

Subject: DevelopSense Newsletter

DevelopSense Newsletter, 26 May 2008



61 Ashburnham Road, Toronto, ON M6H 2K4

(416) 656-5160

info@developsense.com

In this issue: **New Tools for the Newsletter, CAST Update, Jerry Weinberg, and New Articles on My Web Site**

[New Tools for the Newsletter](#)

[The CAST Conference Update](#)

[An Interview with Jerry Weinberg](#)

[Updated Articles on My Web Site](#)

New Tools for the Newsletter

First, please note that you're receiving this newsletter because I believe that at some point you expressed an interest in receiving it, and because to the best of my knowledge and ability to stay organized, you haven't expressed an interest in *not* receiving it. If either of these assumptions is false, please let me know so that I can remove you right away. My first priority is to avoid pestering people who aren't interested.

That said, thank you, dear reader--because the newsletter now has a larger base of subscribers worldwide than I can handle with my own tools, so I'm now using a service called Constant Contact to help with composing and distributing it. I hope that this will make it easier for you and me both to make sure that you subscribe and receive the newsletter--and to unsubscribe if you don't want it any more. As always, I appreciate your feedback. Now to the real news.

The CAST Conference, Toronto, July 14-16, 2008: An Update



Since last I sent out a newsletter, the program for the CAST Conference has been released. I'm thrilled by the work that the conference committee has done.

As Early Bird registration ends this week, I thought it would be a good idea to send a reminder--get those registrations in soon!

A colleague recently pointed out that an important mission of our community is to remind people--and ourselves--that testing doesn't have to suck. Well, neither do testing conferences. The Conference for the Association for Software Testing 2008 is the kind of conference that I've always wanted to attend. The theme is "Beyond the Boundaries: Interdisciplinary Approaches to Software Testing", and the program is incredibly eclectic and diverse.

Start with the [keynotes](#):

- Jerry Weinberg on Lessons from the Past to Carry into the Future
- Cem Kaner on The Value of Checklists and the Danger of Scripts: What Legal Training Suggests for Testers
- Robert Sabourin (with Anne Sabourin) on Applied Testing Lessons from Delivery Room Labor Triage (there's a related article in this month's Better Software magazine)
- Brian Fisher on The New Science of Visual Analytics

Then there are the [one-day tutorials](#):

- *The Tester's Communication Clinic*, with Jerry Weinberg
- *Testing Mobile Applications*, with Julian Harty
- *Performance Testing Software Systems: Analyzing Performance Test Data*, with Scott Barber
- *From Craftsmanship to Leadership*, with Hung Nguyen

[Track sessions](#) include talks relating testing to

- improv theatre (Adam White), to music (yours truly and Jonathan Kohl)
- finance and accounting (Doug Hoffman)
- wargaming and Darwinian evolution (Bart Brokeman, author of *Testing Embedded Software* and one of the co-authors of the *TMap Next* book)
- civil engineering (Scott Barber)
- scientific software (Diane Kelly and Rebecca Sanders),
- magicians and magic tricks (Jeremy Kominar),

- file systems (Morven Gentleman),
- data warehousing (Steve Richardson and Adam Geras), and
- data visualization (Martin Taylor), AND
- four-year-olds playing lacrosse (Adam Goucher).

There will also be lightning talks and a tester competition.

Yet another feature of the conference is that Jerry is launching his book on testing, *Perfect Software and Other Illusions About Testing*. I read an early version of it, and I'm waiting for it with bated breath. It's a book that we'll all want to read, and after we're done, we'll want to hand to people who are customers of testing. For some, we'll want to tie them to a chair and *read it to them*.

The [conference hotel](#) is inexpensive, the food in Toronto is great, the nightlife is wonderful, the music is excellent...

More Information

You can find details on the program at <http://www.cast2008.org/Program>.

You can find information on the venue and logistics at <http://www.cast2008.org/Venue>.

Those from outside Canada should look at <http://www.cast2008.org/Venue#customs>

Paying the Way

If you need help persuading your company to send you to the conference, check out [this article by Jon Suzuki](#).

