How to Apply Real World Agile Practices to Your Own Testing Projects

John Watkins, IBM Partner Software Architect Team
johnwatk@uk.ibm.com
Overview

- Introduction
- Setting the Scene for Agile
- Overview of Popular Agile Approaches
- Agile in the Real World
- An Agile “Retrospective”
- Summary and Conclusions
Overview

- Introduction
- Setting the Scene for Agile
- Overview of Popular Agile Approaches
- Agile in the Real World
- An Agile “Retrospective”
- Summary and Conclusions

“If you try to make the software Fool Proof, they will just invent a Better Fool ”
Dorothy Graham, Grove
Setting the Scene for Agile
Setting the Scene for Agile

- The Need to Respond to Extreme Competitive Pressures
- The Need to be Responsive to Customer Requests
- Need to Deliver Quality Solutions in Shorter Timescales
- Frequent Legislative Changes
- Agile Development puts Testers Under Pressure
- Testing has to be Effective & Efficient
- Developers and Testers are Increasingly Challenging the Role and Use of Heavy-Weight Processes

“Testing is Never Completed, Its Just Abandoned”, Simon Mills, MD Ingenuity
Overview of Popular Agile Approaches
A Brief Thought Before We Continue …

“Fact of the matter is, there is no hip world, there is no straight world. There’s just this world, you see, which has people in it who believe in a variety of different things.

Everybody believes in something, and everybody, by virtue of the fact that they believe in something, use that something to justify their own existence!”

Frank Zappa
Overview of Popular Agile Approaches

- There Are Many Agile Development and Testing Approaches
- Rapid Application Development
- The Dynamic Systems Development Method
- eXtreme Programming (XP)
- Scrum
- Agile Enterprise / XBreed, Ruby on Rails, Evo, RUP, Etc, etc, etc
- www.testing.com/agile/agile-testing-essay.html

“Nanos Gigantum humeris insidentes;

We are but Dwarfs standing on the Shoulders of Giants”,
Bernard of Chartres
The Rapid Application Development Method – RAD
RAD - Background

- Developed by James Martin some 30 Years Ago, and Inspired by Earlier work by Barry Boehm and Others, RAD was Developed in Response to the Perceived “Failure” of Waterfall Methods

- The Key Goals of RAD Are:
  - delivery of high quality systems
  - rapid development and delivery
  - low costs

- RAD Adopted an Iterative Development Approach, Combined with Rigorous Project Management

- Requirements would not be Set in Stone, and Customer Expectations Managed through Prototyping

- Supports Early and Frequent Testing Throughout Development
RAD - Benefits

- RAD Provided Benefits Over Traditional Methods in Number of Areas:
  - Iteration Supported Early and Frequent Testing
  - Project Progress Kept on Track Through Rigorous PM
  - Rapid Development of Prototypes Allowed Early Review by the Customer and Helped to Manage Customer Expectations
  - Frequent Planning and Design Workshops Ensured Deliverables More Closely Met Customer Expectations
RAD - Issues

- Considered to Have Poor Rigor; Many Traditional Developers Thought RAD provided a “Hackers Charter”
- Prototyping Approach Often Criticised for Wasting Developer Effort and Jeopardising Progress
- Prototyping Approach Often Poorly Implemented; Failed to Involve Customer Early Enough, and Poor Planning for Reuse
The Dynamic Systems Development Method – DSDM
DSDM - Background

- Developed in the 1990’s by a Consortium of Organisations and Agile Practitioners, and Building on the “Success” of RAD
- Based on an Iterative Model that Seeks to be Responsive to Changing Customer Requirements
- Seeks to Implement the Customer Business Requirements on Time, to Budget & to Acceptable Levels of Quality
- Typically Applied to Projects with Challenging Timescales and Budgets
- Looks to Address Quality Issues, Lack of Customer Involvement, and Lack of Senior Management Commitment
- Perceived as More Rigorous than RAD; Founded on 9 Key Principles, and Structured under 3 Phases
DSDM - Benefits

