Agile Testing in the Enterprise

Agile 2015
Washington, D.C.
Janet Gregory, DragonFire Inc.
Copyright 2015

@janetgregoryca
With input from Lisa Crispin
A little about me

Agile Testing; 2009
Co-authored with Lisa Crispin
Website:
www.agiletester.com
www.agiletester.ca

DragonFire Inc.
www.janetgregory.ca
@janetgregoryca
janet@agiletester.ca
Introductions

Let’s learn a bit about your experiences
- Less than 100 people
- Larger than 1,000 employees
- Larger than 5,000
- Different cities, same time zone (or 1 hr. apart)
- Time zones more than 7 hrs difference
- More than 3 remote teams
- Others?
My Message Today

to show how agile testing activities can span large teams...

which usually (not always) means distributed teams

And hopefully,
The lessons I share can help you ...
Agile Testing for Distributed and Enterprise Teams

- Intro
- Challenges
- Plan for Testing
- Key Testing Practices
- Wrap-up
Small co-located teams: Agile’s sweet spot
What I mean by “Enterprise” organizations?

Large organizations that have enterprise applications and solutions
Struggles for large agile teams

• Bureaucracy
• Reporting needs (real or imagined)
• Orders handed down ....
• Too many concurrent projects
What are distributed teams?

• Global organizations
• Many teams in many locations
  ◦ Not necessarily dispersed
The limitations of distributed teams

- Extra communication efforts
- Time zone differences
- Slower feedback
Cultural Issues (geographical & organizational)

Language

History, habits

Interactions
For example, if I said ..... “Play nice in the sandbox”
Simple Rules

• Share everything.
• Play fair.
• Don’t take things that aren’t yours without asking.
• Throwing sand is never OK.
• Being mean will eventually result in you playing, unhappily, on your own.
• No kicking or breaking other people’s sandcastles.
• Playing nice with others is best.
Other Challenges

• Splitting work between teams
• Longer time to implementation
• Lack of face time
• Lack of trust
Why is testing hard?

- Organizational controls – regulatory, internal
- Working with third parties
- Involving customers in other locations
- ERP system integration
Other Testing Specific Issues

- Throw it over the wall (out-sourcing)
- Separate test team
- Too many dependencies
- Testing integration
- Risks: don’t know who to ask
Is there anything I missed?
For successful testing, plan for ...

• Testing crosses boundaries
  ◦ Teams: physical locations, feature teams
  ◦ Different layers: architecture, levels of detail ...
  ◦ specialties

• Visibility

• Education / learning

• Tools and practices
Turn limitations in to benefits

- More diversity
- Follow the sun
- More kinds of questions
Testing is a team problem

Consider the feature

Create testable stories

Who tests what?
Think in Levels of Detail
Release Planning

Team A Planning

Team B Planning

Team C Planning

Team D Planning

Release Test Planning

Test Mind Map

Test Matrix
Dependencies between teams

Remove them before the iteration, but remember to test them
Multiple Teams - Integration

Timeline for Release 1 to Customer

Release Planning

Team A

Integrated Potentially Shippable Product

Team B

Integrated Potentially Shippable Product

Team C

Integrated Potentially Shippable Product

Team D

Integrated Potentially Shippable Product

End Game Activities (10 minutes – weeks)

Release to Production

Release / Test Planning

Post-Development Testing; workflows, “ilities”; full regression

Iteration 1

Iteration 2

Iteration 3

End Game

Support Activities – demos by project teams, involved in decision making, participating in training
Testing is a team problem, but who is your team?
Who is in your extended family?
Each team responsible for delivering their feature -- completely

Specialist teams report to the “owner” feature team.
Coordinate Tools & Practices
Consider practices for effective testing

• What is the state of testability of your architecture?
  ◦ Can you simplify it for automation?

