IOSR Journal of Computer Science (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727
PP 01-08
www.iosrjournals.org

Challenges of Software Development Team in Agile Development

Syed Rehan[1], Manish Assudani[2], Ritesh Shrivastav[3] Swapnil Deshmukh[4]
[1,2,3,4]*(Computer Science & Engg. Department, AnjumanCollege of Engineering& Tech./R.T.M Nagpur University, India)*
email: sayedrehan_1@yahoo.com

ABSTRACT :*There is a common perception that, while there may be some 'teething' problems experienced during the initial transition to agile, people are much happier, engaged and ultimately more productive in these environments. This study shows that this belief may not always hold true, identifying many serious 'people' challenges experienced by large multinational organisations, all using agile for years. The cases provide an interesting insight in that they involve instances where agile was implemented in a top-down manner across the organisations or at least across business units. This is in contrast to most accounts of agile which involve voluntary, bottom up adoption on small co-located teams developing systems deemed to be suitable for agile development. The people issues uncovered include a broad range of problems from recruitment of agile staff, to training, motivation and performance evaluation among others. The paper will conclude with a set of actionable recommendations as to how organisations can overcome these challenges, based on the better practices uncovered in the cases studied.*
Keywords: *agile, adoption, methodologies, people factors, Whiteboards*

I. INTRODUCTION

While agile methods have been in use for quite a while, there are a number of reasons why it is important to examine the 'people issues' implications of utilising agile approaches in this context. Firstly, the growing popularity of agile methods is clearly evident and they "are fast becoming the adopted methodology commercially". Secondly, the boundaries of agile are now changing, no longer restricted to small co-located teams and increasingly applied in environments outside of their 'comfort zone' , thus presenting new people and human resource management challenges. Finally, in the early years, the decision to adopt agile was typically an insular, bottom up, voluntary one, where the project team could decide to embrace or rebuke the transition 'on its own terms'. Increasingly, suppliers, consultants, partners and customers and even public sector bodies are coercing the use of agile, either through a formal requirement to do so, or through necessity to ensure inter-organisational process alignment. The increasing prevalence of agile approaches, the lowering of traditional agile boundaries and growing pressure to adopt agile, all contribute to the need for human resource departments and project managers to address any associated skill and people challenges. An analysis of the literature e.g. Nerur, Mahapatra et al. (2005) and Schuh (2004), shows that agile environments are significantly different in context to environments where more traditional approaches are used (Table 1), although very often the distinction between the two is not so black and white.

For all of these reasons, there is a need to identify the problems that the agile transition may cause. While numerous studies document challenges in isolation, this study builds a comprehensive list of the most important challenges and develops a set of recommendations for how these might be addressed.

Project Component	**Traditional**	**Agile**
Control	Process centric	People centric
Management Style	Command-and control	Leadership-and-collaboration
Knowledge Management	Explicit	Tacit
Role Assignment	Individual – favours specialisation	Self-organising teams – encourages role interchangeability

Communication	Formal and only when necessary	Informal and continuous
Customer's Involvement	Important, usually only at the analysis phase of the project	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 Organisational Form/Structure	Mechanistic (bureaucratic with high formalisation)	Organic (flexible and participative encouraging cooperative social action)
Technology	No restriction	Favours object-oriented technology
Team Location	Predominately distributed	Predominantly collocated
Team Size	Often greater than 10	Usually less 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 available
Documentation	Substantial	Minimal

Table 1: Contrasting differences between Traditional and Agile Approaches (Adapted from Nerur, Mahapatra et al. (2005) and Schuh (2004))

II. THE RESEARCH PROCESS

A two-phased approach was used in this study. Firstly, we conducted focus group discussions with software development executives, senior project managers and agility experts In addition to identifying an initial set of challenges, this phase acted as a test bed to evaluate the case study protocols used for the second phase. In the second phase we conducted case. The cases include organisations that have embraced agile development very effectively harvesting benefits such as reduced costs, higher quality systems and more satisfied software development staff and customers. The studies also include some organisations that have experienced significant problems and even project failures directly attributable to the agile transition. We intentionally selected cases with such opposing experiences, which allowed us to compare and contrast, thus identifying the distinguishing skills and challenges related to agile adoption. We now present the key people challenges identified across the cases, along with practices uncovered to address these challenges. Where possible, we also try to identify how prevalent each issue was across the cases studied. For example, we may indicate how 5 of 17 cases encountered a certain issue while 12 of 17 may encounter another.