- Good Success in Delivering “Problem Projects” on Time, to Budget and with Quality through Rigorous Project Management
- Early and Frequent Testing Supported by Iterative Approach and by Explicitly Supporting Testing as 1 of the 9 Key Principles
- Customer Expectations Managed by a Focus on Early and Frequent Delivery, and by Mandating Active User Involvement
- Responsive to Changing Customer Needs; Requirements Baselined at a High Level with Ongoing Opportunities for Customer Input
- DSDM Perceived as a Rigorous Approach with Structured 3 Phase (Pre-Project, Life Cycle & Post Project Phases) Method
DSDM - Issues

- Many “eXtreme” Agile Practitioners See DSDM as Being too Formal, too Complex and too Prescriptive
- Although Focused on Rigorous Project Management, DSDM Projects Frequently Adopt Other Project Management Solutions (such as PRINCE2)
- Little Uptake, Adoption and Use of DSDM by Small, Low Budget, Short Duration Projects
eXtreme Programming – XP
XP - Background

- Developed in the Early 1990s by Kent Beck, who had begun to consider how software development could be made more simple and efficient, culminating in 1996 with Extreme Programming.

- XP emphasises customer satisfaction as one of its key drivers, and aims to deliver the software the customer needs when they need it.

- XP looks to improve project success in four main areas:
  - Communication
  - Simplicity
  - Feedback
  - Courage

- Testing is an explicit (although simplified) aspect of the method.
XP - Benefits

- Early and Frequent Testing Supported by Iterative Approach, and within a Simplified Testing Framework – Unit Test and Acceptance Test
- Improved Quality of Developed Software; Techniques such as Pair Programming Improve Functional Quality and Reduce Bugs
- Improved Quality of Delivered Software; Testing is Key to XP, and it employs Techniques Such as Test Driven Development, and Insistence on all Code having Tests
- Ongoing Focus on Quality Throughout Development; Whenever a Defect is Detected, it is Fixed and a Suitable Test Developed and Stored for Later Reuse
XP - Issues

- Management Mistrust; Too Often Senior Managers Consider Extreme Programming to be an Excuse for Hacking
- Practitioner Abuse; (rarely) Individuals and Teams Claim to be Using an Agile Approach but Actually Using Ad-hoc Approach
- Extreme Programming Often Thought of as Excellent on Practitioner Practices, but Being a bit Light on Project Management Guidance
Scrum
Scrum - Background

- As Early as 1986, Takeuchi & Nonaka had Observed that Projects Employing Small, Cross-Functional Teams Were Typically Most Successful – Coining the Phrase “Rugby Approach”

- In 1991, DeGrace & Stahl Coin the Phrase Scrum in the Context of Software Development

- Schwaber & Sutherland Deliver the Definitive Scrum Description at the OOPSLA ’96 Conference

- Scrum is a Project Management Method for Agile Projects, that Enables the Creation of Small, Self-Organising, Co-located Teams, Employing Effective Verbal Communications

- A Key Principle of Scrum is the Recognition that Customers are Likely to Change Their Minds Frequently During a Project, and that the Team Must Respond Quickly to Changing Requirements
Scrum - Benefits

- Accurate, Clear and Unambiguous Communications Achieved Through Frequent Short and Focused Formal Meetings
- Well Understood and Accepted Division of Project Participants into Pig and Chicken Roles (Helps Focus Meeting Comms)
- Good Management of Customer Expectations Through Frequent Intermediate Deliveries of Working Code
- Good Control of Project Risk by Making Everyone Responsible for Identifying and Highlighting Project Issues
- Excellent Feedback on Progress Through High Visibility Means (such as project Dashboards)
- Project Lessons Learnt and Acted On
Scrum - Issues

- Strong on Agile Project Management Approach, But Weaker on Specific Agile Development Practices
- Frequently Claimed to Only Work on Small, Well Defined Projects, with Co-Located Customer Staff and Good Agile Practitioners
- May Have Issues with the “Right” Co-Located Customer Representative(s)
- Pure Scrum Approach May Be Deflected/Hi-Jacked by Controlling Managers with a Traditional Methods Bias
- Scrum Sometimes Seen as a “Bit Mystic” or “Secret Society”
- How Can Scrum Work in a Large, Complex, Off-Site/Off-Shore Project, with Practitioners With Little Experience
Other Agile Methods
Other Agile Methods

- Agile Enterprise / X-Breed (a Fusion of Scrum & XP)
- Evo (Tom and Kai Gilb)
- Ruby on Rails
- Eclipse Way
- RUP, EssUP, SPEM

“Agile Development and Testing is Arguably a 30 Year Old Overnight Success!”

