

***The Agile Difference for
SCM***

Robert Cowham &
Brad Appleton

Contents

- | What is Agility?
- | Standard SCM
- | Introducing Agile and the SCM Implications
- | Areas for Concern
- | Case Study
- | Summary

12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 2

What is Agility?

- | The ability to both create and respond to change in order to profit in a turbulent business environment.
- | What is new about agile methods is not the practices they use, but their recognition of *people* as the primary drivers of project success, coupled with an intense focus on effectiveness and maneuverability.

-- James Highsmith, *Agile Software Development Ecosystems*

12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 3

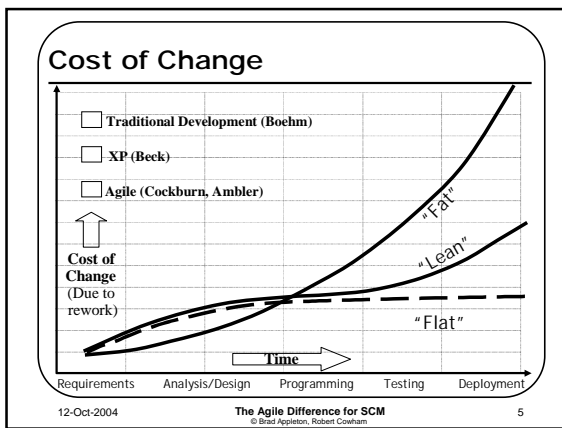
The Agile Difference for SCM

Implications of Change

“Requirements changes late in the lifecycle are competitive advantage, **IF** you can act on them!”
– Mary Poppendieck

*“Change is not the enemy – stagnation is!
Don’t try to prevent change! Plan for it!”*

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 4



Achieving Agility

Responding quickly & effectively to change requires minimizing:

- ▮ The cost of knowledge-transfer
- ▮ The cost of knowledge capture (documents!)
- ▮ The time between making a decision, and exploring its results to learn the consequences of implementing it

▮ Close collaboration and frequent iteration are critical for success!

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 6

The Agile Difference for SCM

Four Principles of Lean Thinking

1. Add nothing but value
– *(Eliminate Waste)*
2. Center on the people who add value
3. Let Customers “Pull” Value
– *(Delay Commitment, Deliver Fast)*
4. Optimize the Value Stream
– *(Optimize Across Organizations)*

Source: Mary & Tom Poppendieck, <http://www.poppendieck.com>

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 7

Seven “Wastes” of Software Development


1. Extra/Unused features (*Overproduction*)
2. Partially developed work not released to production (*Inventory*)
3. Intermediate/unused artifacts (*Extra Processing*)
4. Seeking Information (*Motion*)
5. Escaped defects not caught by tests/reviews (*Defects*)
6. Waiting (including Customer Waiting)
7. Handoffs (*Transportation*)

Source: Mary & Tom Poppendieck, <http://www.poppendieck.com>

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 8

What is SCM? (Traditional View)

- | Configuration Identification
- | Configuration Control
- | Status Accounting
- | Audit & Review
- | Build & Release Management
- | Process Management, etc



12-Oct-2004 © 2004 Brad Appleton & Robert Cowham The Agile Difference for SCM 9

Standard Definition

- | "Configuration Management is a discipline applying technical and administrative direction and surveillance to **identify and document** the functional and physical characteristics of a **configuration item**, **control changes** to those characteristics, **record and report** change processing and implementation **status**, and **verify compliance** with specified requirements."
- | IEEE-Std-610 (revision and redesignation of IEEE-Std-729-1983)

12-Oct-2004 © 2004 Brad Appleton & Robert Cowham The Agile Difference for SCM 10

What is Agile SCM?

The *pragmatic* application of

- | sound *CM principles & practices*
- | in accordance with *Agile Values*
- | using *Lean Thinking*

to serve the needs of the *Business!*

12-Oct-2004 The Agile Difference for SCM © Brad Appleton, Robert Cowham 11

Introducing Agility

(into development)

The Agile Difference for SCM

Adding agility without extremes

- | Automate and share the build process
- | Test framework/unit tests
- | Adopt a continuous integration process
- | Plan and deliver in short iterations/small releases
- | Identify and collaborate with your customer
- | Manage your test data; don't let it manage you
- | Embrace collective ownership and share code

Source: Peter Schuh, Better Software Jul/Aug 04

12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 13

Automate & Share Build Process

- | Automate repetitive and mundane tasks
- | Saves chasing compilation/convergence issues
- | Reduces dependencies and bottlenecks

12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 14

SCM is a "whole team" responsibility

- | SCM and development must closely collaborate toward the shared goal of successfully meeting a project's business and technical objectives
 - Remove "Great Walls of Ire" between SCM & developers!
- | SCM is part of every team member's day-to-day tasks and activities:
 - Integrate/build/test in sandbox before check-in
 - If the build breaks, the whole team takes ownership
- | Everyone understands and appreciates the needs of both development and SCM because they experience the needs and benefits of both every day

12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 15

Concerns

Agile vs. Traditional SCM

Coordination and Automation

SCM tools & practices/processes cannot hinder development or they won't get used!

- Add nothing but value
- Minimize Artifacts
- Eliminate Waste
- Center on the people who add value

I Tools and processes need to be simple, pragmatic, and enhance communication and coordination or reduce rework.

