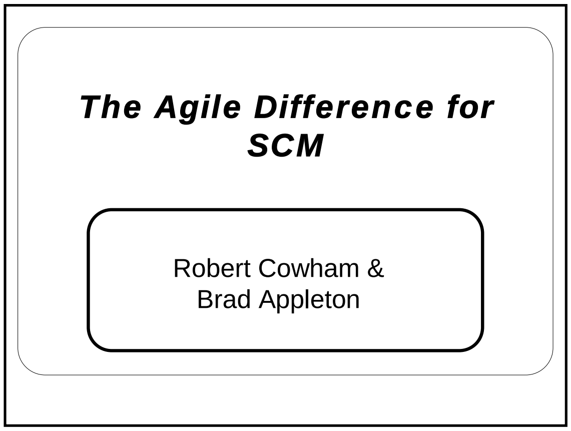
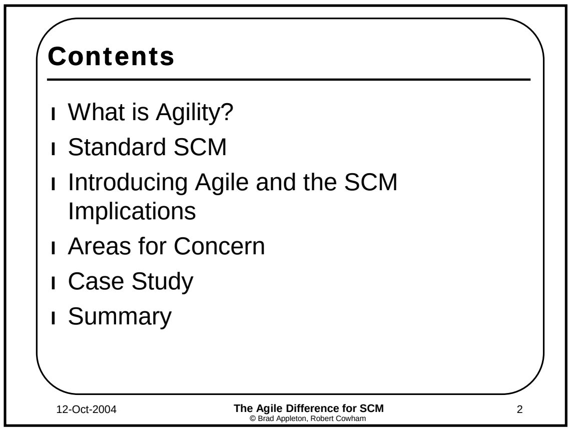


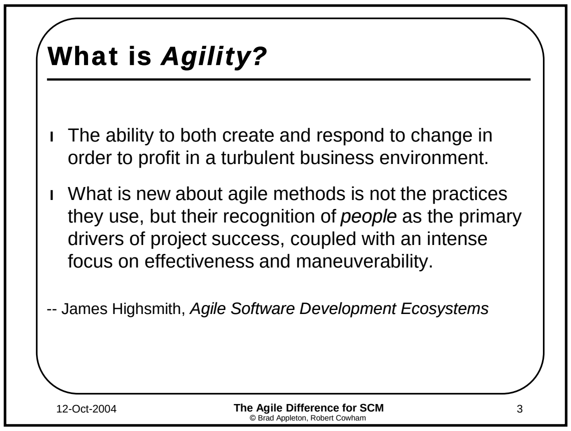
The Agile Difference for SCM

Robert Cowham &
Brad Appleton



Contents

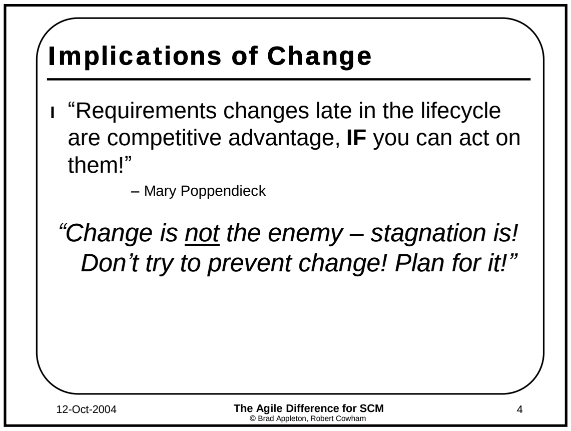
- What is Agility?
- Standard SCM
- Introducing Agile and the SCM Implications
- Areas for Concern
- Case Study
- Summary



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*



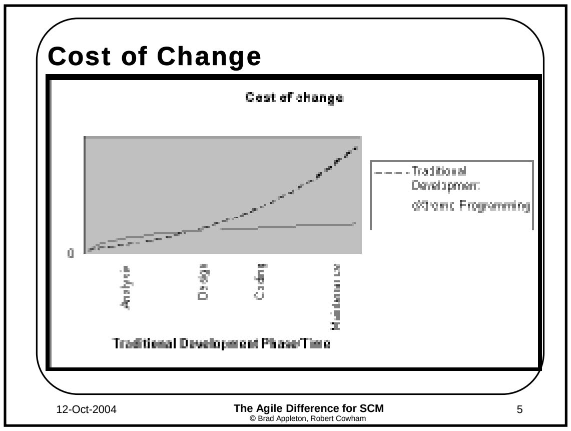
Implications of Change

- "Requirements changes late in the lifecycle are competitive advantage, **IF** you can act on them!"

