important software tools are in your day-to-day work. Automated testers rely solely on powerful suites of tools, but even manual testers use word processors, spreadsheets, file comparison utilities, format converters, screen capture utilities, diagnostic software, and bug tracking systems (to name but a few). What this article will do is remind you to get out there and explore what's new in software testing utilities and then try to get you on the path to continuous improvement by utilizing these new tools.

## **Learn What You Have First**

My first job in testing was as an automated tester. The company I was working for had just switched to the Rational Suite of testing tools and was only using Rational's basic tools (TestManager, Robot, and ClearQuest), which at the time, may or may not have been the only tools available. Regardless, over time Rational released new versions, and with each version there were new tools. My team, due to deadlines, turnover, and time constraints, failed to explore those tools as they came out.

It was not until the end of my second year (four versions later) that things slowed down and I started to look at some of the other tools. I discovered that Rational, those wacky programmers, had created tools that automated or simplified some of the most common tests we performed by manual or self-automated methods. Had we been using these tools when they came out we would have saved time, money, and sanity. We quickly kicked off a group-wide effort to learn the new tools and to develop a plan for including them into our testing processes. Our team was able to get involved in more projects and was able to provide more targeted types of testing.

It's important to take stock of what you have. Constantly review what you use. This will keep you up-to-date on what you have, allowing you to choose the best tool for the job, and will insure you don't waste effort re-inventing the wheel.

## **See What's Out There**

Now that you know what you have, it's time to go out and find what you need (or find something you didn't know you needed). For those of you with cash to waist, I recommend visiting the websites of tool manufactures. Here you can find powerful and user-friendly tools with great documentation and lots and lots of training resources. Even though you may not have the budget at the time, check them out periodically, see what they offer and compare that too your needs.

My favorite place to find testing tools is [www.shareware.com](http://www.shareware.com) (or some other shareware site). Free tools are the best (except for that whole lack of documentation, bugs, and viruses thing...). Regardless of the drawbacks of random downloads, get a box not connected to the rest of the network and try these things out. If you've ever thought to yourself "Gee, a simple tool that would do this would be nice. Maybe I could write one...." then rest assured that someone has. All you have to do is find it. Some of the best tools I have found have been free and were very limited in functionality, but they did exactly what I needed them to do.

By reviewing both commercial and free utilities you will have knowledge of, and access too, a greater arsenal of solutions to your testing dilemmas. This will allow you to do more types of testing, test faster, and reach a higher level of quality in your application-under-test. Try to set aside a couple of hours on Fridays when your mind is shot. Cruise the internet and see what you can find. If you find something, bookmark it, and either review it later or have someone on the team review it.

## **Using Them**

Now you know what's out there, but you still need to use them. This is not a small task. If you're not the manager of the group it can be even tougher. Most people are scared of change and consequently don't want to see their project become a pilot for a new technology. Phase in a new tool one step at a time by devoting only one or two test cases to it. Once you have a success and you can show someone it worked, you can start to build more of a following for it. Soon you will find project managers requesting the targeted testing the new tool offers.

Once you get in the habit of reviewing and implementing, it will become easier and easier. Just remember that you are doing this to find faster and better ways to test, not to simply implement technology for technology's sake. Only keep the tools that demonstrate that they add value. Also be sure not to settle for a tool that "kinda" does what you want it to do. If you don't like it, replace it.

Technology changes daily. Testing has fads just like any other field in technology. New tools that test specific technologies or are used in a specific software development methodology are released every day. Make sure your aware of the changes that can save your team time and money. Just like most things in software, a little bit of an investment up front will pay off big in the end. It may be hard to find the hour or two now, but down the road it will only become harder if you don't.