

# A Tester's Role on an Agile Project



Montreal Agile User's Group  
April 16, 2009

Janet Gregory, DragonFire Inc.  
With material from Lisa Crispin

# Introduction - Me



First agile team – 2000

Currently – coaching, testing

Agile Testing: A Practical Guide for Testers and Agile Teams; Addison Wesley 2009



# Topics

---

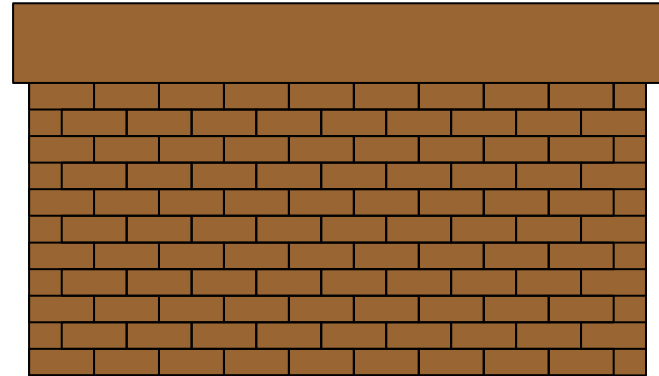
- Challenges
- What is agile testing
- Life of a tester during an iteration
- Why and how testing must “keep up”
- Continual improvement



# The “Wall” of Challenges

---

- Separate test team
- No common language
- Fear – loss of identity
- Testing always at the end
- Organizational culture
- Don't know enough about agile concepts
- Lack technical and tool skills
- Physical limitations – work space



# What is Agile Testing?

---

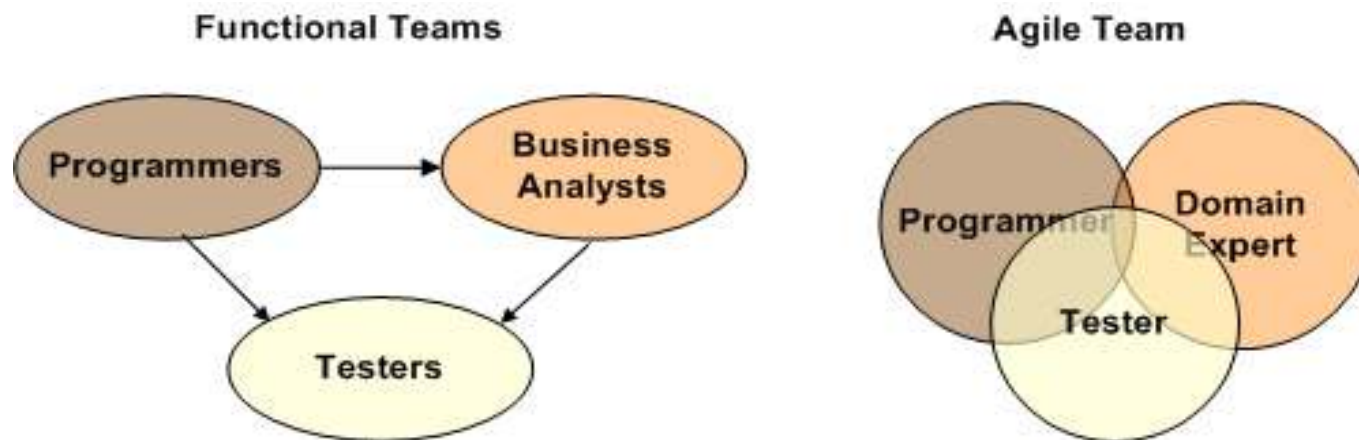
- Testers are part of the team
- Tests each story as it is complete
- Work with customers to define acceptance tests for each story / feature
- Provides continuous feedback to the team
- Practices pair testing
- Provides constructive skepticism



# Blurred Lines Between Roles

---

- Agile developers are “test infected”
- Agile testers and programmers collaborate
- Agile testers and customers collaborate
- “Whole Team” responsibility for testing
- Everyone understands the business



