Whose Is It Anyways

By: Vibhu Srinivasan
{vibhu@solutionsiq.com}
Stand up if you are a developer

Stand up if you are a tester, Developers
Stand up if you are a tester, Developers keep standing
Keep standing if you have zero defects in the software you are building, else sit down.
Who Am I?
Issues or reasons
Why Scale Scrum

Coordination & People

Objectives
VIBHU SRINIVASAN

Started doing XP in 1995

20 plus real projects as developer, architect - Largest was 300 members

Love to test and code - Java, C#, C++, Rails

Member _ SIQ Global Agile Consulting Team

Teach, Practice TDD, Agile Software methods like Scrum

Coached at Yahoo, Microsoft, EBay, Cognizant, Gaming companies, Services firms, regulatory firms etc in US and India
About Us

SolutionsIQ

Placement ← Development

Training

→ Consulting
• 30 years plus
  {Young at heart}
• 10 years agile
development for
fortune X
customers
• 200 plus
customers/ teams
coached
• 3000 plus certifed
So whose bug is it?

Developers?
Developers?
Testers?
Managers?
Customers?
When I grow up, I want to be a Software Developer
I Am A Software Tester
I Can Reduce Your Code To A Twisted Heap In Minutes
TRUST ME, IM A

QUALITY ASSURANCE MANAGER
We are like this only
Over 50% of Scrum teams do not have software tested at the feature level and no known work at the end of a sprint.

Jeff Sutherland - Cocreator Scrum
Untested code is a Lean waste.
Bugs = $$$
Difference between Real Life debt and software debt?
Accountablity
Accountablity

Anyone can pay software debt, mostly our customer or the teams after us pay back quality debt
Anyone can pay software debt, mostly our customer or the teams after us pay back quality debt.
Agile Myths:

- The Debt is not possible to pay.
- Developers cannot test everything.
- Everything is only start.
- Iteration update everything is not possible to update.
- Steps are not needed.
Testers are not needed

It is not possible to complete everything in a iteration

Testing can only start when coding is complete

Developers cannot test

Testers cannot code

It is not possible to create Debt free software
Case studies
Complex algorithms for auto deploying thousands of virtual machines on a cloud get written, tested and deployed monthly
They deploy production software montly to all nodes.
One developer, one tester ratio. Testers can code and fix issues
Testing happens when dev and test sit together and test.
Fearless refactoring,
Team self motivated on quality
Testers interact with product owners and are part of requirement definition.
Friday Bug Bash
If build breaks, the team stops the dev line and focuses on fixing the build.
They also do Scrum but that is not all that important.
Number of production defects
GE Aircraft - Durham facility
Here is the secret
One defect in four engines (only cosmetic)
What makes them different
Flat Org
Importance to Team
No Bonus Scheme
High bar for entering
Daily Meetings and Open Communication
Everyone does everything
ABC communication structure
Not Convinced
Reducing Quality Debt
Unlearn. Doing the usual will not bring a change
Hire people not resources

Focus on Engineering excellence
Hire people not resources

Focus on Engineering excellence as much as you do on Scrum or Lean
Recognize TESTING not the tester
Recognize DEVELOPING not the developer

Add a Zero Defect Clause to your definition of developer
Add a Zero Defect Clause to your definition of done
Focus on the upstream NOT downstream
Focus on the upstream
NOT downstream

Requirements not understood cause most of the confusion. Eliminate too much of handovers.
Aim to reach Executable Specification

Maintain a Quality Debt W
Maintain a Quality Debt Wall

Bugs in a iteration are fixed within the sprint, only outside of sprint, log it in tracking system
Know when to stop
Maintain the flow.
Develop only that much that you can really test in a sprint.
XUnit
TDD
Learn Fitnesse
RSpec
Cucumber
Selenium
My code never has bugs it just develops random unexpected features
Make it your Bug, Pay off Debt now
Visit us at http://solutionsiq.com
Thanks for listening