

“Agile Methodology – Past and Future”

Warren W. Tignor
SAIC

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE MAY 2011		2. REPORT TYPE		3. DATES COVERED 00-00-2011 to 00-00-2011	
4. TITLE AND SUBTITLE Agile Methodology - Past and Future				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Science Applications International Corp,1710 SAIC Drive,McLean,VA,22102				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 32	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

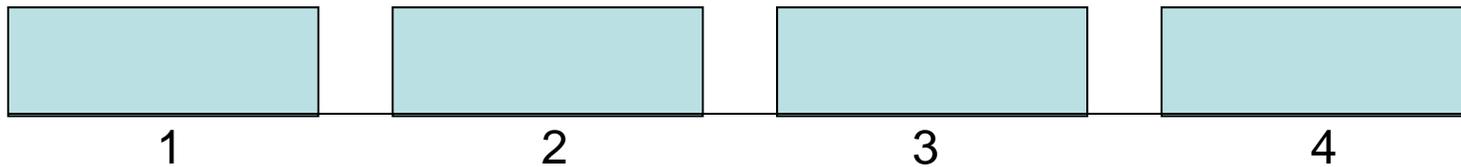


The New Development Game

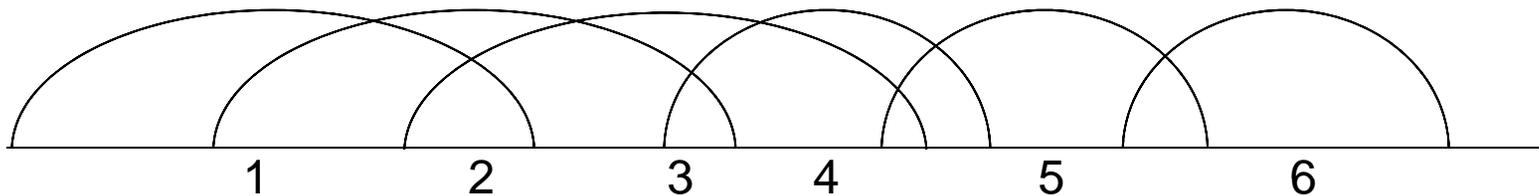
Hiroataka Takeuchi & Ikujiro Nonaka published

- "The New New Product Development Game" HBR Jan-Feb (1986)
- Holistic approach with six characteristics:
 - Built-in instability
 - Self-organizing project teams
 - Overlapping development phases
 - "Multilearning"
 - Subtle control &
 - Organizational transfer of learning

Examples of New Product Development Types *



Linear - Waterfall-like Product Phases



Overlapping - Agile-like Product Phases

* Adapted from Takeuchi & Nonaka HBR 1986, p139

RUGBY

Waterfall-Red vs. Agile-Black Team



Manifesto 2001

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Through this work we have come to value:

Individuals and interactions over processes and tools

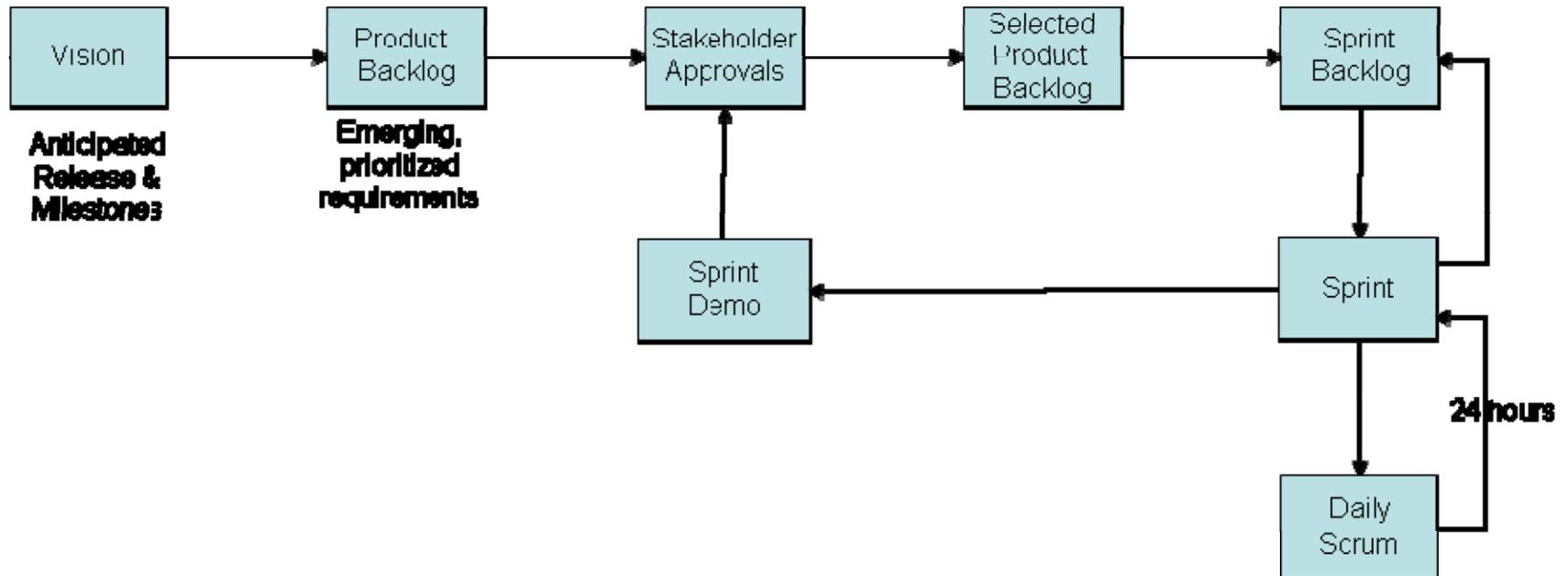
Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

