People over Process: Key Challenges in Agile Development

Kieran Conboy and Sharon Coyle,
National University of Ireland, Galway

Xiaofeng Wang, Lero

Minna Pikkarainen, VTT Technical Research Centre of Finland

// Case studies of 17 organizations that have used agile methods for more than three years uncovered many serious "people" challenges including recruitment, training, motivation, and performance evaluation. //

ALTHOUGH DEVELOPERS HAVE been using agile methods for quite a while, it's important to examine the related "people" challenges. First, these methods' growing popularity means they "are fast becoming the adopted development methodology commercially."[1] Second, agile methods are no longer restricted to small, collocated teams and are increasingly applied in environments outside their comfort zone, thus presenting new personnel and human-resource-management challenges.[2,3] Finally, adopting agile methods is no longer an insular, bottom-up, voluntary decision, in which the project team can choose to embrace or rebuke the transition on its own terms. Increasingly, suppliers, consultants, partners, customers, and even public-sector bodies are coercing the use of agile methods, through formal requirements and as a way to ensure interorganizational process alignment.[4,5]

Agile methods' increasing prevalence, the lowering of traditional agile boundaries, and growing pressure to adopt agile methods all contribute to the need for human resources departments and project managers to address associated skill and people challenges. Sridhar Nerur and his colleagues[6] and Peter Schuh[7] have shown that agile environments differ significantly in context compared to traditional environments (see Table 1), although the distinction often isn't so black and white.

So, it's important to identify the problems that the transition to agile methods can cause. Here, on the basis of case studies of 17 organizations, we describe the most important challenges and offer recommendations on how to address them.

The Research Process

We used a two-phased approach. First, we conducted focus group discussions from June to September 2008 with software development executives, senior project managers, and agility experts. We identified an initial set of challenges and evaluated the case study protocols for the second phase.

0740-7459/11/$26.00 © 2011 IEEE

TABLE 1

Contrasts between traditional and agile methods.[6,7]

Project component	Traditional	Agile
Control	Process centric	People centric
Management style	Command and control	Leadership and collaboration
Knowledge management	Explicit	Tacit
Role assignment	Individual—favors specialization	Self-organizing teams—encourages role interchangeability
Communication	Formal and only when necessary	Informal and continuous
Customer involvement	Important usually only during project analysis	Critical and continuous
Project cycle	Guided by tasks or activities	Guided by product features
Development model	Life-cycle model (waterfall, spiral, or some variation)	The evolutionary-delivery model
Desired organizational form or structure	Mechanistic (bureaucratic with high formalization)	Organic (flexible and participative, encouraging cooperative social action)
Technology	No restriction	Favors object-oriented technology
Team location	Predominantly distributed	Predominantly collocated
Team size	Often greater than 10	Usually fewer than 10
Continuous learning	Not frequently encouraged	Embraced
Management culture	Command and control	Responsive
Team participation	Not compulsory	Necessary
Project planning	Up front	Continuous
Feedback mechanisms	Not easily obtainable	Usually numerous mechanisms available
Documentation	Substantial	Minimal

In the second phase, we conducted 17 case studies from October 2008 to January 2009, using in-depth interviews with senior personnel (see Table 2). Some of these organizations had effectively embraced agile methods, harvesting benefits such as reduced costs, higher-quality systems, and more satisfied software development staff and customers. Others had experienced significant problems and even project failures directly attributable to the transition to agile methods. Selecting cases with such opposing experiences let us compare and contrast, thus identifying the distinguishing skills and challenges related to adoption of agile methods.

Key "People" Challenges

We identified nine key people challenges and practices to address them. Where possible, we show each challenge's prevalence among the cases.

Developer Fear of Skill-Deficiency Exposure

In all 17 companies, developers feared that the agile process could bring their own deficiencies to light. Interviewees outlined how procedures such as stand-up meetings, onsite customers, and the use of storyboards and whiteboards made developer shortcomings visible to the rest of the team because these practices require direct and constant communication and collaboration. For example, storyboards track the status of user stories and make a developer's lack of progress obvious. Whiteboards, which agile teams use to communicate design issues, can highlight developers' technical and communication challenges because they must regularly present their ideas in front of

TABLE 2

The companies studied.