Bob Bartlet, MD SQS UK
Common Themes

- Good Communications
- Acceptance that Requirements Will Change
- Co-location of staff & customer representatives
- Iterative Style of Development
- Agile Meetings, Start-ups & Close-Downs
- Test Design before Coding
- Well Trained, Motivated & Empowered staff
- Close Collaboration between staff – such as Pair-Programming
- Team Ownership and Responsibility for Quality
- Agile Automation

“The Easiest way to get any Agile Methods to Fail, is to expect too much from it”
Ryan Concannon, Scrum Master
Agile in the Real World
Agile in the Real World - Overview

- Based on a review of some 20 Agile Case Studies
- Covers Many Different Agile Methods
- Covers a Variety of Different Project Environments
- Conducted as Research Towards a book on Agile Testing for publication in May 2009
- Reports Varied Views on the Successes and Failures of Agile Projects

“There was a feeling that with this agile “Testing Driven” change program, that the changes were tolerated as long as they didn’t make any difference to the way the company operated!”

Graham Thomas, Independent Test Practitioner
Analysis of the Case Studies

- Summarises Highlights of Case Study Research
- Reviews Successful Aspects of Agile Projects
- Organises Findings Under Agile Practices For:
  - All Agile Projects (Foundation Agile Practices)
  - Agile Practices for Small Projects
  - Agile Practices for Medium Projects
  - Agile Practices for Large Projects
  - Agile Practices for Off-Site/Off-Shore Projects

“If you could just stop finding errors, we could put the time in to make sure you get the full scope of the system delivered”, MD of an Agile Testing Company to their Customer, via Geoff Thompson of Experimentus
Structure of Results

- **Within Foundation, Small, Medium, Large & Off-Site/Off-Shore, the Practices are Organised Under:**
  - Agile Development and Testing Practices
  - Agile Process and Project Management Practices
  - Agile Requirements Management Practices
  - Agile Communications and Meetings Practices
  - Agile Automation Practices
Foundation Agile Practices
Results of the Analysis of the Case Studies, Agile Development and Testing … Foundation

- Iterative Development (with Well Bounded Iterations)
- Early Involvement of Test Resources
- Every Day is Test Day
- Test Driven Design
- Fix All Defects Immediately
- Collective Code and Test Ownership
- Agile Exploratory Testing
Results of the Analysis of the Case Studies, Agile Process & Project Management … Foundation

- Co-Location of Project Stake Holders
- Agile Estimation
- Progress Measurement
- Progress Visualisation
- Process Improvement
- Everyone is a Software Engineer
Results of the Analysis of the Case Studies, Agile Requirements Management … Foundation

- Employ Use Cases / User Stories
- Look Beyond Requirements to Customer Needs
- Ensure All Requirements Have Tests
Results of the Analysis of the Case Studies, Agile Communications and Meetings … Foundation

- Agile Project Start-Up Meetings
- Agile Iteration Start-Up Meetings
- Daily Stand Up Meeting
- Agile Retrospectives
Results of the Analysis of the Case Studies, Agile Automation … Foundation

- Automated Unit Test
- Automated Configuration Management
Agile Practices for Small Projects
Results of the Analysis of the Case Studies, Agile Development and Testing … Small Projects

- Consider the use of Pair Testing
- Consider the use of Rapid Prototyping
- Adopt a Continuous Integration Approach
Results of the Analysis of the Case Studies, Agile Process and Project Management … Small Projects

- Consider the Role and Use of Metrics
- Consider Making Agile Process Guidance Available
Results of the Analysis of the Case Studies, Agile Communications and Meetings … Small Projects

- Consider How to Improve Interpersonal Communications
- Consider Holding Interim Iteration Meetings
- Consider Agile Project Close Down Meetings/Extended Retrospectives
Results of the Analysis of the Case Studies, Agile Automation … Small Projects