  – Mary Poppendieck

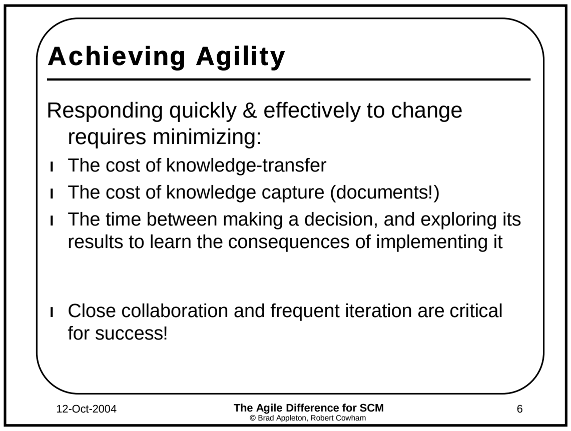
"Change is <u>not</u> the enemy – stagnation is! Don't try to prevent change! Plan for it!"

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



Cost of Change

12-Oct-2004 **The Agile Difference for SCM** 5
© Brad Appleton, Robert Cowham



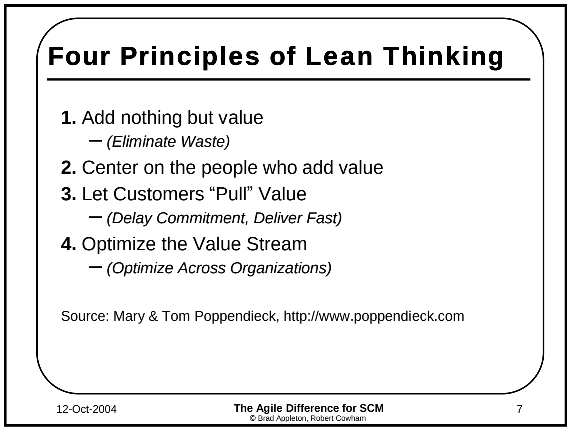
Achieving Agility

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** 6
© Brad Appleton, Robert Cowham

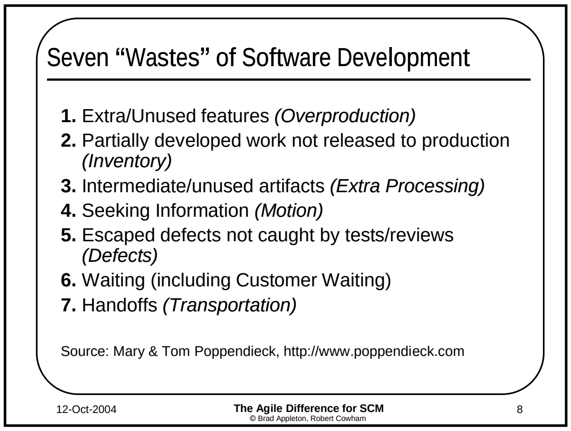


Four Principles of Lean Thinking

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

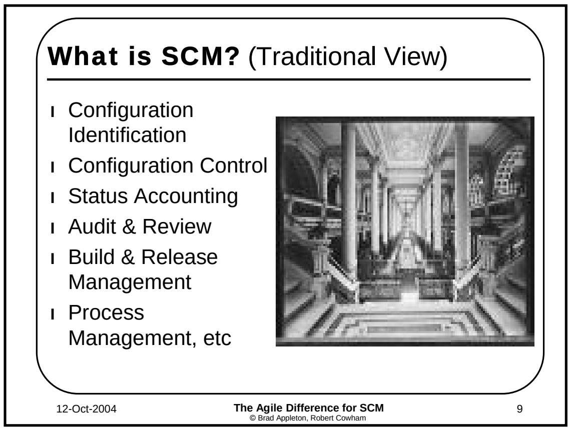


Seven “Wastes” of Software Development

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

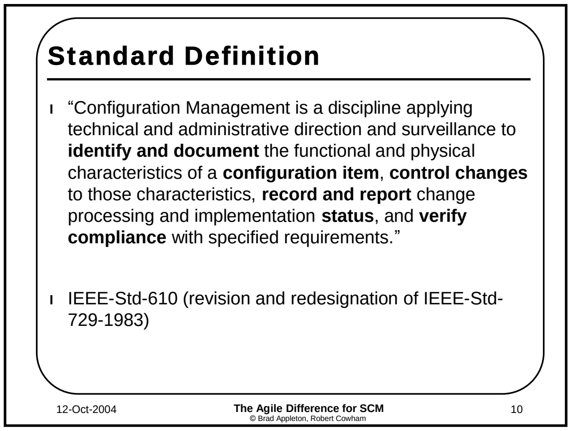
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 8



What is SCM? (Traditional View)

