Agile Adoption:
Does it have to be All In or Fold?

Rebecca Wirfs-Brock
©2011 Wirfs-Brock Associates
Who I Am
We don’t want a revolution.

We can’t have a manifesto!
“There Has to Be A Better Way”

Jon Spence, Sr. Principal Engineer and Technical Fellow, Medtronic, Inc.

Agile 2005 Experience Report

- Their software development struggle
  - Plan-driven, waterfall-based, document-centric
  - Company believed detailed requirements can be completed before implementing and can be used to accurately estimate resources and outcomes
  - But ……SW doesn’t have a neat, clean “manufacturing-like” flow
  - Hard to comprehensively and accurately plan and document SW years ahead
Enter Agile

- Invited external outside speakers
- Went to conferences, read the “good books”
- Learned about extreme programming, scrum, and lean software development
- Recognized potential to leverage several practices
Who’s Responsible for Change?

Jon pledged to colleagues that he was “putting all my chips in the game” to influence others to try agile on a mainline product development effort.

He had allies: Department Director, Mainline Product Development Manager, the Process Architect, key Developers.
Their Initial Proposal

Drawn from extreme programming and scrum

- Process ownership: team owns, management defers
- Process discipline: rigid adherence to team’s process
- Small teams: 4-6 person feature area subteams
- Customer: Project manager + sys engineers
- Daily meetings: short, barrier ID, refocus, synch
- 2 week increments
- Reflection: At end of each increment

- Simple design: Don’t speculate
- Test-first for new or changed non-trivial, non GUI code
- Continuous Integration: daily build, automated regression tests
- Collective code ownership: eliminate ownership bottlenecks
- Refactoring: To maintain simple design, if supported by unit tests
- Pair programming on new, non-trivial code
- Co-located teams: dev lab per feature area
Their Approach:
Empirical Adoption

- Try → Inspect → Adapt → Try

- Changes:
  - Team size: From 12 to 4-6
  - Customer: Originally had project mgr. Now systems engineers interact with project mgr.
  - Planning: Originally all teams planned independently. Now: teams meet for goal setting, then plan individually, then reconvene as whole group to share goals/discuss dependencies
  - Product backlog: Entire backlog visible to dev teams to avoid local sub-optimizations
are we in or out?
“A Tale of Two Writing Teams”
Stacia Broderick and Melody Locke

- Company X
  - Project manager for 85 person software dev team during transition to Scrum
  - Moved into consulting
  - Consulted at BMC

- BMC Software
  - Lead writer producing online and hardcopy docs for 5 times
  - Worked with several dev teams who produced integrated product
Company X’s Story
Beginnings

- 85 person project. Cross-functional teams delivering functionality to the same code base. 5 writers.

- Writers’ lives pre-agile:
  - “marred with negativity and path-of-least-resistance existence”
  - Documentation an afterthought
  - Releases usually delayed to document late features
Company X’s Move to Agile

- 10 Scrum team. Each writer assigned to two different Scrum teams
- 2 daily stand ups, new online help system, lots of new technology and tools
- First 3 sprints teams demoed online help with product features
- After three months, several teams co-located to practice XP
  - Everyone enthusiastic except writers
  - Writers tasks switched to updating legacy code docs…and the writing team reverted to former practices
- Writer’s eventually reformed their own group
  - Own task backlog
  - One iteration behind code development
- “Functional team cohesion over whole team discomfort”
BMC’s Story…Beginnings

- Development VP announced Scrum for a flagship product
- Five teams: 14 QA, 35 developers, 2 lead writers
- Layoffs and reorganizations occurred early in project
- Writers a scarce resource
  - Lead writer attending 5 daily Scrum meetings
  - Time fragmented
  - Early meetings focused on dev tasks not QA or docs
- Retrospectives led to change
BMC Actions

- Documentation as a feature.
- “Overhead” added to other tasks
  - Story cards and tasks for hardcopy. Help-related tasks added to development story cards.
- Editorial process made more agile.
- Pre-planning gave writers a chance to see the territory ahead.
- Results: QA and writers still not working at a sustainable pace. Hardcopy docs have a slight lag. Online docs track dev schedule.
It’s Not Easy…

What’s tracked on the backlog?

What does “done” mean?

Should writers use product “hardening” time to finish?

Limited interactions means non-code deliverables can lag.

Work outside or alongside the “whole team”? 
are there different rhythms to our work?
Agile in the large
“Enterprise Agile Transformation: The Two-Year Wall”
Chuck Maples, Borland

- Plan to convert most of development organization to agile over three years, one site at a time.

