

Users as Testers

Adrian Kosmaczewski

2007-06-24

Both users and black-box testers approach software with apparently similar attitudes; however, they have fundamentally different goals. With this article I will try to expose some of my ideas about this dichotomy.

Users and Testers

When it comes to black-box software testing, there are two main differences between users and testers, in my opinion; the first one, maybe the most obvious, is that the user usually paid for the software, while the tester is paid to use (and break) it. This is very important, since all efforts should be done to make that the final user gets the best possible experience, and the smallest possible number of bugs. After all, the user is paying for the software, and she needs it to fulfill some business problem or some particular need, be it run a company, play a game or write a mail.

The tester uses the software with the sole objective of **finding problems**. The user tries to **avoid them** at all price.

The second main difference, is that if your users become (by any chance) de facto testers of your software, it means that you have shipped buggy code, or even worse, you might consider that having your users test your product is the right thing to do. Both of which are bad ideas anyway:

The worst thing about this form of testing is the remarkably bad impression you will make of your company. When Userland released the first Windows version of their flagship Frontier product, I downloaded it and started working through the tutorial. Unfortunately, Frontier crashed several times. I was literally following the instructions exactly as they were printed in the tutorial, and I just could not get more than 2 minutes into the program. I felt like nobody at Userland had even done the minimum amount of testing, making sure that the tutorial works. The low perceived quality of the product turned me off of Frontier for an awfully long time.

(Spolsky, 2000)

The web brought a whole new medium for software deployment; through a web browsers, users were suddenly able to access new versions of applications almost

instantaneously, and this brought the blurry line between users and testers to another level.

The “Eternal Beta” Situation

Since the late 90s, several companies have relied on users’ input to find flaws in their systems. Even nowadays, several popular applications used by millions of people are presented as “beta”; for example, Google’s webmail, Gmail:

(Source: Gmail.com, as of May 17th, 2007)

Gmail’s own “Terms of Use” specify clearly that

In addition, you understand and agree that the Service is provided on an AS IS and AS AVAILABLE basis. Google disclaims all responsibility and liability for the availability, timeliness, security or reliability of the Service. Google also reserves the right to modify, suspend or discontinue the Service with or without notice at any time and without any liability to you.

(Gmail)

Through the above statement, Google is able to modify the appearance, functionality and usefulness of Gmail, in any way, at its sole discretion; in this environment, end users appear as ad hoc testers, sometimes making the big headlines:

Seems like every couple of months reports surface about outages with Gmail (and Yahoo Mail too). Lately reports about Gmail accounts being disabled and people being unable to access their mail for hours, if not days, have been heating up. Google says it has received feedback from some Gmail users who were having trouble accessing their accounts but that the incidents were isolated and have been fixed. (...) Problems like these, even sporadic and eventually fixed, can damage the reputation of any Internet service provider, but particularly one that’s increasingly marketing itself as the go-to company for hosted applications aimed at consumers and businesses.

(Mills, 2006)

This phenomenon has taken the web development industry as a storm:

Following Google’s lead, many companies stick “beta” on their logos and leave it there for months or years. Gone are the betas that get released to a limited set of known but external testers, with formal product management follow-up interviews. The concept of “beta” as a time period or stage of development has fallen away, and been replaced with beta as a way of setting expectations, or excusing faults, about the current state of the application.

(Hedlund, 2006)

And sometimes, this attitude is absolutely and completely explicit:

We don't have the resources to do full usability tests, so you are our worldwide usability assessment.

(Jotlet Blog, 2006)

Conclusion

Black-box testers are an intermediate kind of user: they click on the user interface, launch processes and input data, and actually they try to get things done with the software like any other user would do; but their ultimate objective is not the same as that of the final user: they are discovering flaws before any user do, which will prove to enhance the perceived quality of the final product, drive up sales, increase the trust in the organization and lower the chances of project failure. And even better, all of these factors will have long-term effects, both for the producers and the clients of the final product.

References

Gmail, "Gmail Terms of Use", [Internet] http://www.google.com/mail/help/terms_of_use.html (Accessed May 17th, 2007)

Hedlund, M.; "Web Development 2.0", February 2006, [Internet] <http://radar.oreilly.com/archives/2006/02/web-development-2-0.html> (Accessed May 17th, 2007)

Jotlet Blog, "So what do you think?", September 2006, [Internet] <http://blog.jotlet.net/?p=11> (Accessed May 17th, 2007)

Mills, E.; "Gmail problems... again?", C|net News.com, March 23rd, 2006, [Internet] http://news.com.com/8301-10784_3-6053454-7.html (Accessed May 17th, 2007)

Spolsky, J.; "Top Five (Wrong) Reasons You Don't Have Testers", April 2000, [Internet] <http://www.joelonsoftware.com/articles/fog0000000067.html> (Accessed May 17th, 2007)