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

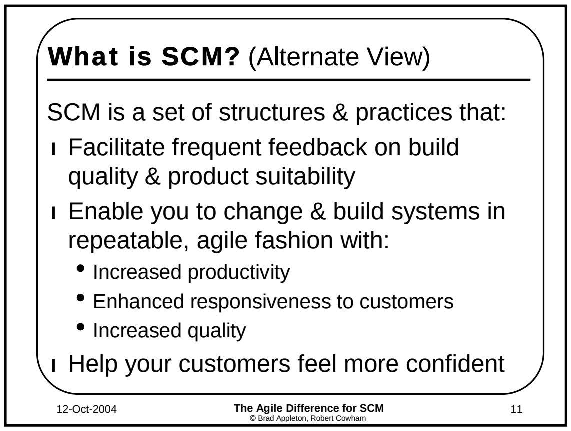
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 9



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 The Agile Difference for SCM © Brad Appleton, Robert Cowham 10

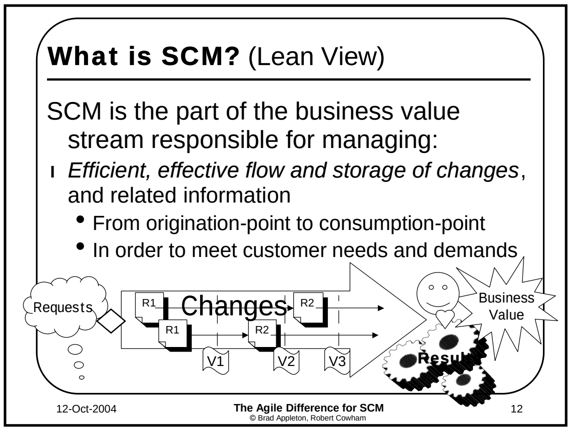


What is SCM? (Alternate View)

SCM is a set of structures & practices that:

- Facilitate frequent feedback on build quality & product suitability
- Enable you to change & build systems in repeatable, agile fashion with:
  - Increased productivity
  - Enhanced responsiveness to customers
  - Increased quality
- Help your customers feel more confident

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



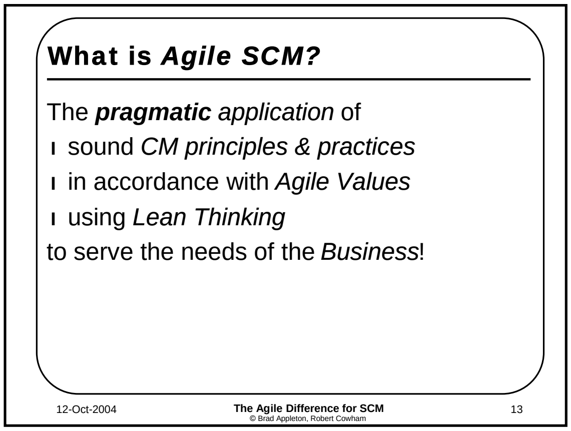
What is SCM? (Lean View)

SCM is the part of the business value stream responsible for managing:

- *Efficient, effective flow and storage of changes*, and related information
  - From origination-point to consumption-point
  - In order to meet customer needs and demands

Requests
R1
R1
Changes
R2
R2
V1
V2
V3
Business Value
Results

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



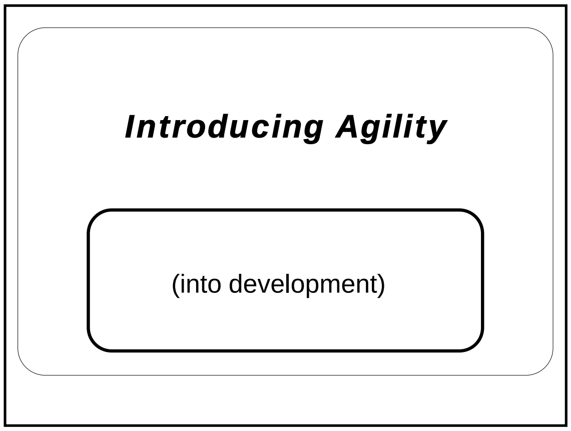
What is *Agile SCM?*

The ***pragmatic*** *application* of

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

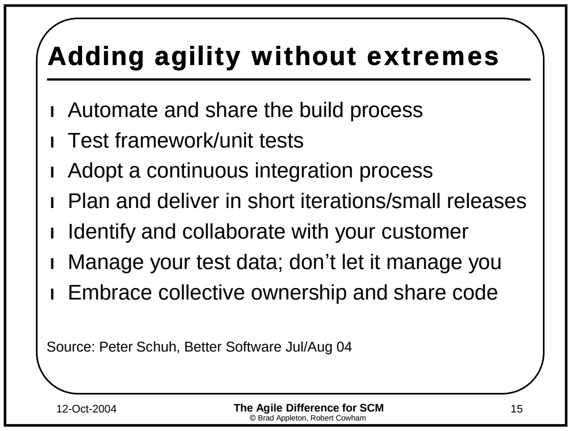
to serve the needs of the *Business*!

