Issues in Adopting Scrum and Agile Practices

Mark C. Paulk, Ph.D.

September 20, 2011
IEEE Huntsville Section

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
Sweet Spots for Agile

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

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.

---

S.W. Ambler, “Agile Adoption Rate Survey Results: February 2008,”
XP Criticisms

From the 1999 IEEE Dynabook eXtreme Programming Pros and Cons: What Questions Remain?

Cannot be scaled to large problems, multi-discipline teams (explicitly software only)

Seems to be targeted to custom software for a specific company where developers and customers can converse

Relies on generalists; current staff are usually specialists

Methods are only briefly described

Refactoring does not work effectively on large systems

“Stories” do not capture non-functional constraints well

Not design-centric, does not adequately capture design information (no design documentation)

Lacks a well-structured review process; quality is through testing

Needs management support and infrastructure
What Are the Issues?

Agile may not fit the business / project environment

Misunderstandings of the agile culture
  • Resistance by those whose jobs are affected
  • Resistance by the customer / end user

Ignorance of how to do the agile practices right

Unable to appropriately adapt the agile practices to the context

Does Agile Fit Your Needs?

Misunderstandings and Resistance

Does my role fit into agile?
  • Project managers
  • Business analysts

Will my experience be nullified by the agile way of doing business?
  • Collaborative culture vs hierarchical
  • Empowerment, participation, ...

Is Agile an Excuse for Hacking?

<table>
<thead>
<tr>
<th>Individuals and interactions over processes and tools.</th>
<th>Talking to people instead of using a process gives us the freedom to do whatever we want.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working software over comprehensive documentation.</td>
<td>We want to spend all our time coding. Remember, real programmers don’t write documentation.</td>
</tr>
<tr>
<td>Customer collaboration over contract negotiation.</td>
<td>Haggle over the details is merely a distraction from the real work of coding. We’ll work out the details once we deliver something.</td>
</tr>
<tr>
<td>Responding to change over following a plan.</td>
<td>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?</td>
</tr>
</tbody>
</table>

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…

What Is “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)
**Tacit vs Explicit Knowledge**

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

*Ivar Jacobson, Software Engineering Conference (Russia), 27-28 October 2005.*

---

**Money for Information / Flexibility**

Many project strategies revolve around spending money for either information or flexibility.

*Money-For-Information* (MFI) – the team can choose to expend resources now to gain information earlier
  • mock-ups, prototypes, and simulations

*Money-For-Flexibility* (MFF) – the team may opt to expend resources to preserve later flexibility
  • incremental delivery with iterative re-work

**What’s a User Story?**

A promise to have a discussion; not every detail needs to be included.

Describes functionality that will be valuable to either a user or purchaser of a system.
- **Card** – written description of the story used for planning and as a reminder
- **Conversation** – about the story that serve to flesh out the details of the story
- **Confirmation** – details that can be used to determine when a story is complete

*Ron Jeffries, XP Magazine, August 30, 2001.*

---

**Building the Foundation**

![Graph showing Effort vs. Sprint with different trends for Architecture & Infrastructure and Business Value](image-url)

17

18
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.


Technical Debt

Ward Cunningham first drew the comparison between technical complexity and debt in 1992.

Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite… The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt.

Activities that might be postponed include
- Documentation
- Writing tests
- Attending to to-do comments
- Tackling compiler and static code analysis warnings
- Knowledge that isn’t shared around the organization
- Code that is too confusing to be modified easily

Sprint Retrospective

A three hour, time-boxed meeting (for one-month Sprints) held after the Sprint Review and prior to the next Sprint Planning meeting where the Team discusses what went well in the last Sprint and what can be improved for the next Sprint.

I’d say the retrospective is the second most important event in Scrum (the first being the Sprint Planning Meeting).

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...

Questions and Answers
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/