Company code	Firm profile	Description of agile project	Agile method	Team location	Number of interviewees and their roles
A	A leading global management consulting, technology services, and outsourcing company, with 91 of the top global Fortune 100 companies as clients	Internal project management reporting system	Extreme Programming and Scrum	UK, US	Four: two partners, one associate partner, and one project manager
B	A leading global provider of end-to-end solutions for real-time infrastructure in electronic, network, communication, and industrial equipment	New enterprise server product	Lean development	Ireland, US	Three: one human resources (HR) director and two project managers
C	A leading global manufacturer of medical supplies, whose devices are used to diagnose and treat conditions in a variety of medical fields	New requirements-gathering tool	Crystal	Ireland	Two: one project manager and one consultant
D	A regulator and protector of the Irish telecommunications, broadcasting, and energy sectors	Internal reporting system	Extreme Programming	UK	Three: one IT director, one HR manager, and one project manager
E	A leading global technology services firm that delivers a broad portfolio of IT and business process outsourcing services to clients	Customer relationship management (CRM) software implementation and customization for external client	Extreme Programming and Scrum	India, Ireland	One project manager
F	Ireland's leading telecommunications group, providing telephone, Internet, and data transfer services	CRM software implementation and customization	Extreme Programming and Scrum	Ireland	Two: one information services (IS) director and one consultant
G	A large international company providing telecom and automotive products	Automotive and telecom products (devices for mobile applications, wireless)	Extreme Programming and Scrum	Distributed in Germany, Finland, China	Two: one project manager and one site manager
H	The world's leading manufacturer of mobile broadband infrastructure and a provider of optical transport equipment	Software package for mobile devices	Lean software development	Sweden, UK, US, India	Two project managers
I	A leading provider of information security systems in global markets	Information security products	Extreme Programming and Scrum	Distributed in Malaysia, Finland	One manager
J	A global technology firm providing infrastructure and business offerings in software and solutions spanning from handheld devices to computer installations	Web-based license registration system	Scrum	Ireland, US	Three: one HR director and two project managers
K	A global blue-chip technology company and major semiconductor manufacturer	New chip development and testing	Extreme Programming and Scrum	Ireland, US	Four: one IT director (Europe) and three project managers
L	A leading global database-management-system provider	Software plug-ins for main database software product	Extreme Programming and Scrum	Ireland, UK, India	Four project managers

TABLE 2, CONT'D

The companies studied.

Company code	Firm profile	Description of agile project	Agile method	Team location	Number of interviewees and their roles
M	A leading global producer of telecom equipment, including core network switching, wireless, and optical systems	Hardware development	Scrum	Ireland, US, India	Three: one HR director and two project managers
N	A leading global producer of cell phones, including devices, services, and software	Mobile devices	Extreme Programming and Scrum	Finland, US	Four: one IS director, two project managers, and one agile coach
O	A leading global electronics and industrial engineering firm, including industrial automation, control systems, power distribution, and transportation systems	CRM implementation and customization	Extreme Programming and Scrum	US	Four: two consultants and two HR managers
P	Leading global manufacturer of consumer electronics, including TV appliances, medical systems, and silicon systems solutions	Internal reporting system	Extreme Programming	The Netherlands, US, India	Three: one IT director and two consultants
Q	A leading provider of fixed and mobile telephony services through telecommunications networks in Spain, Europe, and Latin America	CRM implementation and customization	Extreme Programming and Scrum	Ireland	Two: one IT director and three project managers

their peers. In addition, continuous integration and automated testing mean that developers can't hide poor, low-quality code.

However, exposing such weaknesses can often prove counterproductive. Eight teams mentioned incidents involving developers with low self-esteem, who, even if performing reasonably well, often felt inadequate in such a transparent environment.

