Editor: Donald J. Reifer ■ Reifer Consultants ■ d.reifer@ieee.org

How Good Are Agile Methods?

The software industry seems to be embracing yet another change to the way it does business. Because of their emphasis on agility and time-to-market, many programming shops are moving to agile methods. Unlike more traditional approaches, these methods focus on generating early releases of working products using mostly collaborative techniques such as pair programming, refactoring, and having customers work on site as team members. Programmers use these releases—which are working products, not prototypes—to demonstrate features and functions to stakeholders involved in their use, marketing, and support.

This article surveys the experience software engineers in a wide range of industries have had in deploying agile methods.

The survey

Fads come and go, in software engineering as in everything else. Practitioners want to know if agile methods are real or just more hype. To answer that question, I surveyed 10 industry segments using the approach illustrated in Figure 1. I designed the survey to

- Determine what practices early adopters of agile methods are using
- Assess the scope and conditions governing their use
- Evaluate the costs and benefits associated with their use

Table 1 summarizes the demographics of the 32 organizations, representing 28 firms, that responded (several large firms had more than one organization trying to use agile techniques). To transfer a technology, these firms use it on a pilot to prove to themselves that it works, use it on a pathfinder to determine how to integrate the technology with their processes, and then move it onto production projects. As expected, five of the 14 firms that responded are involved in e-commerce and e-business applications. The information these early adopters supplied gives us insight into how to tap the power of these emerging practices.

The 14 firms using agile methods cited a laundry list of practices as agile: collective ownership, concurrent development, continuous integration, customer collaboration, daily standup meetings, product demos instead of documents, Extreme Programming (XP), frequent product releases, full stakeholder participation, individuals and interactions, just-in-time requirements, metaphors instead of architectures, nightly product builds, pair programming, rapid application development, refactoring, retrospectives, stories for requirements, team programming, and test-driven development.

The database's 31 projects showed that those firms pursuing agile methods were motivated because they had a poor record of delivering acceptable products to market on time and within budget. Most projects were relatively small (typically fewer than 10 participants) and were pursued as pilots or pathfinders. All projects were in-house de-

0740-7459/02/$17.00 © 2002 IEEE

velopments (as opposed to contracted out), lasting one year or less and involving low-risk methods.

Furthermore, the firms characterized their projects as having stable requirements, established architectures, and a high degree of development flexibility. Products under development were mostly quick-to-market applications (generally Web-based and client-server oriented). Teams were cohesive and staffed with motivated, experienced performers, most of whom were relatively young and thus perhaps more open to new ideas. Although there was some skepticism, most practitioners involved with agile methods were enthusiastic about the prospects.

Although software engineers in the various industries differed on what constituted best agile practices, invariably they agreed that a project's process must be cyclical and involve builds and increments done in parallel. Furthermore, they said, these projects must involve collaborative organizations that include participation by all stakeholders during development. These projects always included full-time participation by customers or users while the work was being done, rather than relying on reviews, and resulted in working product demos, not documents or prototypes that are often thrown away.

Differences arose in the actual form of the process used, such as spiral, incremental, or similar methods, as well as on how informal or flexible the process should be. The engineers in different industries disagreed on who the stakeholders were and how deep their involvement should be. Opinions differed as well on what practices fell under the category of agile methods—Extreme Programming, rapid application development, team programming, and so on.

The biggest surprise was that most responding organizations were at Level 2 or greater under the Software Capability Maturity Model (see Table 2). For the most part, these advanced organizations were willing to try something new because they were having problems meeting delivery expectations even though their pro-cesses were mature. Also, most of the organizations trying agile methods were modifying their processes to incorporate those that worked into their way of doing business.

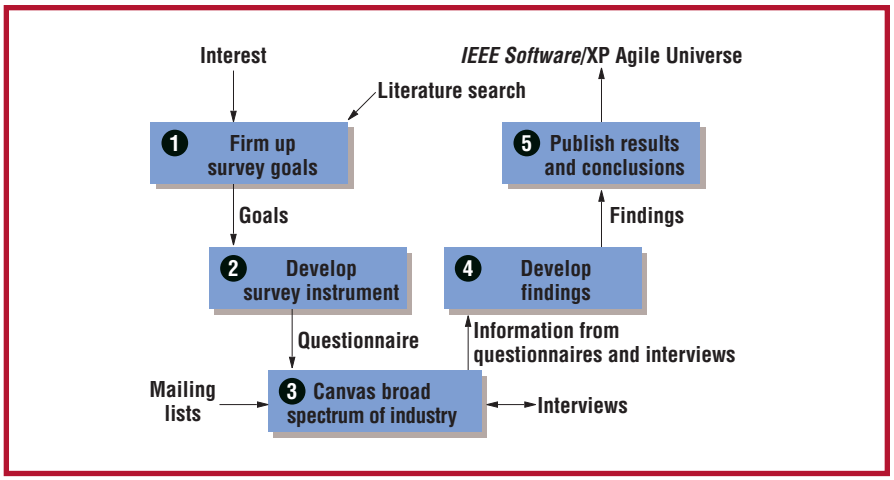

Figure 1. Survey approach showing steps taken to access industry response to XP methods.

