Remember Yesterday

Scott Barber

If you don't have a reference folder, I highly recommend starting one. That's the folder where I stow all the files that I can't quite figure out where else to put, but know that I will want again someday. Admittedly, I rarely find what I go in there looking for, but I never fail to find something interesting enough to make me forget what I went in there to find in the first place…at least for a little while.

I stumbled across a zip file in my reference folder the other day. This particular file caught my eye because it was simply called "Savoia.zip." It was dated 2/2/2004. "Hmmm…," I thought as I opened the file, "What do we have here?" There I found four articles and a presentation written by Alberto Savoia. The oldest was from STAREAST in May of 2000, when he was still CTO of Velogic Inc. The newest was from the July/August 2001 issue of STQE Magazine (now called Better Software) after he had become CTO of Keynote (still before his days at Google, and probably before he had even dreamed of Agitar).

As I skimmed through the documents, I couldn't decide whether I was depressed by the fact that we performance testers are still struggling with many of the same challenges he addressed, or encouraged by the fact that, by and large, it is no longer just the elite performance testers who are trying to educate developers, testers and managers about the importance of addressing these challenges while wrestling with them in virtual isolation. While we still have a long way to go, it seems that currently these are the challenges that software professionals worldwide are discussing, innovating around and collaborating over. Indeed, these are the challenges that have ever-increasing numbers of performance testing experts championing for better methods, tools and heuristics to convert today's stumbling blocks into tomorrow's building blocks for quality performance testing. That is a far cry from debating their importance.

The topics might surprise you, both because I'm certain that you are facing at least some of the challenges related to these topics on your current project and because of just how far ahead of the rest of us Savoia was in 2000-1. I spent that period finishing up my first "official" performance testing projects. During the ones that actually made it far enough into the development cycle for the stakeholders to start focusing on performance, I did a respectable job of finding critical performance issues that could have added to the list of spectacular launch day performance disasters that seemed so common at the time. Of course, thinking back, I know now that I found them by applying nothing more scientific than sheer dumb luck. On one project it was insufficient network bandwidth, on another it was a dramatically undersized thread pool, once it was an 8-second round trip to the mainframe, located on another continent. My all-time favorite was when someone forgot to install the permanent license keys on the servers! It was years before I figured out just how lucky I actually had been in terms of averting performance disasters.

These were issues that could have easily been detected by sending an e-mail to the network admin to request bandwidth stats, monitoring resources while the five functional testers were doing their initial inspection of the prototype, pinging the mainframe or validating software installs. Today, these are the kind of issues that development teams detect before they even tell the performance tester that they have put out a build that is stable enough to start building performance scripts against.

Savoia's presentations—"The Science and Art of Web Site Load Testing," "Predicting How Your Web Site Will Respond to Stress," "Three Web Load Testing Blunders and How to Avoid Them," "Understanding and Measuring Performance Test Results" and "The Science of Web Site Load Testing"—were all based on the same underlying themes:

- Acceptable performance is defined by system users, not by stakeholders or metrics.
- System usage models have to be realistic in order to generate meaningful results.
- To create realistic usage models and interpret the results, we need to understand and account for both human psychology and technically complex systems.
- Without at least a reasonable knowledge of and aptitude in mathematics and statistics, results will be misinterpreted.
- Not accounting for user abandonment will render your results not just inaccurate, but deceptive.
- The real reason we performance test is to proactively address and mitigate business risks.

While Savoia didn't specifically mention usability studies, he did refer to online behavior profiling and user demographics, which are fundamental components of a usability study. In the same presentation, he categorized numerical response times as "Good," "Borderline," "Unacceptable" and

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

“Virtually offline,” then left the numbers behind to focus instead on these user perception categories. In a subsequent article, he expressed users’ tolerance in terms of their expectations, perception and tendency to stick with what they know, as well as their uncanny ability to blame the owner of a Web site for poor performance even if the issue was their own 2400-baud modem. These messages clearly demonstrate that he was incorporating usability information collected by someone.

One of Savoia’s signature contributions to performance testers is his Web site Usage Signature (WUS) technique, used to create usage models for performance testing. This technique has been referenced in several books and has been used as a basis for subsequent models and techniques, because it captures the essence of what matters most in designing performance tests: realism. More than a few vocal individuals (and at least a couple of tool vendors) still challenge the importance of this realism in usage modeling by quoting anecdotes about critical performance defects that were uncovered by executing unrealistic models.

These critics seem to overlook the fact that Savoia advocated these “unrealistic” stress and endurance tests as well for just that reason. The point that Savoia makes about realistic usage models is that without realism in the usage model, there is no way to determine what the performance will actually *be* in production. In the same way that users don’t care whose fault poor performance is, they also don’t care how many performance issues were found and resolved if their overall experience is unacceptable.

Even when Savoia focused on entire systems, he never forgot to acknowledge users as an integral part of each system. This total system approach to performance testing is part of what makes Savoia’s work so valuable. Certainly, performance testing is not a “black box” activity, but the vast majority of the training and instructional materials available today focus on a particular system component. The way that Savoia focused on the entire system was what impressed me about his work when I first encountered it in late 2002.

Recently, another hot topic has been the mathematics behind performance testing. In my opinion, no other single item causes more performance-related surprises than “fuzzy math” (for example, not understanding the not-so-subtle differences between averages and percentiles, the relationship between outliers and sample sizes, or the intuitively misleading effect of averaging averages). Savoia only addressed this topic explicitly once, when he said “the greatest opportunity for voluntary, or involuntary, misuse of statistics is related to average page response time,” but of all the books, articles and presentations I’ve encountered about performance testing, his have the best balance of necessary and appropriately applied math. One of the few complaints I do hear about Savoia’s work is about the math involved, yet it really is nothing more than basic algebra. People are always trying to simplify his math further. The problem is that Savoia has already simplified the math from the integral calculus that it’s derived from to make it field-expedient enough to apply “in Internet time,” yet still accurate enough to be reliable.

Malesuada a, aliquam at, tincidunt at, urna. In imperdiet est. Curabitur interdum enim sed sem

Several years ago I had the pleasure of applying some of Savoia’s work on user abandonment to the load generation tool I was using at the time, which was how our ongoing dialogue began. Savoia didn’t demonstrate how to code user abandonment in tools. He did something more important by explaining the implications of poorly accounting for—or not accounting for—user abandonment on performance test results. This is the area of Savoia’s work in which I’ve noticed the least progress being made since I became active in the performance testing community. Maybe I should ask Alberto to contribute a guest column to reinvigorate this area.

Finally, Savoia never forgot to tie lessons back to the reason we do performance testing in the first place by demonstrating how a poorly performing Web site can adversely affect business via “major eBusiness disasters and/or chronic minor losses.” We all know stories about tragic software performance failures, yet most of our clients and/or bosses still ask us for return on investment analysis before they approve the budget for performance testing their applications. I’ve yet to read or develop an approach to calculating the ROI for performance testing that doesn’t involve some “fuzzy math” of its own. Maybe it would be more effective to use Savoia’s examples in response to the ROI requests; he is a corporate executive at least four times over, so if the examples are good enough for him, they should be good enough for other corporate executives, right?

You know, I think I should drop Alberto a note and let him know that I think his “old” performance testing articles just added about six items to my list of column topics for this year. I’m hearing lots of folks speculating that ’06 is going to be a significant year in terms of advancements in performance testing. I hope what that means is that we will close out this year significantly closer to actualizing the vision Alberto Savoia put before us more than five years ago.

Now, what *was* I looking for in my reference folder?