The Most Important Thing

How Mozilla Does Security and What You Can Steal

Johnathan Nightingale
Human Shield
Mozilla Corporation
johnath@mozilla.com

So you want to steal a security architecture...

Do you actually want to get better?

Do you care about responsiveness?

Can you let go of secrecy?

Why steal from us?

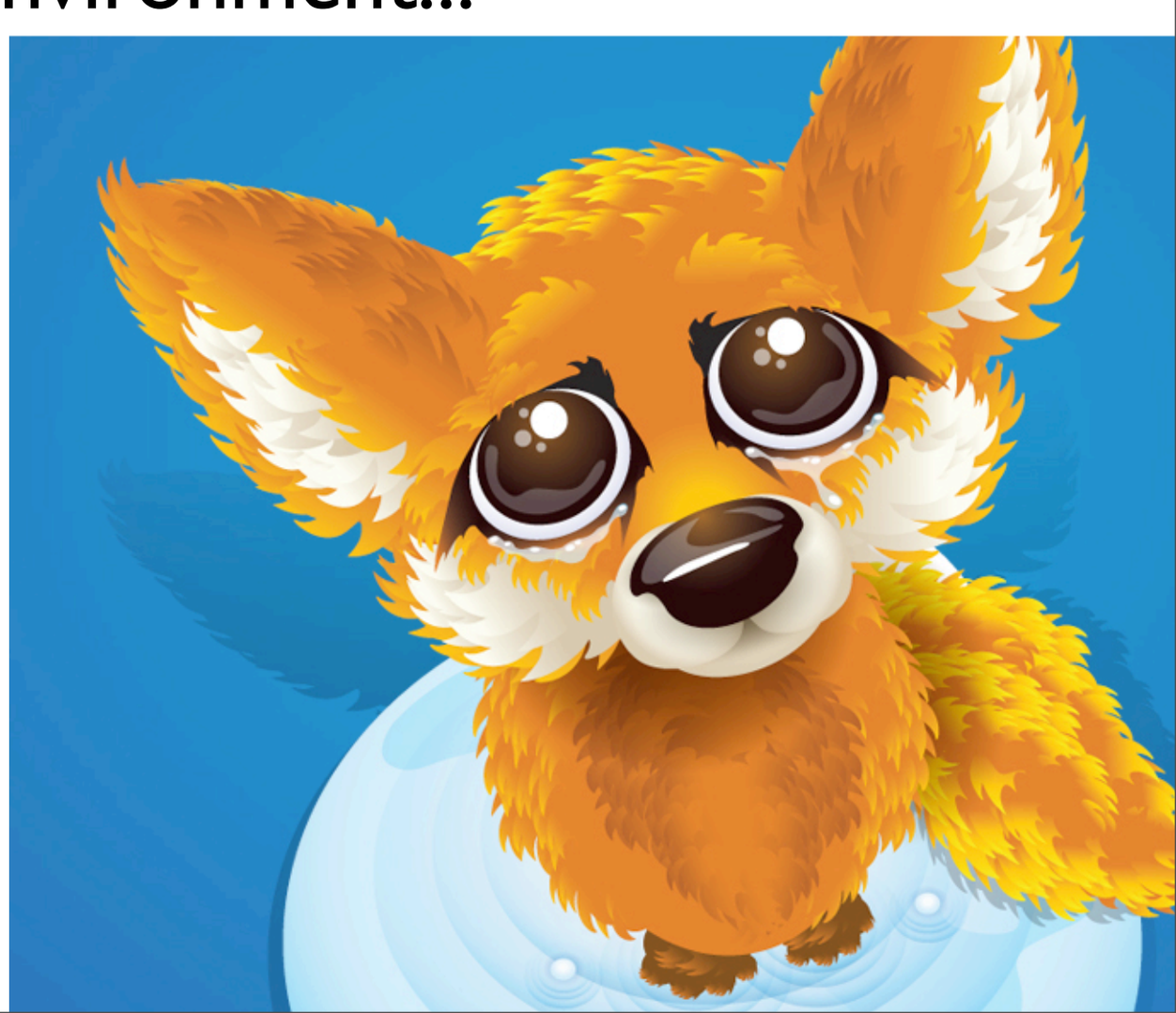
We have been at it for a while...

in a phenomenally hostile environment...