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



Introducing Agility

(into development)

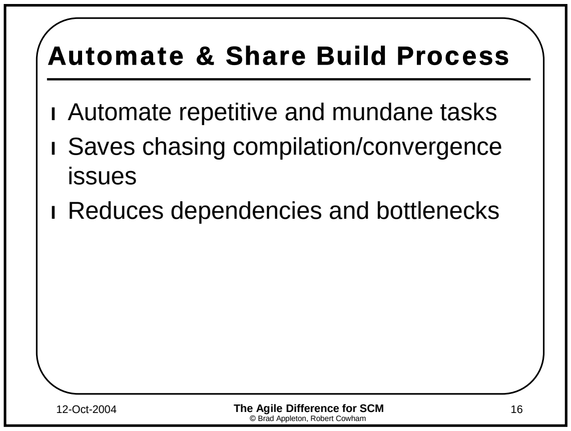


Adding agility without extremes

- 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

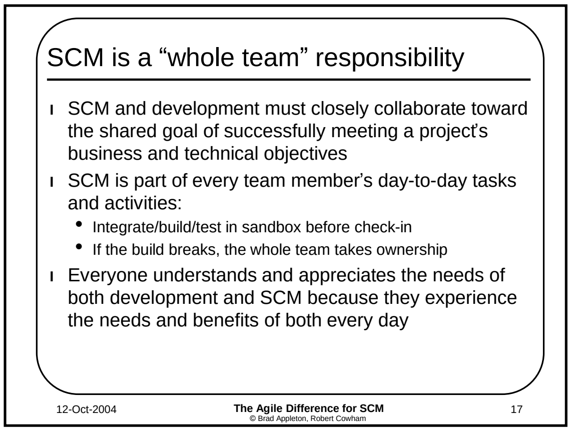
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 15



Automate & Share Build Process

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

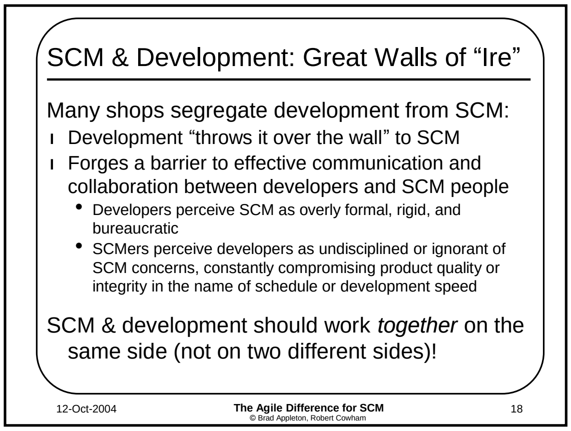
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 16



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
- 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 17



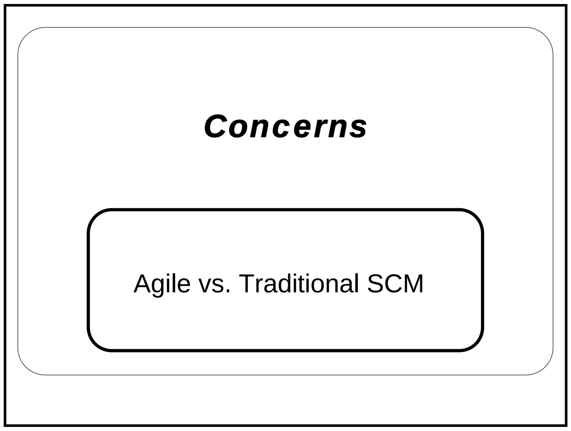
SCM & Development: Great Walls of “Ire”

Many shops segregate development from SCM:

- Development “throws it over the wall” to SCM
- Forges a barrier to effective communication and collaboration between developers and SCM people
  - Developers perceive SCM as overly formal, rigid, and bureaucratic
  - SCMers perceive developers as undisciplined or ignorant of SCM concerns, constantly compromising product quality or integrity in the name of schedule or development speed

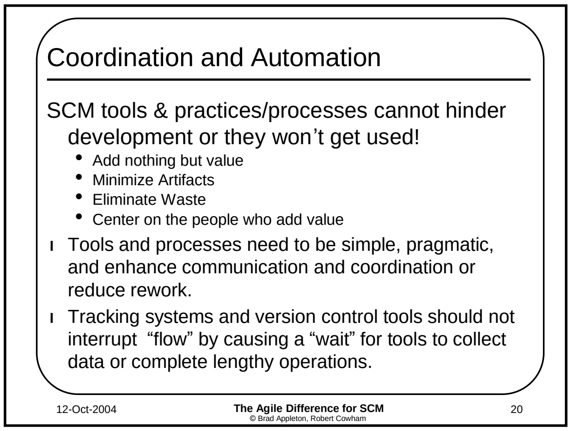
SCM & development should work *together* on the same side (not on two different sides)!