# Whole Team Approach

---

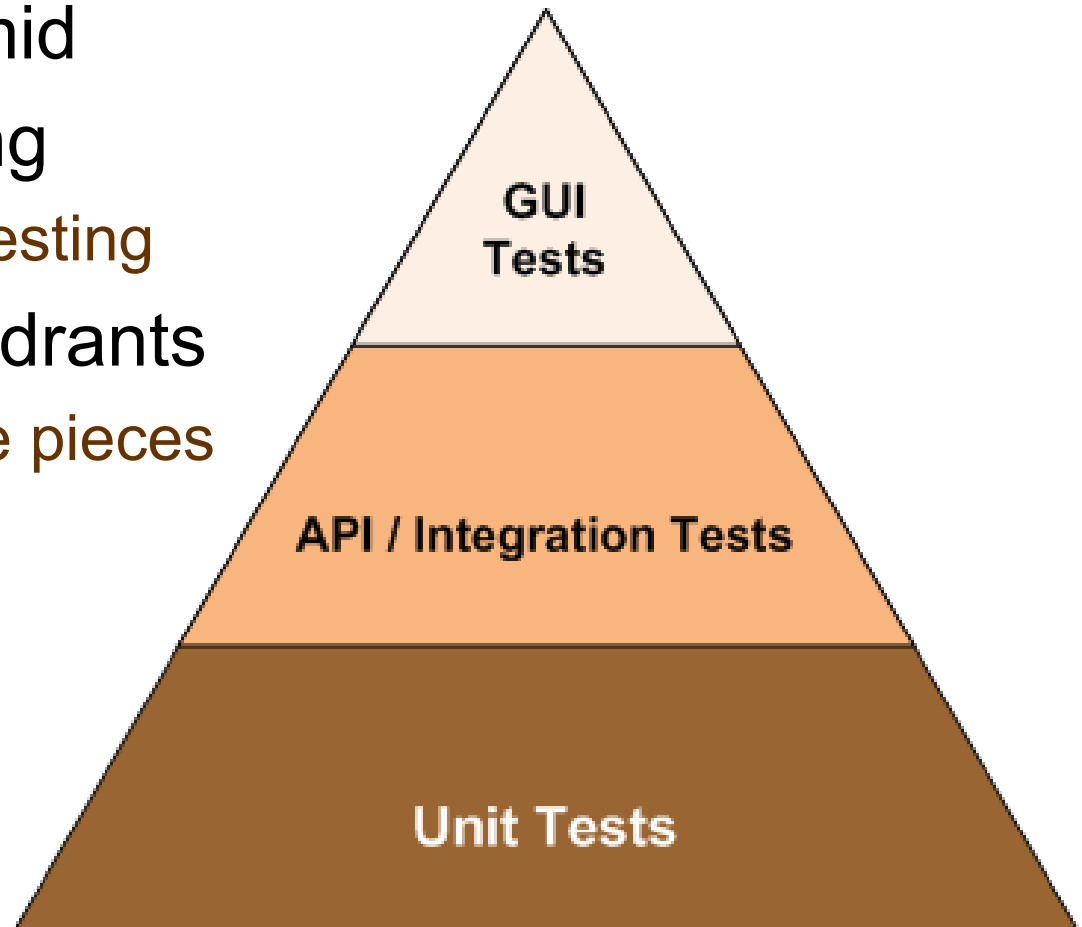
- Anyone can pick up a testing task
- Run automated regression tests regularly
  - Ideally on continuous build
  - Minimally on daily build
- Otherwise....
  - Engage the whole team in manual testing



