Issues in Adopting Agile Methods

Mark C. Paulk, Ph.D.

May 11, 2011
Pittsburgh SPIN

Important Agile Methods

Extreme Programming (XP)
Scrum
Crystal Light Methods, specifically Crystal Clear
Feature Driven Development (FDD)
Adaptive Software Development
Dynamic Solutions Delivery Model (DSDM)
Lean Development
The Agile Manifesto

“We are uncovering better ways of developing software by doing it and helping others do it. 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 on the items on the right, we value the items on the left more.”

http://www.agilealliance.org

Rakitin’s Translation

Individuals and interactions over processes and tools. Talking to people instead of using a process gives us the freedom to do whatever we want.

Working software over comprehensive documentation. We want to spend all our time coding. Remember, real programmers don’t write documentation.

Customer collaboration over contract negotiation. Hagglng over the details is merely a distraction from the real work of coding. We’ll work out the details once we deliver something.

Responding to change over following a plan. Following a plan implies we have to think about the problem and how we might actually solve it. Why would we want to do that when we could be coding?

Why Embrace Agility?

Agility helps us manage change and uncertainty...

Change does not yield to rational analysis; responding to change requires a leap of faith...

There is never one obvious option, but a multitude of options that seem reasonable.

There is never enough information, and the information in hand is often contradictory.

Change creates ambiguity, uncertainty, doubt, and indecision that lead to floundering.


Principles of Agile Software Development

A. Cockburn, “Learning From Agile Software Development”

1) Different projects need different methodology trade-offs.
2) A little methodology does a lot of good; after that, weight is costly.
3) Larger teams need more communication elements.
4) Projects dealing with greater potential damage need more validation elements.
5) Formality, process, and documentation are not substitutes for discipline, skill, and understanding.
6) Interactive, face-to-face communication is the cheapest and fastest channel for exchanging information.
7) Increased communication and feedback reduces the need for intermediate work products.
8) Concurrent and serial development exchange development cost for speed and flexibility.
9) Efficiency is expendable in non-bottleneck activities.
10) Sweet spots speed development.
**Six Sweet Spots**

Dedicated developers  
Experienced developers  
Small co-located team  
Automated regression tests  
Easy access to users  
Short increments and frequent delivery to real users

---

**Agile vs Plan-Driven Tradeoffs**  

S.W. Ambler, “Agile Adoption Rate Survey Results: February 2008,”
Extreme Programming Practices

Planning game
Small releases
Metaphor
Simple design
Testing
• (test-driven development)
• (customer tests)
Refactoring
• (design improvement)

Pair programming
Collective (code) ownership
Continuous integration
40-hour week
• (sustainable pace)
On-site customer
• (whole team)
Coding standard

(http://www.xprogramming.com/xpmag/whatisxp.htm)

The 2004 XP Practices

Primary Practices
• Sit together
• Whole team
• Informative workspace
• Energized work
• Pair programming
• Stories
• Weekly cycle
• Quarterly cycle
• Slack
• Ten-minute build
• Continuous integration
• Test-first programming
• Incremental design

Corollary Practices
• Real customer involvement
• Incremental deployment
• Team continuity
• Shrinking teams
• Root-cause analysis
• Shared code
• Code and tests
• Single code base
• Daily deployment
• Negotiated scope contract
• Pay-per-use

Scrum’s 3 + 3 + 3

Three roles
• Product Owner
• ScrumMaster
• Development Team

Three ceremonies
• Sprint Planning Meeting
• Daily Scrum Meeting
• Sprint Review Meeting

Three artifacts
• Product Backlog
• Sprint Backlog
• Burndown Chart

No Silver Bullet

Scrum will not solve your problems.
Scrum will make your problems visible.
You will have to solve your problems.
Potential Barriers to Agile

Skepticism about agile intent *a la* Rakitin

Doubts about agile implementation
- as intended by the agile gurus

Concerns about impact on existing jobs
- Project managers, business analysts, product line owners, ...

Concerns about impact on customer-supplier relationships
- Custom software development vs software products

The normal resistance to change, diffusion of innovation...

Three Important (Empirical) Questions

How has using agile methods helped projects succeed or fail in your company?
- What’s usefully different about agile?

How do projects measure business success in your company?
- Predictability, short cycle time, customer satisfaction, low cost, innovation, ...

What factors affected adoption of agile methods?
- What caused you to consider agile?
- What barriers hindered adoption?
- What tailorings were necessary to fit your culture?
Scrum Practice Effectiveness

The Scrum method in principle should be used without much variation – significant variation probably indicates a “ScrumBut” implementation.

Practices that are perceived as working will continue to be used...

Were any Scrum practices perceived as not being feasible?

Did any Scrum practices not work well when tried?

Measuring Business Success

Does the project have a “product vision” that characterizes success?

How does the project measure success?
• financial / market (profit, market share)
• quickly responding to changing customer needs
• cost and schedule drivers
• quality
• customer satisfaction / delight
• innovation (building for the future)
Factors Affecting Adoption

Some general diffusion of innovation / change management / marketing / technology adoption / process improvement factor issues
• sponsorship – addressing business problems
• resistance to change
  - We need a requirements specification.
• shelfware (facades)
• training, user groups, conferences
• scope of piloting / adoption
• politics, responsibility, accountability

Some cultural issues
• power and control
• uncertainty (when will we be done?)
• confrontation vs compromise
• risk aversion

Tacit vs Explicit Knowledge

“The difference between agile methods and the Unified Process is knowledge management – agile is tacit, UP is explicit.”

Architecture Breakers

For very large systems, our Pareto analysis of rework costs at TRW indicated that the 20% of the problems causing 80% of the rework came largely from “architecture-breakers.”

Over-focus on early results in large systems can lead to major rework when the architecture doesn’t scale up.


Barriers to the Success of Agile

A customer who insists on the big specification...

A culture that requires long hours to prove commitment...

Projects that are too big (more than about ten programmers)...

An environment with a long time to gain feedback (e.g., realistically test the software)...

The wrong physical environment (e.g., team members on different floors, not co-located)...

We do “agile”, just not most / any of the practices...
Cultural Misfits
(Using the DoD as an Example...)

Regulatory requirements for a level playing field raise challenges for evolutionary and incremental development...

The need by the contracts officer for a requirements specification...

Progress payments defined from a waterfall mentality...

Barriers – regulatory and cultural – to a collaborative customer relationship...

Protests from competitors...

Scrum Adoption Research Plans

Survey of Scrum projects

Interviews and case studies of selected projects

Confirming / testing “what everyone knows”
  • Scrum practices, variations, and associated practices
  • business value of Scrum
  • factors affecting Scrum adoption
Questions and Answers

World Conference on Quality and Improvement

The Software Division of ASQ invites you to attend the Institute for Software Excellence (ISE) held concurrently with the ASQ World Conference on Quality and Improvement

May 16 – 18, 2011
Pittsburgh, PA
David L. Lawrence Convention Center
Room 330

http://wcqi.asq.org/
Contact Information

Dr. Mark C. Paulk
Institute for Software Research
Carnegie Mellon University
Wean Hall 5101
5000 Forbes Avenue
Pittsburgh, PA 15213 USA

Email: mcp@cs.cmu.edu
or Mark.Paulk@ieee.org
Web: http://www.cs.cmu.edu/~mcp/