

Perils & Pitfalls of the New “Agile” Tester



SQDG - March 2008

Janet Gregory

Agenda

- Introduction
- Expectations of an Agile Tester
- The 5 Perils
 - How to identify them
 - The Risk of falling
 - Techniques to avoid them
- Wrap-Up
- References
- Q & A



Introduction – Agile Vocabulary

- Incremental and Iterative
- Stories
 - Build on one another
- Iterations
- Storyboard
 - Shows progress
- Feedback
 - Drives development
- Continuous Integration



Expectations of an Agile Tester

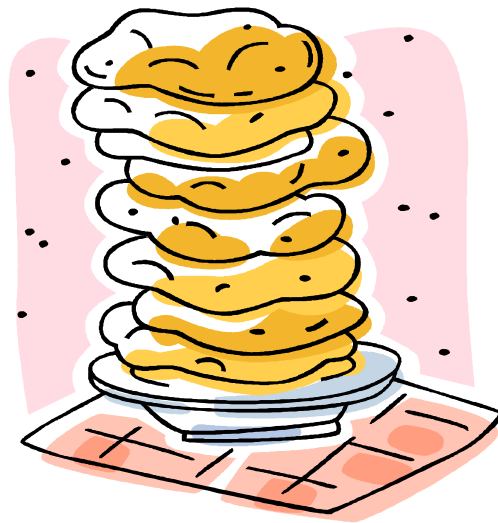
- Test without formal requirements documents
- Test in real time
- Test on changing code
- Test on changing requirements
- Automate “all” tests
- Be part of a close knit team



1-a. Waiting for Tuesday's Build

How to Identify the Peril

- Stories are stacking up in “To Test”
- Stories are not tested in the same iteration
- Deploys aren't tested regularly



1-b. Waiting for Tuesday's Build

The Risks

- Stories don't get tested
- Testers lose credibility
- Velocity is affected
- Feedback is not fast enough to be useful
- Bugs aren't discovered until developers are on to the next story
- Testing runs late into the 'end game'



1-c. Waiting for Tuesday's Build

Techniques to Avoid the Peril

- Be proactive - if you wait, you will be too slow
- Work with Build Master to get regular builds
- Plan for your test infrastructure
- Test immediately - as soon as story is ready
- Test on developer's machine if necessary – pair test
- Immediate feedback is key – get them used to it
- Anyone can sign up for testing tasks
- Include testing tasks in velocity
- Plan so that testing is balanced with coding
- **Hint:**



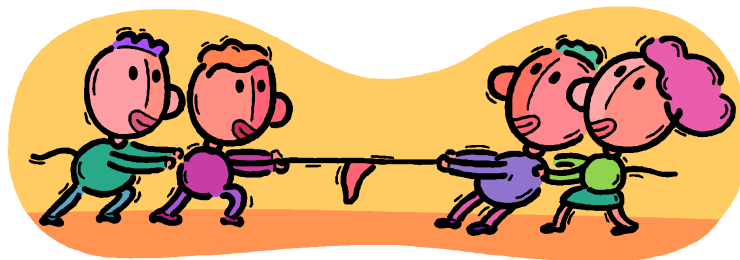
Running faster and harder is not the way



2-a. You're Not "Really" Part of the Team

How to Identify the Peril

- Testers not included in iteration planning sessions
- Discussions held between developers & business users with no testers involved
- Testers not talking during iteration planning meetings
- Business Users define stories by themselves
- Testers don't understand the stories



2-b. You're Not "Really" Part of the Team

The Risks

- Assumptions are not uncovered early
- Impacts to the system are found late
- Not the best use of everybody's skills
- You don't know what is going on!!!
- The team becomes divided



3-c. You're Not "Really" Part of the Team

Techniques to Avoid the Peril

- Push the whole team attitude
- Logistics – sit with the developers
- Become a necessity; invite yourself
- Power of 3: make it mandatory
- Ask to “just try new ideas” for an iteration or two
- Be useful – Constantly be testing and giving feedback
- Help customers develop stories
- Understand your role as tester



3-a. Maintaining a “Quality Police” Mindset

How to Identify the Peril

- All bugs are put into defect tracking system
- A separate test team
- The test team sits apart
- Test team can stop production



3-b. Maintaining a “Quality Police” Mindset

The Risks

- The team does not buy into build “quality” in concept
- Developers start using the testers as a safety net
- Communication is through bug tracking system
- The team never ‘jells’



3-c. Maintaining a “Quality Police” Mindset

Techniques to Avoid the Peril

- Get the whole team to own the “quality” of the product
- Show how each role adds value & quality
- Develop a relationship with the developers
- Use a ‘consulting’ mindset – lend your expertise
- Ensure testing happens during the iteration
- Define “Done” up front with the whole team
- Try cards on the wall for bugs found on stories



4-a. Trying to Test Everything Manually

How to Identify the Peril

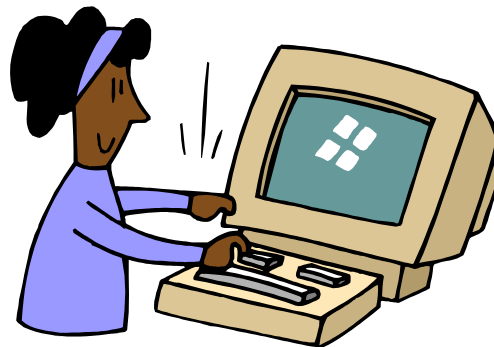
- Spending all your time retesting features already tested
- Not getting to new features
- Need more and more testers
- Not contributing to implementation / design discussions
- Regression suite not run regularly
- Making excuses to miss meetings



