BETTER SOFTWARE

INSIDE

SOA

A Look at the Process that Defines the Services

BUT WAIT, THERE’S MORE!
Get user buy-in on process improvement

SOME RESTRICTIONS MAY APPLY
Deciphering TDD’s fine print

January/February 2008 $9.95 www.StickyMinds.com

The Print Companion to StickyMinds.com
Is There a Problem Here?

by Michael Bolton

Suppose you were testing an application that you had never seen before with no time to prepare, no specification, no documentation, no reference programs, no prepared test cases, no test plan, and no other person to talk to. Can you test the product, recognize potential bugs, and assess their significance? I can.

This is the true story of how I tested a Web application without any preparation or test data—only the dates have been changed. Suppose today is June 1. I’m trying to book a flight to Europe for a conference that starts on October 22. I go to my airline’s Web site. The default departure date shown is June 8, a week from today; the default return date is June 15, one week later. I click the calendar icon. A calendar pops up showing the months of June and July. I navigate to the month of October, choose 21—the day before the conference—and close the calendar. The departure date now shows as October 21. I click the calendar icon for the return flight. The calendar shows June and July. I’m surprised.

I pause for a moment and ask, “Why am I surprised?” Is there a problem here? On a moment’s reflection, I recognize a principle: Return dates for a round trip should be later than the departure date, and the calendar that the site offers should reflect that. I’m irritated. I click the “next month” link several times until I see October 21. I click the calendar icon for the return flight. The calendar shows June and July. I’m surprised again.

I navigate back to the date selection page, set the return date to November 3, and click Search. A list of itineraries appears with a departure date of October 20 and a return date of October 27. I’m shocked. I never selected October 27 as my return date. I’m puzzled.

Why am I shocked and puzzled? Is there a problem here? I realize that when I changed my departure date, the software set my return date to exactly one week later. Is that what really happened? I’m curious. I navigate back to the date selection page, set the return date to November 3, and click Search. Another long delay, a list of itineraries appears with flights departing October 20 and (this time) returning November 3.

I look at the list. There are sixteen itineraries from which to choose. I don’t understand how they’re ordered. I’m confused. Why? Problem or no problem? After a little looking—I have to scroll to the bottom of the list—I see a very small set of radio buttons that allows me to rank the list by price, schedule, or price and schedule. Those buttons could be more prominent. I’m annoyed. Again.

Why annoyed? The people who designed and developed this site have been less than helpful at anticipating reasonable things that I might want to do. I click “Schedule.” The 6:00 p.m. outbound flight is just the one I want. The return flight is at 10:00 a.m., and in the other itineraries, I can see an afternoon flight that I’d prefer. None of the other itineraries offers the same morning flight, though. There is apparently a link by which I can select outbound and inbound flights separately. I click the morning flight link that says “Choose this flight,” anticipating a chance to choose inbound flights. I’m confronted with a page that says “No flights match your request.” I’m flabbergasted. I’m amused. I’m curious. I go back to the previous page and click all of the “Choose this flight” links. Every one returns “No flights match your request.” I’m frustrated. I’m getting
I am furious. I also notice that there’s a link at the top of the page that says “Canadian point of sale.” I’m in Toronto; couldn’t the software have sniffed my IP address and offered the Canadian site by default? I click the link, and I’m taken to the homepage for the site. Everything that I’ve done so far has been forgotten, and I have to start over from scratch. I’m furious.

Why am I furious? Is there a problem here? You bet. There are a lot of problems with this site, and my emotions are the first indicators. At every turn the site is delaying, confusing, frustrating, and annoying me—and amusing me, albeit not in a good way. Had I been a developer or tester of the site, I might have felt embarrassment, too.

A bug is something that bugs somebody who matters. Testers can’t be sure that something will bug someone, and so we must apply heuristics—useful but fallible ways of solving a problem or making a decision. Oracles—heuristic principles or mechanisms by which we might recognize a bug—including comparable programs, documents, mathematical principles, personal experience, and people—“live oracles.”

Emotional reactions are associated with a psychological and physiological phenomenon called “arousal,” which is the state of becoming alert or awake. For good testers, emotional reactions are oracles—trigger heuristics that wake us up to the possibility of a problem. Although emotional reactions don’t come with a guarantee, they can often aid our decisions about the existence or the significance of a problem.

We sometimes depend on logical and objective assessments of software at the expense of other ways of thinking—ways that include feeling. Most of the results that this system is returning are functionally correct, but that correctness doesn’t matter when there are problems that interfere with the customers’ goals. If we engaged emotions and empathy more often and more consciously, they would point us rapidly to things that are important to people. Try testing a program while simply pretending that we have some skin in the game.

Our customers constantly find bugs in our systems without any preparation, documentation, reference programs, or live oracles. Customers test the product by using it, and they recognize bugs and assess their significance and the associated value of our products. They can do that. So can we.

Michael Bolton lives in Toronto and teaches heuristics and exploratory testing in Canada, the United States, and other countries. He is co-author, with James Bach, of Rapid Software Testing and a regular contributor to Better Software magazine. Contact Michael at mb@developsense.com.