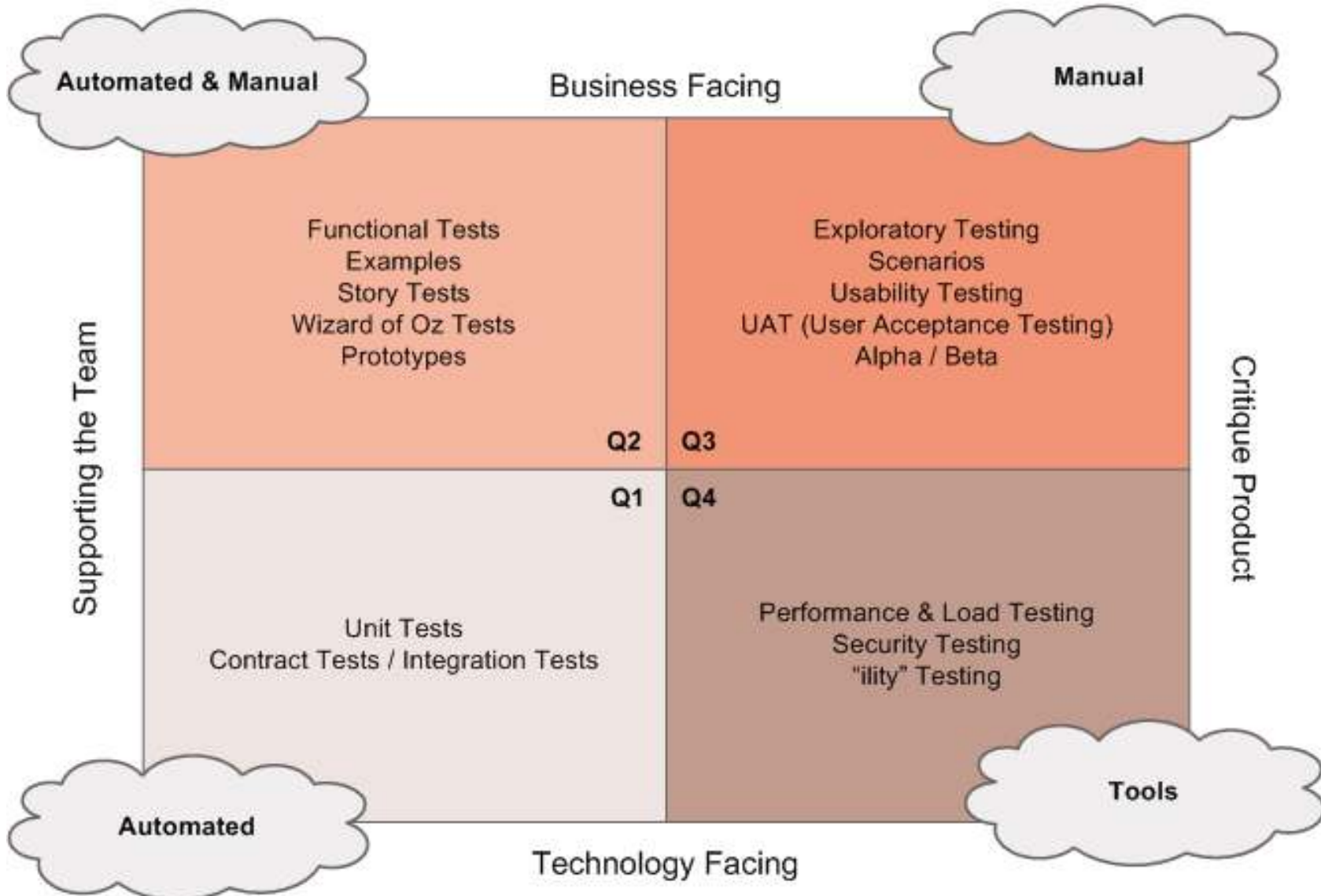
# Testing Terminology

---

- Acceptance Tests
- Automation Pyramid
- Exploratory Testing
  - Session-based Testing
- Agile Testing Quadrants
  - Think about all the pieces



# Agile Testing Quadrants



# Preparing for Release - Visibility

---

- Consider a story (or task) board
- Consider a burn down chart
- Brainstorm with development team about test strategies, automation approach, tools