- Consider the Role of Static Analysis Tools
- Consider the Role of Test Harness Tools
- Consider the Role of Requirements Management Tools
Agile Practices for Medium Projects
Results of the Analysis of the Case Studies, Agile Development and Testing … Medium Projects

- Employ Pair Testing
- Employ a Continuous Integration Process
- Consider the Role of Test Refactoring
- Consider Identifying the Targets of Test
- Consider Employing Code Coverage Metrics
- Strongly Consider Rapid Prototyping
Results of the Analysis of the Case Studies, Agile Process and Project Management … Medium Projects

- Be Receptive to the Reuse of Traditional Testing Techniques
- Make Agile Process Guidance Available
- Ensure Metrics are Collected and Used
- Reduce Documentation Overload
- Make Best Efforts to Ensure Co-Location of Project Stake Holders
Results of the Analysis of the Case Studies, Agile Communications and Meetings … Medium Projects

- Focus on Improving Interpersonal Communications
- Hold Interim Iteration Meetings
- Hold End of Iteration Retrospectives
- Consider the Role and Use of Agile Workshop Meetings
- Hold Agile Project Close Down Meetings
Results of the Analysis of the Case Studies, Agile Automation … Medium Projects

- Consider the Role and Use of a Formal Tool Selection Process
- Adopt and Use Static Analysis Tools (with Build Process)
- Strongly Consider the Role and Use of Test Harness Tools
- Consider the use of Functional Test Tools
- Strongly Consider the Role and Use of Requirements Management Tools
- Consider the Role and Use of Build Management Tools
- Consider the Use of Change Management & Defect Tracking Tools
Agile Practices for Large Projects
Results of the Analysis of the Case Studies, Agile Development and Testing … Large Projects

- Implement a Pair Testing Approach
- Adopt a Continuous Integration Approach
- Adopt and Encourage Test Refactoring
- Identify Targets of Test
- Utilise Code Coverage Metrics
- Adopt Rapid Prototyping
Results of the Analysis of the Case Studies, Agile Process & Project Management … Large Projects

- Be Sensitive to the Reuse of Traditional Testing Techniques
- Adopt Accurate Solutions to Progress Measurement
- Adopt Effective Solutions to Progress Visualisation
- Provide Effective Access to Process Guidance
- Employ a Thorough Metrics Scheme
- Reduce Documentation Overload
- Make Best Efforts to Ensure Co-location of Project Stakeholders
Results of the Analysis of the Case Studies, Agile Communications and Meetings … Large Projects

- Adopt Formal Methods of Improving Interpersonal Communication
- Hold Interim Iteration Meetings
- Where Appropriate Hold Agile Workshop Meetings
- Hold Agile Project Close-Down Meetings
Results of the Analysis of the Case Studies, Agile Automation … Large Projects

- Strongly Consider the Role and Use (reuse) of a Formal Tool Selection Process
- Strongly Consider the Adoption of Process Enactment Tools
- Adopt and Use Static Analysis Tools (with Build Process)
- Adopt and Use Test Harness Tools
- Strongly Consider the use of Functional Test Tools
- Adopt and Use Requirements Management Tools
- Adopt and Use Build Management Tools
- Adopt and Use Change Management & Defect Tracking Tools
Agile Practices for Off-Site/Off-Shore Projects
Results of the Analysis of the Case Studies, Agile Process & Project Management … Off-Site/Shore Projects

- Adopt Effective Distributed Real-Time Progress Measurement Solutions
- Adopt Effective Distributed Real-Time Progress Visualisation Solutions
- Ensure the Availability of Distributed Process Guidance
- Adopt an Effective Distributed Real-Time Metrics Scheme
Results of the Analysis of the Case Studies, Agile Communications and Meetings … Off-Site/Shore Projects

- **Adopt Effective Solutions to Manage Non-Co-Located Teams:**
  - Set up and use an effective project communications infrastructure
  - Consider Harmonising off-shore time zones
  - Consider Co-location of an off-shore representative

- **Adopt Effective Solutions to Support Agile Meetings**
  - look to Web 2 tools to assist

- **Improve Interpersonal Communications**
Results of the Analysis of the Case Studies, Agile Automation … Off-Site/Shore Projects