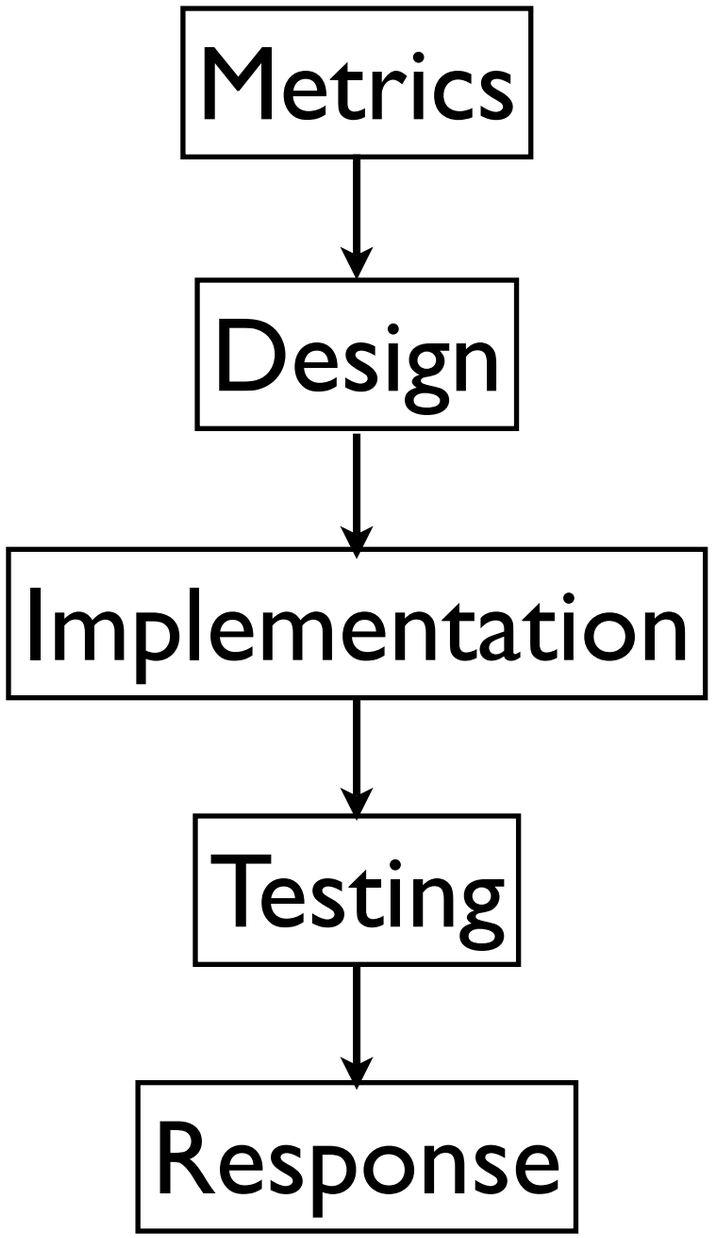
with 180 million users...

and we seem to be doing a lot of things right...