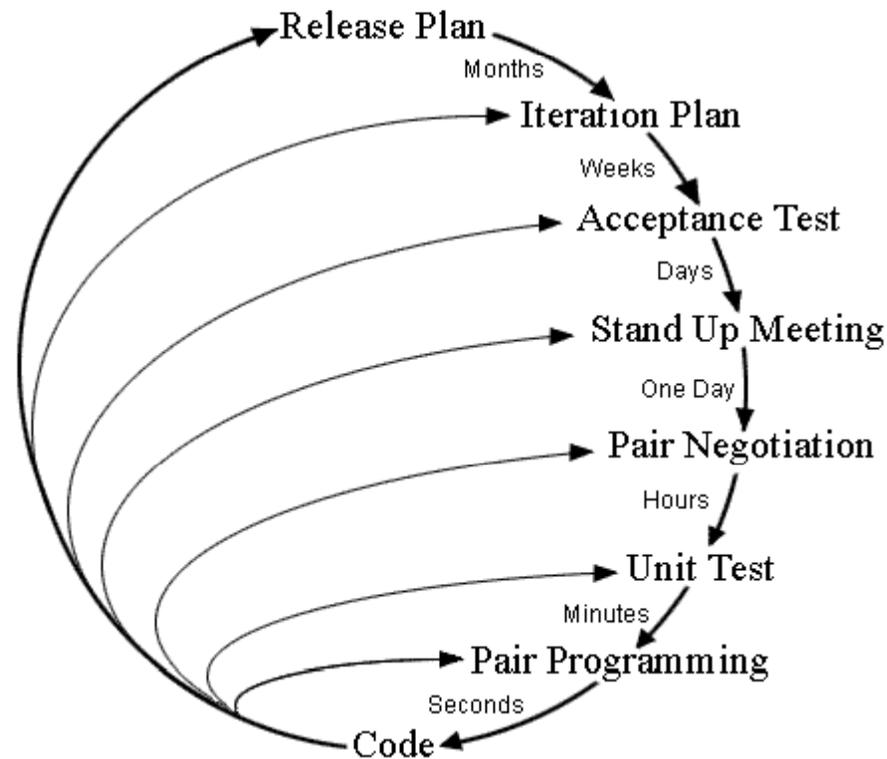
SCRUM GRAPHIC*



* Adapted from Schwaber (2007)

Agile Extreme Programming (XP)

