



## AT THE STPCON WITH DAN DOWNING

*An ATI Interview by Dion Johnson*

If you have a question about performance testing, there's a good chance that Dan Downing has the answer. With 28 years of technical and leadership experience as a programmer, sales engineer, product manager, senior manager, and consultant, he's a subject matter expert in load testing and creator of the Five Steps of Load Testing methodology.

Downing delivered two talks at the Software Test & Performance Conference this year, and after one of them, I was able to catch up with him and ask him a few questions. Those questions and a portion of his responses are provided in this article.

**Q** **DION JOHNSON** I just came out of a presentation that you gave on performance testing and data patterns and it was a pretty good presentation. I thank you for giving that. Would you like to talk a little bit about performance testing and these data patterns?

**A** **DAN DOWNING** Sure. I'd love to. I'm not exactly sure where to start. But anybody that's involved with performance testing knows that data is one of the biggest hurdles to get around. "What data am I going to use?" Typically we think of generating



our own data. Sometimes that's easy... But many times, that's not possible. There are complex systems where cultivating the data, or sort of generating on the fly isn't necessarily the easiest thing to do. [This was the topic of the presentation I did] with Ross Collard. He's a lead thinker in data patterns. He's published a dozen articles or so in STP Magazine and other places over the last year or so... And we talked about live data; the concept of live data being real data harvested from a production system. You have the benefit of doing this, of course, when you have a production system. If it's a brand new app, it's a little tougher to do that. So the notion is [that] live data is the best place to go to. And the notion of advanced patterns of data is [that] if you have the luxury of starting with harvested data from production, how do you then enhance it? Or what are some of the mechanisms for ensuring that you're truly covering the rich messiness of the real world so that your data is fine tuned to match your testing practice.



**Q** **DION JOHNSON** So you just mentioned the “rich messiness” which was an interesting concept that you talked about at the presentation; “rich messiness” and also “pristine data”. Could you talk a little bit about what that is and the importance of this “rich messiness”?

**A** **DAN DOWNING** Absolutely. If you talk to some of the luminaries in functional testing, [for example] James Bach [who's] here at the conference, he's promoted a notion of exploratory testing as being way better and more effective at finding real bugs quicker than the more traditional [scripted] test cases that you plod through. And one of the things that you encounter in exploratory testing is the idea that you're looking for the corner cases, the boundary conditions, the unexpected data inputs. “Nobody would ever type a string of 32 characters for a password.” But why not?

Let's try that and see if we get a SQL injection error and get the web server to sort of cough up some information that we can use to do a security breach on that app. So the notion of exploratory testing brought into the performance testing realm - the idea is that if you have the luxury of cultivating real data from a production system without messing with it, that is what we would call pristine data. [You don't exclude certain data transactions because they're full of errors or because they don't occur very often]. If you cultivate data from a production system it contains, by its inherent nature, the rich messiness of the real world. So people doing expected things, entering expected values and entering unexpected values. And when you can use that kind of data in

a performance test, then the realism quotient that we strive for in performance testing is maximized. But keeping in mind that there's this other set pattern or set way to take this data and use it or enhance it [so] you really always have to be very conscious of [the objective of the test] and what do I need to do to this cultivated data to enhance my test objective to ensure that I meet it. So that's kind of what we talked about today.

**Q** **DION JOHNSON** One other question that I have is, I think during the presentation there was the concept of spending about 20% of your time just dealing with test data. Do you feel that's a good amount of time? Do you need more time? Do you think it should be less time? I didn't really hear it [qualified] whether or not that was a good number, or the ideal number.

**A** **DAN DOWNING** I think Ross kicked that notion off by asking the audience, “so how many of you out there spend about 20% of your time dealing with the problem of data,” and only a couple of hands went up. And then somebody in the back of the room said, “Sometimes I wish I had more time,” because there's recognition, particularly in performance testing, of the

[Continue on page 22](#)

## At STP With Dan Downing

*(Continued from 21)*

---

importance of data. And yet, when you're dealing with the realities of time constraints and deadlines, and [limited] time before the go-live milestones that you're trying to achieve, and having enough react time to correct the problems that may surface, I think the audience was basically saying that they've discovered they don't have enough time to deal with the data, and that not too many people get to spend that much time really tuning the data to test. ■