and you can see how we do it



This Diagram is Stupid



Good Security is a Feedback Loop

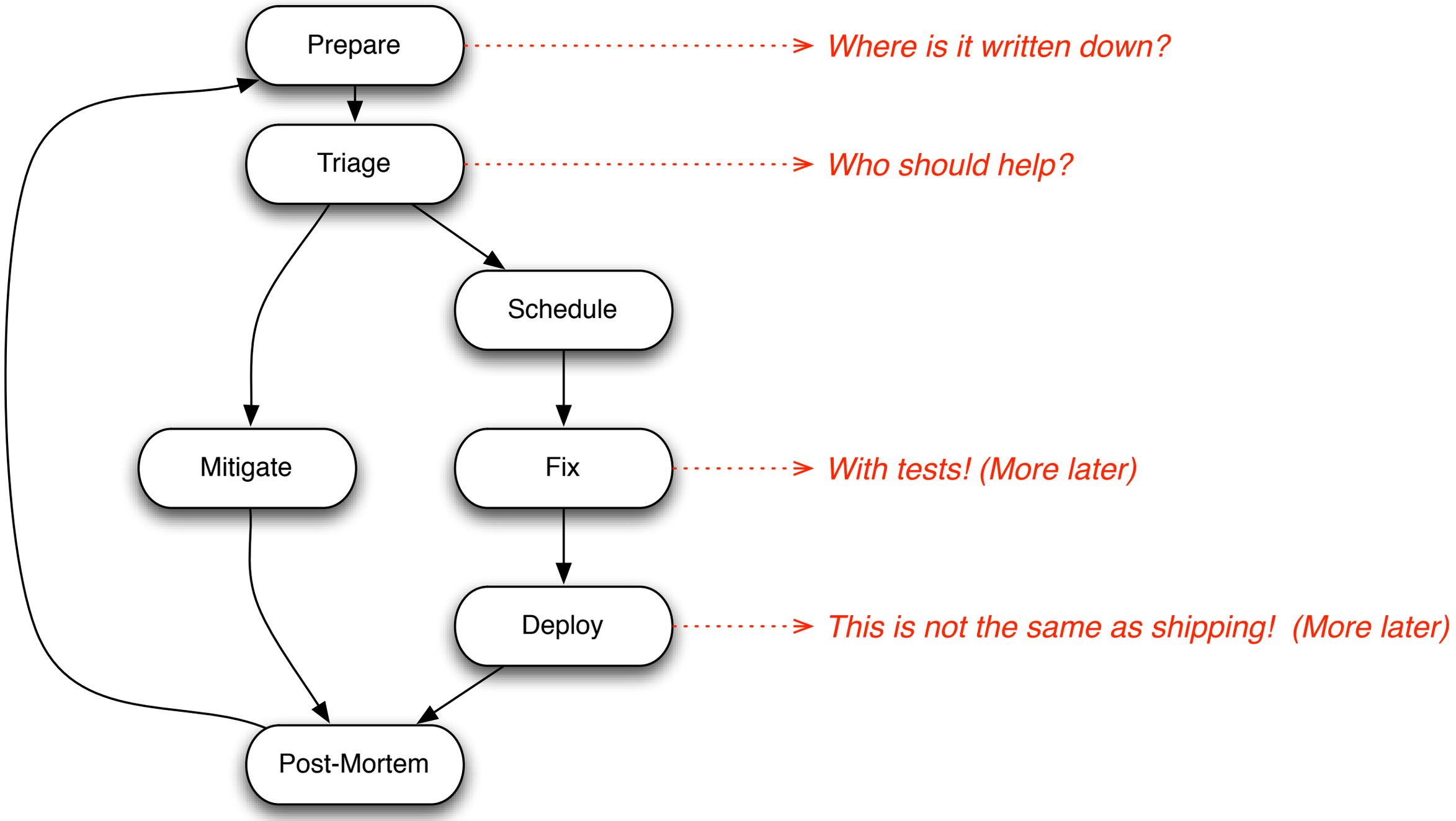
- The idea that security can be wholly top-down, with discrete one-way steps in an orderly flow from start to end is the worst kind of process management fiction
- Your security process should instead ask at every step, "How can we make sure problems like this never happen again?"

The single most important thing you can do is find ways to capture expensive knowledge so that you never pay for the same lesson twice

Response

A security compromise is the most expensive knowledge of all

Response


Prepare
Triage
Schedule
Mitigate
Fix
Deploy
Post-Mortem
Where is it written down?
Who should help?
With tests! (More later)
This is not the same as shipping! (More later)

Learning from Response

- It's okay for post-mortems to be short
- It's not okay to skip them
- If you make them into blame-finding, they stop being useful (even for blame-finding!)

