signals in testing
jason arbon
engineering director, uTest
quick context on me
finally here

- Testers are alligators, we haven't evolved much :(
- Signals show the face of test is changing fast
- How to turn the scary signals to your advantage
- Share findings in testing and quality
mounting pressure

- speed. “move fast and break things”
- appification. “mobile first”
- voice. customer’s voice is getting louder
- fragmentation. OS, Hardware, SoLoMo
ride these waves!
testing judo

- don’t hide or run away, use these trends to your advantage!
- We’ll look at each area, how it’s scary and how we can leverage it
- “Judo: .... resisting a more powerful opponent will result in your defeat, whilst adjusting to and evading your opponent's attack will cause him to lose his balance, his power will be reduced, and you will defeat him.”
signal: speed

our world is speeding up
continuous integration/build/test
speed: test planning

test plan is outdated
All about risk
< 30min!
Always test the most dangerous stuff
Tie it to metrics
Team owns risk
speed: continual testing

- Stop trying to compress the standard product cycle
- I tried when I first moved from Bing->Google
- Don’t have a stabilization period
- Continuous deploy means continuous testing
- You are never done, so pace yourself
on regression testing...

- Don’t automate everything

- If you are shipping often and innovating...your tests will break!

- Be very careful with technical and manual test debt...it will be the end of your value add.

- If needed, defer this busy work to the crowd
Alan Page said it well

AlanPage @alanpage
Humans fail when they don't use automation to solve problems impossible or impractical for manual efforts.
Retweeted by uTest seattle

AlanPage @alanpage
Automation fails when it tries to do or verify something that's more suited for a human evaluation.
Retweeted by uTest seattle
speed: dev owns quality

Dev owns quality
Invest heavily in “Unit+” testing
Handoffs too expensive
Dev does most testing
Don’t be an enabler
Dev investigates all breaks. first.
Key to how Google ships continuously
Slows down the buggier devs :-)

Sunday, March 11, 12
speed: features

Quick builds mean quick fixes
Quality features a must:
* Auto update + rollback
* Flights, A/B testing, Channels
* Instrument app for user actions
* Look for statistical changes in user behavior
Leverage early adopters and crowd
They will tell you when things break
Opt-in
Lots of code ‘baking’ going on
Forget magic of secrecy
Channels versus versioning
speed: deprecate your tail

signal: mobile

![Graph showing units shipped per year for different devices including PC, Android, iPhone, iPad, Macintosh, and Other. The graph highlights a significant increase in mobile devices compared to traditional computing devices.]

Mobile won overnight

Sunday, March 11, 12
mobile: vs the web

Big but slowing and standardizing
Most growth in mobile web
Growing fast. Feedback is easy
Ship early and often. Fail often, but cheaply.

signal: appification

600 new apps/day

User Reviews

<table>
<thead>
<tr>
<th>Rating</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 star</td>
<td>1216</td>
</tr>
<tr>
<td>4 star</td>
<td>278</td>
</tr>
<tr>
<td>3 star</td>
<td>181</td>
</tr>
<tr>
<td>2 star</td>
<td>113</td>
</tr>
<tr>
<td>1 star</td>
<td>586</td>
</tr>
</tbody>
</table>

Average rating: 3.6

2,374

Your apps 9 apps

Sunday, March 11, 12
appification: the good

Customer Ratings

Average rating for the current version: ★★★★★ 2177 Ratings

Great Update ★★★★★
by seanybryan - Version 1.9.1 - Feb 14, 2012
Great update. Next update bring on the survival endless then you will have reached perfection!

2 out of 2 customers found this review helpful

Fantastic as always ★★★★★
by David Crume - Version 1.9.1 - Feb 14, 2012
I always find myself playing this game when I got nothing to do. Even though I have tons and tons and tons of other games. Sooo Addicting!!

1 out of 1 customers found this review helpful

awesome!! ★★★★★
by stiny111 - Version 1.9.1 - Feb 14, 2012

well made game

1 out of 1 customers found this review helpful

Q: Does this look like a good app?
appification: the bad

User Reviews

- 5 star: 1,216
- 4 star: 278
- 3 star: 181
- 2 star: 113
- 1 star: 586

Average rating: 3.6

Daniel on February 5, 2012 (SEMC Xperia Play with version 1.3.0)

★★★★ Star broken after update, instant crashes now
2 stars for trying but still almost instant crashing.

steven on February 10, 2012 (Samsung Galaxy S with version 1.3.0)

★★★★★ given up with this crap...
never worked and will never wrk because the developer simply dont give a crap... emailed 5 times no reply had this game when it came out round xmas and only managed to get it to work a few times.. will never buy frm dev ... using galaxy s

Q: Would you pay for this app? ...on Galaxy S?

Sunday, March 11, 12
appification: same game

- bad thing: fragmentation
- bad thing: your customers can kill an app within hours
- good thing: you will know why
- good thing: schematized feedback (yeah!)
- good thing: most issues preventable
- good thing: we have an oracle for ‘good enough’
appification: use the data!

Understand your market quality data
Understand your competition
Test for common market issues before launch!
appification: use the data!

“Done is better than perfect”
appification: use the data!

STUFF USERS HATE
(BASED UPON USER COMMENTS)

- Install: 40%
- Pricing: 31%
- Performance: 16%
- Crash: 11%
- Hang: 35%
- UI Controls: 2%
- Security / Privacy: 1%

*Based upon crawling 250,000 users reviews of the most downloaded apps.

what to test for...

Powered by: uTest
Where the best test
signal: fragmentation
fragmentation: matrix

\[ ] = \text{app} \times \text{OS} \times \text{device} \times \text{location} \times \text{user context} \times \text{time of day} \times \text{system state} \times \text{carrier/netconnect} \times \ldots

Use sampling methods
Beyond pairwise :) 
Just deploy to the matrix! crowd source early and often
fragmentation: advantage

Stop emulating users, leverage them directly
Engage early adopters and crowd
Argument for fewer features
Argument for simpler features
less testing
less risk

use the crowd to your advantage!
HUDs for testers

How to deal with lots of data. quickly.
Make every tester a good bug finder and filer
role changes

SDET/SET -> Dev. But not only Test dev!
Test Management -> Dissolve into Dev or PM
Test Architects -> Technical Fellow. Advisory.
Reseach
new continual, crawl-based workflow

machines continually scan
humans review machine diffs
route those to developer
all in <10 minutes
bugs are best when code is fresh
findings from Google years

Crowd finds more bugs, earlier and cheaper
Don’t automate UX
Test code in language of application
Kill regression automation, esp UX based.
Dev has to own quality
Lower your ratio of test:dev
Lots of unit tests with continual
crowd/exploratory
No top-down rules on testing
...or metrics
buy the book :-)  
much of this is in the book  
how google does it  
early experiments  

co-authored with Dr. James Whittaker (who now uses windows phone 7)  
AMZN & http://www.howgoogletestssoftware.com
looking forward: cloud
all testing should be in the cloud
funny we still test web pages with client side java :) Ok, its not funny....

local binaries slow you down
move to the cloud:
  all artifacts
  all test results
  all devices
  all humans
crowd benefits

Test bandwidth to complete test passes
On-demand
Diversity in context
Real testing ‘In the wild’
No 1:1‘s, reviews, or career path discussions :)”
For the cost of one bad fulltime hire: have 5 people on demand, and with a click, keep them or drop them.
leverage uTest :)

what could you do with 50k testers on-demand, and “In the Wild”?
Evolve quickly...

To ->
contact

jarbon@gmail.com
jarbon@utest.com
@jarbon , @utestsea
+JasonArbon