Google Testing Blog

Introducing "Testing on the Toilet"

Sunday, January 21, 2007

We want you to write more tests. Yes, you. You've already been told that tests are the safety net that protects you when you need to refactor your code, or when another developer adds features. You even know that tests can help with the design of your code.

But, although you've read the books and heard the lectures, maybe you need a little more inspiration, tips, and prodding. And you need it to be in a place where when you see it, you can't ignore it.

That's where we can help. We're the "Google Testing Grouplet," a small band of volunteers who are passionate about software testing.

We're unveiling the public release of "Testing on the Toilet": one of Google's little secrets that has helped us to inspire our developers to write well-tested code. We write flyers about everything from dependency injection to code coverage, and then regularly plaster the bathrooms all over Google with each episode, almost 500 stalls worldwide. We've received a lot of feedback about it. Some favorable ("This is great because I'm always forgetting to bring my copy of Linux Nerd 2000 to the bathroom!") and some not ("I'm trying to use the bathroom, can you folks please just LEAVE ME ALONE?"). Even the Washington Post noticed.

We've decided to share this secret weapon with the rest of the world to help spread our passion for testing, and to provide a fun and easy way for you to educate yourself and the rest of your company about these important tricks and techniques.

We'll be putting episodes on this blog on a regular basis and providing PDFs so you can print them out and put them up in your own bathrooms, hallways, kitchens, moon bases, secret underground fortresses, billionaire founders' Priuses, wherever. Send your photos

and stories to TotT@google.com and let us know how Testing on the Toilet is received at your company.

And meanwhile, keep writing those tests.

Labels: TotT

58 comments :

Maciej January 24, 2007 at 3:59:00 PM PST

Will you be posting the back catalog of past TotT installments?

Reply

Alberty Pascal January 25, 2007 at 4:32:00 AM PST

Great idea. Thanks guys !

Reply

Slyoldfox January 25, 2007 at 4:36:00 AM PST

This comment has been removed by the author.

Reply

Doesn't Work At McDonalds January 25, 2007 at 3:51:00 PM PST

Uncle John's Bathroom Reader watch out! :-)

Reply

Pepijn January 25, 2007 at 4:28:00 PM PST

A similar thing was/is in use at a place I worked. People made an effort to stay in the toilet for a long time (doodling on their PDA) and later claimed they were reading the work notes... :)

Reply

Dom De Vitto January 25, 2007 at 5:31:00 PM PST

Some of my best agile development has been done whilst 'at the porcelain hot-desk', so this is a great idea.

I'm sure it will be highly absorbent.

Reply

Kyle January 25, 2007 at 5:37:00 PM PST

when i'm on the toilet, i really just want to do my business, not read about boring test advice. something more thought-provoking would be nice..

Reply

Jason R. January 25, 2007 at 5:51:00 PM PST

I've just installed a Wi-Fi printer in my bathroom. Can Google just print the publications directly?

Reply

Surfer51 January 25, 2007 at 6:06:00 PM PST

My new Dihydrogen monoxide powered mini laptop has helped me tremendously when harold calls...

Now I have TotT...whoa...life is good.

Reply

apersaud January 25, 2007 at 7:10:00 PM PST

Can you make it a podcast like feed so that I can download the pdfs whenever a new one is available on the iTunes store? That would be awesome!

Reply

Elo January 25, 2007 at 7:31:00 PM PST

Testing: "Bombs, away!"
listen

plop!
Success!

Reply

bucky January 25, 2007 at 7:32:00 PM PST

Sounds great to already have reading material available rather than having to print something out, get it from the printer, then carry it in

Reply

Jared January 25, 2007 at 8:29:00 PM PST

Google doing this...mmmmkay....

However, I shudder to think about the PHBs who will use this to other ends...

Reply

doubleA January 25, 2007 at 8:38:00 PM PST

Saves toilet paper too!

