When Waterfall and Agile Collide: Managing the Balance

Pete DuPre’
Mark Kulak
Bryan Fangman

Seattle, Washington
January 23, 2013
Borland Strategic Solutions
Chief Architect

PETE DUPRE’
Overview

- The current state of software project success and industry trends
- What’s wrong with traditional requirements management methods
- Latest innovation and concepts in Requirements Definition & Management with Agile delivery
Trends - Software project success

The Standish Group International, Inc. 2011 CHAOS Manifesto
"2012 results represent the highest success rate in the history of the Research. The reasons for the increase in success rate are:"

- Agile Process ↑ 22% CAGR
- Modernization ↑
- Enterprise packages ↓
What does Agile really mean?
Top Ten CIO Issues

Technology
1. Analytics
2. Mobile
3. Cloud
4. Collaboration
5. Legacy modernization
6. IT management
7. CRM
8. Virtualization
9. Security
10. ERP Applications

Business
1. Enterprise growth
2. Deliver operational results
3. Reduce costs
4. Attract/Retain new customers
5. Improve IT applications and infrastructure
6. Innovation
7. Improve efficiency
8. Attract/Retain workforce
9. Implement analytics
10. New markets and geographies
“...we delivered productivity that was five times the industry average while defect rates were just a quarter of similar projects...”

“This has enabled eight consecutive quarters of on-time, on-budget product deliveries with new functionality added quarterly”
Comprehensive risk management is the single most compelling argument for Agile.

“We found the transparency, control and flexibility of Agile have been of critical importance to both successful delivery and early highlighting of problems.”
Agile Adoption: Typical Roadblocks

- Functional silos
- Lack of collaboration from the business
- Unwilling to invest – gain isn’t free
- Management Skeptics
- Automation to support scale and distributed teams
Agile In Context

Improve quality
Manage change
Increase satisfaction
Speed time to market

Sources: Forrester, 2009 and 2011
Borland Caliber – Requirements Definition & Management
Director of Product Development

MARK KULAK
Waterfall lifecycle

Systems Development Life Cycle (SDLC)

Life-Cycle Phases

Requirements Definition & Management

- **Initiation**
  - Begins when a sponsor identifies a need or an opportunity.
  - Concept Proposal is created.
  - Defines the scope or boundary of the concepts.

- **System Concept Development**
  - Develops a Project Management Plan and other planning documents.
  - Provides the basis for acquiring the resources needed to achieve a solution.

- **Planning**
  - Analyses user needs and develops user requirements.
  - Create a detailed Functional Requirements Document.

- **Requirements Analysis**
  - Transforms detailed requirements into complete, detailed Systems Design Document.
  - Focuses on how to deliver the required functionality.

- **Design**
  - Converts a design into a complete information system.
  - Includes acquiring and installing systems environment; creating and testing databases; preparing test case procedures; preparing test files, coding, compiling, refining programs; performing test readiness review and procurement activities.

- **Development**
  - Demonstrates that developed system conforms to requirements as specified in the Functional Requirements Document.
  - Conducted by Quality Assurance staff and users.
  - Produces Test Analysis Reports.

- **Integration and Test**
  - Includes implementation preparation, implementation of the system into a production environment, and resolution of problems identified in the Integration and Test Phases.

- **Implementation**
  - Describes tasks to operate and maintain information systems in a production environment.
  - Includes Post-Implementation and In-Process Reviews.

- **Operations & Maintenance**
  - Describes end-of-system activities, emphasis is given to proper preparation of data.

- **Disposal**
Requirements Definition & Management

- Elicitation
- Definition
- Management
- Approval

© 2013 Borland
Waterfall Project Management

**Strengths**

1. Clear understanding of expected scope
2. Produces complete design and documentation
3. Clearly visible project history and evolution
4. Progress through stages is clearly visible
5. Definition up front allows understanding of what is fully expected
6. Enforces discipline by requiring completion of one stage before moving to the next
7. Staged approach is relatively easy to implement and understand

**Weaknesses**

1. Mistakes in requirements leads to significant wasted effort
2. Business needs change as defined requirements wait for implementation
3. Great difficulty in adapting to change across phases
4. Defining full features up front is difficult
5. Estimating up front is often inaccurate
6. Visibility of detail across stages is limited (isolated by project roles)
7. Work occurs in cycles creating peaks and valleys of activity
8. Monumental review process

