Categories » Products » News & Events » Community » About Us » SmartBear.com

Top 5 Software Blunders of 2013

January 17, 2014 by Gregory Mooney Leave a Comment

赞 24 g+1 13 Tweet 28 Share 7 Stu

HealthCare.gov
Take health care into your own hands
Dropbox is experiencing issues.
Check out @DropboxOps for more information.
INDIANA FAMILY & SOCIAL SERVICES ADMINISTRATION
FSSA

Happy New Year! It's that time of year that we like to take a look back at what's transpired over the last 12 months and find ways to improve ourselves for the future—which, as history has shown, not enough of us actually do.

2013 was a tumultuous year in the world of software. If you ask me, last year goes down in the books as the year of the defect—as if "13" wasn't unlucky enough.

连接错误 [重试]

The following list of tech blunders gives us a lot to think about as we enter 2014, not just for betterment software testers, but for the sake of the entire planet.

5. United Airlines Hands Out Free Tickets

This first testing blunder is chuckle-worthy and, unlike most events on this list, actually negatively impacted the company more so than it did its customer base.

So, a few lucky users of the United Airlines website found a bug and ended up with tickets that would usually cost thousands for what amounted to pocket change. Luckily for UA, they discovered this glitch early enough to avoid any serious losses and honored those tickets when the news hit the media waves in September. Great PR move, United Airlines!

That is, if the story had ended there.

A month later, users who signed up for a frequent flier card but canceled out of the process were able to go buy tickets usually worth thousands of dollars for a few dollars – yes, again.

United Airlines had a hissy fit this time around, blaming the people who discovered the bug, and decided not to honor those tickets because they felt

Search Our Blog

Search this website… Search

Subscribe to our Blog

Email *

Select topic(s):

□ Functional Testing

□ Performance Testing

□ All Topics

For more subscription options, click here.

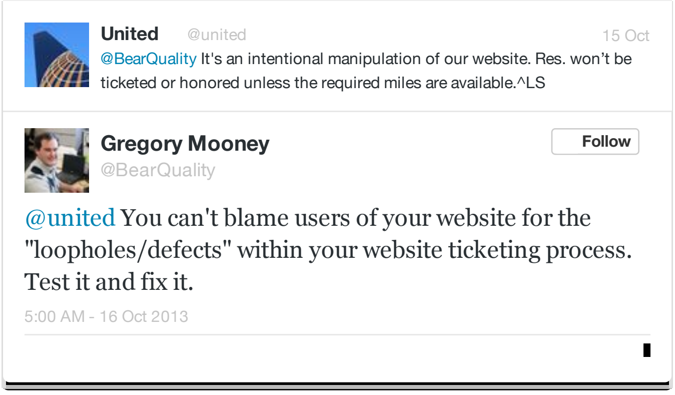
Subscribe!

Stay Connected With Us!

Popular Posts

14 Ways to Contribute to Open Source without Being a Programming Geniu…
A Taste of Salt: Like Puppet, Except It Doesn't Suck
There Ain't No ROI in Software Testing
C11: A New C Standard Aiming at Safer Programming
5 Reasons It's Time to Ditch MySQL
Fabrice Bellard: Portrait of a Super-Productive Programmer
13 Things People Hate about Your Open Source Docs
A Short History of Computers in the Movies: Panel Lights, Spinning Tap…
Creating your own TestSteps in soapUI
15 Code Editors For the iPad – For Free or Very Cheap

their customers were “intentionally manipulating” their site. Generally, companies are grateful to those who find issues with their website and let them know. Apparently, not in this case. Bad PR move, United Airlines.



United @united 15 Oct

@BearQuality It's an intentional manipulation of our website. Res. won't be ticketed or honored unless the required miles are available.^LS

Gregory Mooney @BearQuality

@united You can't blame users of your website for the "loopholes/defects" within your website ticketing process. Test it and fix it.

5:00 AM - 16 Oct 2013

At the end of the day, no one really cared as much as United Airlines, since we just expect “air-bus” services to not get it anyway.

4. Dropbox Outage

Do you get cold sweats about uploading all your data into the public cloud? Are you worried it could be hacked or you won’t have access to it someday? Frightening thought, isn’t it?

Those nightmares became a reality for Dropbox customers back in May when the site went down for, wait for it… wait for it, yeah, an hour.

The *Dropbox Panic of 2013* will go down in the books as the most inconvenient time in history to upload puppy GIFs or download those last-minute reports you did for your impatient boss.

Seriously though, this brief outage did get us thinking about living in a world in which everything is uploaded to the cloud. It’s an idea that companies have struggled to accept for years and it still seems to be a hot point. Maybe the cloud is better served as a backup of our files and/or to develop and test software remotely. The fact of the matter is, your data is more secure in the cloud than on your home PC, but forget it if you can’t access it for more than a minute.

3. CBOE Defect

So wait, you said you knew about a major defect in your software, didn’t fix it, and you still released it? For minor software bugs, this isn’t a big deal (it’s impossible to have zero bugs anyways), but if there is a defect that eliminates the user’s ability to “use” the product and you release it anyways, you’ve just shot yourself in the foot.