At the other end of the spectrum, in four companies, full transparency created unhealthy environments involving "exhibitionists" (according to a consultant in company P), "show-offs" (manager, company L), and "bullies" (consultant, company P). Repercussions included developers experiencing discomfort (16 cases) and hostility (7 cases), changing teams (14 cases), and leaving the organization (5 cases). Attributing these cases to "weak" developers is too simplistic. We found that "weakness is relative" (manager, company L) and that some highly respected and high-performing developers felt inadequate when compared to those performing at an even higher level.

To address this challenge, developers need an environment where they feel safe to expose their weaknesses. In company C, all developers completed short forms every two weeks in which they could document any fears, issues, or concerns they didn't feel comfortable discussing in an open forum. In company D, listing problems at stand-up meetings was voluntary for junior developers.

In companies B, D, and M, junior or new staff had separate, lengthier stand-up meetings with dedicated mentors. Developers should also know that they can get help to improve their skills. In at least nine cases, pair programming teamed weaker developers with more experienced developers; thus, joint responsibility dissolved the public exposure of any potential weaknesses.

Broader Skill Sets for Developers

In all 17 companies, agile environments seem to blur the boundaries among developers' roles and require competence in a broad range of skills, as opposed to specialization in one.

> *To be a successful agile [developer] you need to be a coder, a tester, an architect, a customer, a quality assurance expert, and a multitude of other things software-related. (manager, company M)*

As a manager in company D described, rather than being a "jack of all trades, master of none," a developer in an agile team must be a "master of all trades." This multifaceted skill set caused numerous problems. First, almost all project managers had difficulty finding developers with all the necessary agile skills, either externally or in their organization.

Training was also more difficult. In four cases, management sent its entire team to all available training courses, incurring high expense. In those cases, before the company adopted agile

development, the developers would have received training only in niche areas directly related to their roles. Because agile development encourages blended roles, depends on voluntary contributions, and emphasizes team as opposed to individual performance, a team member might become a jack of all trades but lack the opportunity to hone a smaller number of key skills—for example, Java certification. Some team members felt this put them at a disadvantage when competing for a promotion or other jobs.

> Being a good communicator is one thing. Knowing what not to communicate is much more important.

To address this challenge, organizations must strike a balance between team members becoming "masters of all" and "masters of none." Developers must have broad knowledge on all aspects of software development but should also specialize and hone their skills in certain areas. As a manager in company G suggested, "An agile developer requires multiple skills but still needs to maintain some degree of specialism." Developers in companies F, L, and M, however, maintained distinct roles (such as tester, Java developer, and database developer) because of large team sizes and the potential for developer conflict.

Increased Social Interaction

Agile practices such as collocation, onsite customers, stand-up meetings, retrospectives, and pair programming increase social interaction and heighten the need for social, communication, and presentation skills. All respondents generally viewed the development of social skills as positive but raised some interesting problems and concerns. First, 15 cases had people who were technically very talented but had inherently weak communication and presentation skills. Whereas all managers saw the benefits of constant face-to-face communication, the degree of communication an agile environment requires clearly diminished some key staff's productivity:

> *When your star player outperforms the rest by five to one, but is not getting the work done because they are losing sleep and breaking into a sweat about standing in front of a group, you need to rethink your approach, and change it for them. (manager, company F)*

Agile development's customer-facing aspect also caused significant problems in eight companies. It was clear that with certain people, "you should never, ever put them in front of a client" (director, company M). In fact, "being a good communicator is one thing. Knowing what not to communicate is much more important" (manager, company O). Managers cited examples of developers revealing politically sensitive and confidential information to customers regarding contracts, salaries, and opinions regarding development teams' weaknesses.

Interestingly, we found that strong social skills might put developers recruited for global software development projects at a disadvantage. One human resource manager noted,

> *When we were hiring home developers, they always presented and communicated really well, but you wondered if they really do have the technical skills they claim to have. The developers in the offshore location presented and communicated terribly, but we were always left feeling their technical skills are better than what was coming across. (manager, company D)*

Social-skills training is an obvious solution to this challenge. company K, however, took a more holistic approach and incorporated social-skills development into a larger training program. The company made videos of all new students' stand-up meeting presentations, which they brought to a required course called "Communication and Presentation in Business." Instructors viewed each recording and integrated it into the course material, letting the students see how their skills improved over time.