III. KEY PEOPLE CHALLENGES EMERGING FROM THE STUDY

3.1 Developer fear caused by transparency of skill deficiencies

Developer fear caused by the transparency of skill deficiency was noted across many companies studied. Interviewees outlined how procedures such as stand up meetings, an on-site customer and the use of storyboards and whiteboards made developer shortcomings very visible to the rest of the team, since these practices require direct and constant communication and collaboration among team members. For example, storyboards track the status of user stories and make a developer's lack of progress very obvious. Whiteboards (used by agile teams to communicate design issues), can also highlight the deficiency of technical and/or communication skills of any one developer since they need to present their ideas in front of their peers on a regular basis. In addition, continuous integration and automated testing means that developers cannot hide poor, low quality code. Exposing weaknesses of developers however can often be counter-

productive. Eight teams spoke of incidents with developers having low self-esteem who even if performing reasonably well, were often made to feel inadequate in such a transparent environment. At the other end of the spectrum, full transparency created unhealthy environments in some companies where egotism was present and *"exhibitionists"* (Consultant, P), *"show-offs"* (Manager, L) and *"bullies"* (Consultant, P) were involved. Repercussions of this included discomfort among some developers, hostility among developers (7 of 17 cases), and developers leaving the organisation (5 of 17 cases) or at least changing teams (14 of 17 cases). It is too simplistic to say that these involved 'weak' developers. In fact, a very interesting finding was that *"weakness is relative"* (Manager, L), and that some highly respected and high performing developers were made to feel inadequate by 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 on a fortnightly basis where they could document any fears, issues or concerns they felt inappropriate for discussing in an open forum. In Company D, listing problems at stand up meetings was voluntary for new junior developers on one large project studied. In Companies B, D and M, junior or new staff had a separate, lengthier stand up meeting with dedicated mentors. Developers also need to be assured that they can get help to improve their skill set. In at least nine cases, pair programming was used where weaker developers were paired with those more experienced, and thus joint responsibility dissolved transparency of any potential weaknesses.

3.2 The need for developers to be a 'master of all trades'

Across all companies, it was found that boundaries between developer roles seem to be less clear in an agile environment and that it was important that developers were competent in a broad range of skills as opposed to being an expert 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, M).

As one manager in Company D described, rather than being a *"jack of all trades, master ofnone"*, a developer in an agile team needs to be a *"master of all trades"*. This multi-facetedskill set caused numerous problems. Firstly, almost all project managers found it difficult to find developers that displayed all of the skills necessary for agile, either externally or within their respective organisation. Training was also found to be more difficult. In four cases, management sent all developers on the team to all training courses, incurring high expense. In all four of these instances, prior to the adoption of agile, developers would only have received training in niche areas directly related to their narrow team role. The fact that agile encourages blended roles, is dependent on voluntary contributions and emphasises team as opposed to individual performance, means that team members may become a 'jack of all trades' but lack the opportunity to hone a smaller number of key skills e.g. Java certification. As a result, in the cases studied, some team members felt they were being disadvantaged when competing for promotion or jobs in the marketplace.

3.3 Increased reliance on social skills

Agile practices such as co-location, an on-site customer, stand up meetings, retrospectives and pair programming were all commonly cited examples that increase social interaction, thus heightening the need for social, communication and presentation skills. While the development of social skills was generally seen by all as positive, some interesting concerns and problems were raised through an analysis of interviewee responses. Firstly, it was evident across the majority of cases (15 of 17) that some personnel were technically very talented but inherently weak in terms of communication and presentation skills.

The customer-facing aspect of agile also caused significant problems in eight companies. It was clear that certain people *"you should never, ever put them in front of a client"* (Director, M). In fact *"being a good communicator is one thing. Knowing what not to communicate ismuch more important"* (Manager, O). Managers cited examples of developers revealing tocustomers politically sensitive and confidential information regarding contracts, salaries and opinions regarding weaknesses within the development team. An intriguing finding of this study is that, although both technical and social skills are required in an agile team, developers with strong social skills might be disadvantaged when they are recruited in a global software development context.

3.4 A lack of business knowledge among developers

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, and so, according to many of those interviewed, an absence of basic domain knowledge among developers becomes very obvious

Quite a few managers spoke of the potential damaging, *"cancerous effect"* (Manager, L) of this problem, citing examples of customer indifference and disengagement because of the resulting perception that *"the team know nothing about our business so they won't deliveranything of value to our business"* (Manager, M). This was regarded as a challenge by 12 ofthe 17 companies studied and seemed to particularly problematic in seven companies where internationally distributed teams were involved. For example, one manager in 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 was seen as one possible solution. some companies held training sessions on basic topics within the problem domain, such as accounting standards, basic management accounting and finance and marketing principles. Typically, such training went some way to addressing the issue but usually failed to consider the client-specific knowledge required. Getting the customer organisation to run the training seemed to be quite effective in solving this problem (2 of 17).

Moreover, almost all companies were making an attempt to resolve the root cause of the problem by recruiting staff and graduates with a combination of IT and business knowledge.

