Practically…Testing
(or a less boring title of your choice)

VIDROHA DEBROY,

OCT 1ST 2016

A little bit about me

- Graduated with a PhD in Software Engineering under the supervision of Dr. Wong (I think you met him) in 2011
- Joined the Windows Team at Microsoft as an SDET and stayed there for close to 2 years
- Returned to DFW and joined Hudson Alley Software as a Senior Software Engineer. Stayed there close to 2 years
- Became a Senior Software Architect at Verizon. But now I am with Varidesk and am loving being a developer again.
- Also an Adjunct Professor of Computer Science at SMU
- and clearly, I have nothing better to do on a Saturday…

Why test?

- Well Dr. Wong told us we should…Congratulations! You will probably get an A ☺.
- Well… if we agree…this lecture could end sooner.
- Well… I didn’t hear the question…but nod head along with others…

Seriously though…

- Testing is an investment with proven returns
- Testing is as fundamental a software development activity as any, and often can drive other activities
- Testing lets you realize things about your product that you wouldn’t otherwise
- On a personal note, testing can help avoid embarrassment

Effective Testing

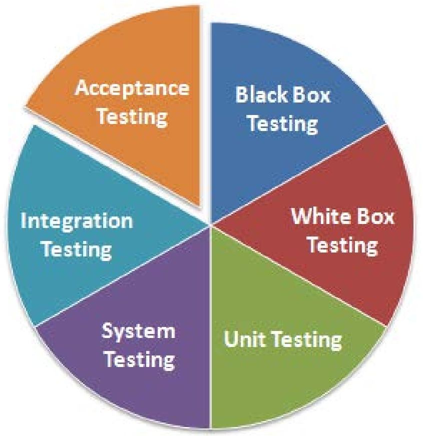
- Testing is not really a question of doing or not doing – it is about understanding; this is as much an art as it is a science
- We need to be familiar with the software we are testing
- We need to plan, scope and prioritize and measure
- We need to treat our tests as first class citizens
- We need to leverage automation as appropriate
- We need to be creative!

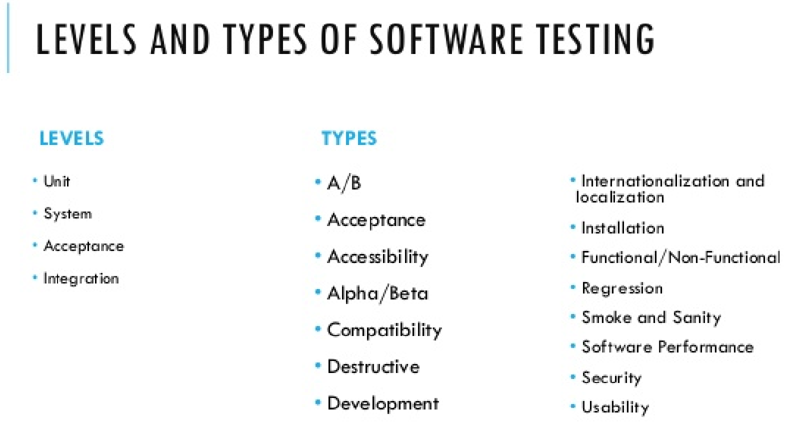
The relationship(s) between dev code and test code

- ❑ What comes first – implementation or tests?
- ❑ Dev code usually has to be modified to facilitate test code, i.e., dev code *needs to be made testable*
- ❑ More test code is typically written than dev code
- ❑ Does test code need to be of an inferior quality than dev code?
- ❑ Ultimately, test code is meant to cover dev code (or is it ^_^)…

Dev code and test code alike need to be under SCM, they are equally important – that is an important takeaway

Categorizing tests helps





LEVELS AND TYPES OF SOFTWARE TESTING

LEVELS

- Unit
- System
- Acceptance
- Integration

TYPES

- A/B
- Acceptance
- Accessibility
- Alpha/Beta
- Compatibility
- Destructive
- Development
- Internationalization and localization
- Installation
- Functional/Non-Functional
- Regression
- Smoke and Sanity
- Software Performance
- Security
- Usability

Manual Testing more suited for Testing Types
Functional Testing
Unit Testing
Exploratory Testing
Ad-Hoc Testing
Integration Testing
UAT (User Acceptance Testing)
Usability Testing
Alpha Testing
Beta Testing
Compliance Testing
www.SoftwareTestingSoftware.com

OK – enough with the stock pictures…we get it you know how to use Google Images…

- Not all tests are created alike…they have:
  - Different goals
  - Different needs (intentionally staying away from the word ‘requirements’)
  - Different results and interpretations of the results

Tools and Automation

- What can we do to automate and what can automation do for us?
  - Believe in this! Let’s look at a popular library: https://github.com/moment/moment
- How do we pick a good testing framework?
- Test execution cannot be an all or nothing effort
- Test results are more meaningful when they are consumable
- When done right, automatic test execution and reporting is either:
  - Triggered
  - Performed on a timely basis
  - Scheduled on-demand by anyone in the team

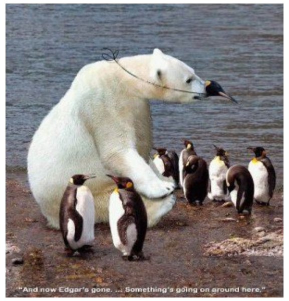
Code coverage

- It is ‘a metric’ to evaluate tests
- It is best taken literally – without inferring too much else
- When used correctly, it is great at identifying untested code (and potentially untestable code)
- When used incorrectly, it is great at instilling false confidence
- The way I think it is best viewed – “*when your tests have covered some code, you know something about the code; when you haven’t covered the code, you know nothing of it*”

When tests fail…

- Believe it or not, its quite normal…
- Reproducibility is key
- Sometimes its just not about the dev code or the test code, its just about the environment
- Ultimately, test failures are a good thing
- But the fix <u>cannot be</u> just about getting the test to pass – there is much more to it than that

Fitting into testing and fitting testing in


"And now Edgar's gone. ... Something's going on around here."

- If I had an answer for you, I wouldn't be here today…I'd pretty much be on my private jet to Hawaii…assuming I wasn't already in Hawaii.
- But I can tell you that the slide title is important to figure out
- In fact…very important to figure out for yourself and those above and below you in the hierarchy of things (I couldn't find a better term).
- In a sense its not that different from other decisions…what's the ROI?
- What would a final bullet point be without a but? – Understand immediate returns versus a dividend style model.

Final thoughts on testing

- We know its important and all that…but its also a great job
- You get to play detective
- You get the final say on whether the product is ready or not
- It is true you have a lot of responsibility…
  - Sometimes more than people give you credit for
  - Sometimes it's a thankless job (or so it seems)
- In the end – have your pick of an example – would you get on a plane if you knew things hadn't been tested?

Thank You

Questions?