And if all that fails, you can likely write off the cost of the conference against your taxes, even if you're an employee. (I am not a tax professional, but INC magazine reports that you can write off expenses to "maintain or improve skills required in your present employment". Americans should see IRS Publication 970 (<http://www.irs.gov/publications/p970/ch12.html>), Section 12, and ask your accountant!)

Come Along and Spread The Word!

So (if necessary) get your passports in order, take advantage of early bird registration (if you register this week), and come join us. In addition (and I'm asking a favour here), please please /please/ tell your colleagues, both in your company and outside, about CAST. We want to share some great ideas on testing and other disciplines, and we want to make this the best CAST ever. And the event will only be improved by your presence.

So again, please spread the word, and come if you can.

An Interview with Gerald M. (Jerry) Weinberg



Being a conference chair has its advantages. Recently I was privileged to chat with Jerry Weinberg on why he's favouring CAST with his only conference appearance of the year, other than the Amplifying Your Effectiveness conference, of which he's a co-founder and host.

Michael: *You've been involved with computers for 50 years, and with giving people advice for almost that long. What do you suggest my first question should be, and how would you answer it?*

Jerry: Ask me why I chose this conference as my one of the year. And other things.

Michael: *Sounds good. So: why did you choose this conference as your one of the year?*

Jerry: Errors have been the principal issue in computing right from the beginning, as John von Neumann pointed out even before I got into the field (and that's really a long time ago). I wrote about testing as the opening topic in my first book, "Computer Programming Fundamentals" way back in 1960--and way back then, I already took flack from some reviewers who didn't think errors was a suitable topic for politically correct people. You'd think I had written about human excrement.

And you'd also think that as our field matured, we would have outgrown that prudish attitude about error--but we haven't. Back then, we had no professional testers. Testing was every developer's job (though they weren't called "developers" back then, or even "programmers"). We fought hard to have testing recognized as a profession of its own, and though we have people called "testers" today, we still have the prudes. In many organizations, testers are, sadly, considered lower-class citizens.

Testing holds a special place in my vision of the future of the computing profession as a whole. Why? Because testing is the first place where we generally get an independent and realistic view of what we are doing right and what we are doing wrong when we build new systems. We do get this view from Support (another area that's considered low-class), but by the time information arrives from Support, the people who put the errors in a product are often long gone and immune to learning from their mistakes.

Quite simply, if we don't learn to learn from our mistakes, we won't improve as a profession. And if we don't improve, we limit whatever good this amazing new (still) technology offers to humanity.

That's why I've made the status of testing and testers my first priority for some years, and why I'm debuting my book on testing fallacies and myths at CAST, the one conference that I feel is a creation of testers, by testers, and for testers.

Michael: *Recently you launched a new Web site, and your banner is "Helping smart people to be happy." Why did you choose that?*

Jerry: Most of the people in the computing professions are pretty smart, at least as measured by tests and the kind of technical work they accomplish. But so many of them haven't learned how to use their smarts on themselves. They can create wonderful systems, but when they use their brains to think about themselves, they often think themselves into depression.

I was like that, for a long time, until I began to figure out what I was doing to myself. I set myself the task of learning how to be happy, and as I began to succeed, I realized that one of the things that makes me happy is working with other happy people. So, selfishly, I decided I would devote

myself to helping my colleagues and students learn to share my happiness. Like most things I do, it's completely selfish--but has side effects that others may enjoy.

Michael: *Why not "Helping happy people be smart?"*

Jerry: If you're happy, you don't need to be smart. Smart isn't the only road to happiness. It's not that I mind helping people be smart, or smarter, but it's just not my primary goal. Nevertheless, I guess there are thousands of people out there who would say I've helped them grow smarter in some way. I think that's true of you, Michael, at least from what you tell me. I hope I've helped you be happier, too.