Although agile methods emphasize minimal documentation, another mediating solution involved using appropriate documentation to facilitate communication. In company E, a manager found it harder to converse with less experienced developers without supporting documentation. An investment in documentation might have merit in projects with many inexperienced developers.

Lack of Business Knowledge

Agile development involves constant, high-tempo interaction between customers and developers. The embedded nature of the customer's role within the team increases interaction with all team members. So, according to many of those we interviewed, an absence of basic domain knowledge among developers becomes obvious. A manager in company L captured the potential implications of this:

> *If they [the developers] don't know the business basics, the customer loses confidence in their overall abil-*

TABLE 3

Contrasting implementations of agile methods at company O.

Practice	Project 1	Project 2
Stand-up meetings	• Time wasted due to late arrivals • Average 50 minutes, up to 1.5 hours • No responsive action • Highly critical atmosphere • Some people (US-based) frozen out	• Time set to include everyone • Time set aside for breakthrough ideas • Highly interactive • Nonthreatening
On-site customer	• "Highly passive" • Not involved in spikes • Only role was user story validation—"more of an editor" • "Them and me" mentality • Averaged 4.3 days to give feedback on user stories • Attended 27 of 113 stand-up meetings and 6 of 14 retrospectives	• Created brainstorming sessions • Consistently engaged other stakeholders (R&D, manufacturing, accounting, and so on) and continually organized meetings • Real-time involvement, live reprioritization • Attended 43 of 45 stand-up meetings

ity, and their technical strengths may be ignored.

Many managers spoke of this problem's potential "cancerous effect" (manager, company L), citing examples of customer indifference and disengagement because of the resulting perception that "the team knows nothing about our business, so they won't deliver anything of value to our business" (manager, company M). Twelve companies regarded this as a problem; seven found it particularly problematic in situations involving internationally distributed teams. For example, one manager with company K recalled her experiences with a distributed project involving the offshore location:

> *They had the technical skills in abundance but no business acumen whatsoever. … Getting the business angle across to the people (in the offshore location) was really tough. If we can break it down into 1s and 0s they are fine, but anything qualitative is very hard for them to work with. The transition to agile really caused problems with this.*

Training in the business domain is one possible solution. Six companies held training sessions on basic business topics including accounting standards, basic management accounting and finance, and marketing principles. Typically, such training addressed some issues but failed to consider the client-specific knowledge required. In two cases, getting the customer organization to run the training solved this problem. In one case, running small modules on a frequent, phased basis seemed more beneficial than delivering training up front in the project's first week before the team became actively engaged. In another case, making the sessions interactive allowed developers to hone in on the niche areas they found particularly troublesome.

Almost all the companies tried to resolve the problem's root cause by recruiting staff and graduates who had both IT and business knowledge. Three companies actually recruited domain experts, which required significant additional investment. However, all three managers believed the cost was justified.

Understanding Agile Values and Principles

Whereas at least 10 projects implemented agile methods "on paper," they didn't achieve agility's ultimate goals. For example, at company O, two teams implemented agile methods at the same time, participating in the same three-day agile training session. Although both teams implemented stand-up meetings and on-site customer practices, they didn't achieve the same goals (see Table 3). According to a manager at the company, no single issue caused the difference—rather, "some intangible combination of staff personality, management style, cultural issues, and other factors."

Although formal training is a typical solution to teach agile practices, it's insufficient for development teams to adequately embrace agile values and principles. Some companies included a provision for training and attendance at agile conferences focusing on values and principles. Continuous hands-on training was preferable to one-off training in helping developers absorb and retain agile values and principles.