4-b. Trying to Test Everything Manually

The Risks

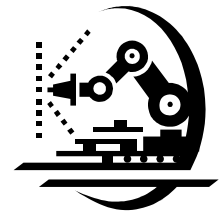
- Bugs already fixed, creep back in
- Testing cannot keep up with new stories
- Features that used to work and are broken aren't noticed
- The testers don't learn and progress using new technology
- Testers get stuck in a rut - boredom
- Developers don't get as involved in the 'customer' testing



4-c. Trying to Test Everything Manually

Techniques to Avoid the Peril

- Use automation to build your regression suite
 - Automated suites become your change detector
 - Include your automation time in estimates
- Design for testability
- Automate as you go – get developers involved
- Help developers to write good unit tests
- Find an automation tool that works for the entire team
- Understand the the power & risks of automation
- Show your skills – testing skills are technical
- Expand your toolkit



5-a. Forgetting the Big Picture



How to Identify the Peril

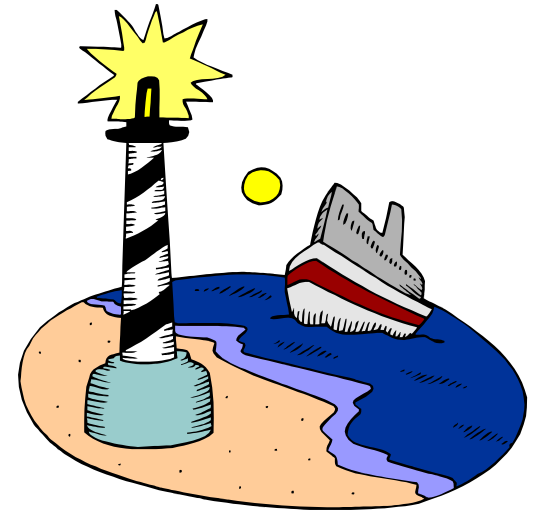
- Testing only individual stories
- Finding story integration bugs late in the release
- Reports aren't getting developed until the end
- All testing is based on what the developer tells you
- Only exploratory testing being done after the code is done
- Testing finds 'requirements' type bugs
- Developers have to add extra code after they are done



5-b. Forgetting the Big Picture

The Risks

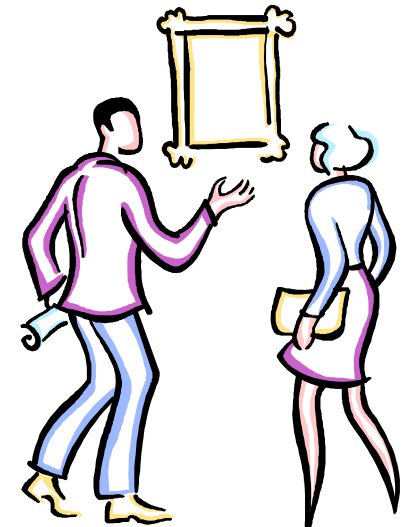
- The right thing isn't built
- Stories don't connect; there are gaps
- Pieces of the puzzle don't fit
- Workflow is not smooth
- Decisions made during coding aren't aligned with the end goal



5-c. Forgetting the Big Picture

Techniques to Avoid the Peril

- Think about impacts to other parts of the system
- Find way to build test data that reflects 'real world'
- Use business-facing tests to help drive development
- Use whiteboards to draw pictures
- Understand the story before coding starts
- Think about workflows
- Use exploratory testing
- Use examples
- Build your acceptance tests first
- Have the developers refuse to code without tests



The Five Perils

1. Waiting for Tuesday's build
 - Not keeping up with developers
2. Not really part of the team
 - No input into requirements
3. Quality Police mindset
4. Trying to test everything manually
 - Thinking you are not technical enough
5. Forgetting the big picture
 - Testing only what the developer's build



Wrap-Up

Agile Testing is full of perils

- Be aware, and watch for them
 - but don't let them scare you
- Problems happen faster so impact is felt faster
- Use an agile coach to help make changes
- Participate in the yahoo agile testing group
- Read articles and ask questions

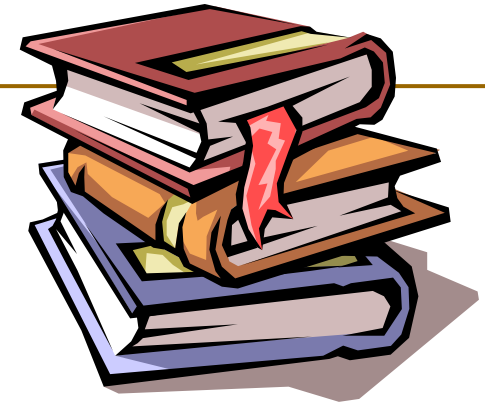
Mostly, have fun and
enjoy the challenge!



Resources

- agile-testing@yahooogroups.com
- lisa.crispin.home.att.net
- www.testing.com

- *Testing Extreme Programming*, By Lisa Crispin and Tip House
- *Fearless Change: Patterns for introducing new ideas*, Linda Rising and Mary Lynn Manns
- *Implementing Lean Software Development: From Concept to Cash*, Mary and Tom Poppendieck



Coming in 2009!

Agile Testing

By Lisa Crispin and Janet Gregory

www.agiletester.ca



Question & Answer Time