# Preparing for Iteration - Stories

---

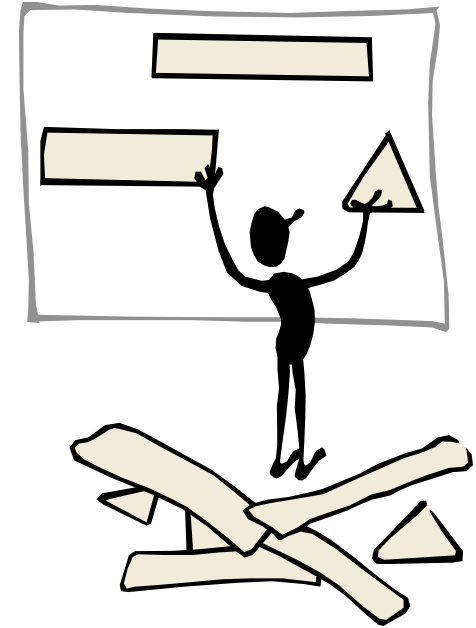
- Be proactive
- Help customer achieve ‘advance clarity’ on stories
  - Customers speak with “one voice”
  - Testable stories
  - Story size



# Laying the Foundation - Collaboration

---

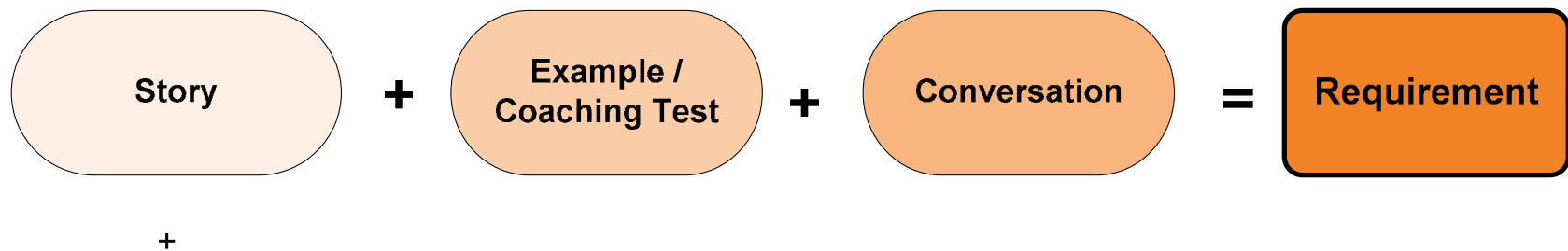
- Collaborate with customer
  - Elicit requirements
  - Find examples
  - Ask open-ended questions
  - Draw on the whiteboard
  - Get the 'big picture'
  - For complex stories, whole team may need to participate
- Consider different points of view: business, programmer, end user
  - Are there conflicts?



# During Iteration Planning

---

- Find hidden assumptions – question assumptions
- Ask “why?” Ask “what if?”
- Define and estimate testing tasks
- Ask what other tasks are required
- Define high level story tests or examples



# When Coding Starts

---

- Write detailed test cases from examples
  - Make them executable if possible
- Write simple, happy path tests to automate
  - Share them with the programmers
- Once simple tests pass, write more test cases
  - Start with happy path
  - Then move towards edge cases
- Identify possible exploratory scenarios



# Testing/Coding: Don't sit and wait!

---

- Is any testable part of a story ready?
  - Test with behind-the-GUI tool such as FIT?
  - Or other harness to bypass GUI
- Pair with programmers
  - Test together before check-in
  - Show them issues
  - Ask questions
  - Bugs found here are cheap and easy to fix



# Testing/Coding: Collaboration

---

- Pair with customers
  - Show customer early
  - Does delivered feature meet expectation?
- Other team members, e.g.
  - Work with DBA to understand database
  - Work with sys admin to get better messaging from continual build



# Exploratory Testing

---

- Good for testing new functionality
- Good when exploring troubled areas
- Sophisticated, thoughtful approach to ad-hoc testing
- Not defined in advance
- Understand your 'mission'
- Time-box the session
- Incorporate new ideas into tests
- When you want to go beyond the obvious
- Look under the rock



# Wrap it up: Last Day

---

- Whole team responsible for delivering tested code
- Show customers what you delivered
  - Could be on first day of next iteration
- This should not be a long or difficult day!
- Standup in afternoon if needed
- Release to staging environment
- Plan production release



