Agile Software Development: The People Factor

Alistair Cockburn, Humans and Technology
Jim Highsmith, Cutter Consortium

In a previous article ("Agile Software Development: The Business of Innovation," *Computer*, Sept. 2001, pp. 120-122), we introduced agile software development through the problem it addresses and the way in which it addresses the problem. Here, we describe the effects of working in an agile style.

AGILE RECAPPED

Over recent decades, while market forces, systems requirements, implementation technology, and project staff were changing at a steadily increasing rate, a different development style showed its advantages over the traditional one. This *agile* style of development directly addresses the problems of rapid change.

A dominant idea in agile development is that the team can be more effective in responding to change if it can

- reduce the cost of moving information between people, and
- reduce the elapsed time between making a decision to seeing the consequences of that decision.

To reduce the cost of moving information between people, the agile team works to

- place people physically closer,
- replace documents with talking in person and at whiteboards, and
- improve the team's amicability—its sense of community and morale—so that people are more inclined to relay valuable information quickly.

Agile development focuses on the talents and skills of individuals, molding the process to specific people and teams.

To reduce the time from decision to feedback, the agile team

- makes user experts available to the team or, even better, part of the team and
- works incrementally.

Making user experts available as part of the team gives developers rapid feedback on the implications to the user of their design choices. The user experts, seeing the growing software in its earliest stages, learn both what the developers misunderstood and also which of their requests do not work as well in practice as they had thought.

The term *agile,* coined by a group of people experienced in developing software this way, has two distinct connotations.

The first is the idea that the business and technology worlds have become turbulent, high speed, and uncertain, requiring a process to both create change and respond rapidly to change.

The first connotation implies the second one: An agile process requires responsive people and organizations. Agile development focuses on the talents and skills of individuals and molds process to specific people and teams, not the other way around.

AGILE IS FOR PEOPLE

The most important implication to managers working in the agile manner is that it places more emphasis on people factors in the project: amicability, talent, skill, and communication.

These qualities become a primary concern for the would-be agile team. Skill development is important, so that each person can deliver more value with time.

The attention to the human issues gives agile projects a particular feel. It is not simply a matter of reducing paperwork and hoping that everything else will fall into place. To paraphrase one developer, "A small, rigorous methodology may *look* the same as an agile methodology, but it won't *feel* the same."

The agile group concept grows to span teams, organizations, and other working relationships. It is very difficult for agile people to function well in a rigid organization—and vice versa.

INDIVIDUAL COMPETENCE

Agile development teams focus on individual competency as a critical factor in project success.

If the people on the project are good enough, they can use almost any process and accomplish their assignment. If they are not good enough, no process will repair their inadequacy—"people trump process" is one way to say this.

However, lack of user and executive support can kill a project—"politics trump people." Inadequate support can keep even good people from accomplishing the job.

The "CHAOS Chronicles Version II," a recent update of the Chaos Report from the Standish Group (http://www.pm2go.com), outlines a recipe for success that includes 10 items. The first three items are executive support, user involvement, and experienced project management.

These items are certainly key, and we agree with having them in the top 10. However, "competent staff" is buried inside the tenth item: "Other."

While it is true that "politics trump people," it is equally true that the developers won't succeed in producing a system if they lack the basic competency for the job.

Interestingly, people working together with good communication and interaction can operate at noticeably higher levels than when they use their individual talents. We see this time and again in brainstorming and joint problem-solving sessions.

Therefore, agile project teams focus on increasing both individual competencies and collaboration levels.

In *Now, Discover Your Strengths*, (Simon & Schuster, New York, 2001), Marcus Buckingham and Curt Coffman discuss the outcome of a long-running research program by the Gallup organization, in which 80,000 managers in 400 companies were interviewed over a 25-year period In this book, they highlight the interplay between talent, skill, and knowledge. Regarding high-performance working environments, they write:

> The larger, establishment camp is comprised of those organizations that legislate the *process* of performance. … They try to teach each employee to walk the same path.
>
> This [people-based] type of organization focuses not on the steps of the journey but on the end of the journey.
>
> The distinction between the two camps is real. Step-by-step organizations are designed to battle the inherent individuality of each employee. Strength-based organizations are designed to capitalize on it.

Extending their ideas, it's not that organizations that employ rigorous processes value people less than agile ones, it's that they view people—and how to improve their performance—differently. Rigorous processes are designed to standardize people to the organization, while agile processes are designed to capitalize on each individual and each team's unique strengths: One-size-fits-one—every process must be selected, tailored, and adapted to the individuals on a particular project team.

Agile processes are designed to capitalize on each individual and each team's unique strengths.

Too often, software engineering and rigorous process adherents confuse process and competence. Process can provide a useful framework for groups of individuals to work together, but process per se cannot overcome a lack of competency, while competency can surely overcome the vagaries of a process—again, "people trump process."

When you look under the cover of XP, Scrum, Adaptive Software Development, Crystal Methods, Feature-Driven Development (FDD), or Dynamic Systems Development Methodology (DSDM), the emphasis on people and their talent, skill, and knowledge becomes evident.

This is a competency attitude, not an elitist one. Whether it's the practices of pair programming and mentoring in XP or chief programmers and inspections in FDD, the goal remains consistent—working jointly to improve the knowledge and skill of individuals. A recent book by Pete McBreen, an agile development practitioner, is therefore appropriately titled not *software engineering* but *Software Craftsmanship* (Addison-Wesley, Reading, Mass., 2001).

TEAM COMPETENCE

Agile teams are characterized by self-organization and intense collaboration, within and across organizational boundaries.