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



Concerns

Agile vs. Traditional SCM



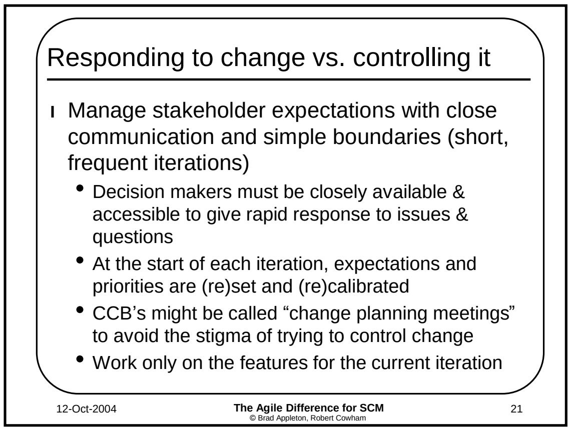
Coordination and Automation

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

- Tools and processes need to be simple, pragmatic, and enhance communication and coordination or reduce rework.
- Tracking systems and version control tools should not interrupt “flow” by causing a “wait” for tools to collect data or complete lengthy operations.

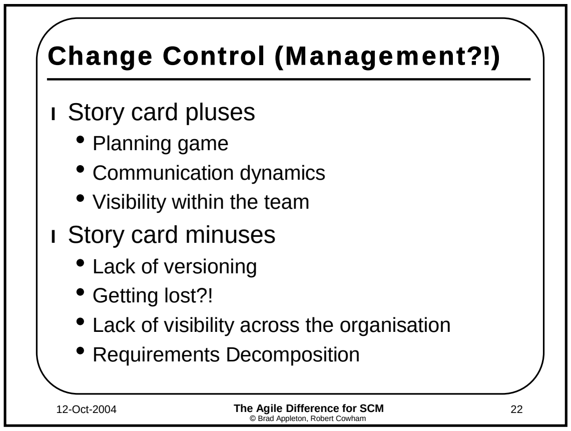
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 20



Responding to change vs. controlling it

- 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

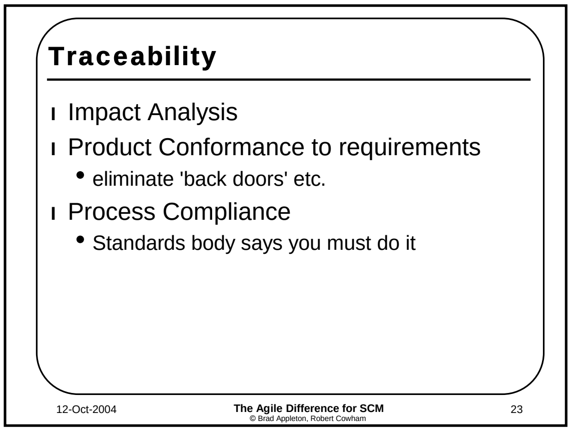
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 21



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

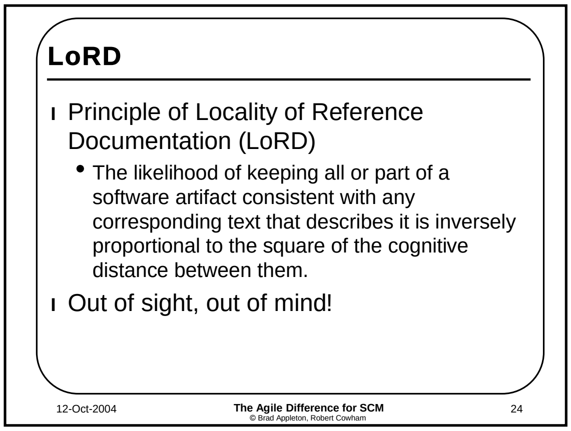
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 22



Traceability

- Impact Analysis
- Product Conformance to requirements
  - eliminate 'back doors' etc.
- Process Compliance
  - Standards body says you must do it

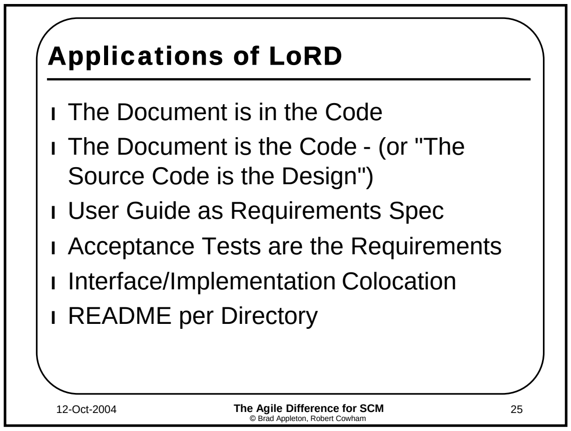
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 23



LoRD

- 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!