> *The real value came from continuous training. (manager, company L)*

In addition, coaching can complement training to assist a team during the transition to agile methods. In 10 cases, senior team members acted as coaches to drive the effort of retaining agile values and principles in the team. Alternatively, company D found that swapping developers among agile

teams can ensure cross-team observation and validation of agile practices, thus identifying "bad habits." Periodically assessing a team's agility using an assessment framework based on a set of agile goals as opposed to practice adherence can also help. Company A adapted and dropped several agile practices as a result of assessment practices.

Lack of Motivation

Five companies encountered developers who lacked motivation to use agile methods. This was more prominent in companies that adopted agile methods top-down. A manager in company G observed, "sometimes they have the competence but are not convinced it [agile development] will work." Many respondents perceived process innovations such as adopting agile methods as overly onerous, complex, and time-consuming.

In some organizations, mechanisms such as strong personnel involvement in the adoption process (two cases), training (eight cases), and sharing agile development experiences (two cases) already existed to convince and motivate developers to adopt agile methods. A manager in company G indicated how the organization continually collected experience reports on successful agile projects and shared them with project teams. Five companies collected experiences from different teams and customers to gain valuable insights. Various respondents said that sharing agile success stories provided encouragement and inculcated belief in the methods.

Devolved Decision-Making

Some respondents reported significant problems with devolved decision-making, a commonly cited aspect of agile methods:

> *People were picking tasks they shouldn't have. It was self-organizing gone mad. (manager, company L)*

Devolved decision-making can also mean problems for project managers:

> *Project managers do not know what their role is. (manager, company N)*

In company L, the manager cited anxiety over losing power as a "problem among some managers."

Several agile practices contributed to devolved decision-making, including pair programming, stand-up meetings, regular retrospectives, and frequent informal communication.

Sometimes, however, team and peer pressure can be too much. Two companies held weekly 15-minute meetings with individual developers and product managers to ensure that all developers had ample opportunity to communicate anything they found difficult to express in an open forum.

Effective team decision-making practices across all 17 cases included a democratic voting system to ensure that everyone had input on every decision. In three cases, project managers acted as agile-team facilitators who made the final decisions. Such role switching lets project managers act as peers of the rest of the team while retaining the final say.

The organization continually collected experience reports on successful agile projects and shared them with project teams.

Implementing Agile-Compliant Performance Evaluation

In all 17 cases, we found that although agile methods advocate interaction, collaboration, mentorship, teamwork, and communication, the performance evaluation of these activities has many issues. Implementing team collaboration isn't easy if the performance evaluation and appraisal mechanisms are based on individual performance. A manager in company L said,

> *We had one guy, who was the guru of the team. While he was happy with agile during the year, he really felt demotivated when he was passed over for promotion. His argument was he spent most of his time giving advice, pairing with weaker developers, and helping the team in stand-ups and retrospectives. In his eyes, none of this had been rewarded, and as his manager, I have to agree.*

In five cases, the performance evaluation criteria (particularly at junior levels) focused on technical skills and the ability to follow directions, while neglecting distinguishing factors in agile development such as social skills, creative thinking, and self-organization. In other instances, agile teams were evaluated largely according to traditional criteria, so the results often didn't reflect the team members' true abilities. Meanwhile, performance evaluation of the onsite customer seemed particularly problematic and highly contentious. In at least four instances, the customer felt aggrieved that he or she wasn't being rewarded properly:

> *At the end of the day, we can say the onsite customer is vital. In reality though, a marketing person is rewarded for their marketing work, and an accountant for what they do with the accounts. Time spent with a development team helping some other*

TABLE 4

A summary of people challenges and recommendations to address them.

