After All This Time, A Tool Maker Listens!

Scott Barber

For over six years now, I've been doing my best to provide meaningful feedback to testing tool vendors—particularly performance testing and load generation tool vendors. I've provided feedback as a paying client, as a services partner, as the author of over 30 articles about a particular tool, as a target of market researchers who call when doing their quarterly stock valuations, and as someone with enough voice in the field that vendors often ask me for endorsements of their products. For over six years now, my feedback has apparently been dismissed or ignored—until now.

About two years ago, a vendor started asking questions—the right questions. And not just of me, but of a relatively large group of software testing consultants, authors and test tool users. I was pleased with the questions that were being asked, but I was understandably hesitant to get my hopes up. I was more pleased when I was hired to do a feature and market analysis for the performance-testing aspect of a tool that was being built from scratch. And I was downright shocked when I arrived onsite to find notebooks full of printed materials with such labels such as *Open Forums*, *Vendor Forums*, *Roland Stens* and *Scott Barber*, all with sticky notes hanging out, each featuring a requirement number. These guys were really doing their homework.

Hope Springs Eternal

I got a chance to evaluate the beta and then to use the version 1 release, and again I was pleased—but not thrilled. I loved the tool's features, pricing model, paradigm and extensibility, but version 1 was still somewhat narrow in its overall capabilities, and a stack of those sticky-note requirements were yet to be implemented. Nevertheless, against my better judgment, I started having hope—hope that this testing tool vendor would build a tool that was actually designed, implemented and marketed to help good testers test even better.

I was a little surprised a couple of months ago when I received an invitation to attend an SDR (Software Design Review) for future releases of this suite of tools, but I'd been to these shindigs before. The vendor gets a bunch of executives from its big clients together in a room for a day immediately preceding its annual users' conference. The product managers talk about how great the next release is going to be; so great, in fact, that "you executives won't mind paying a little more for all these new features, right?" Then the "technical sales representative of the year" starts "demonstrating" the next version of the product by clicking through and reciting an obviously scripted set of words and activities intended to impress the roomful of executives who've probably never even seen the previous version in action. All the while, the product manager is virtually chanting, "See, we heard your feedback and we added this new feature... See, we heard your feedback and we fixed that rough spot..." Finally, the executives are invited to fill out their feedback forms with wish lists for next year—which they dutifully do, with at least some of the items on the list that their test managers gave them before the SDR. Then someone announces that the bar in the reception hall is open, at which point the executives dutifully submit their half-completed wish lists of items they barely understand anyway and go fetch their free beer. Same old, same old—right?

Not Just Business as Usual

Wrong. Let me tell you, this was different—and it was different in a lot of surprising and positive ways. This wasn't a one-day meeting in conjunction with a conference; it was three full days in conjunction with absolutely nothing else. And it wasn't a one-time event tied exclusively to the next release of the product; it was one of a semi-annual series of SDRs focused on everything from hot fixes to the five-year product vision. Not only that, but the vendor covered the hotel expenses for the attendees.

On the first morning when I took a seat and looked around the room, I noticed that instead of executives and sales partners, I was surrounded by the tool's actual users. How did I suss this out, you ask? Two ways. First, I actually knew several people in the room from other venues. Second, the people I didn't know weren't wearing ties, carrying day planners or exchanging business cards—they were wearing whatever was clean when they packed, were busy on their laptops taking care of whatever emergency had come up overnight, and were asking the people next to them if they knew a better way to solve the problem. Yes, these were genuine consultants and clients from around the world with actual firsthand experience using this tool on real projects—and many had also used the tool's competitors. To tell the truth, I don't think they were all fans of version 1, either. What I know for sure is that every single person in the room had his or her very own personal wish list for version 2 that he or she felt very

Scott Barber is the CTO at PerfTestPlus. His specialty is context-driven performance testing and analysis for distributed multi-user systems. Contact him at sbarber @perftestplus.com.

strongly about and wanted to share with the group.