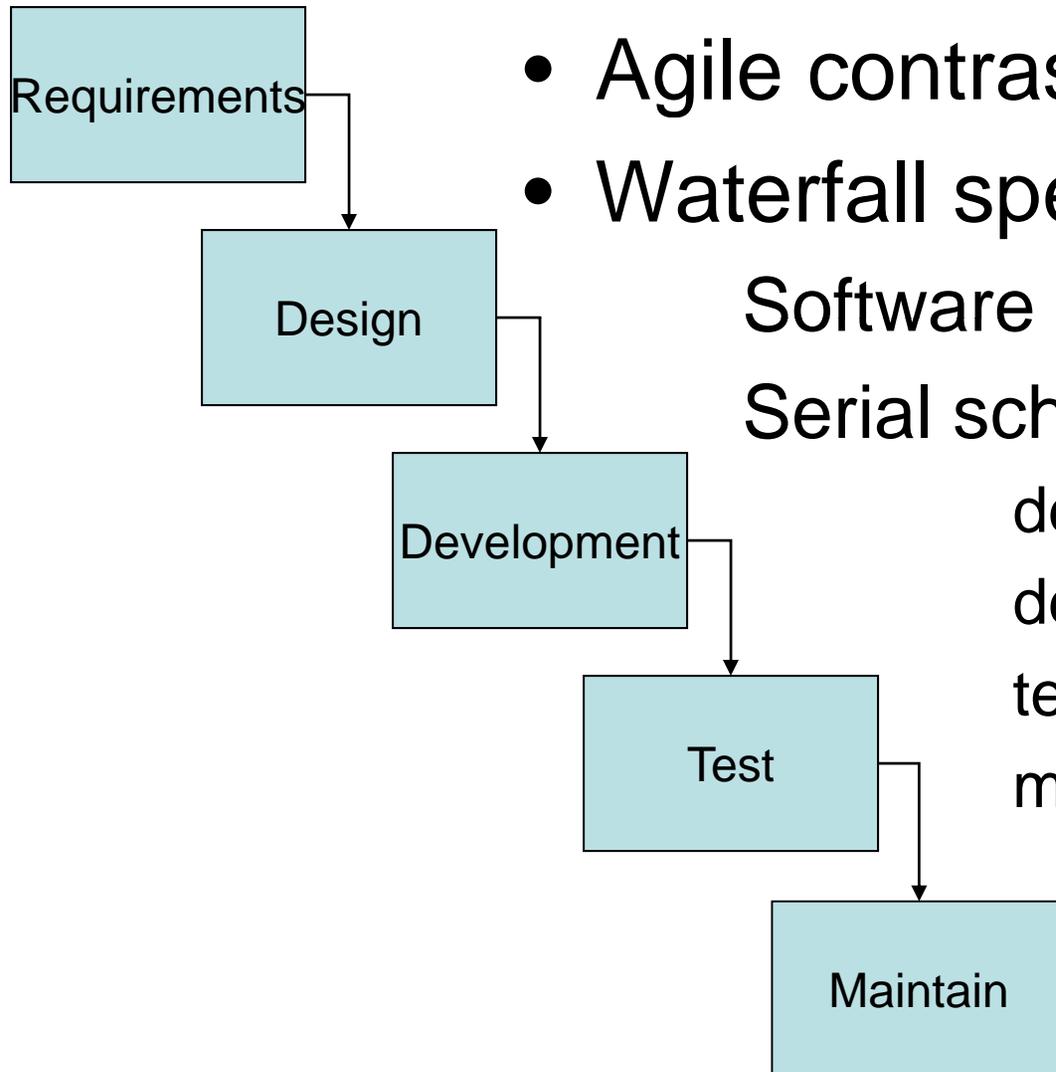
Planning/Feedback Loops



Attributed to Don Wells (<http://en.wikipedia.org/wiki/File:XP-feedback.gif>) without endorsement of me or my use of the work.

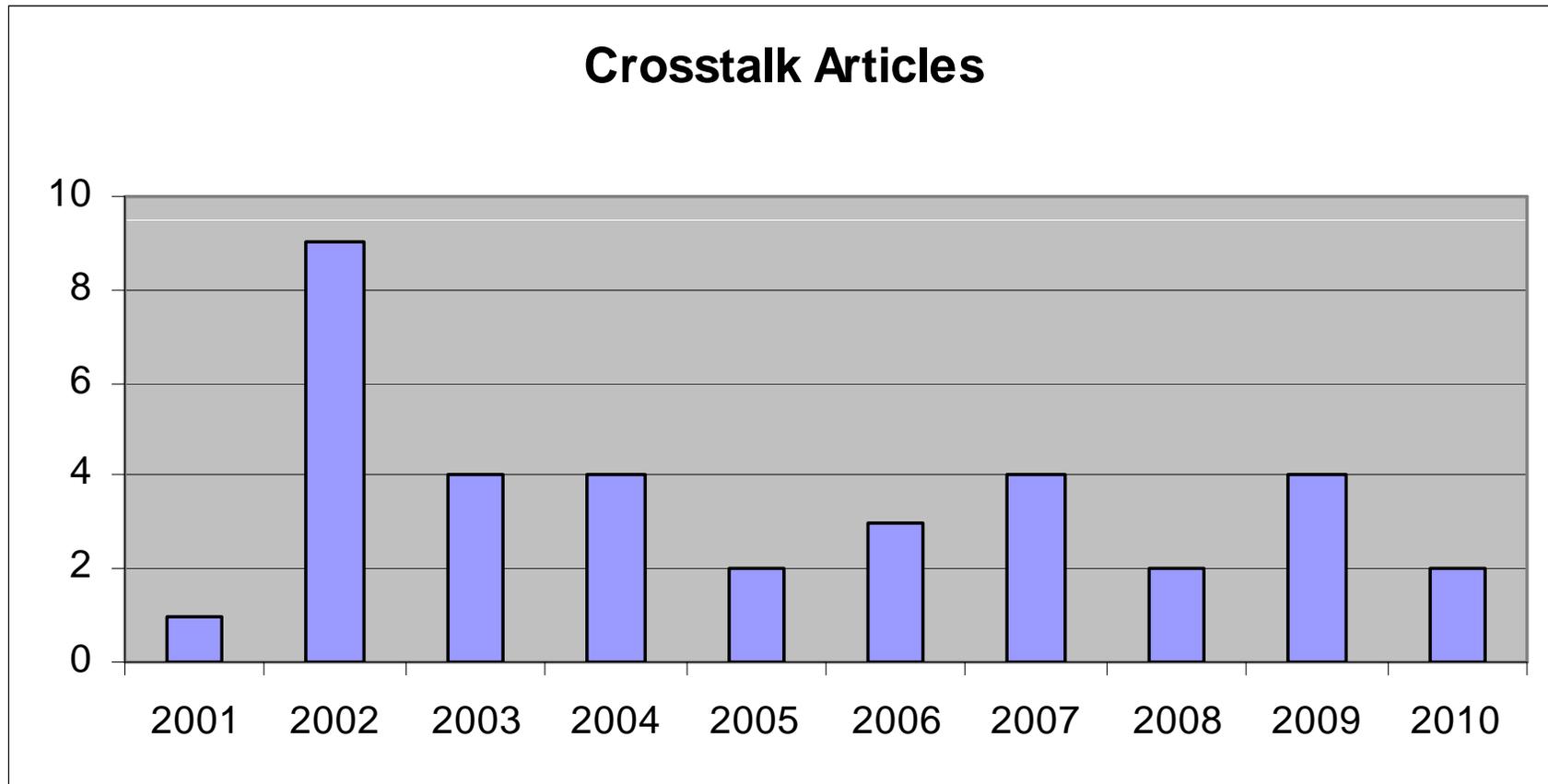
This file is licensed under the [Creative Commons Attribution 3.0 Unported](https://creativecommons.org/licenses/by/3.0/): **to share & to remix.**