# “Keeping Up”

---

- Understand team’s definition of “DONE”
- Understand the quality level required
- Stories need to be testable
- Estimate test tasks with development tasks
  - So the team understands the effort
- Keep track of “big picture”
  - Consider how each story affects rest of application
  - Does it affect other systems?
  - Don’t forget end to end tests
- Collaborate with customer to determine minimum success criteria



# “Keeping Up” - Automate

---

- Automate all regression tests
  - Consider collaborative approaches such as FIT / FitNesse
  - Consider ROI
  - Automate at lowest level possible
- Automate time consuming tasks
  - Calculations, parsing files...
- Make time for exploratory testing
  - Involve customer team if possible



# Retrospective – Continuous Improvement

---

- Solve problems as a team
- Is there one single limiting factor?
  - something holding you back
  - something causing a blown iteration
  - anything causing other big problems
- Action items
  - Focus on 1 or 2 things to improve
  - Create task cards for future iterations
  - Need new resources?
    - E.g.. new team member with particular expertise



# The Agile Tester Role - Review

---

- Get moving! Be proactive!
  - Don't sit and wait for things to come to you
- Who does what testing?
  - Understand the “whole-team” approach
- Collaboration is key
  - Customers/product owners/business experts
  - Developers
  - Other team members



# What's different about the agile tester mindset?

---

Instead of

- We're here to break the software!



Think

- What can we do to help deliver the software successfully?



# Now Available

## *Agile Testing: A Practical Guide for Testers and Agile Teams*

By Lisa Crispin and Janet Gregory

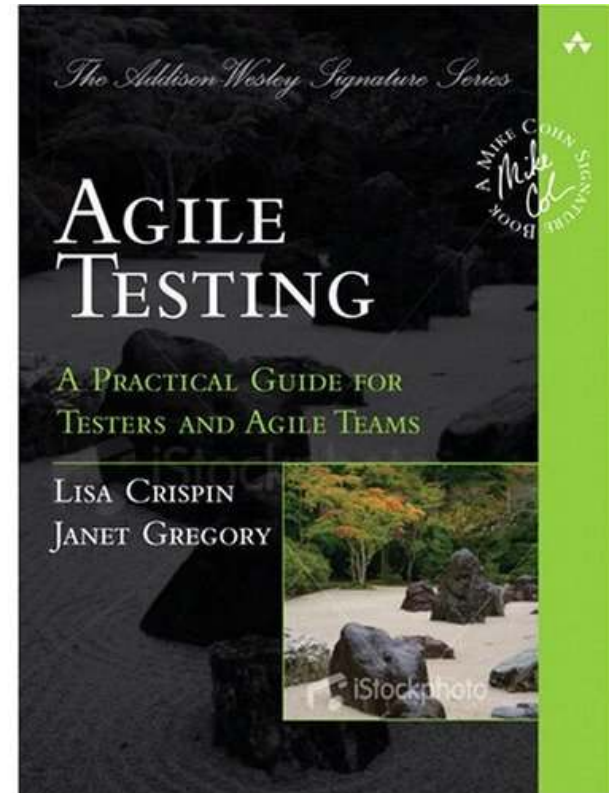
[www.agiletester.ca](http://www.agiletester.ca)

My contact info

[www.janetgregory.ca](http://www.janetgregory.ca)

<http://janetgregory.blogspot.com/>

Email: [janet@agiletester.ca](mailto:janet@agiletester.ca)



# Resources

---

- [www.lisacrispin.com](http://www.lisacrispin.com)
- [agile-testing@yahoogroups.com](mailto:agile-testing@yahoogroups.com)
- Lisa Crispin and Tip House, *Testing Extreme Programming*, Addison-Wesley 2002
- Mary Poppendieck and Tom Poppendieck, *Lean Software Development*, Addison-Wesley 2003
- [www.mountangoatsoftware.com](http://www.mountangoatsoftware.com) – Mike Cohn's web site

