Agile Testing
What is it? Can it work?

bret@pettichord.com

www.pettichord.com

October 2002

The Agile Alliance Values

“We have come to value:

Individuals and interactions *over* processes and tools
Working software *over* comprehensive documentation
Customer collaboration *over* contract negotiation
Responding to change *over* following a plan

That is, while there is value in the items on the right, we value the items on the left more."

www.agilemanifesto.org

What is Agile Testing?

- 1. Testing practice that follows the agile manifesto, treating development as the customer of testing
  - In this light the context-driven manifesto provides a set of principles for agile testing.
- 2. Testing practice for projects using agile methodologies.
  - What is the role of the tester on an agile project?

Context-Driven Principles

1. The value of any practice depends on its context.
2. There are good practices in context, but there are no best practices.
3. People, working together, are the most important part of any project's context.
4. Projects unfold over time in ways that are often not predictable.
5. The product is a solution. If the problem isn't solved, the product doesn't work.
6. Good software testing is a challenging intellectual process.
7. Only through judgment and skill, exercised cooperatively throughout the entire project, are we able to do the right things at the right times to effectively test our products.

www.context-driven-testing.com

Agile Development Methodologies

- Extreme Programming (XP)
- Crystal
- Adaptive Software Development (ASD)
- Scrum
- Feature Driven Development (FDD)
- Dynamic Systems Development Method (DSDM)
- XBreed

XP Practices

- Test-First Programming
- Pair Programming
- Short Iterations & Releases
- Refactoring
- “User Stories"
- Acceptance Testing

The Role of Testing

- Testing is the headlights of the project
  - Where are you now? Where do you headed?
- Testing provides information to the team
  - This allows the team to make informed decisions
- A “bug” is anything that could bug a user
  - Testers don’t make the final call
- Testing does not assure quality
  - The team does (or doesn’t)
- Testing is not a game of “gotcha”
  - Find ways to set goals, rather than focus on mistakes

Test-First Programming

- Developers write unit tests before coding.
  - Motivates coding
  - Improves design (reducing coupling and improving cohesion)
  - Supports *refactoring*
- Many open-source test tools have been developed to support this
  - xUnit

Refactoring: Improving the Design of Existing Code

"Changing a software system in such a way that it does not alter the external behavior of the code yet improves its internal structure"

- Make the simplest design that will work.
- Add complexity only when needed.
- Refactor as necessary.
- Refactoring requires unit tests to ensure that design changes (refactorings) don't break existing code.

Acceptance Testing

- User stories are short descriptions of features that need to be coded.
- Acceptance tests verify the completion of user stories.
- Ideally they are written before coding.

Should Testers Go Along with This?

- Some say that XP is an invitation to poor quality and an excuse for hacking.
- I think that XP is exciting and will improve the practice of testing in the industry.
  - XP developers are writing unit tests and unit testing tools
  - Kent Beck's next book is *Test-Driven Development*
  - XP projects are finding new ways to build in testability and support automated tests.

Testers Should Embrace Agile Programming
http://www.io.com/~wazmo/papers/embrace_agile_programming.html

A Practice for Agile Testing

- Conversational Test Creation
- Coaching Tests
- Providing Test Interfaces
- Exploratory Learning

Conversational Test Creation

- Who should write tests?
  - Customers are often too busy.
- Defining tests is a key activity that should include programmers and customer representatives.
- Don't do it alone.

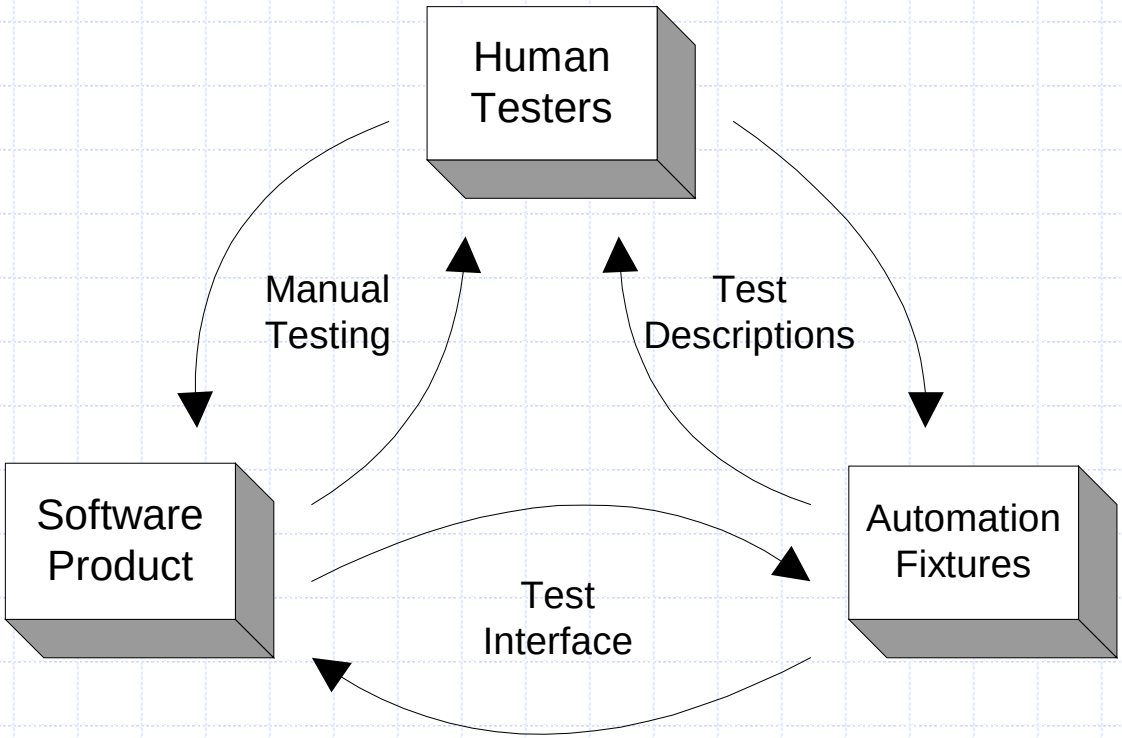
Coaching Tests

- A way of thinking about Acceptance Tests.
- Turn user stories into tests.
- Tests provide:
  - Goals and guidance
  - Instant feedback
  - Progress measurement
- Tests are specified in a format:
  - That is clear enough that users/customers can understand
  - That is specific enough that it can be executed
- Specification by Example

Providing Test Interfaces

- Developers are responsible for providing the fixtures that automate coaching tests
- In most cases XP teams are adding test interfaces to their products, rather than using external test tools

Test Interaction Model



Exploratory Learning

- Plan to explore the product with each iteration.
- Look for bugs, missing features and opportunities for improvement.
- We don’t understand software until we have used it.

Further Study

- *Lessons Learned in Software Testing*
  - www.testinglessons.com
- Ward Cunningham's acceptance testing framework
  - fit.c2.com
- Agile Testing Papers
  - www.testing.com/agile
  - www.pettichord.com