I Tracking systems and version control tools should not interrupt "flow" by causing a "wait" for tools to collect data or complete lengthy operations.

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 17

Responding to change vs. controlling it

I Manage stakeholder expectations with close communication and simple boundaries (short, frequent iterations)

- Decision makers must be closely available & accessible to give rapid response to issues & questions
- At the start of each iteration, expectations and priorities are (re)set and (re)calibrated
- CCB's might be called "change planning meetings" to avoid the stigma of trying to control change
- Work only on the features for the current iteration

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 18

Change Control (Management?!)

- | Story card pluses
 - Planning game
 - Communication dynamics
 - Visibility within the team
- | Story card minuses
 - Lack of versioning
 - Getting lost?!
 - Lack of visibility across the organisation
 - Requirements Decomposition

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 19

Traceability

- | Impact Analysis
 - To communicate to other stakeholders
 - To assist with estimation & risk assessment
- | Product Conformance to requirements
 - Eliminate 'back doors' etc.
 - Increasing response to Sarbanes-Oxley in U.S.
- | Process Compliance
 - Standards body says you must do it
 - Contracting party mandates it

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 20

The LoRD Principle

- | Principle of Locality of Reference Documentation (LoRD)
 - The likelihood of keeping all or part of a software artifact consistent with any corresponding text that describes it is inversely proportional to the square of the "cognitive distance" between them.
- | Out of sight, out of mind!

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 21

Applications of LoRD

- | Documentation in the Code
 - (e.g. JavaDoc, Perl5 PODs)
- | Documentation in the Model
 - (e.g., UML with formal+informal text annotations)
- | Documentation in a Database w/ links+metadata
 - (e.g. Telelogic DOORS, IBM/Rational RequisitePro)
- | User Reference Manual as Requirements Spec
- | Acceptance Tests are the Requirements
- | Interface/Implementation Co-location
- | README per Directory


12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 22

“Lean” Documentation and Traceability

- | Minimize Traceability and Eliminate Redundancy by
 - using Fewer Artifacts
 - Separation of Concerns
- | Track features/use-cases and classes/modules
 - instead of their individual requirements/routines
- | Fewer items, means fewer items to track/trace
 - Detailed requirements/use-cases may serve double-duty as acceptance test-cases
 - Hi-level requirements/features may be simple feature/change requests and/or release notes (automatically generated)
 - Some end-user documents may even be used as use-cases or functional requirements documentation

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 23

Case Study



WDS GLOBAL
it simply works...

The Agile Difference for SCM

WDS Global

- | Integrated Service Framework to the wireless industry:
 - Nokia, Vodafone, AT&T Wireless, Singtel, Telstra, Siemens
- | Web based services, mainly java
- | 25 developers: Seattle, Poole & Singapore
- | Transitioned to XP 2½ years ago
- | Daily handovers via conference call
- | Story card planning
- | Migrated from CVS

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 25

Mainline model

The diagram shows a horizontal arrow representing a branch named 'main'. There are four vertical tick marks along the arrow. Three of these tick marks have arrows pointing upwards to the label 'Release points'.

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 26

Release via live branch

The diagram shows a horizontal arrow representing a 'main' branch. A curved arrow branches off from 'main' and is labeled 'live'. There are four vertical tick marks along the 'live' branch, each with an arrow pointing upwards to the label 'Release points'. Dotted arrows point from the 'live' branch back to the 'main' branch.

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 27

WDS Global Experiences

- | Working well
 - Visibility dramatically better
- | Multi-site fine
- | Still use only story cards – from Product Backlog spreadsheets

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 28

Summary

Agile SCM is *not*...

- | A hackers charter!
- | About throwing out good SCM principles and practices
- | Losing control

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 30

Agile SCM requirements

An Agile SCM solution should:

- | Serve its practitioners and not vice-versa!
 - while still serving the needs of the business
- | Unite SCM and developers
 - collaboration
- | Respond to change vs. prevent changes
- | Track, guide/facilitate & coordinate vs. control
- | Be transparent and "frictionless"
 - automation & process-enactment

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 31

Becoming More Agile

- | Introduce One Practice
- | Adapt
- | Evaluate
- | Repeat!


And hopefully reap the rewards!

12-Oct-2004 The Agile Difference for SCM
© Brad Appleton, Robert Cowham 32

Resources

The Agile Difference for SCM

Other Agile/SCM References



- | **Software Configuration Management Patterns**, Stephen Berczuk and Brad Appleton
- | **Configuration Management Principles and Practice**, Anne Mette Jonassen Hass
- | **Software Configuration Management Strategies and Rational ClearCase**, Brian White
- | **Requirements by Collaboration**, Ellen Gottesdiener
- | **Lean Software Development**, Mary & Tom Poppendieck
- | **The Laws of Software Process**, Phillip G. Armour
- | **Agile Software Development Ecosystems**, James Highsmith
- | **Agile Software Development**, Alistair Cockburn
- | **Domain Driven Design**, Eric Evans
- | **Refactoring: Improving the Design of Existing Code**, Martin Fowler
- | *Addison-Wesley's XP Series of Books*

12-Oct-2004 **The Agile Difference for SCM** 34
© Brad Appleton, Robert Cowham