Waterfall



- Agile contrasts with Waterfall
- Waterfall specifies up-front Software to be developed
Serial schedule of events, e.g.,
design,
develop,
test, &
maintain.

CROSSTALK Articles Reflect Agile's DoD Emergence





Agile & CMM® Process

- Glazer (2001/11) investigated the Agile (XP) and CMM® Myth/Reality/Bridge
- Kane & Ornburn (2002/10) declared Agile is not a return to days of cowboy programmer



Agile & CMM® Process

- Paulk (2002/10) noted Agile advocated many good engineering practices - some controversial and counterproductive
- McCabe and Polen (2002/10) questioned how could bad things continue to happen to good programs where CMM® was applied - implying maybe Agile might help



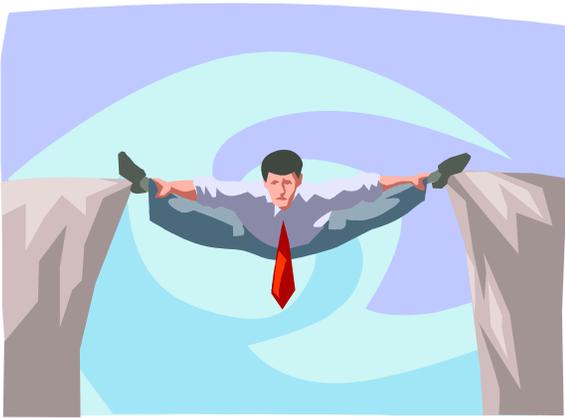
Agile & CMM® Process

- Highsmith (2002/10) wrote Agile & CMM®/CMMIsm are different conceptual frameworks
- They drive organizations to different behaviors
 - Agile best when in equivalent of a “battle zone”
 - CMM®/CMMIsm best in defined process with defined task



Agile & CMM® Process

- Jacobs (2004/03) used Agile to instantiate CMM®
 - Avoided tendency to over-process with multiple forms, plans, and procedures
 - Accelerated getting processes in place quickly
 - Concentrated on improving processes over time
- The Perez & Ambrose (2007/08) used Agile to instantiate CMMIsm
 - Moved from no formal process capability CMMIsm ML2
 - Prototyped processes
 - Defined processes 30% faster



Agile & CMM® Process

- Glazer (2010/01) says Agile and CMMIsm complete each others' capabilities - lead to fast, affordable, visible, & long-term benefits
- Dutton (2010/01) writes that practices contained in the CMMI-DEV have migrated to enable Agile approaches
- SEI CMU/SEI-2010-TR-033 include guidance for Agile methods



Agile & Waterfall

- Cockburn (2002/10 part 1) wrote Agile means prioritizing for maneuverability
 - Requirements
 - Technology, and
 - Understanding of the situation
- Cockburn (2002/11 part 2) wrote plan-driven can borrow from Agile
 - Streamlining
 - Improving Predictability
 - Hedging Bets
 - Lowering Costs



Agile & Waterfall

- Willison (2004/04) described Army's Maneuver Control System (MCS Lite)
 - Software process struck balance between Agile & Waterfall
- Turner & Boehm (2003/12) say critical success factors are generally people factors
 - Staffing, culture, values, communication, & expectations management



Agile & Waterfall

- Cockburn (2004/11) reported Agile scorned models & schedules for
 - Emphasized collaboration social tools
 - Used feedback tools, e.g., CM, automated testing
- Surdu & Parson (2006/4) say development method depends on the program, for OneSAF
 - Followed CMMIsm Level 5 (Waterfall) & individual interactions (Agile)
 - Focused on tacit knowledge & social collaboration in contrast with Waterfall's impersonal milestones



Agile & Project Management

- Sleva (2002/10) noted Hill AFB used Agile for an auditable “unplanned work” approval tracking system - responded to change over following plan
- Mekelburg (2003/04) wrote traditional and agile approaches assume success is features delivered – but projects are successful only when they have met the stakeholders’ expectations
- McMahon (2004/05) discussed case study of conflicts where a company that used Waterfall collaborated with a company using Agile – needed lightweight project management framework



Agile & Project Management

- McMahon (2005/05) presented a case for using key Agile practices along with recommended extensions on a broad range of projects - large and distributed
- Miller (2005/12) says Agile at Microsoft® uses personas, shadowing, and test thresholds.