Michael: *Happier for sure, and smarter I hope. I've learned about both from conversations that I've had with you and other smart people. I remember once that Joshua Kerievsky asked you about why and how you tested in the old days--and I remember you telling Josh that you were compelled to test because the equipment was so unreliable. Computers don't break down as they used to, so what's the motivation for unit testing and test-first programming today?*

Jerry: We didn't call those things by those names back then, but if you look at my first book, and many others since, you'll see that was always the way we thought was the only logical way to do things. I learned it from Bernie Dimsdale, who learned it from von Neumann.

When I started in computing, I had nobody to teach me programming, so I read the manuals and taught myself. I thought I was pretty good, then I ran into Bernie (in 1957), who showed me how the really smart people did things. My ego was a bit shocked at first, but then I figured out that if von Neumann did things this way, I should.

John von Neumann was a lot smarter than I'll ever be, or than most people will ever be, but all that means is that we should learn from him. And that's why I go to a select number of conferences, like CAST and AYE, because there are lots of smart people there to learn from. I recommend my tactic to any smart person who wants to be happy.

Updated Columns on the DevelopSense Web Site

I've recently posted a whole whack of columns (from my Test Connection series in Better Software magazine) to the DevelopSense Web site.

Tests are often performed as isolated, atomic events, but the real world doesn't always have tidy preconditions, carefully defined activities, and readily observable postconditions. If we want to find problems effectively, sometimes we have to model the operation of software more realistically. [Go With The Flow](#) is about flow testing--doing one thing after another after another without resetting the system.

When we think about people who use our software, we usually focus our attention on favoured users--people that we like, that are well-trained, and who want to get useful information out of our product. [Users We Don't Like](#) starts with the Consider the Opposite heuristic, and asks what happens when we think about users that we don't like, that might not be so well-trained, or that might want to use our products maliciously.

As testers, we're often dealing with novel or innovative products. [McLuhan for Testers](#) examines Marshall McLuhan's Laws of Media--a set of four general questions that we can ask about a product to identify risks, problems, usage models and benefits.

Software development has long used engineering as a dominating metaphor for how we do things, but there are other disciplines to which we can turn for idea. [How Testers Think](#) looks at some of the ideas explored in Jerome Groopman's [How Doctors Think](#) and identifies parallels between testing and diagnostic medicine.

One of the big problems in testing is reification, the act of treating ideas and concepts as though they were material, countable things. [What Counts](#)--the title can be read in a number of different ways--notes the opportunities for deception and self-deception associated with trying to count requirements, test cases, and bugs.

One of the great myths of testing is that every test should have a specific, expected, predicted result--but we can find plenty of problems in software when our expectations aren't necessarily specific or predicted in advance. [Is There A Problem Here?](#) talks about the kind of valuable testing information we can obtain when we use a product and build expectations on the fly.

When should a given test stop? When we're exploring an application without scripted test cases, how do we decide whether to pursue the current line of investigation or move on to something new? [How Much Is Enough?](#) discusses a number of heuristics that suggest whether to stop testing or keep going.

[Learning the Hardware Lessons](#), despite the title, isn't really about hardware; it's about testing lessons learned in a hardware store. On a recent shopping trip, I encountered all kinds of problems with the larger system, only a few of which involved software--and yet I couldn't help but think that better software testing and management might have addressed them.

These articles and more are all archived and available at <http://www.developsense.com/articles>. Enjoy!

Other Stuff...

[DevelopSense Web Site](#)

[Rapid Software Testing](#)

[My Articles](#)

[My Blog](#)

Contact Information

Michael Bolton
61 Ashburnham Road
Toronto, ON M6H 2K4

CANADA

mailinglist@michaelbolton.net
+1 (416) 656-5160



Did you get this newsletter from a friend? Would you like to join the DevelopSense mailing list to receive our sporadically-published newsletter? Click on the icon to the left, or drop a line to mailinglist@developsense.com.

[Forward this newsletter to a colleague!](#)

 [SafeUnsubscribe®](#)

This email was sent to constantcontact@michaelbolton.net, by mailinglist@developsense.com

[Update Profile/Email Address](#) | Instant removal with [SafeUnsubscribe™](#) | [Privacy Policy](#).

Email Marketing by



DevelopSense | 61 Ashburnham Road | Toronto | Ontario | M6H 2K4 | Canada