Challenges	Recommendations
Developer fear caused by transparency of skill deficiencies	• Allow feedback outside stand-ups to document any fears, issues, or concerns inappropriate for discussion in open forum • Make stand-up meetings voluntary for junior developers • Assign mentors to new staff • Pair weaker developers with more experienced developers, giving them joint responsibility for requirements
The need for developers to be masters of all trades	• Use pair programming and pair rotation to distribute knowledge and facilitate learning • Encourage task self-assignment to let developers work in different areas and learn new skills • Reintroduce specific roles when it benefits teams—for example with large teams or when conflicts exist among developers
Increased reliance on social skills	• Combine development and training programs to provide customized training materials on social skills, using developers' own examples • Use proper documentation to back up communication
A lack of business knowledge among developers	• Have customer company run training sessions on basic topics within the company's business domain and in specific areas • Provide small, frequent, interactive training modules to let developers acquire the niche business knowledge the project requires • Recruit staff and graduates with a combination of IT and business knowledge
The need to understand and learn agile values and principles, not just the practices	• Ensure multiple members get agile training or attend agile conferences • Encourage agile coaching and championing • Ensure cross-team observation and validation of agile practices • Assess agility in terms of agile values not practice adherence
Lack of developer motivation to use agile methods	• Include motivated developers on each team • Collect and share successful adoption stories and positive experiences
Implications of devolved decision-making	• Build a sharing and learning environment to empower team decision making • Implement a democratic voting system • Assign project manager role of facilitator
The need for agile-compliant performance evaluation	• Ensure performance evaluations consider breadth of skills, not just depth • Ensure performance evaluations apply higher weighting for mentoring, voluntary contributions, and so on • Establish 360° feedback
Lack of agile-specific recruitment policies and suitably trained IT graduates	• Develop specific recruiting practices tailored for agile methods to hire the right people • Use team recruiting to find the right person for the team • Put newly recruited graduates on agile projects so they gain hands-on experience

department be a success does not help them much in their own reviews. (manager, company E)

So, developing team-based performance evaluation with indicators tuned to agile attributes can foster team collaboration and use of agile practices. For example, three companies developed a team-based bonus program. To make team-based performance evaluation more effective, team members can act as both evaluators and those being evaluated. Six companies introduced *360-degree feedback*, in which all team members evaluate one other (as opposed to managers appraising subordinates), thus capturing voluntary contributions and mentorship.

Recruiting Challenges

A lack of agile-specific recruitment policies makes it difficult for most companies to find the right people for agile development. A manager in company G described this challenge succinctly:

The policies that we use in recruiting people do not really take into account

RELATED WORK ON "PEOPLE" CHALLENGES

Previous research has addressed some of the challenges we identified in the main article. However, that article is the first to bring them all together in one place.

Michael J. Gallivan and his colleagues considered the need for developers to be masters of all trades in a traditional development environment.[1] Mike Cohn and Doris Ford looked at this challenge during the introduction of an agile process to an organization.[2]

Andrew Begel and Nachiappan Nagappan[3] and Dirk S. Hovorka and Kai R. Larsen[4] researched the increased reliance on social skills in an agile environment.

Numerous researchers have considered the repercussions of developers' lack of business knowledge.[5–12]

Pekka Abrahamsson[13] and Ritu Agarwal and J. Prasad[14] addressed the need for developers to understand and learn agile values and principles, not just the practices.

Cohn and Ford,[2] Lucas Layman and his colleagues,[15] and Kai Petersen and Claes Wohlin[16] all reported on the lack of developer motivation to use agile methods.

Cohn and Ford[2] and Subhas Chandra Misra and his colleagues[17] studied the implications of devolved decision-making in agile environments.

Ronald L. Thompson and his colleagues[18] and Asif Qumer and Brian Henderson Sellers[19] addressed the need for agile-compliant performance evaluation.

Finally, Gallivan and his colleagues[1] and Juhani Iivari and Magda Huisman[20] discussed the lack of specific recruitment policies and suitably trained IT graduates in traditional software development environments.