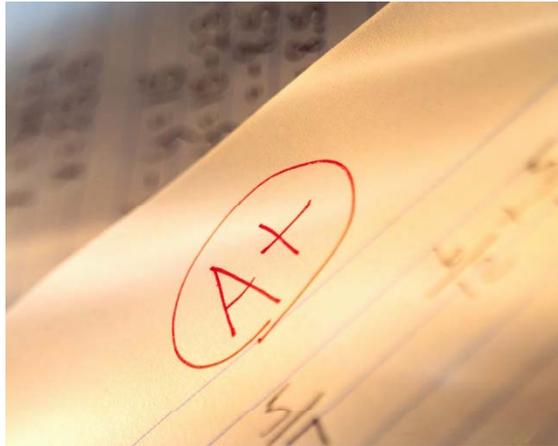
Agile Performance & Metrics

- Reiffer (2002/6) examined Agile & software estimating
 - Concluded estimating software size and duration was feasible using Web objects
- Manzo (2002/10) provided some Agile performance statistics compared to projects conducted before adopting Agile
 - Showed cost per line of code & defect rates drastically reduced
 - Development velocity was significantly increased
- Opperthausen (2003/9) discussed Agile requirements & implementation defects prevention & management
 - Concluded Agile focused on prevention and repair
 - Included both requirements and implementation defects



Agile Performance & Metrics

- Cockburn (2006/02) describes governance metrics
 - True value, expected vs. actual progress
 - Used combinations of waterfall, incremental, concurrent, and Agile strategies
- Derby (2007/04) looks beyond Agile technical skills
 - Cites interactions & collaboration skills for peak performance
- McMahon (2008/05) says to question whether measuring the right things:
 - Are you seeing the results of your process improvement efforts?
 - If not, do you understand your real “as-is” process?



Agile & Testing

- Daich (2003) discussed testing using combinatorial coverage & Orthogonal Array Testing Strategy (OATS)
 - Provided better integration test coverage, whether following CMM® or applying Agile testing methods
- Siddiqi (2008) studied Web Service (WS) standards & strategies for interoperability
 - Examined open source, service-oriented architecture (SOA), & Agile techniques
 - Allowed the team to more efficiently review and test
- Crowe & Cloutier (2009) Agile supported the DoD's Evolutionary Acquisition (EA) policy to rapidly provide operational capabilities to the warfighter
 - Used a rapid test approach to get feedback & resolve problems

Agile & Other Domains



- McMahon (2006/05) says U.S. defense contracts experienced **systems engineering** breakdown
 - Agile is not a short-cut around systems engineering
- Turner (2007/04) says traditional **systems engineering** may not fit Agile systems
 - Inherent Waterfall orientation in system engineering
- Cockburn (2007/04) writes that Agile software engineering is similar to **agile manufacturing**
 - Analogy leverages lessons learned studies (100 yrs)



Agile & Other Domains

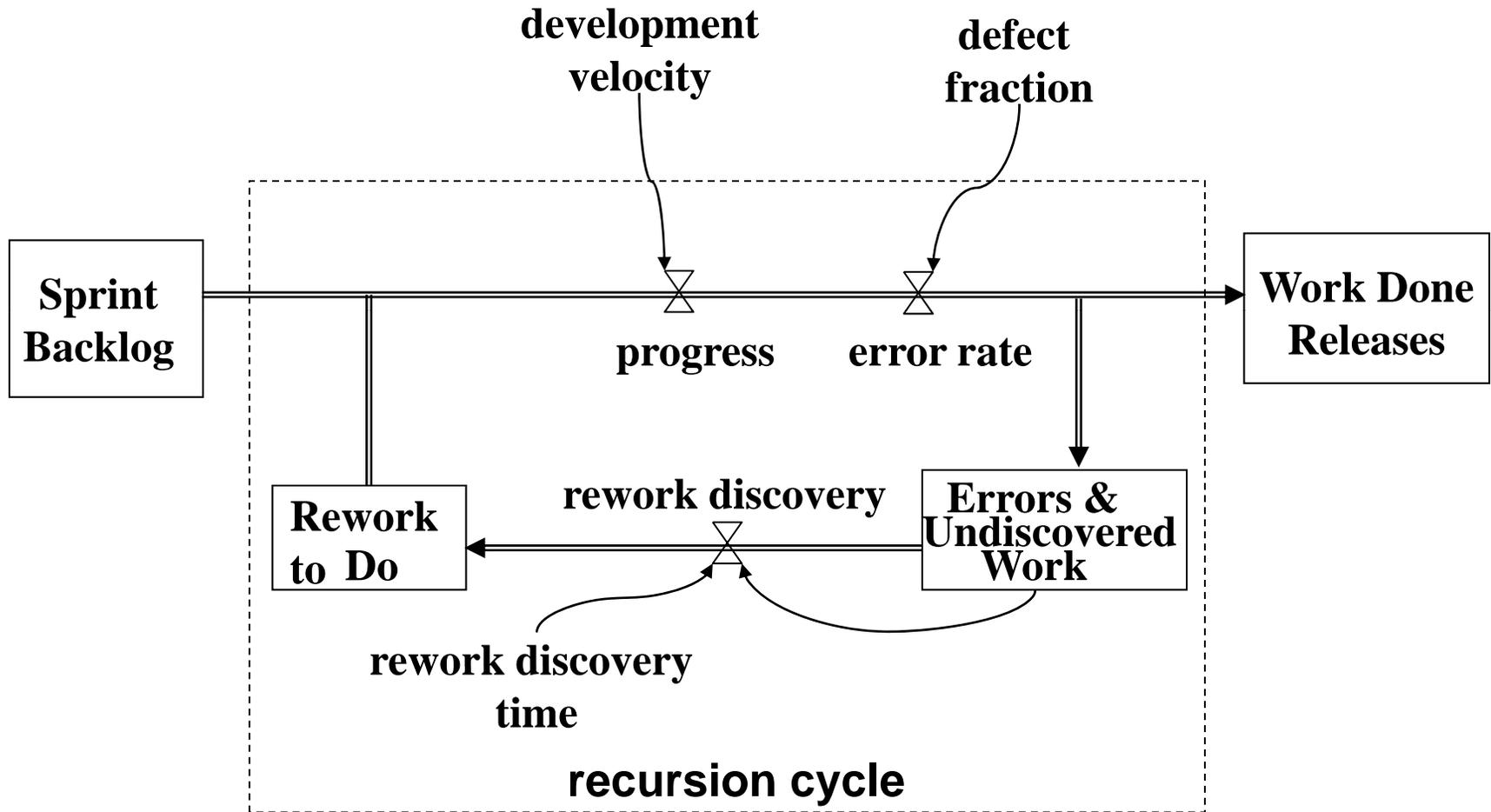
- Derby (2009/01) advises **evidence-based management**
 - Looks at what actually works rather than relying on common practices, or fads
- Brown, Nord & Ozkaya (2009/01) say Agile practices often overlook critical role of **architecture**
 - Architectural Agility allows architectural development to follow a “just-in-time” model
- McMahon (2009/02) applied Agile to address shortfalls under defense **acquisition** regulations, DoD/National Security Space Acquisition Policy 03-01.
 - Funding for Risks/Deferring Non-Key Items/Defining Readiness