The results so far

In summarizing the results either measured or observed by these early adopter organizations, seven of the 14 organizations that used agile methods captured hard cost, productivity, and quality data. Five of these had benchmarks that they could use for comparisons. Hard data included

- *Productivity improvement*: 15 to 23 percent average gain based on published industry benchmarks.[2]
- *Cost reduction*: 5 to 7 percent on average based on published industry benchmarks.[2]
- *Time-to-market compression*: 25 to 50 percent less time compared to previous projects in participating firms.
- *Quality improvement*: Five firms had data showing that their defect rates were on par with their other projects when products or applications were released.

These numbers normalize contributions of all participating firms independent of their CMM levels.

In addition, the seven organizations that didn't capture hard data used soft data to justify their move to

Table 1

Characteristics of responding firms

Industry	Firms using agile methods	Projects	Year first tried	State of progress	Average size (KESLOC)*
Aerospace	1	1	2001	Pathfinder	23
Computer	2	3	2000	Pilot	32
Consultants	1	2	2000	Pilot	25
E-business	5	15	2000	Production	33
Researchers	1	1	2000	Pilot	12
Scientific	0	0	2001	Pilot	N/A
Software	2	4	2000	Production	25
Telecom	2	5	2000	Production	42
Total	**14**	**31**		**Average**	**31.8**

*KESLOC = thousand equivalent source lines of code computed using formulas that normalize reused and modified code in terms of new lines of code (see Barry Boehm's discussion of the mathematical approach involved).[1]

Survey Summary and Recommendations

Questions asked and responses

What do users think agile methods are?

- Devised list of variants and invariants based on user perceptions, not on a search.

Who's using agile methods?

- Small, in-house teams developing software for quick-to-market applications.

Do they provide added value?

- Although reports from the field were positive, the sample was too small to make any broad conclusions.

What are the issues plaguing users?

- Most issues revolve around classical problems in managing technology change.

Recommendations

- Clearly define what "agile methods" means.
- Build a business case for agile methods using "hard" data to justify the move.
- When adopting agile methods, recognize that you are changing the way your organization does business.
- Provide those moving to agile methods with support for making the transition. Support should include startup guidelines, "how to" checklists, and measurement wizards; a knowledge base of past experience accessible by all; and education and training, including distance education and self-study courses.

agile methods. Most used some form of survey to capture stakeholder opinions, and all used recruitment, morale, and other intangibles to build a case for trying and retaining agile methods. All argued passionately for continued use of agile methods based on qualitative factors, and all pressed for help in resolving the issues that revolved around technology transfer.

In any case, the jury is still out because the sample size (14 organizations and 31 projects) is just too small to derive any firm conclusions. In addition, the "hard" data gathered might be tainted by the Hawthorne effect common in efforts of this type, which relates to the small sample size. (These were small, low-risk projects staffed by select teams under controlled situations, so the results might neither scale to larger projects nor reflect higher-risk situations.) We will just have to see if agile methods can scale to address larger efforts.

The percentages I've cited can be deceptive. Although cost, schedule, productivity, and quality are related, they must be considered separately. For example, as we all well know, decreasing cost by accepting reduced quality can accelerate schedule but result in lost market share. And increasing productivity could increase a company's cost as software staff is busily producing the wrong product. In such cases, rework increases as does schedule.[1]

In the "Survey summary and recommendations" sidebar, I've organized my findings by the questions the survey sought to answer. Recommendations are aimed at addressing key issues identified by early adopters. I hope this initial report from the field on agile methods prompts others to put their experiences in the public domain. I am currently preparing a paper detailing this survey's findings for presentation at XP Agile Universe. If you're interested in this topic but cannot attend, contact me for a copy of that paper.

References

1. D.J. Reifer, *Making the Software Business Care: Improvements by the Numbers*, Addison-Wesley, Reading, Mass., 2002.
2. D.J. Reifer, "Let the Numbers Do the Talking," *Crosstalk*, Mar. 2002, pp. 4–8.

Donald J. Reifer is president of Reifer Consultants and a visiting associate at the Center for Software Engineering at the University of Southern California. Contact him at d.reifer@ieee.org. He'd like to hear from you.

Table 2

Software CMM ratings of responding firms

Industry	Number of agile projects	Level 1	Level 2	Level 3	Level 4	Level 5	No Rating
Aerospace	1				1		
Computer	3		3				
Consultants	2			1	1		
E-business	15		6	1			8
Researchers	1		1				
Scientific	0						
Software	4		2	2			
Telecom	5			2	2	1	
Totals	**31**		**12**	**6**	**4**	**1**	**8**