Is it Time for Execs/Leaders to think differently about what QA teams do ? And produce ?

It is time for QABall…?

BY MAS KONO

MEDASSETS : VP DEVELOPMENT SHARED SERVICES

Who is this Guy ?

- Mas Kono
- VP for Shared Services at MedAssets
  - QA, QA automation
  - Performance /Load Testing
  - PMs, SCM, Tech docs, Instructional Designers etc..

- ~20 years of SW/Dev/QA experience in various positions…
- Pretty much a geek, but like to get out and talk to people once in a while..

- You can find me on Linkedin or email me at maskono@outlook.com

Anti-view (instead of an overview)

- This will not solve all of your problems
- This is not a prescriptive solution, so this may not work for you
- This is not a how to implementation guide
- Benefits / Mileage will vary

- This is just one person's opinions / observations
- This could all be crazy talk, but seems to be working for me… ☺

Money Ball ….

- Bill Beane talks about "we need to replace" 112 RBI…
- Traditional scouts in the room.. Scouts - we want this guy, that guy……
- Billy B… no we need 112 RBIs..
- Billy B… Because he gets on Base…
- HMMM… A new way of looking at things… QABall

What am I trying to present ?

▶ QABall – Attempt to Change / Modify the thinking of Exec/Leaders asking questions about QA activities and results (information, data, metrics…)

▶ The traditional questions Execs ask, are they outdated ?

  ▶ How many test cases did you run ?

  ▶ How many are automated ?

  ▶ How many bugs did you find ?

  ▶ Defect metrics / Trends ?

  ▶ Are these like asking "how many HRs" in Money Ball ?

What am I trying to present… ?

- If we need to communicate differently then..

- We need to effectively communicate what typically goes on in a QA organization

- Attempt to level set Leaders outside of QA about the difficulties faced by a typical QA organization

- Attempt to educate and have realistic expectations of a QA organization

- Then present some ideas on what alternative data points are out there..

- Hopefully some of this data helps you communicate up/down/sideways etc…

Why QABall ?

- Many of the leaders may have a Dev/Engineering based background,

- Very few have any true understanding of QA, or worse they think they understand QA

- They ask question like How many test cases are run or automated ? (why do you care……?)

- Lets turn this around by getting leaders to focus on what is important by educating and presenting possible alternate data points……

- Think differently … Money Ball - "because he gets on base"

  - Not Batting Avg, or HRs

  - On base % = manufactured runs…

  - Instead …Batting Avg with runners in scoring position

Communicating with Execs ?

- Who runs into this issue ? (when talking to execs after a release goes bad)
  - Why did "QA MISS THIS BUG" !!
  - How many test cases did you run ? How many are automated ?
  - Why wasn't there a test case for this?
  - This is so blatantly obvious how did you miss that ?
  - If we knew about this defect why did we release ?
  - I didn't understand what you meant by this defect.
  - Oh that's what you meant by that metric.
  - Why didn't you tell me not to release ?
  - Etc….. Etc….

# Let us level set the conversation…

Testing Modern Software is impossible !!
(OK maybe not impossible but hard)

- ▶ Software is getting more and more complex everyday

- ▶ As much as the PR machine for tooling companies say how easy it can be to test.....

- ▶ The reality is that it is really, really hard and in some cases impossible to achieve the expected coverage

- ▶ Everyone keeps saying 100% coverage

  - ▶ "Test coverage is of little use as a numeric statement of how good your tests are" – Martin Fowler

- ▶ Why ?

  - ▶ Less and Less access to relevant production data (HIPAA, Audits)

  - ▶ More and More complex Production environments

  - ▶ Ridiculous number of paths/configurations

 - ▶ Oh and we have 2 days to regress everything...

No one wants to hear whining but ...

▶ Lets educate !!

▶ Execs don't want to hear about how hard something is to do…

▶ But what if they have 0% understanding to what QA really does on a daily basis…

▶ How effective can they be as leaders ?