3.5 The need to understand and learn values and principles of agile, not just the practices

It was evident in at least ten cases that while 'on paper' they were implementing agile practices, the ultimate goals of agility were not being achieved. Company O in particular captures this point. Two teams in Company O implemented agile at the same time, participating in the same three day agile training session. As can be seen from Table 2 below, both teams implemented the stand-up meeting and on-site customer practices but it is clear that, while they technically implemented the same practices, they did not achieve the same underlying goals. According to a manager in Company O, there was no single issue that caused such a difference between the two projects, but rather *"some intangible combinationof staff personality, management style, cultural issues and other factors".*

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 Non-threatening
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 ups, 6 of 14 retrospectives	Created brainstorming sessions Constantly hassled other stakeholders (R&D, manufacturing, accounting etc) and organised meetings continually Real-time involvement, live reprioritisation Attended 43 of 45 stand ups

Table 2: Contrasting Implementation of Agile Practices

Although formal training is seen as a typical solution to teach agile practices, it is not sufficient for development teams to adequately embrace agile values and principles. Some procedures identified across the companies studied included the provision of training and attendance at agile conferences focusing on values and principles. With regard to training, continuous and hands-on training was preferable to once-off training as a way to help developers absorb and retain agile values and principles where a manager in Company L claimed how "*the real value came from continuous training*".

In addition, coaching can complement training to assist a team along the journey of agile-transition. Generally across at least ten of the companies studied, senior team members played the role of coach, whose primary goal was to drive the effort of retaining agile values and principles within the team. However, the effect of coaching can also be obtained through swapping developers across agile teams which ensures cross-team observation and validation of agile practices, thus identifying "*bad habits*" (Company D). Periodically assessing the agility of a team using an assessment framework based on a set of agile goals as opposed to practice adherence may also help. Company A had adapted and dropped several agile practices as a result of assessment practices.

3.6 Lack of developer motivation to use agile methods

A lack of motivation to use agile methods among developers was a challenge encountered by five companies studied. It was more prominent in companies where agile methods were adopted in a top-down manner. A manager in Company G observed that *"sometimes theyhave the competence but are not convinced it (agile) will work".* In many of the cases studied,there was a perception that process innovations like adopting agile are often viewed as overly onerous, complex and time consuming.

In some organisations, mechanisms such as strong people involvement in the adoption process (2 of 17), training (8 of 17) and sharing of agile development experiences (2 of 17) were already in place to convince and motivate developers to adopt and use agile methods. A manager of Company G indicated how they continuously collected information regarding successful agile projects, in the form of multiple experience reports and then shared them among different project teams. Five companies collected experiences from different teams and customers and have gained valuable insights from them. According to various respondents, the sharing of agile 'success stories' provided encouragement and belief.

3.7 Implications of devolved decision-making

While devolved decision-making is a commonly cited aspect of agile, it has caused significant problems among the companies studied: *"People were picking tasks they shouldn't have. Itwas self-organising gone mad"* (Manager, L). Devolved decision-making also means achange for project managers in some cases causing problems as "*project managers do notknow what their role is*" (Manager, N). In Company L the manager cited anxiety of losing thetraditional 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. A practice discovered in two companies is a 15-minute meeting between individual developers and the manager every week to ensure that all developers have ample opportunity to communicate any views or concerns they have which may be difficult for them to express in an open forum.

3.8 The need for agile-compliant performance evaluation

Across the seventeen cases studied, it was found that while agile methods advocate people interaction, collaboration, mentoring, teamwork and transferring knowledge, there are many issues associated with the performance evaluation of these activities. Team collaboration is not easy to implement if performance evaluation and appraisal mechanisms are based on individual performance

In five cases the criteria for performance evaluation (particularly at junior levels) focused on technical skills and the ability to follow direction whereas distinguishing factors in agile development, such as social skills, creative thinking and self-organisation, were neglected. In other instances, agile teams were largely evaluated according to traditional criteria and so according to those interviewed, results of performance evaluation were often not indicative of the true abilities of the team members. Meanwhile, performance evaluation of the on-site customer seemed particularly problematic and highly contentious. In at least four instances, the actual person doing the job felt aggrieved that they were not being rewarded properly:

Developing team-based performance evaluation with indicators tuned to agile attributes, therefore, can foster team collaboration and use of agile practices. For example, 3 of the 17 interviewed companies have developed a bonus program that is team-based rather than rewarding individuals. To make team-based performance evaluation more effective, team members can act as evaluators as well as being evaluated. Six companies have introduced "360º feedback" where all team members evaluate each other, as opposed to managers appraising subordinates, thus ensuring that voluntary contributions and mentoring are captured in the appraisal.