Ask Questions

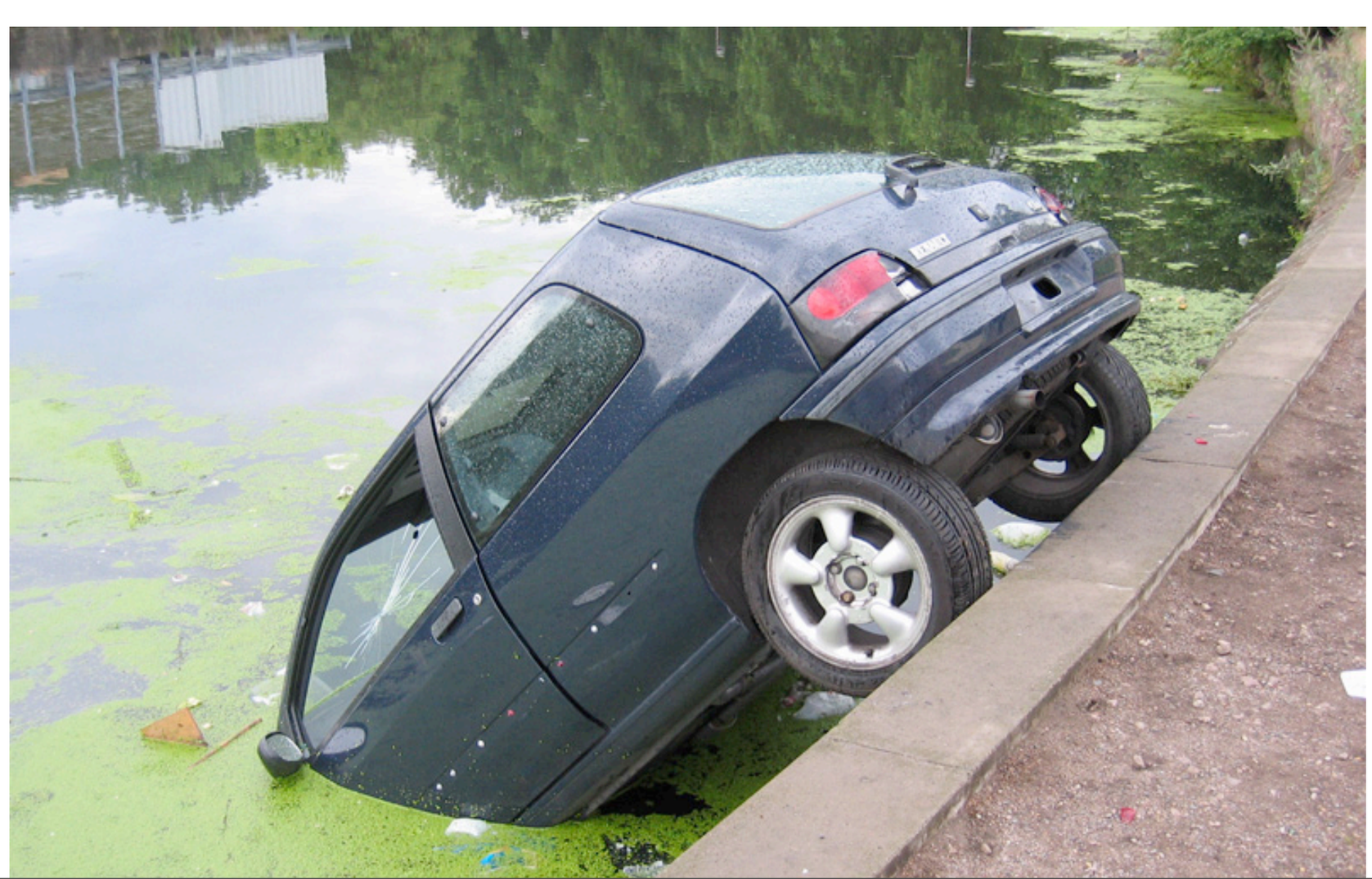
- Who did we have to bring in late?
- Why didn't we notice that we broke the internet?
- How could we have dealt better with the original reporter?
- What were our bottlenecks?