What's Next

- Key to Agile's future
 - Empirical feedback
 - Double-loop learning

What's Next



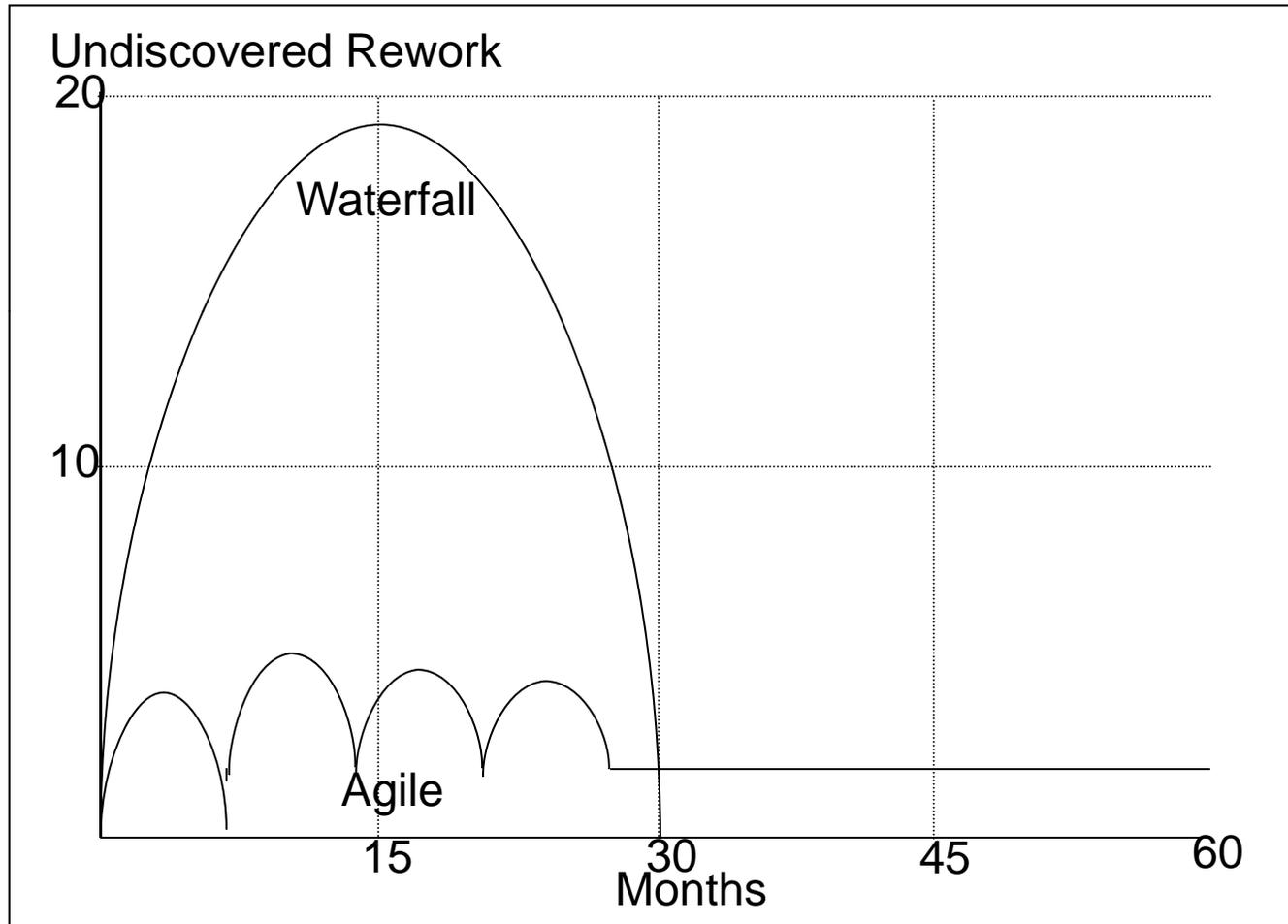
Conceptual Model (Adapted from: Lyneis & Ford, 2007, p161)