Reply

Joe #302 January 25, 2007 at 9:43:00 PM PST

Hm. I have to say, I think this is misguided.

I've worked for a long time in the testing/QA engineering field for software (seems like every company calls it something different, and there are different expectations of each, but that just leads to my point). My point is that QA needs to have a bright line between it and development. No doubt, developers need to be good at creating good unit testing. By the same token, good QA/Testers need to be good at development.

But the two should remain separate. Go to any mature industry's engineering department, and ask how much of the release testing is done or defined by the actual development engineers. The answer will be you'll be laughed out of the office. There's good reason for this.

This is a sign of the immaturity of the software development industry. We're

getting there, though. I've worked for a couple of the truly big Seattle software interests, and they are slowly getting more mature in how they deal with QA. The key is, (for as good and happy as developers doing a good job developing unit testing for their own work): SDEs are not QAEs, and the distinction should not be blurred, if you want to maintain true quality.

Reply

robin January 25, 2007 at 10:06:00 PM PST

and providing PDFs
How about just a printer friendly version without comments?

Reply

Eric January 25, 2007 at 10:08:00 PM PST

When I go to the john, I create the Bush cabinet, I don't worry about anything more than watching Cheney and Rice being flushed to a place they belong.

Reply

Shona January 25, 2007 at 11:58:00 PM PST

I think you may be missing a certain level of irony in your "positive comments"... You may also be missing a certain level of fibre from your diet if you are regularly on the loo long enough to read through so much...

Reply

Ortwin January 26, 2007 at 1:37:00 AM PST

A picture of those bathrooms would be very informative and entertaining. Please...

Reply

Alessio Pace January 26, 2007 at 2:25:00 AM PST

Simply great!! :-)

A related curiosity: a friend of mine had in his bathroom some maths formulas he wasn't able to remember easily :-)

Reply

netron January 26, 2007 at 3:47:00 AM PST

i guess this gives a whole new meaning to a "Googlebomb".

Reply

Joel Odom January 26, 2007 at 3:57:00 AM PST

These folks obviously need to eat more fiber.

Reply

amit January 26, 2007 at 5:52:00 AM PST

Holy shit !!!

Reply

Major Tom January 26, 2007 at 6:10:00 AM PST

To respond to Joe #302
You're wrong. I'm a developer in a fortune 500 company and I test everything I do. By the time it goes to Alpha test, only field anomalies break my code. So to say "SDEs are not QAEs, and the distinction should not be blurred, if you want to maintain true quality." you're, well, full of crap. Your statements should be flushed. Not everyone has poor software developers working for them. A true programmer can take the project from cradle to grave, or more aptly from napkin scribblings to deployment and not have problems.
I have 3000 sites with my software for over three years and have had 2 bugs related to errors on my part. Proper Software Engineering includes heavy testing during development.
I also know of one company that spends Billions (Microsoft) on everything you stated, and well, let their end product speak for itself. It does not support your conclusion.
I rest my case.
-T-

Reply

Michael January 26, 2007 at 6:52:00 AM PST

One word: awesome. (okay, I lied) And, thanks.

Reply

strimpster January 26, 2007 at 6:55:00 AM PST

To respond to Joe #302
I think that you may want to rethink your position and help usher your company into quicker development. Everyone in your company should be involved in QA activities because it is everyone's job. If the developers learn from their own mistakes, then they should be constantly improving and not have to bother you with errors that could be fixed by them. The QAE should be worrying more about integration testing rather than unit testing, but the SDE should also ensure that their code is scalable and will not cause errors due to an inability to forsee problems.

Reply

Robert January 26, 2007 at 6:56:00 AM PST

ewwwwwww. possible marketing backfire? "google" & "crap"...

Reply

Michael January 26, 2007 at 7:41:00 AM PST

I have a simple question as a former-programmer, now business guy, between dev and QA.

Why not put pay performance targets on both sides as incentives per testing release?