Write down the answers for next time

(there's always a next time)

Testing

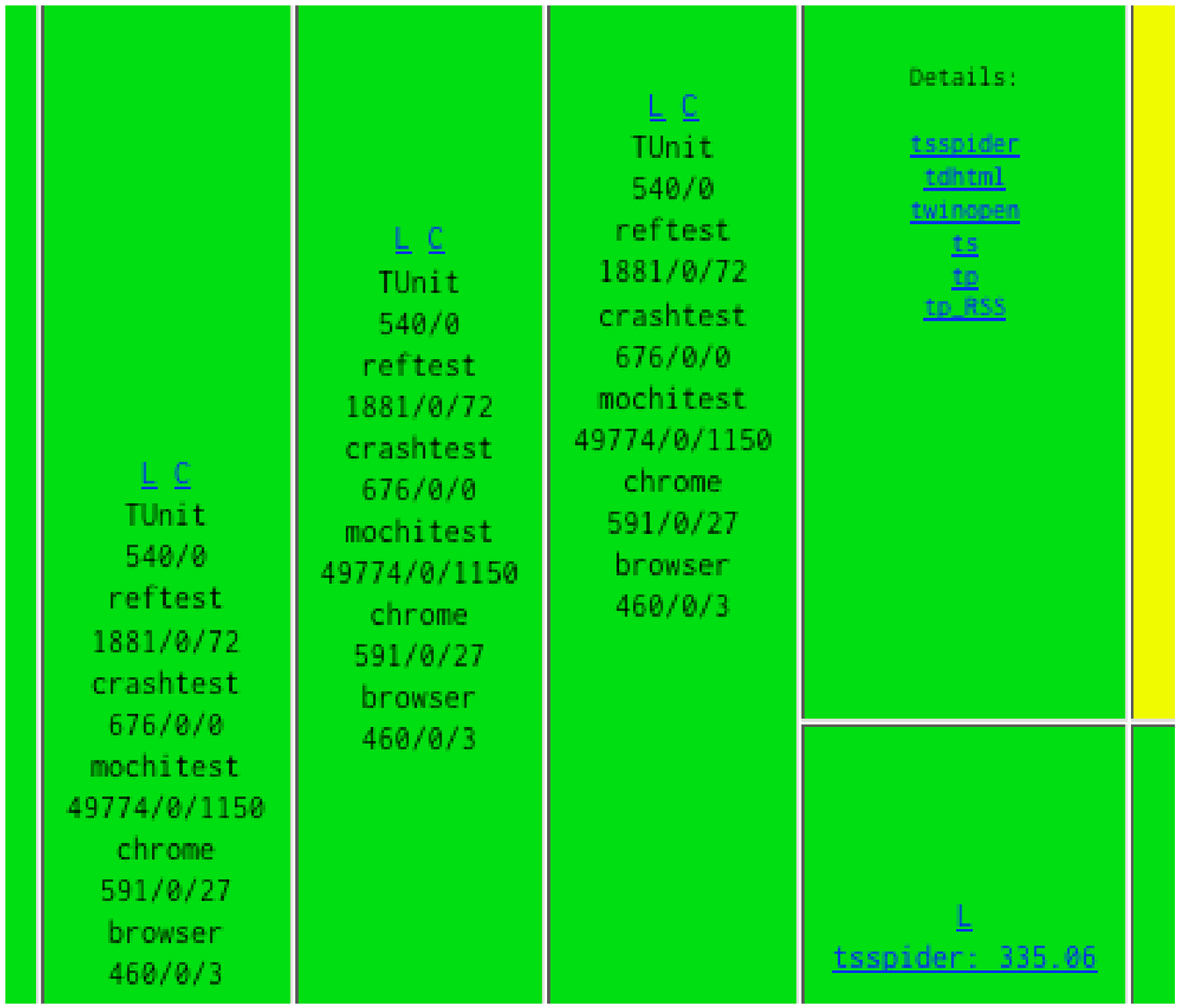
Testing is your best defense against forgetting, because you will forget



Data Point

We run:

- 55,000 automated tests
- in 6 test frameworks
- on 4 platforms
- at least 20 times a day



You Already Know Why

- Tests protect your features from security-based changes
- Tests protect your security from feature-based changes
- Tests capture and transfer expensive knowledge
- Tests reduce Bus Factor

Now Make It Happen

- It must be easy to add new tests
  - Yes, this is tricky at first
- Money can be exchanged for goods and services!
- Nothing lands without tests
- Nothing. Lands. Without. Tests.

