Ten Things About Testing

Every Manager and Developer Should Know

RBCS

TIME TESTED.

TESTING IMPROVED.

www.RBCS-US.com

Introduction

- Did you ever wonder why your colleagues sometimes look at you funny when you're explaining your test results?
- Do they ask you strange questions like, "Why don't you find all the bugs in testing?"
- Maybe they don't understand what you do or even why you do what you do
- Let's examine ten common misunderstandings that software professionals have about testing…

*Testing **Can't** Find All the Bugs*

- Probably the most common misconception is that testing **can**—and testers **should**—find all the bugs
- Typical defect detection effectiveness (DDE) is more like 85%
- Really great test groups get 95%
- Many upstream problems can reduce DDE, including delivering too many defects
- Tester should try to find almost all the important bugs, and excellent test groups do

But Testing ***Can*** *Show You the Risks*

- So, if we can't find the all the bugs, what can we do?
- One thing we can do is show managers and stakeholders the risks
- Reducing risk to an acceptable level, and helping stakeholders understand the risks, are key objectives for testing
- If you do proper risk-based testing, you can report on which risks have and haven't been mitigated

It's Not True that "Anyone Can Test"

- Another very common misconception is that anyone can test
- This has unfortunately been reinforced by some agile advocates
- In fact, testing requires special skills to do well
- This is especially true for particular types of testing
  - Security
  - Performance and reliability
  - Test automation
  - Structural (white-box) testing

TDD Doesn't Eliminate Regression Risk

- Just because you have automated unit tests, doesn't mean you can change code without risk
- Unit tests range from 10% to 50% (at best) DDE
- This includes regression bugs
- Some very outlandish claims have been made by certain Agile advocates in this regard
- TDD can't find all bugs—no form of testing can!

There ***Still*** Are No Quality Silver Bullets

- Throughout the history of software engineering, people have claimed to have found a solution to software's challenges
  - Object-oriented programming
  - Six sigma
  - Total quality management
  - The list goes on
- While Agile does help with certain software challenges, it's not a silver bullet
- No reputable broad-based studies show an appreciable increase in quality due to use of any particular lifecycle

Test Tools Work…For Skilled People Only

- Many powerful test tools are available now
- Both commercial and open-source
- However, most tools are not self-explanatory
- I've seen numerous examples of misuse of tool
  - Developers not knowing code coverage
  - Performance testing
  - GUI automation
- Misuse of the tools leads to confusing and/or misleading test results
- Bad project decisions follow

Coverage Means More than Requirements

- Traceability of tests to the test basis is important for proper test results
- Too often, people think this means only requirements coverage
- Test coverage should be considered as a multidimensional problem
- Requirements, yes
- But also risks, configurations, data, personas, and other relevant dimensions
- Requirements alone are incomplete and imperfect as a test basis

What That "Test Sign-off" Really Means

- Too often, organizations misuse sign-off
- If a test team approves a release, that doesn't mean there are no bugs
- At best, it means that thorough testing built adequate confidence that the system will work—when used as tested
- If the test basis is inadequate, it doesn't even mean that
- Managers and developers must have realistic expectations about test sign-off

Test Data and Environments Matter

- Test results are predicated on the test data and environments used for testing
- When testing is done with non-representative test data or environments, important bugs will slip past
  - Performance
  - Reliability
  - Security
  - Functionality
  - Scalability
- I've seen many instances of misleading test results due to inadequate test data and/or environments
- Ideally, test data is anonymized production data
- Ideally, the test environment is a realistic replica of production
- Both of these are serious challenges to testing in this era of big data and diverse environments

False Positives Aren't About Testers Skills

- A high rate of false positives can impose serious inefficiencies and risks
- Some false positives result from test skill issues
- However, other factors are significant
  - Incomplete, flawed, or contradictory requirements
  - The lack of a single, consistent test oracle
  - Test data and environment issues
- False positives, like many other testing problems, have diverse and complex upstream causes

Conclusions

- We’ve seen ten examples of areas where managers and developers don’t understand key testing realities
- As a tester, you need to help them understand these realities
- Perfect testing is not possible, but clarity about testing and its results is
- Identify and resolve situations where misunderstandings exist

To Contact RBCS

For 20 years, RBCS has delivered consulting, outsourcing and training services to clients, helping them with software and hardware testing. Employing the industry's most experienced and recognized consultants, RBCS advises its clients, trains their employees, conducts product testing, builds and improves testing groups, and hires testing staff for hundreds of clients worldwide. Ranging from Fortune 20 companies to start-ups, RBCS clients save time and money through improved product development, decreased tech support calls, improved corporate reputation and more. To learn more about RBCS, visit www.rbcs-us.com.

Address:	RBCS, Inc. 31520 Beck Road Bulverde, TX 78163-3911 USA
Phone:	+1 (830) 438-4830
E-mail:	info@rbcs-us.com
Web:	www.rbcs-us.com
Twitter:	@RBCS
Facebook:	RBCS-Inc