Why Didn't You Find That Bug?

Michael Bolton
DevelopSense
http://www.developsense.com
michael@developsense.com

But first…

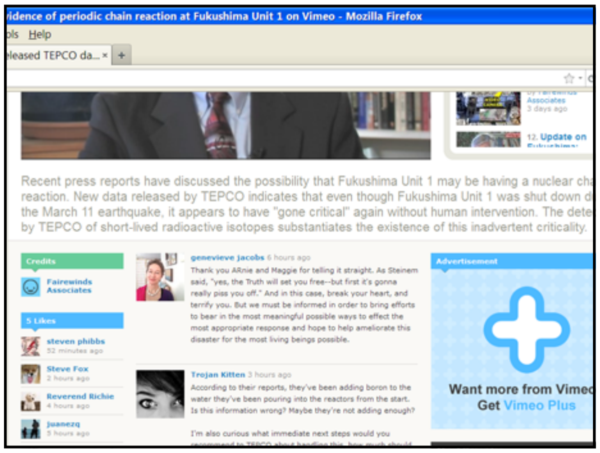
…a few Burning Issues.

When Agile testing focuses on confirmation, it becomes old testing wine in new Agile bottles.

Acceptance tests are misnamed. You don't know you're done when they pass; you know you're NOT done when they fail. They should be called "rejection checks".

Testing never ends; it only stops. So it's folly to think that ALL testing for a given amount of work can fit into that work's iteration.

Testing quality into a product is like measuring height into a basketball player.



idence of periodic chain reaction at Fukushima Unit 1 on Vimeo - Mozilla Firefox

ols Help

leased TEPCO da...

Associates
3 days ago

12. Update on

Recent press reports have discussed the possibility that Fukushima Unit 1 may be having a nuclear cha reaction. New data released by TEPCO indicates that even though Fukushima Unit 1 was shut down d the March 11 earthquake, it appears to have "gone critical" again without human intervention. The dete by TEPCO of short-lived radioactive isotopes substantiates the existence of this inadvertent criticality.

Credits

Fairewinds Associates

5 Likes

steven phibbs
52 minutes ago

Steve Fox
2 hours ago

Reverend Richie
4 hours ago

juanezq
5 hours ago

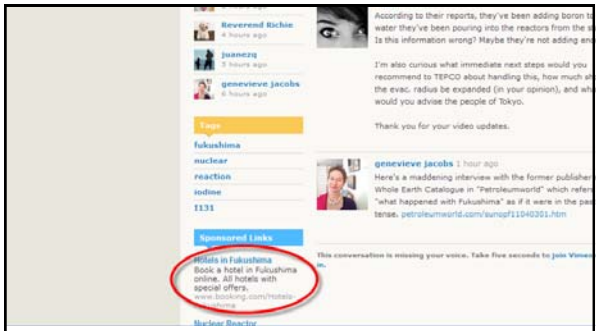
genevieve jacobs 6 hours ago
Thank you ARnie and Maggie for telling it straight. As Steinem said, "yes, the Truth will set you free--but first it's gonna really piss you off." And in this case, break your heart, and terrify you. But we must be informed in order to bring efforts to bear in the most meaningful possible ways to effect the most appropriate response and hope to help ameliorate this disaster for the most living beings possible.

Trojan Kitten 3 hours ago
According to their reports, they've been adding boron to the water they've been pouring into the reactors from the start. Is this information wrong? Maybe they're not adding enough?

I'm also curious what immediate next steps would you

Advertisement

Want more from Vimeo
Get Vimeo Plus



Reverend Richie
4 hours ago

juanezq
5 hours ago

genevieve jacobs
6 hours ago

Tags

fukushima

nuclear

reaction

iodine

I131

Sponsored Links

Hotels in Fukushima
Book a hotel in Fukushima online. All hotels with special offers.
www.booking.com/Hotels-Fukushima

Nuclear Reactor

According to their reports, they've been adding boron to water they've been pouring into the reactors from the s Is this information wrong? Maybe they're not adding en

I'm also curious what immediate next steps would you recommend to TEPCO about handling this, how much sh the evac. radius be expanded (in your opinion), and wh would you advise the people of Tokyo.