References

1. M.J. Gallivan, D.P. Truex, and L. Kvansy, "Changing Patterns in IT Skill Sets 1988–2003: A Content Analysis of Classified Advertising," *ACM SIGMIS Database*, vol. 35, no. 3, 2004, pp. 64–87.
2. M. Cohn and D. Ford, "Introducing an Agile Process to an Organization," *Computer*, vol. 36, no. 6, 2003, pp. 74–78.
3. A. Begel and N. Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study," *Proc. 1st Int'l Symp. Empirical Software Eng. and Measurement* (ESEM 07), IEEE CS Press, 2007, pp. 117–125.
4. D.S. Hovorka and K.R. Larsen, "Enabling Agile Adoption Practices through Network Organizations," *European J. Information Systems*, vol. 15, no. 2, 2006, pp. 159–168.
5. J.D. Becker, R.G. Insley, and M.L. Endres, "Communication Skills of Technical Professionals: A Report for Schools of Business Administration," *ACM SIGCPR Computer Personnel*, vol. 18, no. 2, 1997, pp. 3–19.
6. H.D. Crockett and C.J. Jeffries, "Preferred Information Systems Skills: Are Undergraduate IS Programs Serving Their Markets?" *Interface*, vol. 15, no. 2, 1993, pp. 9–13.
7. S. Cusack, "Moving Up to Management," *Computerworld*, vol. 22, no. 21, 1988, p. 82.
8. D.A. Fischer, "It's 2 P.M.: Do You Know Where Your Help Desk Is?" *Infoworld*, vol. 16, no. 24, 1994, p. 67.
9. C. Hildebrand, "Managing the Aftermath: Orchestration Skills Needed," *Computerworld*, vol. 25, no. 31, 1991, p. 58.
10. R.A. Pastore, "Decided Slant," *CIO*, vol. 7, no. 4, 1993, pp. 50–56.
11. B. Pitman, "Stop Wasting Training Dollars: Train for Outcomes, Not Outputs," *J. Systems Management*, vol. 44, no. 6, 1994, p. 25.
12. L.S. Ridgeway, "Read My Mind: What Users Want from Online Information," *IEEE Trans. Professional Communications*, vol. 30, no. 2, 1987, pp. 87–90.
13. P. Abrahamsson, "Commitment Nets in Software Process Improvement," *Annals Software Eng.*, vol. 14, nos. 1–4, 2002, pp. 407–438.
14. R. Agarwal and J. Prasad, "A Field Study of the Adoption of Software Process Innovations by Information Systems Professionals," *IEEE Trans. Eng. Management*, vol. 47, no. 3, 2000, pp. 295–308.
15. L. Layman, L. Williams, and L. Cunningham, "Motivations and Measurements in an Agile Case Study," *J. Systems Architecture*, vol. 52, no. 11, 2006, pp. 654–667.
16. K. Petersen and C. Wohlin, "A Comparison of Issues and Advantages in Agile and Incremental Development between State of the Art and an Industrial Case," *J. Systems and Software*, vol. 82, no. 9, 2009, pp. 1479–1490.
17. S.C. Misra, V. Kumar, and U. Kumar, "Identifying Some Important Success Factors in Adopting Agile Software Development Practices," *J. Systems and Software*, vol. 82, no. 11, 2009, pp. 1869–1890.
18. R.L. Thompson, C.A. Higgins, and J.M. Howell, "Personal Computing: Toward a Conceptual Model of Utilization," *MIS Q.*, vol. 15, no. 1, 1991, pp. 125–143.
19. A. Qumer and B. Henderson-Sellers, "A Framework to Support the Evaluation, Adoption and Improvement of Agile Methods in Practice," *J. Systems and Software*, vol. 81, no. 11, 2008, pp. 1899–1919.
20. J. Iivari and M. Huisman, "The Relationship between Organizational Culture and the Deployment of Systems Development Methodologies," *MIS Q.*, vol. 31, no. 1, 2007, pp. 35–58.

agility. I do not even know how we should do it.

Further compounding the problem, few third-level institutions (that is, colleges or universities) significantly incorporate agile methods and skills. For instance, a manager in company L said,

We cannot seem to find any graduates who have done anything hands-on or even gone beyond one or two lectures on agile methods.

Additionally, degree programs tend to lean heavily (if not entirely) toward intense technical or business skills but rarely incorporate both.