What's Next

- Tignor (2009) explored agile project management relative to Lyneis & Ford (2007) generic rework structure
 - Reviewed 17 agile articles
 - Identified agile feedback
 - Allocated feedback to generic rework structure

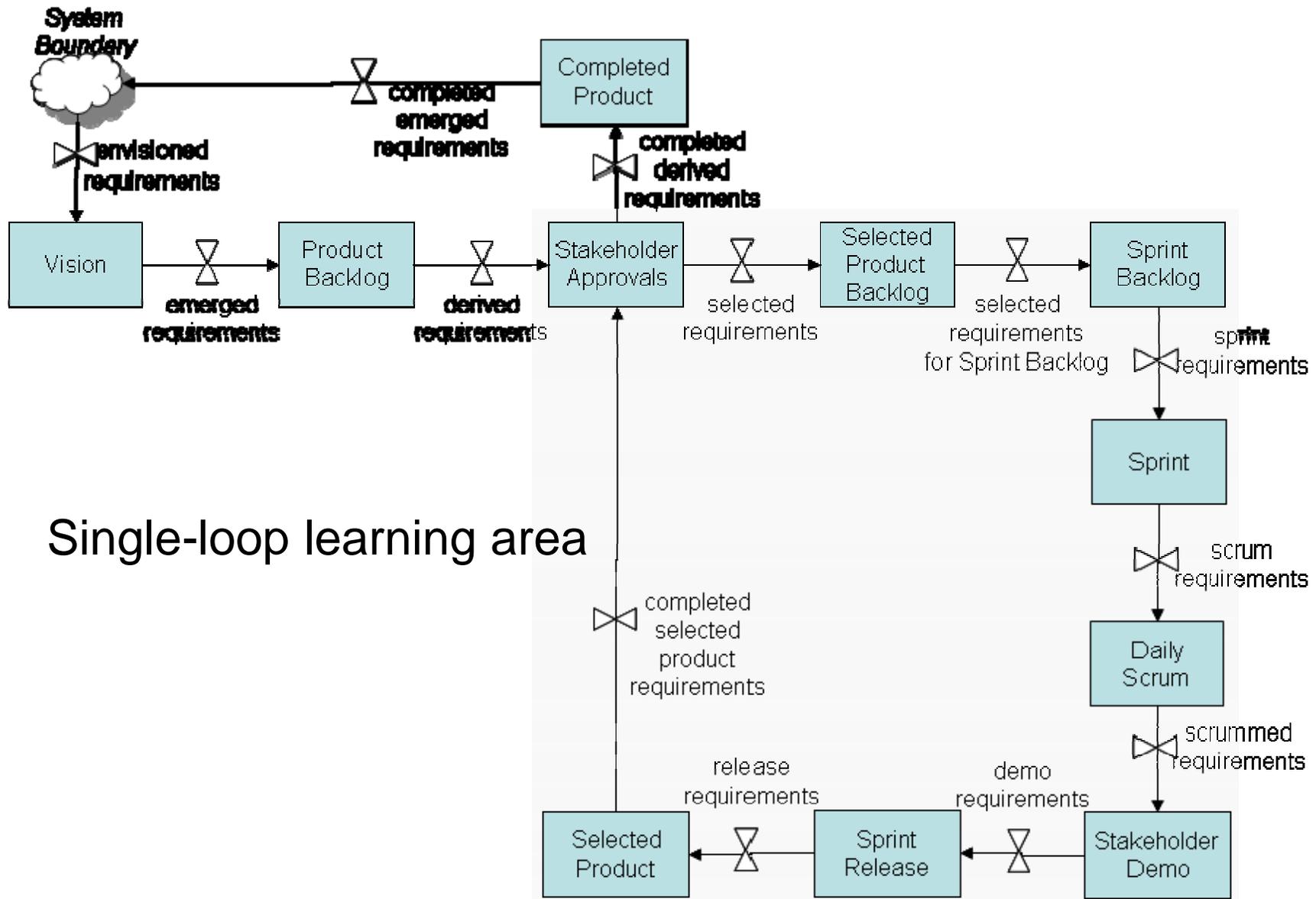
Rework Cycle	11
Original Work To Do	4
Rework To Do	2
Work Done	3
Undiscovered Work	2
Rework Discovery	3
Progress	2
Error Generation	1
Controlling Feedback	3
Ripple Effects	1
Knock-on Effects	2

What's Next



Undiscovered Rework adapted from Chichakly (2007), (Courtesy: Chichakly)