▶ Maybe just maybe they could understand the challenges better and ask the "right" questions and even facilitate the changes needed…OMG what a concept ☺
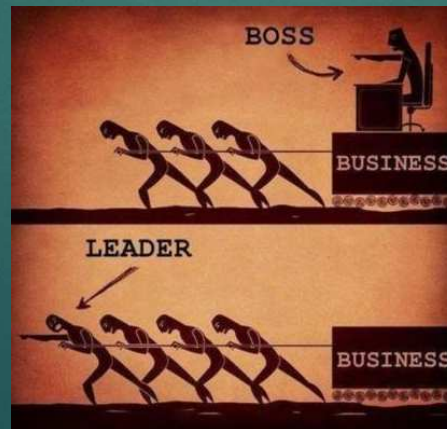
Lets point out an obvious one…

▶ Simple Web site.. Multi select + Submit Button
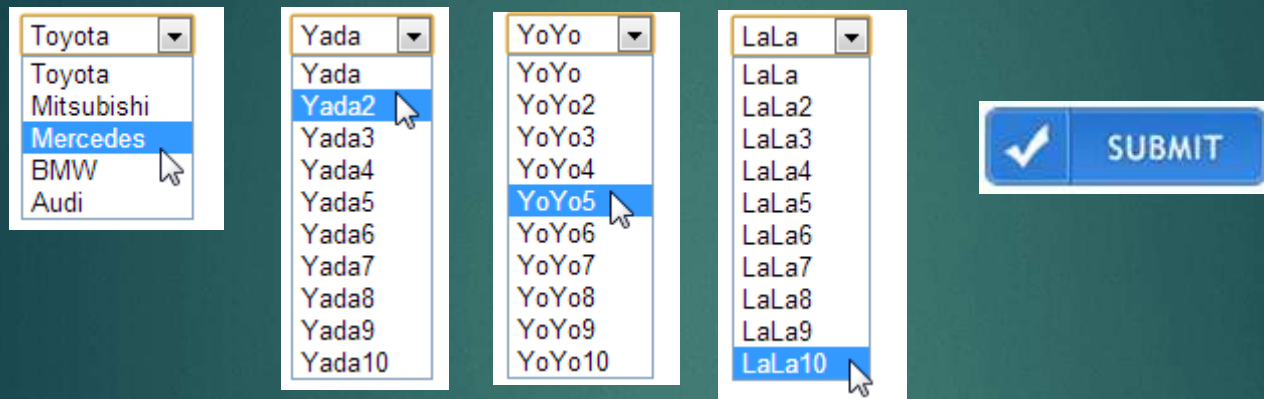
▶ Easy right ?   How long would you like to test that ? 10 min ?

▶ Maybe not.. that is a lot of combinations for coverage…

	Menu item 1	Menu Item 2	Menu Item 3	Menu Item 4	Combinations
Single	5	10	10	10	5000
Multi Select	25	100	100	100	25,000,000

CIO Magazine said you guys need to Automate that…

Automation can help…

- Yes, It can help… but it is NOT the solution to everything..
- Mathematical /Time impossibilities are going become apparent

- Example: 3 million possible tests to run.. On a simple page
  - On a Web page (slow to load)
  - On 10 different browser configs , so 30 million cases
  - At 5 seconds a test (very optimistic)
  - 30M x 5 Seconds = 4166 Hours of testing.. (173 days)

- Sort of like asking a travelling sales man to hit a quota that asks him to drive 2000 miles a day. (you can't without breaking the law)

Let me force you to test small…

- Example: DB performance issue example
    - Test report generation in a QA environment , works fine
    - But the QA env only has a small subset of production level data
    - No performance issue but production blows up since it has 30TB of data…
- How come QA "missed" this one…
    - This has more to do with environmental restrictions and limitations
    - Does the audience/decision makers "really" know what QA's effective limitations are ?
    - Has this been communicated ? Risk / Rewards calculated ?