© 2013 Borland
Agile Software Development

Time for Change
<table>
<thead>
<tr>
<th>Agile Manifesto</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.</strong></td>
</tr>
<tr>
<td><strong>Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.</strong></td>
</tr>
<tr>
<td><strong>Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.</strong></td>
</tr>
<tr>
<td><strong>Business people and developers must work together daily throughout the project.</strong></td>
</tr>
<tr>
<td><strong>Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.</strong></td>
</tr>
<tr>
<td><strong>The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.</strong></td>
</tr>
<tr>
<td><strong>Working software is the primary measure of progress.</strong></td>
</tr>
<tr>
<td><strong>Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.</strong></td>
</tr>
<tr>
<td><strong>Continuous attention to technical excellence and good design enhances agility.</strong></td>
</tr>
<tr>
<td><strong>Simplicity—the art of maximizing the amount of work not done—is essential.</strong></td>
</tr>
<tr>
<td><strong>The best architectures, requirements, and designs emerge from self-organizing teams.</strong></td>
</tr>
<tr>
<td><strong>At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.</strong></td>
</tr>
</tbody>
</table>

- Individuals and interactions... over process and tools
- Working software... over comprehensive documentation
- Customer collaboration... over contract negotiation
- Respond to change... over following a plan
Agile / Iterative / Scrum...
Agile Project Management – Requirements

**Strengths**

1. Adapts to changing business needs in short cycles
2. Allows progress with limited information
3. Mistakes in requirements lead to limited wasted effort
4. Inclusive of all stakeholders across project
5. Provides clear status in the form of working software
6. Work occurs in a sustainable manner

**Weaknesses**

1. Limited requirements change management / visibility
2. "Fail fast" approach knowingly introduces regular rework
3. Non-functional requirements are difficult to capture as user stories
4. "Agile" often implemented as un-governed methodology
5. Project scope not clearly understood
6. Lack of structure and necessary documentation
7. Delivery focused
Agile is now widely recognized as a software lifecycle

- Agile was a pullback from very tightly managed and highly governed waterfall lifecycle

- Manifesto was required
  - A radical change required a radical action
  - Agilista

- Agile needs to fit for the enterprise...
  - Visibility
  - History
  - Traceability
  - Governance

- Time to adapt... “be agile”, don’t “do agile”
Agile / Iterative / Scrum...

Requirements Definition

- Elicitation?
- Definition?
- Collaboration?
- Review?
- Versioning?
- Change Management?
Requirements Definition & Management

Elicitation

Definition

Management

Approval
Two Life Cycles... Working as One

As structured as you need to be... as Agile as you want to be
Agile requirements definition

- Requirements backlog
  - Managed to enable consensus, visibility and governance

- Incremental definition
  - Detail defined according to relative priority
  - Definition work spent on low priority requirements is waste
  - Definition work beyond that required for communication is waste

- Iteratively turn over "well formed" requirements / user stories
  - Defined, reviewed and approved... Manage the backlog

- Success measured against “business need” not against “development unit”
  - When is a product releasable?... Minimally marketable features
Defining Business Need - Requirement

- Idea
- High level goals
- Basic visualization
- Use cases
- Functional requirements
- Non-functional requirements
- Fully formed Requirement

Business Need Driven

Increasing Detail
Defining Development Unit – User story

Feature → User story → Breakdown for work → Development Need Driven → Task
Agile Requirements Bell Curve

Defining Business Features

Defining Development Detail

Only define up front what is required for process and communication
Incremental Definition = Practical Fidelity

Definition beyond what is required for communication is waste
Pragmatic definition

Defining Business Features

WASTE

Defining Development Detail

Detail required for business clarity

© 2013 Borland
Different for every organization, project, requirement

Regardless of level of rigor in the organizational process, requirements must be managed
Agile Requirements Definition and Delivery

• Two lifecycles working as one
  – Iterative requirements definition
  – Iterative requirements delivery (Agile development)

• Less waste by assuring business and delivery are in alignment
  – Fail fast... and less frequently

• Pragmatic definition - only define enough detail to satisfy organizational needs and delivery clarity
  – Anything beyond is waste
Borland Caliber – Requirements Definition & Management
Senior Product Manager

BRYAN FANGMAN
The Future of RDM and Agile Tools

- Visualizations will become increasingly important for requirements elicitation and validation
- Testing will cover requirements at the time of definition as well as the incremental deliverables in each supporting story
- As Agile practices become more managed, Agile requirements/stories will trace to an increasing number of related elements
- More stakeholders will become actively involved in the elicitation and approval process as tools improve efficiency
The Future of RDM and Agile Tools (cont)

- The majority of individuals participating in the requirements management process will use web-based and mobile solutions.
- Organizations will become more efficient in centralizing, standardizing and reusing requirements.
- The accuracy of the initial and ongoing time/cost estimations will improve and occur earlier in the lifecycle.
Borland’s Approach

We are **Open**... to the way you want to work and to the investments you have already made in your ALM tooling

We are **Agile**... in how we work with you, responding quickly to your changing requirements, bringing the benefits of Agile to existing processes

We are **Enterprise**... in our scale and understanding, satisfying even the largest of organizations
Our Own Award-winning Agile Transformation

Borland received an Info World Top 100 Innovation Award for transitioning its own development organization to agile