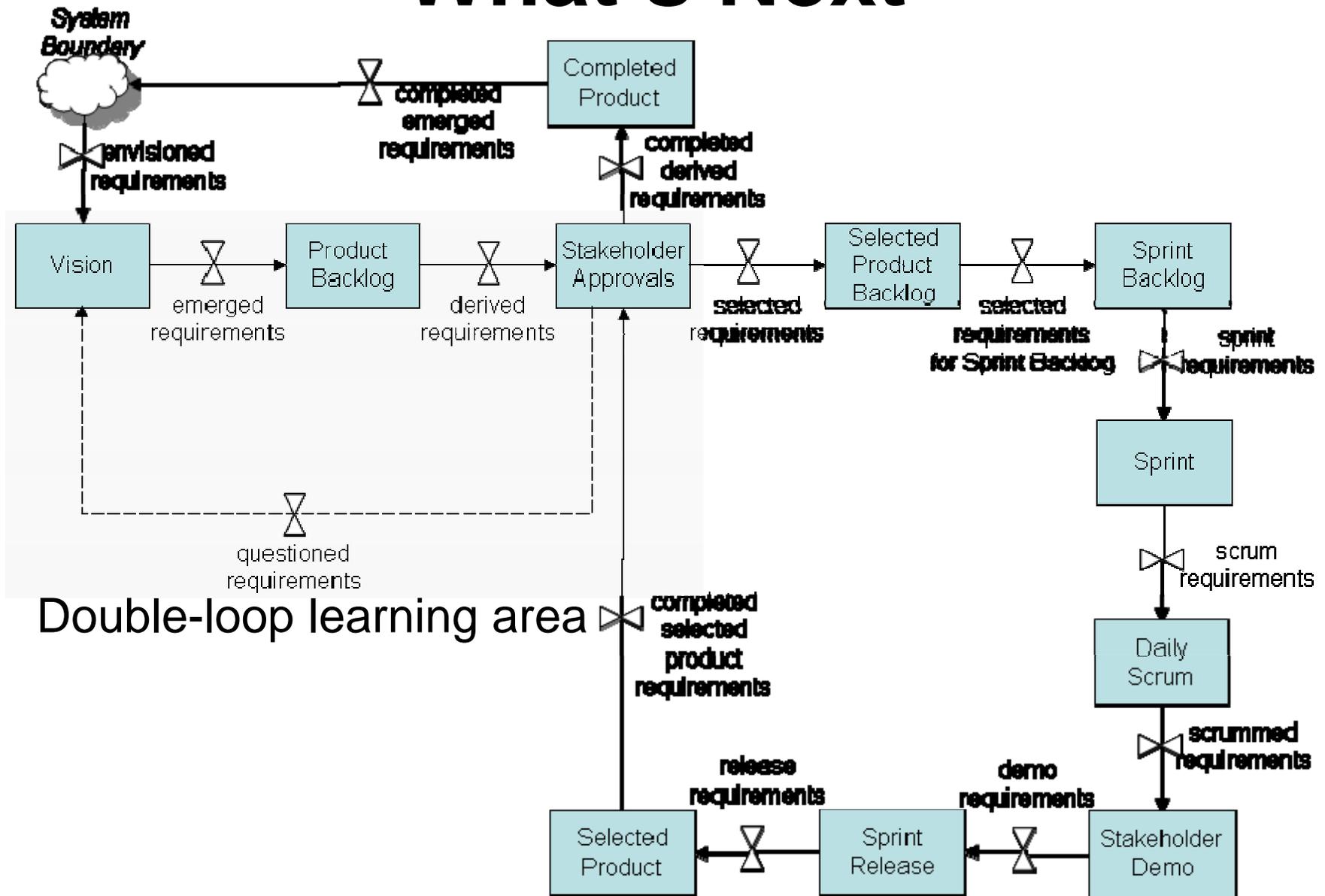
What's Next



Single-loop learning area

Adapting the Schwaber (2007) SCRUM graphic to a "rework" model

What's Next



Adapting the Schwaber (2007) SCRUM graphic to a "rework" model

Summary

- Agile solves complex problems based on its adaptive, iterative, and incremental properties
- Agile has the flexibility to cross over to other domains, e.g., CMM®, Waterfall, system engineer, ...
- Agile acknowledges that feedback plays a role, but feedback is generally overlooked as a detail
- The degree that feedback underpins Agile is significant upon closer inspection
 - Single-loop learning will help Agile manage its backlogs
 - Double-loop learning will help Agile manage its vision
- **Rugby: All Blacks 36 v England 12 Auckland, NZ (6/19/04)**

Glossary

- AFB – Air Force Base
- CMMI-DEV – CMMI for Development
- CM – Configuration Management
- CMM® - Capability Maturity Model
- CMMIsm - Capability Maturity Model Integration
- EA - DoD's Evolutionary Acquisition policy
- HBR – Harvard Business Review
- MCS - Maneuver Control System
- OATS - Orthogonal Array Testing Strategy
- OneSAF - One Semi-Automated Forces
- SEI CMU – Software Engineering Institute Carnegie Mellon University
- SOA - Service-oriented Architecture
- WS - Web Service
- XP – Extreme Programming