Thank you for your video updates.

genevieve jacobs 1 hour ago
Here's a maddening interview with the former publisher Whole Earth Catalogue in "Petroleumworld" which refer "what happened with Fukushima" as if it were in the pas tense. petroleumworld.com/sunopf11040301.htm

This conversation is missing your voice. Take five seconds to join Vimeo in.

When managers ask, "When are we going to be finished testing?" they really mean "When are we going to be finished *developing*?"
So why ask the testers?

Waiters don't tell the chef when food is ready to serve. Why are testers asked to tell managers when products are ready to ship?

One purpose for testing is to move known unknowns toward the known; unknown unknowns toward known unknowns.

Testers are not in the confidence-building business. Testers are in the unwarranted-confidence demolition business.

Counting yesterday’s passing test cases is as relevant to today’s project as yesterday’s good weather report is to today’s picnic.

The “expert” stage of the Dreyfus Model of Knowledge Acquisition “transcends reliance on rules, guidelines, and maxims”. So, presumably, if you’re a real expert, you’ll ignore the Dreyfus Model.

“Works as designed” means exactly the same thing as “Doesn’t work…as designed.”

When a manager asks “Why didn’t you find that bug earlier?” ask “Why didn’t you remind me to find it earlier?”

Managers often ask the testers…

Why didn't you find that bug?

Some answers…

- because we didn't know where it was
- because the people who put it in didn't know where it was either
- because it was well hidden
- because the people who put it in did a bunch of things to *make sure* it was hidden
- *because of the illusion that letting it stay hidden would save time*

How to Win at Hide and Seek

- Play it in a tidy room
- Keep the count short
- Choose your opponent wisely

Another answer…

Because we were busy looking for bugs that aren't there.

Is Regression Your Biggest Risk?

- Before the Agile Manifesto was declared, a group of experienced test managers reported that regression problems ran from 6-15% of discovered problems
- In Agile shops, we now (supposedly) have
  - TDD
  - unit tests
  - pairing
  - configuration management
  - build and version control
  - continuous integration
- Is regression a serious risk?
- If so, can testing (whether automated or not) fix it?
- Is regression really a symptom of problems elsewhere?

Regression Problems Are *Symptoms*

- If you see a lot of fires in the neighbourhood
  - any given fire is not your biggest problem
  - you might want to notice construction problems or arsonists
- If you see a consistent pattern of regression
  - the failing tests are not your biggest problem
  - you might want to note a favourable environment for regression

Another answer…

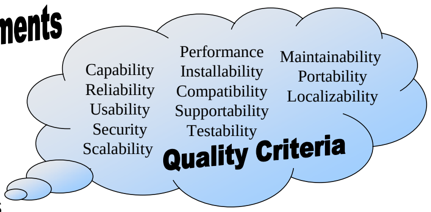
Because we used a set of models that was too limited.

One Way to Model Coverage: Product Elements and Quality Criteria)

SFDPOT -- San Francisco Depot

Product Elements

- Structure
- Function
- Data
- Platform
- Operations
- Time



Another answer…

Because we were busy finding other bugs.

Test Session Effectiveness

- A "perfectly effective" testing session is one entirely dedicated to test design, test execution, and learning
  - a "perfect" session is the exception, not the rule
- Test design and execution tend to contribute to test coverage
  - varied tests tend to provide more coverage than repeated tests
- Setup, bug investigation, and reporting take time away from test design and execution

Modeling Test Effort

- Suppose that some burst of test activity takes two minutes
  - this is a highly arbitrary and artificial assumption—that is, it's *wrong*, but we use it to model an issue and make a point
- Suppose also that it takes an extra eight minutes to investigate and report a bug that we found with a test
  - another stupid, sweeping generalization in service of the point
- In a 90-minute session, we can run 45 feature tests—*as long as we don't find any bugs*

How Do We Spend Time?
(assuming all tests below are *good* tests)

Module	Bug reporting/investigation (time spent on tests that find bugs)	Test design and execution (time spent on tests that find no bugs)	Number of tests
A (good)	0 minutes (no bugs found)	90 minutes (45 tests)	45
B (okay)	10 minutes (1 bug, 1 test)	80 minutes (40 tests)	41
C (bad)	80 minutes (8 bugs, 8 tests)	10 minutes (5 tests)	13

Investigating and reporting bugs means….