As the meeting kicked off, I realized too that this event wasn't run by a product manager and a technical sales representative, but an entire staff. In fact, over the course of the three days, 30 members of the company's development, product management and marketing teams gave specific presentations, fielded challenging questions with honesty and grace, and solicited positive and negative feedback like professional facilitators. All the while, another 200-plus members of the company team were listening in via conference call. And just in case anyone who should have been there couldn't or didn't make it, audio and video were recorded throughout the entire three days. No, they didn't tape the demos. (Yes, we did get some demos, but I had to sign a comprehensive NDA, so I'm probably not even supposed to tell you that one of the new features they previewed was so cool that it actually left me speechless... much to the amusement of the others in the room. Luckily, I got the vendor to review this column and approve it as "suitable for public consumption.")

What they did tape was all of the attendees providing feedback, opinions, praise, rants, expectations and wishes. And not only did they record it, they had some mechanism (that I'm going to have to get them to explain to me someday) to index the recording by keywords so folks could actually do searches on the recording later.

Somebody's Listening

As if all that wasn't enough, it was clear throughout that the entire event was designed to solicit feedback—not to impress us with the next generation of "cool, but otherwise fairly useless" new features. In fact, the facilitators went so far as to list all of the potential enhancements under consideration on flip charts and have everyone in the room vote for a top 10 and bottom 10 to help them determine feature priority for the next release. Interestingly, by the end of the third day, it was obvious that on some features, the attendees had a very different recommendation than the vendor expected, and the vendor seemed to take this message to heart.

Of course, this all could have been a show to make us feel validated and listened to so that we'd all go start singing the praises of this vendor and the new version of its tool—but I don't think so.

Why? First, most of the SDR attendees are now part of a volunteer advisory board that has a virtual meeting twice a month and receives pre-beta software to test and provide feedback for, administrated by many of the same 30 individuals who ran the SDR. Second, many attendees are being recruited to develop training materials for the tool suite, compose contextually relevant help to ship with the product, and even write books in support of the next release. My third reason is personal: I know, respect and trust the man driving the vision of this tool. I don't always agree with everything this gentleman says, writes or envisions, but I do believe that he is a brilliant man driven to do whatever he can to enable everyone involved in creating software to create better software.

Over time, this feedback method will end up helping to develop a tool that users like and want to use.

So do you want to know who this guy is? I suspect that many of you are already familiar with Sam Guckenheimer's contributions to RUP (the Rational Unified Process). Now that Sam works for Microsoft, the company has presented the professional opportunity of a lifetime, inviting him to use his 25 years in software development to help teams develop software better, and to do it the way it should have been done in the first place. Further, Microsoft offered to build a supporting tool set from scratch based on Sam's plans. (To find out more about Sam's vision for software development, read his book "Software Engineering with Microsoft Visual Studio Team System" [Addison Wesley, 2006].)

I'm sure you already figured this out by now: The vendor I've been crowing about is Microsoft, and the version 1 tool is Visual Studio Team Suite 2005—which includes tools designed to assist and enable every member of a software development team to do his or her job more effectively and efficiently.

I felt compelled to share this with you not simply because I'm a fan of the method Microsoft's using to collect feedback about their tool.

What had the biggest effect on me is the end result of the method as compared to what I have witnessed from some other vendors in the past.

Over time, Microsoft's feedback method will end up helping to develop a tool that users like and want to use. In contrast, with the methods similar to the one I described at the beginning of this column, the vendor ends up developing a tool that's easy to sell to executives.

Naturally, these vendors see "easy to sell" as a sign of success. But the inherent problem with "easy to sell" as a measure of success is that it doesn't take into account the percentage of the time that the tool becomes either shelf-ware or the beté noire of the folks who are forced to use it.

If Microsoft stays the course with this method of collecting and applying feedback, going ahead to develop high-quality software addressing that feedback, this tool won't have time to sit on the shelf due to its huge following of happy and loyal users. Certainly, none of this is a given, but it makes Visual Studio Team Suite well worth keeping an eye on.

If the version 2 tool has an official name, designation or target release date, I don't know about it. I do know the tool's code name, but I'm not allowed to tell you. So you'll just have to wait until the beta release, when I'll give you a full feature review—and maybe then Microsoft will let me divulge the code name!