The defect really came back to requirements traceability. The exact issue being:

“preliminary staging work related to the planned reconfiguration of our systems in preparation for extended trading hours on CBOE Futures Exchange (CFE) and eventually CBOE options.”

You see, there was a requirement that was needed and, in order to meet it, the CBOE scrambled to release, skipping test procedures in the process. In the end, CBOE was fined $6 million for regulatory failures.

The point here really is that, yes, sometimes we all cut corners in order to get our software released on time. However, depending on the magnitude of software we are releasing it may be best to simply request a later release date.

A good example is medical devices. Would you develop and release buggy software that may affect the life or death of a human being? The point is, depending on the software, you may be able to get away with a few bugs, but don't be surprised if they come back to bite you. It's like stuffing food down the drain without a disposal; a few months down the road, all those fruit flies are going to come in swarms.

2. FSSA Bug Handlings

Federal regulations state that encryption and authorization techniques are to be used to securely send and receive confidential information. Unfortunately for about 188,000 people in May, the Indiana Family and Social Services Administration (FSSA) disclosed bundles of private information, including social security numbers, to the wrong recipients. Oops!

What's even more curious is that it took the FSSA about a month (and some change) to fix the error and to reach out to those affected to let them know that they were indeed impacted by the error. Not only are people in their system struggling as is (the organization's name is proof of that), but now they have to worry about their personal information being handed out to strangers who could possibly use it to exploit them.

Not only did it take a considerably long time to fix what seems like a fairly obvious bug, but the fact of the matter is that this is the type of defect that should be found quickly during testing procedures during development of the data system. The truth is that software bugs will always find ways to make it past the development stages. The question is which ones make it through—you need to make sure it isn't the bugs that count.

1. Healthcare.gov Disaster

This software testing blunder takes the cake at No. 1, and if you're a software tester you may revere this event as the "defect" that put software testers to the forefront of the media — even if it was only for a couple of weeks.

Hundreds of thousands of users flocked to the site on October 1 (the launch day of the healthcare.gov website). Signup processes failed, browsers crashed, and only a miniscule amount of the people who entered this debacle in online technology successfully signed up and bought health insurance—all while the US government was shut down, due to Obamacare provisioning. We want our tax dollars back!

Turns out this debacle occurred because there was a very brief period of testing done after the website had been developed. The contractors who developed the website said they only had two weeks to test when they knew they needed months. Many people were to blame, but what made things even worse is the Republicans tried to use this as a reason why this Obamacare system can't work.

No, actually it doesn't work because no one in the government knows how to project manage a software development project properly.

There are two reasons why we should be happy this happened:

1. It squelched all those conspiracy theorists who actually think our government is smart enough to conspire anything. *then again, maybe that's what they want us to think…
2. It finally opened the world's eyes to who software testers are and why they're so important. When the words "Agile" and "software testing" were being spoken in the same sentence on national news, software testers everywhere rejoiced.

Learn From Our Mistakes and Move On

2013 was the year that software testers came into the limelight for many reasons. Our world now entirely relies on the software that runs it, and it's time that we all take a long, hard look at what this software is doing.

Everyone (including those who aren't naturally technically inclined) needs to become informed before they start typing their information into what seems like a harmless form on a seemingly secure site. It's very easy to be fall into a false sense of security, but that's a very dangerous (and naive) way to go about life.

If 2013 brought anything good, I hope it birthed a more informed society and a need to take software testing more seriously than ever. Testers know the drill, but too often their friends and colleagues question what it value they actually bring to their company. The fact of the matter is, if you don't take software testing seriously, you are either doing the world a disservice or you shouldn't be developing software.

This post was originally published in Testing Circus magazine.

See also:

- Top 10 Web Outages of 2013
- Top 5 Software Quality eBooks of 2013 (Free for Download)
- Get ready for TX: Tester Experience

Filed Under: Bugs, Software Quality, Test Automation, TestComplete, Testing

0 comments ★ 0

Start the discussion…

Best SmartBear Software Blog Share Login

Be the first to comment.

ALSO ON **SMARTBEAR SOFTWARE BLOG** WHAT'S THIS?

10 Live Wallpapers to Gussy Up Your Android Device

1 comment • a month ago

Redhead Fae — Totally getting Moonlight Live.. thanks~

Exploring Programming Languages' Science and Folklore

2 comments • a month ago

Eamon Nerbonne — You cannot draw the conclusion that "There isn't a large ROI from worrying about …

How to Build a Web Service in 5 Minutes with Go

3 comments • a month ago

Bruno Albuquerque — Meanwhile, all the code can be found here anyway: https://github.com/brunoga/go-...

10 Signs You're a 'Woman in Tech'

2 comments • 12 days ago

Wanza — Yep. That describes me. I do think there is a broad generalization that applies to …

Subscribe Add Disqus to your site

SmartBear.com | Forums | Support | SoapUI.org | LoadUI.org |
Partners | Contact Us | Shopping Cart

Privacy | Terms of Use | Site Map

Contact Information
Phone: +1 978-236-7882
Email: ben.austin@smartbear.com

Return to top of page