SLOWER TESTING or…

REDUCED COVERAGE …or both.

- In the first instance, our *coverage* is great—but if we're being assessed on the number of bugs we're finding, we look bad.
- In the second instance, coverage looks good, and we found a bug, too.
- In the third instance, we look good because we're finding and reporting lots of *bugs*—but our *coverage* is suffering severely. A system that rewards us or increases confidence based on the number of bugs we find might mislead us into believing that our product is well tested.

What Happens The Next Day?
(assume 6 minutes per bug fix verification)

Fix verifications	Bug reporting and investigation today	Test design and execution today	New tests today	Total over two days
0 min	0	45	45	90
6 min	10 min (1 new bug)	74 min (37 tests)	38	79
48 min	40 min (4 new bugs)	2 min (1 test)	5	18

Finding bugs today means….

VERIFYING FIXES LATER

…which means….

EVEN SLOWER TESTING or…

EVEN LESS COVERAGE …or both.

•…and note the optimistic assumption that all of our fixed verifications worked, and that we found no new bugs while running them. Has this ever happened for you?

With a more buggy product

- More time is spent on bug investigation and reporting
- More time is spent on fix verification
- Less time is available for coverage

Not only do we do more work…

…we also know less about the system

With a *less* buggy product…

(that is, one that has had some level of testing already)

- We've got *some* bugs out of the way already
- *Those* bugs won't require investigation and reporting
- *Those* bugs won't block our ability to test more deeply

So, programmers, please consider this heuristic:

Test early, and test often!

Test Early and Often!

- Recurrent themes in agile development (note the small A)
  - test-first programming
  - automated unit tests, builds, and continuous integration
  - testability hooks in the code
  - lots of customer involvement
- The ideas are
  - to increase developers' confidence in and commitment to what they're providing ("at least it does *this*")
  - to allow rapid feedback when it *doesn't* do *this*
  - to permit robust refactoring
  - to increase test coverage and/or reduce testing time

Test a product as you build it!

Testing Is Questioning

- Note that I just gave you a compelling-looking table, using simple measures, but notice that we still don't really know anything about…
  - the quality and relevance of the tests
  - the quality and relevance of the bug reports
  - the skill of the testers in finding and reporting bugs
  - the complexity of the respective modules
  - luck

…but if we *ask better questions*, instead of letting data make our decisions, we're more likely to *learn important things*.

To ask better questions…

To answer them more quickly…

To find bugs more easily…

To fix found bugs faster…

Focus on testability!

We Testers Humbly Request… (from the developers)

- Developer tests at the unit level
  - use TDD, test-first, automated unit tests, reviews and inspections, step through code in the debugger—whatever increases your own confidence that the code does what you think it does

A less buggy product takes less time to test.

We Testers Humbly Request…
(from the whole team)

- Focus on testability
  - log files
  - scriptable interfaces
  - real-time monitoring capabilities
  - installability and configurability
  - test tools, and help building our own
  - access to “live oracles” and other forms of information

Speed up the decision:
"Problem or no problem?"

Acknowledgements

- James Bach
- Dale Emery
- James Lyndsay
- Cem Kaner

Want to know more?
Some other resources

- “How Much Is Enough?” (Better Software column)
  - http://www.developsense.com/publications.html
- Why Is Testing Taking So Long?
  - http://www.developsense.com/blog/archive/2009_11_24_archive.html
  - http://www.developsense.com/blog/archive/2009_11_25_archive.html
- Testing vs. Checking
  - http://www.developsense.com/blog/archive/2009_08_29_archive.html, and linked posts
- Disposable Time
  - http://www.developsense.com/blog/archive/2010_01_17_archive.html

Michael Bolton
DevelopSense
http://www.developsense.com
michael@developsense.com