- Ahas!
  - Once dev teams became productive, feature requests flooded in from the field.
  - Need for sophisticated “enterprise testing” strategy and tactics.
    - Scripts generated by sprint teams, handed off to central testing.
  - Need to commit to certain features at a certain date, with more always possible.

- “Reality on the ground”
“Ongoing Quality Improvement or: How We All Learned To Trust XP”
Mark Striebeck, VA Software

- SourceForge Enterprise Edition re-write
- Developed automated acceptance/system tests
- Constantly test application on multiple platforms
- Complete system test takes 7 hours—so can’t be run immediately against the build
- Define significant subset that “almost” guarantees good build
- Two dev test rhythms…build tests and thorough system tests
Shifting Rhythms After Demonstrated Results

- Initially QA only tested after code freeze.
- Both manually executed and automated test scripts.
- After quality was demonstrated in first release… QA got on board
- Now part of initial planning
- QA starts when development does
- Teams still separate, but communicate
“Case Study of Customer Input for a Successful Product”

Lynn Miller, Director of UI Development, Alias

- The products: 3D software for design, game creation, and graphical special effects
- The structure: Usability engineering is part of the product development group
- The process: Each interaction designer is assigned to a single product at a time
- The scope of work: Each interaction designer is responsible for market research, gathering user requirements, interface design, and usability testing
Dual Tracks

Planning
Data Gathering

Independent dev of low UI features

Implement Designs

Implement Designs

Cycle 0

Design for cycle 2
Gather customer input for cycle 3

code
design
customer data
customer data
code
code

Test cycle 1
code
Design for cycle 3
Gather input for cycle 4

Cycle 1

Cycle 2

Cycle 3

Gather input for cycle 5

Developers

Interaction Designers
does incremental adoption work?
“Introducing Agile Development into Bioinformatics”
David Kane, SRA International

- Visualization tool for National Cancer Institute scientists
- Dynamic requirements
- Exploring scientific research
- New tool requirements constantly emerging
- Results and tools published in scientific publications
Agile Practices and Dependencies

On-site customer → Configuration Mgmt

Collective Ownership → Planning Game

Continuous Integration → Team

Automated Builds → Time start

Open workspace → Automated Tests

Coding Standards → Refactoring

Pair Programming → Code Reviews

Incrementally added over 16 months
“Subclassing XP: Breaking its rules the right way”
Greg Luck

- Removed authority for some refactoring from individual developers or pairs to the team.

- Two refactorings types:
  - Refactoring to fit in code for a new story. Minor and narrow in scope.
  - Refactoring that effects a design change, or affects threading code. Usually major impact and wide in scope.

- Refactoring replaced with Team Refactoring
Team Refactorings

- Refactoring list maintained on JIRA.

- The team evaluates suggested refactorings and ranks them in priority order. Design and risks considered.

- Refactoring task done at the end of the iteration, when a developer has completed all user stories, provided someone else does not need a hand.
“The Bold New Extreme Programming Experiment – Now in Its Ninth Year”
Brian Spears, Follett Software

- Non-standard XP
  - No story cards. Now stories that contain use cases with sample screen shots.
  - Far less than a detailed requirements document
  - Business Analysts “own” consistency of look and feel
  - Sometimes effort is wasted
- Running tests against the databases
- Not everyone is a generalist. But they try not to depend on a single expert.
- Added Story Champions- Dev and tester to fit story into current design and break into tasks.
Agile 2011 Newsflash
Trend: Agile + Lean/Kanban

“Agile & Kanban in Coordination”—Ryan Polk, WMS Gaming Inc.

“Breaking the Rules to Increase Customer Satisfaction”—Mike Prior, SumTotal Systems, Inc.
How can you grow a thriving agile practice?
Keep it Fresh

“If you are doing XP and your process is exactly the same as it was a year ago, you probably aren’t as agile as you were a year ago.”—Brian Foster
Connect to Corporate Values

“Our entire corporation is strongly connected to its mission statement and the principles that derive from it, so we emphasize Agile principles that supported that mission.” —Kelly Weyrauch, Medtronics

“To strive with reserve for the greatest possible reliability and quality in our products” → Inspect and adapt, test-driven design, customer focus

“To recognize the personal worth of employees…” → Process transparency, collaboration, reflections, co-location
Reinforce What You Value

- **Team review** of major code changes
- **Pair of eyes** on all code checked in at “final fix”
- **End-to-end testing** of database transactions
- **Realistic tests with different test rhythms** (build and overnight)
- **Adding Kanban** to improve flow
- ...