3.9 Lack of agile-specific recruitment policies and suitably trained IT graduates

D ue to a lack of agile-specific recruitment policies in place in most companies it can be difficult to find the right people that are needed for agile development. A manager in Company G described this challenge succinctly: *"The policies that we use in recruitingpeople do not really take into account agility. I do not even know how we should do it?"* Theissue can be worsened by the fact that very few third level institutions incorporate agile methods and skills to any significant degree
3 of the 17 companies developed agile compliant recruiting practices. For example, in Company L job applicants are required to refactor a piece of code and develop a set of user stories and acceptance tests based on an interview with a fictional customer. In

II. RECOMMENDATIONS & CONCLUSIONS

A key output from this study is a set of practices that effective organisations are using to overcome the challenges identified in this study. These are discussed throughout this article but are summarised below. While success is a hard thing to measure, all of the practices below were implemented effectively in at least one organisation, according to the respondents in each respective case. Many of these were implemented effectively in multiple cases.

Challenges	Recommendations
Developer fear caused by transparency of skill deficiencies	Feedback outside stand ups, allowing the documentation of any fears, issues or concerns inappropriate for discussion in open forum Stand up meetings voluntary for new juniordevelopers Dedicated mentor for new staff Weaker developers paired with those who had more experience, taking joint responsibility for requirements
The need for developers to be a 'master of all trades'	Use pair programming and pair rotation to distribute knowledge and facilitate learning Encourage task self-assignment to allow developer work in different areas and learn new skills Reintroduce specific roles when it is perceived beneficial to teams with e.g. large team size, conflicts between developers
Increased reliance on social skills	Combine development and training program to provide customised training materials on social skills, using developers' own examples. Using proper documentation to back upcommunication
A lack of business knowledge among developers	Customer company runs training sessions on basic topics within the business domain and on company specific area(s) Provide small training modules (on a frequent basis), making it interactive to allow developers acquire niche business knowledge required by the project Recruit staff and graduates with a combination of IT and business knowledge
The need to understand and learn values and principles of agile, not just the practices	Ensure multiple members get agile training or attendagile conferences Agile coaching and championing Ensure cross-team observation/validation of agile practices Assess agility in terms of agile values not practice adherence
Lack of developer motivation to use agile methods	Try to have multiple 'bought-in' developers on each team Collecting and sharing successful adoption stories and positive experiences
Implications of devolved	Build a sharing and learning environmentto empower team decision-

decision-making	making Implement a democratic voting system Project manager plays the role of facilitator
The need for agile- compliant performance evaluation	Performance evaluation needs to consider breadth of skills, not just depth Performance evaluation to apply much higher weighting for mentoring, voluntary contributions etc
	360° feedback a must
Lack of agile-specific recruitment policies and suitably trained IT graduates	Develop specific recruiting practices tailored for agile methods to hire right people Use team recruiting to find right person working in the team Put newly recruited graduates on agile projects to get hands on experience

Table 3: A summary of people challenges and recommendations to address them

We believe that the findings from this study can be used for a variety of purposes. The list of issues can be used by organisations to assess the issues that they may be susceptible to when considering whether or not to implement agile, or to determine what problems they may be encountering if agile is already in place. This can be a very insightful exercise, given many of the problems identified in this study exist 'under the radar' or are, as referred to by one manager, acting *"as silent killers".* The best practices used to overcome the challenges (listed above) could be used as a starting point for developing a recruitment or training strategy. This would be particularly appropriate where an organisation is undertaking a transition to agile. It is important to note that such practices may reduce or at least surface people issues, but it is unlikely to remove them altogether. It was clear from the many interviews conducted that the management of people issues is an art more than a science, that the source of the problem can be the organisation, the project, the team, or the individual and there is no technique that can solve all problems. Also, it is clear that some organisations may not be in a position to implement all of the recommendations due to cost, cultural issues, organisational structure limitations or a variety of other reasons. Of course some of these issues may be largely outside their control, with the lack of university graduates being a key example. Also, the study was limited in that those interviewed were typically in managerial positions and so it may be interesting to conduct a similar study ascertaining the views and experiences of developers. Identifying contrasts and conflicting opinions between developers and managers and reasons for those opinions may be insightful.

While exacerbated in an agile environment, not all of the issues raised in this study are new and in reality many have plagued project managers, HR staff and trainers for many years.

REFERENCES

[I] Nerur, S., R. Mahapatra, et al. (2005). "Challenges of Migrating to Agile Methodologies." Communications of the ACM 48(5): 72-78.

[II] Schuh, P. (2004). Integrating Agile Development in the Real World, Charles River Media Inc. Tan, C. H. and H. H. Teo (2007). "Training Future Software Developers to Acquire Agile

[III] Development Skills." Communications of the ACM 50(12): 97-98.UXResearch September (2008). "DEEWR Tender Win." From