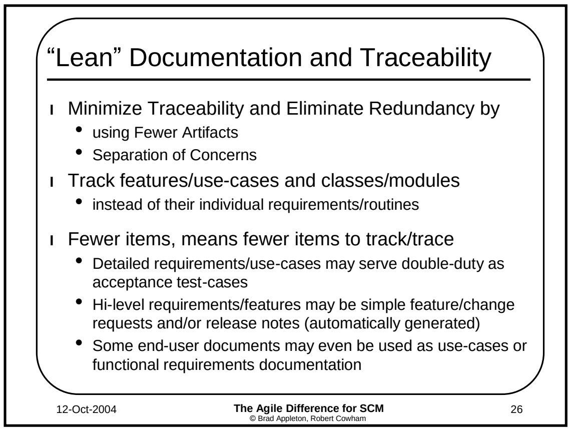
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 24



Applications of LoRD

- The Document is in the Code
- The Document is the Code - (or "The Source Code is the Design")
- User Guide as Requirements Spec
- Acceptance Tests are the Requirements
- Interface/Implementation Colocation
- README per Directory

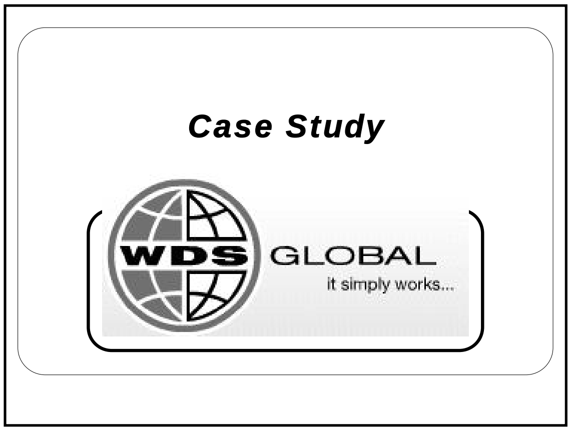
12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 25



“Lean” Documentation and Traceability

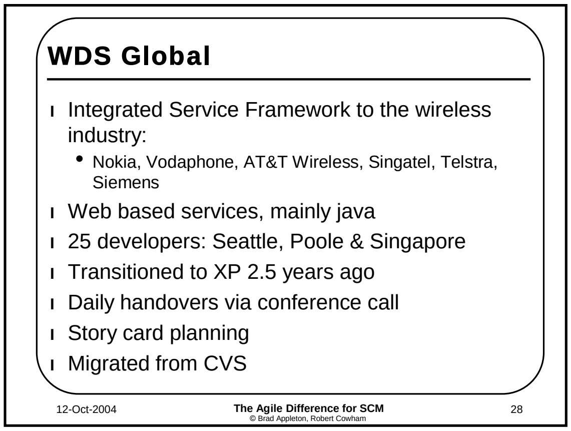
- Minimize Traceability and Eliminate Redundancy by
  - using Fewer Artifacts
  - Separation of Concerns
- Track features/use-cases and classes/modules
  - instead of their individual requirements/routines
- Fewer items, means fewer items to track/trace
  - Detailed requirements/use-cases may serve double-duty 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 26



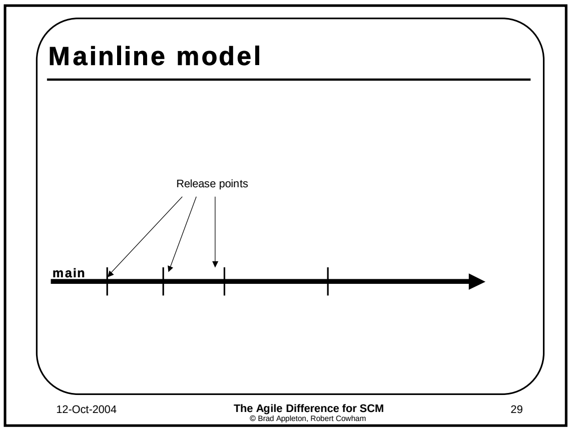
Case Study

WDS GLOBAL
it simply works...