In other words, each QA person gets $10 for each bug they find (up to 10, or whatever). Each development person gets $10 for the number of bugs not found under a certain target (10, or whatever).

Seems like an easy way to get people more motivated about the whole testing process.

Reply

stewart January 26, 2007 at 8:23:00 AM PST

To respond to Major Tom:

I commend you on having only 2 bugs in your portion of code with a 3000 site deployment. However, I'm wondering if you're an anomaly and am concerned that this methodology sets a dangerous precedent. Considering the approach of a development engineer should be, and in the best companies is, fundamentally different from that of a test engineer. The DE is solving a positive functional problem, while the TE is addressing the validation and exceptions associated with the DE's solution. When presented with requirements from (in most cases) Marketing PMs, the DE considers how to make the feature/product work, while the TE considers where and how the feature/product will be deployed (functionality and performance testing) and under what circumstances it might break (stress, robustness, and negative testing). As much as we try in our smaller organization to unite the DE and TE mindset to save headcount and to arguably improve unit testing prior to the hand-off to TEs, they will always remain firmly divided.

Reply

fool January 26, 2007 at 8:59:00 AM PST

Its a great idea ... Well as long as I am not asked to do 'pairing' inside the toilet.

Reply

Russell January 26, 2007 at 10:15:00 AM PST

I had very good success writing my own tests as I was writing libdomainkeys. As I wrote the code, I would create the test. Every branch was labelled with a comment naming the test which forced that branch to be executed. Too bad our text editors don't (usually) support hot-linking between source files.

Reply

leskippeur January 26, 2007 at 11:44:00 AM PST

Gives a new meaning to "toilet training" !

Reply

BPDecaf January 27, 2007 at 4:37:00 AM PST

Thank you for your sane and sensible comments Major Tom.

Reply

Jeff Fry January 27, 2007 at 6:56:00 PM PST

Michael said...

I have a simple question as a former-programmer, now business guy, between dev and QA.

Why not put pay performance targets on both sides as incentives per testing release?

In other words, each QA person gets $10 for each bug they find (up to 10, or whatever). Each development person gets $10 for the number of bugs not found under a certain target (10, or whatever).

Seems like an easy way to get people more motivated about the whole testing process.

This sounds like a well intentioned idea, but it leads inevitably toward disastrous consequences.

Reply

Ken Kuhlman January 27, 2007 at 7:35:00 PM PST

And here I've always thought of testing as a crappy, #2 priority. Thanks for throwing another log on the fire & helping me see the light!

-Ken

Reply

Jeffrey January 27, 2007 at 10:22:00 PM PST

Just reading material? You can do better.

(From eXtreme Feedback)

Reply

Phil January 27, 2007 at 10:37:00 PM PST

Re: SDE's are not QAE's.

I think a point to be made here is that engineers of any sort tend to get accustomed to an idea or algorithm or whatever they've been working on. They see it as working just fine. The QA people are there to ensure that your idea of fine agrees with everyone elses. Sure, maybe it doesn't have bugs, but maybe the interface isn't as intuitive as it might be, or whatever. The point is, the more people that look at it before production, the more likely it is to be useful and useable. If you want to streamline your development timeline, you include QA people in the development cycle, to add their input as the project progresses. That way problems can be identified and resolved before it becomes a major problem. But you already knew that - it's your basic concurrent engineering model you learned about in college. I hope.

Reply

Adamm January 27, 2007 at 11:13:00 PM PST

I actually googled for "Linux Nerd 2000" to see if I was missing out on some cool new magazine. Heh.

Reply

untactical000 January 28, 2007 at 9:27:00 AM PST

Before anyone of you reaches for the toilet roll,make certain that you've read the fine print first!

Reply

Pradeep Soundararajan January 28, 2007 at 9:56:00 PM PST

Michelle from Google suggested to share the following with you all:

Kudos to the effort of brainstorming the idea of testing bumper(Harry Robinson style) stickers and tests in toilets.

