

Getting Software RITE

Jeff Patton

Something has always troubled me about designing and developing software. Let me explain.

It's both a sad and a lucky thing that my profession is also my hobby. (It's sad because I honestly need a real hobby.) I find software design and development extremely challenging and enjoyable. I love learning about a complex problem, noodling around with possible designs, and then in a quick flash arriving at what feels like a truly elegant solution. I then build the solution I've designed, making slight changes and improvements as I go, finally ending up with something I'm really proud of. Here comes the troubling part. It comes when the software is placed in front of my end users. They knit



their brows in confusion as they stare at the user interface, then say apologetically, "I don't get this." And that's when things are going well.

If you've never sat with or observed people using software you've designed, you're in for a real surprise. For most designers and developers, it feels like being hit with a brick. If you can't empathize, I admire you.

But this column is actually about a solution to the problem.

Validate your software design with users

I find that, in traditional software development environments, many designers don't routinely put their software in front of users. If the ultimate measure of well-designed software is how effectively it serves its intended purpose, then validating that it does so is critical, and the sooner the better. Many user-centered design practitioners do this with usability testing.

A formal usability test is a rather sciencey affair. It involves a number of different people in different roles, and goes a bit like this:

- Prototypers or developers build a user interface

prototype or piece of working software.

- Usability testers select user tasks to validate. A user task is an objective that users might have relative to the software they're using. "Sending an email message" in an email client is a task. "Finding a digital camera" on a shopping Web site is also a task. These tasks are what we'll ask users to perform.
- A facilitator introduces a user to the prototype he or she will be using and explains a little about the test, reinforcing the idea that it's the software being tested, not the user. The facilitator then asks the user to perform the task, or series of tasks, being validated.
- Observers then watch, recording observations about missteps the user makes and places where the user is confused or is struggling with performing the tasks.
- In a very formal usability test, observers are behind one-way glass or in a separate room watching the test as it is recorded with a camera. Being present in the room might bias the results by tipping off the user if he or she makes mistakes. Formal usability testers might record the face of the user, the computer screen, the user's hands on the keyboard and mouse, and, using special devices, even the user's eye motion on the screen.
- Several users are tested by the team.
- Observers record problems during each test, then results are combined.

This is all hard work. It benefits from well-trained and skilled testing practitioners. Furthermore, we're left with a bundle of well-justified fixes that must be carried out on our software. If this is a prototype, that's not so bad. But when we make fixes and run another batch of users through testing, we often find that our fixes need fixes—or that one fixed item reveals another problem. If we've been testing with working software, the cost of change is even higher, and making changes might be difficult. We often ship software with known problems because we're out of time and money to make changes.

Lighten up validation

Now, validation needn't be this formal or difficult. I'll admit I don't have the skills, patience, and equipment to perform this sort of testing.

Alternatively, usability testing often involves lightweight paper prototypes, with a human who's moving bits of paper around to simulate computer interactions. This approach is surprisingly quick and effective. I've done a fair bit of this.

When doing this, I experience the same letdown I described at the article's beginning, when my best design efforts are met with confusion by my user. But here's where the story gets better. In paper, I can quickly change the prototype to fix problems, grab another test subject, and try again. And I'm always pleased with how quickly things begin to improve using iterative testing.

Turn your validation process into your design process

This test-fix-test-fix cycle continues until the software starts to feel pretty tight. This approach is called RITE (Rapid Iterative Testing and Evaluation). The name was coined by Michael Medlock and his colleagues at Microsoft, who documented the approach in their paper "Using the RITE Method to Improve Products." (See the sidebar for more information on this and other further reading.)

However, the approach those authors described didn't use paper prototypes; they tested the working software. They described bringing test subjects in, testing, and having usability practitioners and developers observe the tests. After usability testers and developers discussed problems, developers went to work fixing critical problems as quickly as possible before the next test subject showed up. They tested again to confirm they had removed some of the identified errors, and repeated the process until the quality of usability felt sufficiently high.

This sounds like a great idea, but while I don't find nearly enough organizations that invest in usability testing, I find nearly none that invest in RITE. But recently I had the pleasure of talking with Catherine Courage and Craig Villamor of salesforce.com. We talked because I was interested in their adoption of very short agile development cycles, which in other cases have often laid waste to rigorous user experience practice. Rigorous user experience practice implies

Further Reading

- C. Courage and K. Baxter, *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques*, Morgan Kaufmann, 2004.
- J. Nielsen, "Paper Prototyping: Getting User Data before You Code," 14 Apr. 2003, www.useit.com/alertbox/20030414.html.
- M.C. Medlock et al., "Using the RITE Method to Improve Products: A Definition and a Case Study," 2002, www.microsoft.com/downloads/details.aspx?FamilyID=3b882eb1-5f06-41d9-baba-d39ad13bc3ff&displaylang=en.
- C. Snyder, *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*, Morgan Kaufmann, 2003.

some amount of user research, some time for design, and some time to validate that design with users before sending it to development. An agile development process often doesn't leave time for a lot of this sort of work. "We've really embraced the RITE method here," explained Catherine. I was surprised.

Catherine went on to describe that they tested all their software using electronic prototypes before it went into development, and much of it again afterward.

Build a pool of ready users

To support this effort, they need a huge pool of users from all over the world, because salesforce.com's product is global. Salesforce.com builds its pool by asking users to volunteer in various ways—for example, through their "idea exchange" (ideas.salesforce.com), invitations in their user-experience blog, and user-experience marketing campaigns. "Thousands have opted in," explains Catherine. Every month, dozens of users from that pool work with salesforce.com staff to remotely test prototypes and working software. "Our employees love our customers, and our customers are incredibly enthusiastic," Catherine says.

Keep testing quick, simple, and frequent

Here are some ways salesforce.com facilitates testing:

- The company uses simple screen-sharing software (GoToMeeting) to share prototypes electronically. Testers pass control of mouse and keyboard to their users to watch them work.
- A simple conference call lets a facilitator talk to users and lets the user explain likes, dislikes, and areas of confusion.

- The company sometimes uses TechSmith's Morae to record the screens and audio as users interact with the software.
- A projector displays the user's interactions with the product in a large format so that all the product team members can easily observe them.

But this is the RITE method, so the goal is to fix the problems, not just identify them. "I'll often start making changes while they're still testing," says Craig.

A typical weekly testing cycle has salesforce.com designers testing Monday, fixing Tuesday, testing Wednesday, fixing Thursday, and then testing again Friday. By the end of the week, things are getting pretty good. User-experience people, product managers, developers, and others sit in on the call and watch the screen. "When the user gets it right, the whole room cheers!" describes Craig. (I assume their conference phone is on mute.)

When everyone participates in testing, they understand how the software works and why the design is the way it is. Catherine explains, "There's no UI spec—the UI spec is the prototype."

Once you've gone RITE, it's hard to go back

Catherine described her experience going back to traditional testing after salesforce.com purchased a product to add to their suite: "It was like [the movie] *Groundhog Day*—we kept seeing the same errors over and over and were dying to fix them."

For me too, iteratively testing and improving my design has become addictive. The real joy of software development for me comes in delivering great software. ☺

Jeff Patton is an independent consultant. Contact him at jpatton@acm.org.