Three companies developed agile-compliant recruiting practices. At company L, job applicants must refactor a piece of code and develop a set of user stories and acceptance tests based on an interview with a fictional customer. Company A monitors applicants during a two-hour "iteration" documenting user stories, estimating, prioritiz-

ing, developing, and refactoring. The company tests them with a stand-up meeting after one hour and a retrospective after two. This mode of recruiting quickly exposes an applicant's lack of technical and social skills. Company I actually drops an applicant into a live team of developers, who then evaluate the applicant's performance.

Organizations can use our findings for a variety of purposes (see Table 4). For example, companies considering whether to implement agile methods can assess challenges they might experience. Companies that already use agile methods can determine what problems they might be encountering. Such an exercise can be insightful, given that many of the problems we identified exist under the radar or act as silent killers. The best practices for overcoming these challenges could provide a starting point for developing a recruiting or training strategy. This is particularly appropriate for organizations transitioning to agile methods. Such practices can reduce or at least expose people challenges, but they're unlikely to remove them altogether.

Managing people challenges is more of an art than a science; the problems' source could be the organization, the project, the team, or an individual. No technique can solve all the problems.

Also, some organizations might not be in a position to implement all our recommendations, owing to cost, cultural issues, organizational-structure limitations, or a variety of other reasons. Some challenges might be largely outside their control, a key example being the lack of university graduates.

Our case studies were limited because the respondents typically held managerial positions; conducting similar studies with developers might prove interesting. Identifying contrasts and conflicting opinions between developers and managers and the reasons for such opinions could prove insightful.

Not all the challenges we raised are new; they're just exacerbated in an agile environment. Many have plagued project managers, human resources staff, and trainers for many years. See the sidebar for previous research on these challenges.

Acknowledgments

A Science Foundation Ireland grant (3/CE2/I303_1) to Lero partly supported this research.

References

1. C.H. Tan and H.H. Teo, "Training Future Software Developers to Acquire Agile Development Skills," *Comm. ACM*, vol. 50, no. 12, 2007, pp. 97–98.
2. C. Poole and J.W. Huisman, "Using Extreme Programming in a Maintenance Environment," *IEEE Software*, vol. 18, no. 6, 2001, pp. 42–50.
3. J. Drobka, D. Noftz, and R. Raghu, "Piloting XP on Four Mission-Critical Projects," *IEEE Software*, vol. 21, no. 6, 2004, pp. 70–75.
4. D. Jamieson, K. Vinsen, and G. Callender, "Agile Procurement and Dynamic Value for Money to Facilitate Agile Software Projects," *Proc. 32nd EuroMicro Conf. Software Eng. and Advanced Applications* (EuroMicro 06), IEEE CS Press, 2006, pp. 248–255.
5. "DEEWR Tender Win," *UXResearch*, Sept. 2008; www.uxresearch.com.au/news/deewr-tender-win.
6. S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of Migrating to Agile Methodologies," *Comm. ACM*, vol. 48, no. 5, 2005, pp. 72–78.
7. P. Schuh, *Integrating Agile Development in the Real World*, Charles River Media, 2004.

ABOUT THE AUTHORS

KIERAN CONBOY is a senior lecturer on information systems at the National University of Ireland, Galway. His research interests include agile methods for systems development and other disciplines, systems analysis, and management accounting in systems development projects. Conboy has a PhD in Information Systems from the University of Limerick. Contact him at kieran.conboy@nuigalway.ie.

SHARON COYLE lectures on information systems at the National University of Ireland, Galway. Her research interests include agile systems development, project management, and group decision-making. Coyle is a doctoral researcher in information systems at NUI Galway. Contact her at sharon.coyle@nuigalway.ie.

XIAOFENG WANG is a research fellow at Lero—the Irish software engineering research centre. Her research interests include the software development process and methods, agile software development, and complex-adaptive-systems theory. Wang has a PhD in Information Systems from the University of Bath. Contact her at xiaofeng.wang@ul.ie.

MINNA PIKKARAINEN is a senior research scientist with the VTT Technical Research Centre of Finland. Her research interests include agile development, software innovation, and software variability management. Pikkarainen has a PhD in agile development from the University of Oulu. Contact her at minna.pikkarainen@vtt.fi.