Self-organizing teams are not leaderless teams; they are teams that can organize again and again, in various configurations, to meet challenges as they arise. Agility requires that teams have a common focus, mutual trust, and respect; a collaborative, but speedy, decision-making process; and the ability to deal with ambiguity.

We distinguish collaboration from communication. Communication is the sending and receiving of information. Collaboration is actively working together to deliver a work product or make a decision.

The DSDM manual, for example, identifies the differences between traditional and agile project managers, one of which is: "A traditional project manager will normally focus on agreeing to a detailed contract with customers about the totality of the system to be delivered along with the costs and timescales. In a DSDM project, the project manager is focused on setting up a collaborative relationship with the customers."

Scrum projects are characterized by intense 15 to 30 minute daily meetings in which participants constantly adjust to the realities of high-change projects.

All too often, project teams are given the ultimate accountability for product delivery, while staff groups—project offices, process management, data administrators—are given decision power. In order to innovate and react to change, agile teams reflect a better alignment of accountability and responsibility. It is again an emphasis, at the team level, on competency rather than process.

AGILE ORGANIZATIONS

An agile team working within a rigid organization has as difficult a time as agile individuals working within a rigid team. Many project teams we have interviewed report that when they responded to customers' requests and to technology platform changes to deliver usable value to the customer, their management admonished them for lack of conformance to the original plan.

Agile organizations and agile managers understand that demanding certainty in the face of uncertainty is dysfunctional. Agile companies practice leadership-collaboration rather than command-control management. They set goals and constraints, providing boundaries within which innovation can flourish. They are macromanagers rather than micromanagers. They understand that who makes decisions isn't as important as collaboration on information to make informed decisions. They understand that agility depends on trusting individuals to apply their competency in effective ways.

Doug DeCarlo has labeled traditional project management "Newtonian neurosis." In this context, neurosis means attacking complex, nonlinear problems with simplistic, linear processes.

AGILE ECOSYSTEMS

A project is built from people having differing personalities and differing skills, working in a physical environment within an organizational culture. The people, environment, and organizational culture all influence one another. When a strong person leaves, the organization rearranges itself to compensate; when the team spreads itself across multiple floors, communications change; and so on. The project is an ecosystem.

Using the term *ecosystem* to convey the totality of the project at hand brings us to the discussion of processes and ecosystems and fitting the two together.

Installing a process within an ecosystem offers two choices: Fit the ecosystem to the process or fit the process to the ecosystem. In reality, both the process and ecosystem change, but the agile approach honors the ecosystem and recognizes that not every process will work in every ecosystem.

AGILE EFFECTIVENESS

Nearly 200 people from a wide range of organizations in North America, Europe, Australia, India, and other locations responded to a survey of agile and rigorous methodologies conducted by the Cutter Consortium in 2001. This survey showed three interesting results.

First, compared to a similar study conducted late in 2000, many more organizations said they were using at least one agile methodology.

Second, agile methodologies showed slightly better delivery performance than rigorous methodologies in terms of business performance, customer satisfaction, and quality.

Third, agile methodologies scored better than rigorous methodologies in terms of employee morale. This is particularly surprising in that 54 percent of the respondents were IT and executive managers and only 12 percent were developers.

One study is, of course, only one study, and we should view the conclusions accordingly. However, these preliminary results reflect the effectiveness and contribution of agile approaches to morale.

AGILE DOMAINS

Agile development is not for everyone. Imposing agile principles on process-centric, noncollaborative, optimizing organizations is likely to fail. Imposing a change-embracing process on sedate project teams may not be reasonable. Attempting to get close user collaboration with organizations that have little time to spend with developers won't work.

Agile development is more difficult with larger teams. The average project has only nine people, well within the reach of the most basic agile processes. Nevertheless, it is interesting to occasionally find successful agile projects with 120 or even 250 people.

Agile development excels in exploratory problem domains—extreme, complex, high-change projects—and operates best in a people-centered, collaborative, organizational culture. This approach has proved to be effective at solving many problems and at forging attractive work environments in many organizations. While it is not suited for everyone, it is suited for many. ✷

Alistair Cockburn *is Consulting Fellow at Humans and Technology. Contact him at arc@acm.org.*

Jim Highsmith *is director of the Cutter Consortium's e-Project Management Practice. Contact him at jimh@adaptivesd.com.*

Editor: Barry Boehm, Computer Science Department, University of Southern California, Los Angeles, CA 90089; boehm@sunset.usc.edu

How to Reach *Computer*

Writers

We welcome submissions. For detailed information, write for a Contributors' Guide (computer@computer.org) or visit our Web site: http://computer.org/computer/.

News Ideas

Contact Lee Garber at lgarber@computer.org with ideas for news features or news briefs.

Products and Books

Send product announcements to products@computer.org. Contact Stephanie Kawada at skawada@computer.org with book announcements.

Letters to the Editor

Please provide an e-mail address or daytime phone number with your letter. Send letters to Letters, *Computer,* 10662 Los Vaqueros Cir., PO Box 3014, Los Alamitos, CA 90720-1314; fax +1 714 821 4010; computer@computer.org.

On the Web

Visit http://computer.org for information about joining and getting involved with the Society and *Computer*.

Magazine Change of Address

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Make sure to specify *Computer*.

Missing or Damaged Copies

If you are missing an issue or received a damaged copy, contact membership@computer.org.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at whagen@ieee.org. To buy a reprint, send a query to computer@computer.org or a fax to +1 714 821 4010.

COMPUTER
Innovative technology for computer professionals