Here are my 2 cents:

Toilet stickers:

1. The better you code, the lesser it stinks.
2. You expect the toilet to be clean, as we expect your code to be.
3. You mess, we spot, you clean.
4. You dropped the paper roll by mistake. Fix it !

5. Thanks for remembering that you need to go back and fix the bug I found.
6. for ($i=0,$i< ($i--1);++$i) - I am not reminding your coding error :)
7. 234/8 - No! It's a cricket match score and not 234 bugs per 8 KLOC.
8. I am your friend but unfortunately our customers aren't.
9. The more time you spend here, the less time you distract me and the more bugs I find.
10. Did you come here running to think about the explanation you want to offer on why you missed that bug?
11. Don't believe me when I say; your code is working fine so far. I just want to give you a bigger surprise.
12. I heard you say, "Please leave me alone" and that is why I am not reminding you of the bug that took the longest time for you to fix it.

Tests:

1. An end user thought of running our program with a platform we no longer support. Should he get an error message - "You are not supposed to that" ?
2. "Arggh! The page is corrupted." - Does this mean you were unable to handle an exception?
3. I am going on a 3 week vacation and while I come back, I will have a report that my script generated on the reliability test. Are you sure, I wont see anything in RED?
4. "Sorry, no donut for you" - Who got the donut then?
5. The voice kept breaking, when I was using a dial up connection. Did ._ _ _ know?
6. I hear too many click sounds when I login. Do I have an option to hear less clicks?
7. I refer to James Bach's 36 heuristics to test. Do you use them to code?
8. When firewall is ON, your code isn't.
9. Proxies do not by pass bugs.
10. I tried closing the window in between. It goes away smoothy without the error message you expected to see.
11. I am always amazed at your coding skills but not the way you build your code.
12. His code never merges with anyone. Beware if you are merging with him.
13. I found a bug with your unit test script and hence I know what to test.
14. Last time my developer forgot to enable the logger. He spent a week convincing me to reproduce the issue.
15. Psst! Don't tell this to anyone. I look at our competitor product known issues and test them on our product. This helped me catch bugs faster :)

[Reply](#)

abhi_3182 January 29, 2007 at 12:37:00 AM PST

aa guys nice idea, when is the release...

Reply

cameron January 29, 2007 at 5:13:00 AM PST

debugging code sucks; testing rocks; debugging test code really, really sucks

Reply

Ben February 1, 2007 at 3:31:00 PM PST

Michael said...

I have a simple question as a former-programmer, now business guy, between dev and QA.

Why not put pay performance targets on both sides as incentives per testing release?

In other words, each QA person gets $10 for each bug they find (up to 10, or whatever). Each development person gets $10 for the number of bugs not found under a certain target (10, or whatever).

Seems like an easy way to get people more motivated about the whole testing process.

This is a horrible idea.

I will not improve quality. It will not improve relations between business, developers, and testers. In fact, it will make relations worse within the company and can lead to real bug escapes that impact customers.

This will only lead to finger pointing and time wasted chasing unimportant and duplicate "bugs".

As a tester, I have often had to make the decision about reporting a bug as one or 100,000. I don't know if the 100,000 errors my automated test found are due to 100,000 different problems in the code or just one. I could report them separately, or I could report them in groups, or I could pick up the phone and talk to a developer than could help me determine how the bug should be reported.

And what if the bug exists because a QA person did a poor job of reviewing the requirements? I could easily provide input to requirements that I know will lead to bugs just to get a bonus.

Not all bugs are due to a developer making a mistake in their piece of code. Some are due to bad requirements given to the developer. Some are due to bugs someone else created. If one developer's code does not properly interact with another developer's code, who is to blame? If the issue to bad documentation and requirements, do we blame QA?

And then we could argue about what is and what is not a bug.