We tested it on
a mini bike

Sure will work
Fine on a Moto GP
Right ?

Late to the party…. (aka why did QA find this so late)

- In the previous example…
    - Lets agree you came up with 100 "tests" to get reasonable coverage
    - Each one takes you say 6 min to test = 600 min or 10 hours
    - Which to a Project manager means we are releasing tomorrow
    - But the build didn't get to you until 6pm time to suck it up
- You are 8 hours into the testing and boom…. You find a killer one
- Crap…
    - Should we stop the release ? No this isn't so bad is it ?
    - Wait why did it take you 8 hours to find this obvious bug ? You stink..
    - Only happens on QA not on Dev boxes, wait what…
    - Screw it release it, it doesn't look bad…
    - Next day … Why didn't QA find this…???

OK I got it… So this stuff is hard…

But someone told me…

Isn't Testing like Polling ?

- Well sort of....
  - US election pollster…
  - 313 million Americans (2012)
  - 220 million eligible to vote
  - And yet, pollsters can poll 1000 people and get pretty accurate right ?
- QA/Testing work we are dealing routinely with
  - Possibilities in the Trillions if not more..
  - And Everyone is able to "cause a disaster"
  - The data to influence who to "poll" is a lot less firm than it is polling…
  - Techniques are valid but….

Toy Rocket
Does not translate into Saturn V

Fine....
So what am I supposed to Ask ?

What Execs probably want is…

You want
to know % WIN!!

Unfortunately
Not that simple

But that is sort of like driving with just this...

I would recommend a bit more information...

I think we actually sometimes provide this ?

OK got it….
What should I ask.. ?

Well, what do you want to know ?

- Do you want to know on base % how many runs ?

- Do you want to know batting avg with runners in scoring position ?

- Do you want to know how many RBIs on cold / warm days what ?

- $ per hit ? P/E ratio,   what ??

- You learned that QA probably can't cover everything…

    - What is important for you ?

    - Performance ? Core functions ?

    - What didn't QA test ?

    - What are the limitations ?

    - What are you afraid of ? (don't need data give me a gut feel)

    - Would more time help or not ?

    - Appropriate Trends/Metrics (if you are up for it…)

Typical things asked of me….

- ▶ % win indicator…

- ▶ % likely to fail in production…

- ▶ If I gave you 2 more days how many more critical defects will be found..

- ▶ The winning numbers for tomorrow's Mega Millions

- ▶ Numbers that I can't easily predict.. At least reliably

- ▶ This is not a card game..

- ▶ So essentially mostly impossible data points..

- ▶ I had to turn it around and present other things..

What has worked for me…

▶ Simple Data points followed up by complex appendix…

▶ Start with a simple Yes or NO (on a release vote)

▶ Gut check, what am I worried about…

▶ Go a bit deeper with feature level data (Check, Check + etc..)

▶ Schedule compression number

    ▶ QA typically never gets the X days requested to test

    ▶ Publish a compression number to indicate risk

    ▶ We only had 10/15 days so 0.67 compression (risk is high)

    ▶ If you had calculated you need a full tank of gas to make it to Houston, would you go with 2/3 ?

▶ Go a bit deeper with regression coverage maps

▶ NO Test cases run, or defect metrics etc…

So what does this look like..

- ▶ Yes , Go Release
- ▶ Areas for concern XYZ
  - ▶ Because we got it late and configuration nightmare to test
- ▶ Schedule Compression = 0.67 (we got crunched badly)
- ▶ Test Case point total = 892/1000 (we got close but not all the way)
- ▶ New feature Coverage Map 2 areas yellow rest 6 Green

- ▶ Appendix:
  - ▶ Test Cases run, defects trends, LOKI, Intermittent defect list etc etc…
  - ▶ Mitigations for all of the above when possible
  - ▶ Looks like a 1 page go/no go/ with a large appendix
  - ▶ Think ESPN Box score, but with full stats later down in the article

Questions….??