• Regulatory
  ◦ simplest thing you can possibly do
  ◦ Consider NECESSARY artifacts – can you automate

• Working with third parties (internal / external)
  ◦ Remove dependencies
  ◦ Create fast feedback loops
Consistent Tooling

• Classes of tools
• Involve teams in choosing tools
• One size does not fit all
  ◦ But ... be aware of too many tools for support or cross-training
Team tools

• Same version control
  ◦ Understand the branching and merging approach

• Same CI environment
  ◦ Use to visualize test results

• Accessible & stable test environments
  ◦ Know what versions of applications are in each environment
Visibility / Transparency

• Share information
• Share awareness of dependencies
• Keep everyone “in the loop”
• Same vision
• Same goal
• Accessible to all
Include your extended family
Educate – Create a Learning Organization

• Common orientation
• Off –shore teams
• Acquired teams
• New remote team member
• Foster a learning culture
  ◦ Tolerate mistakes
  ◦ Personal safety
• Give time to experiment, to try
Use communities of practice for sharing
Keep the training feedback cycle short
How can testing help?
Key planning practices

• Aim for short feedback cycles
• Remove dependencies early
• Start with a test to remove assumptions
Use tests and examples for shared understanding and common language
Create a Shared Understanding

- Question
- Draw pictures
- Test ideas

- Power of 3
- Repeat verbal conversations in writing
ATDD
Acceptance Test Driven Development

- Feature (with examples)
- User Story
- Fix Defects
- High-Level AT
- Accept Story
- Explore Examples

Code, test & automate story
Pair - Share your desktop
Implicit vs explicit needs

Over-riding test strategy on how .... but

• Understand who is testing what
• What are the overlaps?
• Where are the gaps?
• Co-ordination between teams?
• ASK: Who, when, how???
Testing is a team problem

• Tests can provide fast feedback
• Examples (tests) show desired behaviour
• Know your team
In summary .... adapt

- Intro
- Challenges
- Plan for Testing
- Key Testing Practices
- Wrap-up
Work together to achieve your goal
Flexibility, Mindfulness

• Schedules
  ◦ Overlapping times
  ◦ Meetings
  ◦ Share the pain

• Use retrospectives
  ◦ Identify problem areas
  ◦ Experiment with solutions
Final note:

I’m not advocating distributed teams or out-sourcing but if you are faced with those problems, I hope you can use some of these ideas to help manage your agile testing.

and

Agile ≠ NO DOCUMENTATION

Know your own context!
And always, strive for quality!
Buy 1 Save 35%
Buy 2 Save 50%

http://informit.com/agile
Discount code: AGILESALE15

Buy the eBook Collection for 35% off or any two books for 50%

Print Books – Available at the Agile Conference Bookstore

DRM-Free eBooks are provided in EPUB, PDF, and MOBI formats – Good for all eReaders and desktop
Agile Testing: A Practical Guide for Testers and Agile Teams

More Agile Testing: Learning Journeys for the Whole Team

By Janet Gregory and Lisa Crispin

www.agiletester.ca
www.agiletester.com

Contact info

www.janetgregory.ca
Email: janet@agiletester.ca
Twitter: janetgregoryca
Instead of saying NO, be the information provider so business can make the decisions.

Questions?
More Reading & References

• *Practices for Scaling Lean & Agile Development: Large, Multisite and Offshore Product Development with Large-Scale Scrum*, Craig Larman and Bas Vodde, Addison-Wesley, 2010

• *Agile Software Development in the Large: Diving into the Deep*, Jutta Eckstein, Dorset House, 2004

• *Fearless Change: Patterns for introducing new ideas*, Linda Rising and Mary Lynn Manns
Blogs and articles

- www.lisacrispin.com
- Rothman Consulting Group, Inc. Vol 9, #5: *Building Trust in any Team*
- http://leadingsoftwaretestinginanagileworld.blogspot.co.uk/2011/03/distributed-agile-test-teams-making-it.html
- http://sites.google.com/site/visualisingquality/
- http://www.slideshare.net/johannarothman/agile-testandtestmgmtatd
More Learning

• Karten, Naomi, "Are You Listening?", http://www.agileconnection.com/article/are-you-listening, Agile Connection, 2009
• Gärtner, Markus, ATDD By Example: A Practical Guide to Acceptance Test-Driven Development, Addison-Wesley, 2012a