The award recognizes Borland’s Agile Transformation as an IT project that exemplifies intelligent, creative use of technology to meet business and technical objectives. Like most organizations considering a large-scale Agile shift, Borland faced a major process renovation while still needing to execute on an aggressive product roadmap. As Borland began to scale its Agile efforts across an organization with more than 350 developers in five geographic locations, teams needed a better way to collaborate, share information and manage their work. At the same time, management needed visibility in order to establish a baseline for performance and be able to measure the progress and benefits of the transition to Agile.
Validated by Partners - Microsoft

The ideas we have shared are being echoed throughout the industry. [http://blog.hinshelwood.com/requirement-management-in-the-modern-application-lifecycle/](http://blog.hinshelwood.com/requirement-management-in-the-modern-application-lifecycle/)

Solution: Caliber from Borland (Partner)

If you have enterprise teams that require more structure while still allowing teams to decide how they are building then you need a tool that embodies that vision so that it will grow into

---

The Modern App Lifecycle

![Diagram of the Modern App Lifecycle]

Waste elimination | Cycle time reduction | Integration & visibility

Define | Ideation | Requirements | Product building | Measure | Actionable learning

Develop | Idea to working software | Working software shared artifacts | Working software in production | Value realization

Operate | Continuous feedback | Continuous quality | Continuous delivery

Figure: Requirement Management is part of Define / Ideation

* See full article in upcoming inaugural issue of ALM magazine
Enabling a Balanced Waterfall/Agile Approach

• Define clear goals

• Adopt a process that conforms to your preferred methodology
  – Ensure the framework does not constrict productivity
  – Respect roles that are not traditionally Agile

• Find a tool that fits within your methodology
  – Allow each role to work with the tools they want
  – Avoid restrictive workflows and rules

• Centralize best practices from successful projects

• Expand visibility between Agile assets and related artifacts for real-time impact analysis

• Focus on the right thing to do at the right time
Conclusion

When Waterfall and Agile collide the result can be

Managed Agile
Waterfall, Agile, others... we can help!

BORLAND SOLUTIONS
Borland Solutions

**Caliber**
- Requirements Elicitation
- Storyboards
- Prototyping
- Test Case Generation
- Glossaries
- Baselines
- Approvals
- Traceability

**SilkCentral**
- Test Management
- Test Execution
- Defect Tracking
- Test Reporting

**StarTeam**
- Source Control
- Issue and Task Management
- Continuous Build and Test
- Development Metrics
- Agile Project Management

**SilkMobile**
- On Device test automation
- iOS, Android, BlackBerry

**SilkTest**
- Regression Testing
- Automated Functional Testing

**Silk Performer**
- Performance Metrics
- Automated Load Testing
- Cloud Execution

Create Test Case Scenarios

Synchronize Requirements and Create Test Cases

Execute Tests and Collect Results

Synchronize Issues

Synchronize and Trace Requirements
Requirements Management and Traceability

Caliber Author

- Reuse requirements across projects
- Integrate with code development and testing tools for verification and validation
- Understand the impacts of change and alignment to business objectives through traced relationships
Scenarios and Simulations
Caliber Visualize

• Visual clarity reduces process ambiguity
• Identifies missing or wrong underlying assumptions
• Stakeholder feedback improves quality of requirements
• Establishes a common vision and shared responsibility
Collaborate, Review and Discuss
Caliber Review

- View requirements, attributes, traces via web

- License free model assures maximum participation of stakeholders
  - Online discussions tied to requirements
  - Same discussions available from Author, Visualize and Review

- Additional functionality
  - Baselines
  - Filters
  - Search
Test Integration and Management
Silk Central Test Management

- Silk Central provides a unified framework for the integration of requirements, test automation tools
  - Unit
  - Functional
  - Performance
- Dashboard Reporting
- Manual execution planning
- Highlight risk mitigation and quality goals
- Configuration testing
- Video and image capture
- SAP Testing
Requirements Delivered
StarTeam Agile

• Organize, prioritize and manage Agile teams’ product backlogs

• Plan sprints, task out the work and then track progress through the sprint

• Get comprehensive visibility of all Agile assets

• Integrate and trace to formal requirements
Today’s Presenters

Pete DuPre’
Chief Architect
peter.dupre@microfocus.com
(301) 838-5106

Mark Kulak
Director of Development
mark.kulak@microfocus.com
(248) 824-1743

Bryan Fangman
Senior Product Manager
bryan.fangman@microfocus.com
(301) 838-5170
Company Information

Borland Website
http://borland.com

For additional information regarding:
• Presentation content
• Scheduling a demonstration
• Integrating requirements and agile delivery
• Products and services
• Speaking engagements

Please contact:

Anthony Gow
Inside Sales Representative
anthony.gow@microfocus.com
(248) 824-1758