It's Hard To Test <X>

- This is terrifying
- Steal another framework
- Don't underestimate manual testing

Power Tools

- Fuzzing
- Penetration Testing
- YMMV

Bug 416834 – Assertion failure after deleting eval 16 times

Status:	VERIFIED FIXED	**Reported**:	2008-02-11 09:04 PDT by Jesse Ruderman
Severity:	major	**Modified**:	2008-04-18 07:05:51 PDT (View Bug Activity)
Keywords:	assertion, testcase		
Whiteboard:			

One More Thing

Tests that don't run are a waste of everyone's time

Option: Automatic Gunfire

Buy a box that sits in a corner and runs tests off trunk every hour. Put a gun on it that shoots people who break tests.

Option: Manual Slog

Make check-in approval contingent on running tests, every single time.

Implementation

"We have tests" is not an excuse
to keep breaking things

Where Mistakes Are Made

- Strategic-level mistakes can be made in design, but *most security bugs come from mistakes not caught during implementation*
- Your ability to profit from expensive knowledge is highest here, but here is where you're probably doing the worst job

No-Brainers

- Static analysis tools
- assert()
- “Public” Betas
  - Alphas?
    - Source?

Tougher

- Non-security bugs point to security bugs
  - Do you have crash reporting?
- No bug happens once
  - Where *else* are you assuming that a null pointer isn't exploitable?
- Bad patterns - knowledge that you get to benefit from more than once.

The Game Changer

The most important change you can make at implementation is mandatory review

- Socializes security knowledge by sharing it
- Gatekeeper against "This is little, it'll be fine"
- $P(Mistake_1) * P(Mistake_2) << P(Mistake_1)$

Design

Every time you eliminate a threat class
an angel gets its wings

Making Things Right

Make it easier to profit from expensive knowledge

- Design for re-use
- Find areas that keep needing “temporary” field patches and fix them for good
- Design for testability
- Threat modelling

Metrics

Measure what matters, not what's easy to measure

Don’t Know What Matters?

- Ask sales
- Ask your users
- Don’t ask your competitors, they are looking for the easy way out

The #1 Grade-A Stupidest Metric of All

Bug Count

- A focus on bug counting creates perverse incentives for security
- Developers hide bugs from management
- You hide bugs from customers

Counting bugs teaches you to bury all the expensive knowledge you should be sharing

Think Harder

- Days of exposure
- Average time to deploy fix
  - Better would be avg. time until > 90% of users are *using* the fix
- Customer downtime

Get Creative

- Number of regressions per update cycle
- Number of all nighters
- Start using similar metrics when judging your own suppliers & platforms
- Tension between metrics can be a good thing, if it pulls people towards awesome

Stupid Criticisms

- This model is totally reactive, not proactive
- This model is steady-state, not innovative

Our tools, let me show you them

Tinderbox	http://www.mozilla.org/tinderbox.html
Mochitest	http://developer.mozilla.org/en/docs/Mochitest
Litmus	http://wiki.mozilla.org/Litmus
MXR	http://mxr.mozilla.org/
Dehydra	http://developer.mozilla.org/en/docs/Dehydra
Bug Policy	http://www.mozilla.org/projects/security/security-bugs-policy.html
Bugzilla	https://bugzilla.mozilla.org/
Fuzzers	http://www.squarefree.com/2007/08/02/introducing-jsfunfuzz/

Remember This Slide

- Capture expensive knowledge everywhere, so that you don't have to re-learn it
- Apply that knowledge everywhere
- Nothing lands without tests
- Nothing lands without code review
- Counting bugs is stupid, try harder

Credits

- Developer Kit, Sean Martell, http://developer.mozilla.org/en/docs/Promote_MDC
- Waterfall, dave.hau, http://flickr.com/photos/davehauenstein/271469348/
- Alarm, Shannon K, http://flickr.com/photos/shannonmary/96320881/
- Oops, estherase, http://flickr.com/photos/estherase/24513484/
- Card House, Bah Humbug, http://flickr.com/photos/gibbons/2294375187/
- Bulldozer, Atli Harðarson, http://flickr.com/photos/atlih/2223726160/