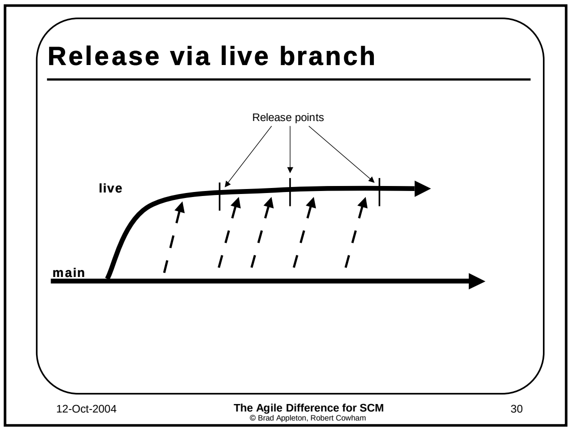


WDS Global

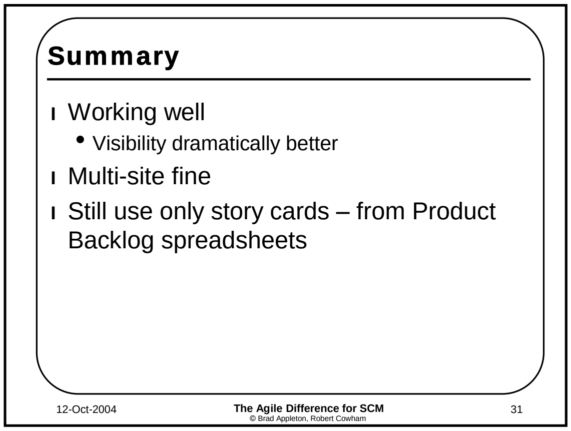
- Integrated Service Framework to the wireless industry:
  - Nokia, Vodaphone, AT&T Wireless, Singatel, Telstra, Siemens
- Web based services, mainly java
- 25 developers: Seattle, Poole & Singapore
- Transitioned to XP 2.5 years ago
- Daily handovers via conference call
- Story card planning
- Migrated from CVS



Mainline model



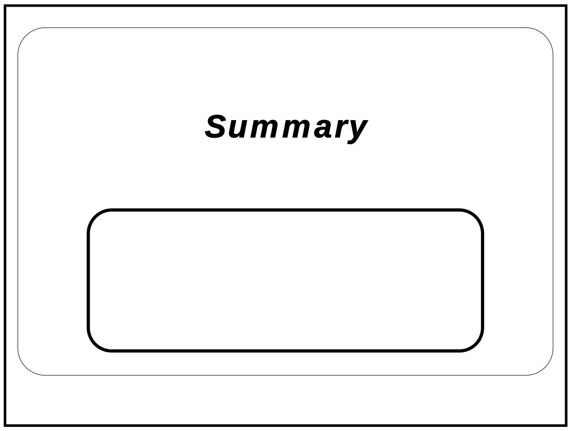
Release via live branch



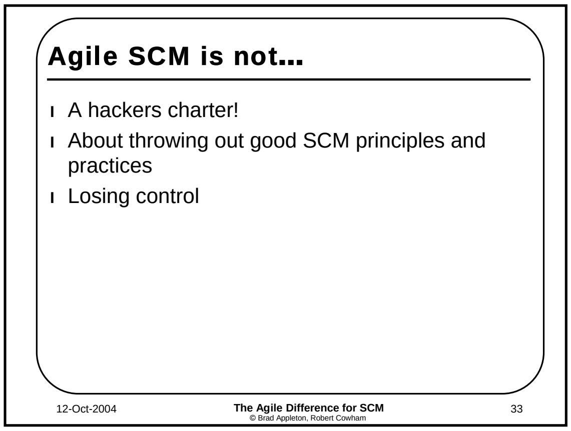
Summary

- 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 31



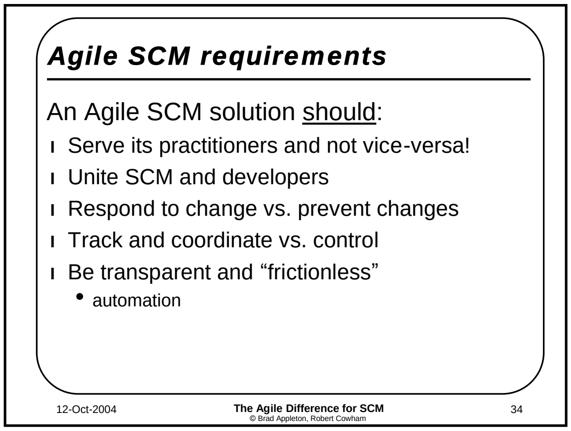
Summary



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 33

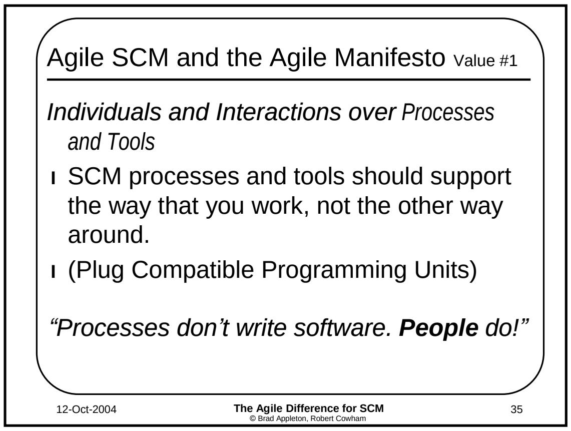


Agile SCM requirements

An Agile SCM solution <u>should</u>:

- Serve its practitioners and not vice-versa!
- Unite SCM and developers
- Respond to change vs. prevent changes
- Track and coordinate vs. control
- Be transparent and “frictionless”
  - automation

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



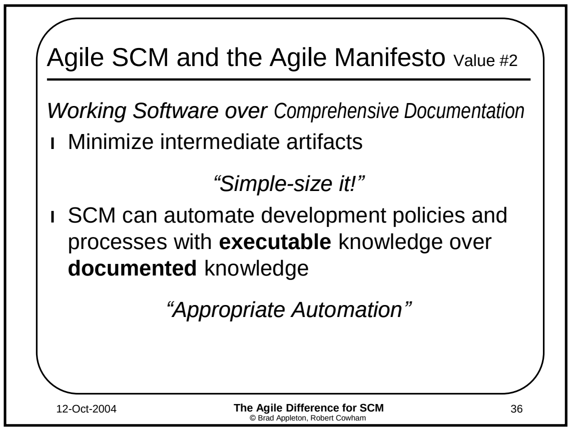
Agile SCM and the Agile Manifesto Value #1

Individuals and Interactions over Processes and Tools

- SCM processes and tools should support the way that you work, not the other way around.
- (Plug Compatible Programming Units)

*“Processes don’t write software. **People** do!”*

12-Oct-2004 **The Agile Difference for SCM** 35
© Brad Appleton, Robert Cowham



Agile SCM and the Agile Manifesto Value #2

Working Software over Comprehensive Documentation

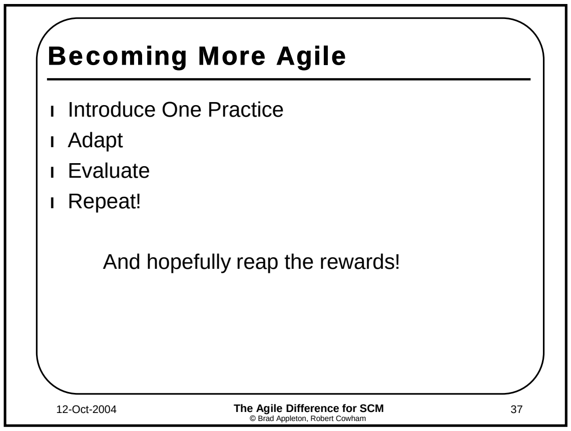
- Minimize intermediate artifacts

“Simple-size it!”

- SCM can automate development policies and processes with **executable** knowledge over **documented** knowledge

“Appropriate Automation”

12-Oct-2004 **The Agile Difference for SCM** 36
© Brad Appleton, Robert Cowham

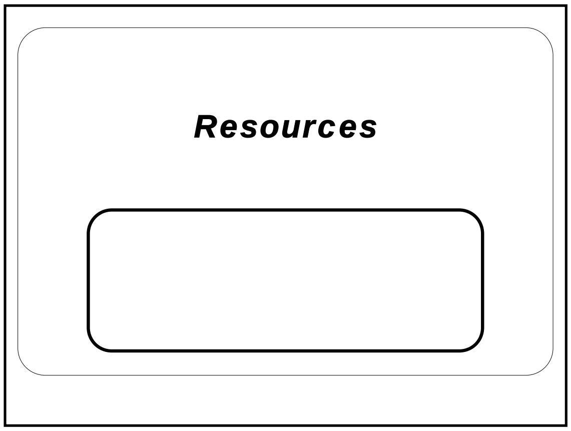


Becoming More Agile

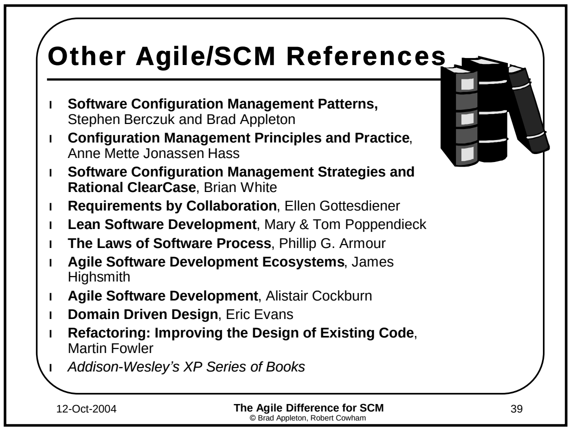
- Introduce One Practice
- Adapt
- Evaluate
- Repeat!

And hopefully reap the rewards!

12-Oct-2004 **The Agile Difference for SCM** © Brad Appleton, Robert Cowham 37



Resources



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** © Brad Appleton, Robert Cowham 39