There was a Dilbert cartoon many years ago about the idea of rewards for fixing bugs: Now, I'm going to code me a new minivan.

Reply

ahoier February 4, 2007 at 4:28:00 PM PST

I like the idea...some of the best ideas come to me when I'm in the "john" ^^ lol

Reply

Seo Practices Guide February 7, 2007 at 7:34:00 AM PST

That is a great idea, I will talk my boss into it, a "SEO Testing Practices Program in the Toilet", amazing, thanks.

Reply

Woody March 6, 2007 at 4:49:00 AM PST

Great idea to share the testing experiences from Google with the whole world!

However you promised to post TotT's "on a regular basis". The last TotT episode was published over a month ago. So what's this regular basis?

Reply

Nate Murray March 7, 2007 at 8:10:00 PM PST

I also had a question about the regularity of this. I think its a great idea, but I've had a hard time finding the collection of these. Is there a place for people to submit these, because I could come up with a few episodes.

Reply

Komail Noori August 17, 2007 at 5:16:00 AM PDT

I like the idea. I will definetely try it out.

Regards,
Komail Noori
Web Site Design - SEO Expert

Reply

M@rce October 2, 2008 at 9:15:00 AM PDT

Excelente aporte a toda la comunidad de testers en todo el mundo; nos dá una visión más cercana del entorno laboral y relacional dentro de Google. Gracias y saludos!!

Reply

Rohit Jain July 10, 2009 at 9:57:00 AM PDT

You get it! It's one serene place for thinking of creative ideas.

Kudos!

Reply

Karit July 11, 2009 at 10:13:00 PM PDT

Seeing Testing on the Toilet is producing something weekly I thought I would have a go at doing something similar with Testing on the Bog. So if others are looking for testing ideas to place on your toilet door please have a look.

Reply

Suresh August 26, 2009 at 5:40:00 AM PDT

It seems funniest activity i ever seen...!! any way time saving work

Reply

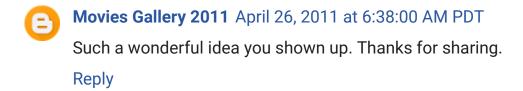**Sam** September 15, 2009 at 3:56:00 PM PDT

Until I saw this post, I always thought it was Google Tricks of the Trade.

I guess sometimes truth is stranger than fiction!

Reply

rnbresearch September 19, 2009 at 5:58:00 AM PDT

Interesting text. You have a nice blog. Keep it up!

Reply

Movies Gallery 2011 April 26, 2011 at 6:38:00 AM PDT

Such a wonderful idea you shown up. Thanks for sharing.

Reply

Movies Gallery 2011 April 26, 2011 at 6:39:00 AM PDT

Nice post.. Glass Art

Reply

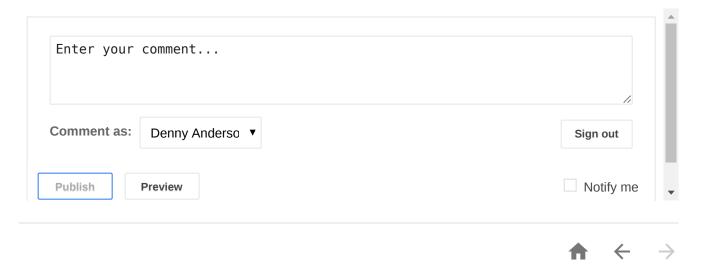**Saenthan 360** February 13, 2013 at 3:03:00 AM PST

Great Idea

Reply

Mohammad Rakib Amin January 21, 2017 at 9:03:00 PM PST

10 Years have passed, you're still the Lighthouse.

Reply

Add comment

The comments you read and contribute here belong only to the person who posted them. We reserve the right to remove off-topic comments.

Enter your comment...

Comment as: Denny Anderso ▾   **Sign out**

Publish   **Preview**   ☐ Notify me

🏠   ←   →

Google

Google · Privacy · Terms