- Any Tools Must be Capable of Distributed Real-Time Use
- Ensure a Formal and Rigorous Tool Selection Process
- Very Strongly Consider the Adoption of Process Enactment Tools
- Adopt and Use Requirements Management Tools
- Adopt and Use Build Management Tools
An Agile Retrospective
An Agile Retrospective …

- Your Turn to Consider the Characteristics of Your Own Testing Projects
- A Chance to Reflect on the Agile Best Practices
- An Opportunity to Pick and Mix from the Agile Best Practices (and Others) to Create “Your Agile Process”

“I am for those means which will give the greatest good to the greatest number ”,

Abraham Lincoln, former President of the USA
An Agile Retrospective …

- **Look for Equivalence:**
  - A Small But Complex Project May Need Large Project Practices
  - Complexity Can be Caused by:
    - difficult technical issues
    - challenging time-scales, budget and/or resources
    - new and/or inexperienced team members
    - challenging political issues (remember Nick Denning)!

“Agile teamwork is not necessarily a natural way of working for all of us, but we can all improve our communication by understanding ourselves, empathising with others and using techniques to structure our communication processes”, Isabel Evans, Testing Solutions Group
Handouts

1. Checklist to Help You Think About the Characteristics of Your Testing Project
2. Checklist of the Agile Practices Discussed Earlier
3. A Form to Compile Your Agile Best Practices

“The easiest way to get any Agile Method to fail is to expect too much from it”, Ryan Concannon, Scrum Master
Wrap Up

- How Did You Get On?
- Any Interesting Findings / Observations?
- Any Additional Practices You Would Recommend?
- What Will You Do Next:
  - Look at Your Testing Project in a Different Way?
  - Consider Influencing Your Management?
  - Consider Running an Agile Pilot?
  - Buy My Book (please ;-) ?
Summary & Conclusion
REMEMBER ...

- There is NO One Size Fits All Test Process
- Everyone is Different:
  - different sized projects
  - different code complexity
  - different budgets, timescales and/or resources
  - different team members of varying experience
  - different project politics
  - different purposes; in-house, bespoke, commercial, etc

- Tailor Your Approach with Knowledge, Experience & Sense
Summary

- Testers are under Increasing Pressure to be as Effective & Efficient as Possible; Use of Heavy-Weight Development & Testing Approaches is Being Challenged

- There are Many Agile Approaches that Need to be Considered; Real World Practitioners often have Mixed Success with Agile Approaches

- There are a Number of Agile Practices that the Case Study Analysis Suggests as being Successful

- Introduction, Roll-out and Adoption of Agile Practices can be Challenging, but the Results can be Rewarding

“The Sprint is King; Long Live the Sprint. Am I getting Old or can you really think through all the testing issues in just 10 days? Developing & Testing Software is an intellectual activity where the “eureka” moment for testers takes time to develop”, Steve Allott, MD Electromind
Conclusion

- So Much of Agile Best Practices Hinge on Good Communications ...
Conclusion

- So Much of Agile Best Practices Hinge on Good Communications …

“You start a conversation you can't even finish it. You're talkin' a lot, but you're not sayin' anything.

When I have nothing to say, my lips are sealed. Say something once, why say it again?”

Psycho Killer, Talking Heads, via Isabel Evans
Thank You
How to Apply Real World Agile Practices to Your Own Testing Projects

John Watkins, IBM Partner Software Architect Team
johnwatk@uk.ibm.com
Rollout and Adoption of Your Agile Process
Roll-Out & Adoption of Your Agile Process

- Don’t underestimate the Challenge of Introducing Agile into Large Organisations with Flat Organisational Structures
- Appoint an Agile Champion
- Report Successes and Promote Benefits
- Ensure Time & Budget is Allocated for Agile Training & Mentoring
- Include Lead Times for Tools Acquisition in any Roll-out plan
- Strategies Must be Developed to Turn-around Objectors (Avoid Us & Them Syndrome)
- Last, But Not Least, Ensure you have Management Buy-in

“As is so often the case, improving quality turns out not to be a technology problem, but a people issue and a leadership challenge”, Nick Denning MD of Diegesis