ABSTRACT

Organizations are transitioning from using the Project Management Body of Knowledge (PMBOK) practices to Agile Project Management (APM) practices. While there are many anecdotal and experiential accounts of issues involved during the transition, there is very little empirical research that systematically investigates those issues. This study relies on an exploratory case study to account, albeit to a small extent, for the gap in the extant literature. Based upon our analysis, we identify certain organization wide and team specific issues when companies adopt APM practices. We conclude with certain lessons learned in the team autonomy, top management support and training of key personnel.

KEYWORDS: Agile Project Management, Scrum, Sprint, Autonomy, Exploratory Case Study

INTRODUCTION

IT investments worldwide are expected to reach a high of $4 trillion in 2015 (Gartner 2012 Q4 Report). A growing number of software developers are turning towards Agile Development Methodology (ADM) for help (up from 35.4% in 2009 to 38.6% in 2010) (West, Gilpin, Grant and Anderson, 2011). More and more organizations are moving towards management by projects (Fernandez and Fernandez, 2009; Sauer and Reich, 2009). While 32% of the projects were successful, a staggering 68% of the software projects were either doomed to fail or severely challenged (CHAOS Report 2009). It is impossible to ignore that a large percentage of software projects do fail. Understanding how software projects are managed is, therefore, of paramount importance.

With the advent of Agile Project Management (APM) practices, academicians and practitioners are reexamining the traditional project management practices as recommended by Project Management Body of Knowledge (PMBOK). The bedrock of the traditional plan driven methodologies is the engineering model. Rigorous processes move the components through well-defined phases from requirements definition to implementation and produce a deliverable (Boehm and Turner 2003). The emphasis is on detailed upfront planning, rigid processes, heavy documentation and stringent contracts over a long time frame. These heavyweight methodologies rely on planning elements including the roles that are played, activities that are performed, and artifacts that are produced in an effort to deliver quality software product at a lower cost on a timely basis (Meso and Jain, 2006). Distributed software development projects have become quite popular, due to a variety of reasons, such as, untapped geographical markets, client vicinity and availability of a large pool of highly skilled manpower (Herbsleb and Grinter, 1999). The plan driven methodologies geared towards supporting such projects were fine tuned to incorporate rigid controls over the people and processes involved in software
Managers of IT projects are often confronted by three enormous challenges, namely, change, complexity and diversity (Jackson, 2003). Markets, technologies and the society are rapidly changing. From the era of standalone computing, we are swiftly moving to mobile computing. Information systems too have to keep pace with this state. The requirements that drive the development of information systems are now changing at what is called the “Internet speed” (MacCormack, Verganti, and Iansiti 2001, Baskerville, Ramesh, Levine, Pries-Heje, and Slaughter 2003). To combat these forces, several leading savants of the software industry came together and put forward the Agile Manifesto. Agile development emphasizes continuous collaboration between the development and the user communities, short iterations, quick planning, frequent validation and feedback from the users and production of working code over a short time frame in as little as two weeks. Proponents of the ADM argue that it is working code produced in a timely fashion that will tackle the enormous demands of the rapidly changing business and technical environment.

The traditional plan and control driven project management methodologies do not work well with the newer ADM because of many reasons. Firstly, ADM is more execution centric. The focus is on repeatedly executing a series of iterations with rapid feedback in order to elevate the status of the developer as a capacity constrained resource (Anderson, 2003). Another key aspect to note is that ADM teams are self-organizing and self-regulating. They manage their workload, and there is very little upfront detailed planning. Users and developers collaborate extensively. There is a heavy emphasis on learning on the go. In ADM, testing is integrated into development, whereas in the waterfall methodology testing is often relegated towards the end of the process. Agile teams have generalizing specialists who accept interchangeable roles, shared leadership and conditions that facilitate emergence. Teams that adopt the waterfall methodology, on the other hand, are staffed with specialists, and are often led in a command and control mode. The traditional project management methodology of detailed planning and control does not emphasize the above aspects and hence do not work well with ADM. APM practices are increasingly becoming very popular, so much so that the Project Management Institute (PMI) recently introduced a certification program in its curriculum. While organizations are transitioning to APM, several projects are still using Traditional Development Methodology (TDM) and are staffed with people who have strong expertise in TDM. This causes a palpable tension, which needs to be understood and managed in order to facilitate organization wide adoption of APM. The goal of this research is to study APM as practiced in industry to enrich our understanding of APM adoption, and draw valuable lessons regarding how organizations are coping with the transition to APM.

The remainder of the paper is organized as follows. The extant literature on traditional and agile project management is reviewed in the next section. It is followed by a description of the research methodology used in this study. The findings are then presented. The paper is concluded with a discussion of the lessons learned.

LITERATURE REVIEW

According to the PMI, a project is “a temporary endeavor undertaken to create a unique product or service” (Meredith and Mantel, 2003). Project management is the use of tools and techniques to ensure that available resources are applied properly to accomplish the objectives of a project, within the constraints of the so called Iron triangle i.e. cost, time and scope and their impact on
quality (Oisen, 1971). The PMBOK has been built around the foundation of the transformation view of production and the theories of management, such as, management-as-planning, the thermostat model of control and the dispatching model of execution (Fernandez and Fernandez, 2009). PMBOK comprises different areas, which include planning, estimation (e.g. schedule, cost, and budget), scheduling, resource allocation, monitoring and control, quality management and risk management (Gray and Larson, 2007). Kwak and Anbari (2009) have recently conducted a review of the top journals of management and allied disciplines in order to explicate how project management has evolved as an academic discipline and as a field of practice. A notable finding of the paper is that the coverage of project management in the Information Technology/Information systems arena has increased by leaps and bounds over the past five decades. IT projects are not simple entities.

The APM practices led by Scrum emphasize a leadership-collaborative style of management where the customers take an active part. Scrum emphasizes how a project is controlled and includes daily standup meetings to discuss what developers are working on, what they have completed and any issues with their tasks (Anderson, 2003). Scrum enables collaborative decision making in teams and encourages adaptability. The objective is to have a cooperative relationship among all stakeholders of the project (Highsmith, 2003).

PMBOK set of practices allow the project resources and time to vary. The scope is fixed at the beginning of the project. On the other hand, APM treats the scope as a variable allowing for the requirements to change as the project is executed. It keeps a tight control over the project resources and time (Owen, Koskela, Henrich and Codinhoto, 2006).

Information Systems are a complex conglomerate characterized by both technical and organizational aspects (Zmud and Cox, 1979). The earliest foray into the project management area in the IS arena was made by Zmud, who studied the management of large software projects and espoused that social processes are deeply involved in the system development life cycle (Zmud, 1980). While a large body of research within the IS discipline has investigated various aspects of traditional project management practices as espoused by PMBOK (see for example, Yoon, Guimaraes, and O’Neal, 1995; Wixom and Watson, 2001), the extant literature has only a few articles on APM.

Here we summarize a few articles on APM adoption that appeared in leading conferences. Pries-Heje and Pries-Heje (2011) use the theories of boundary spanning, social capital and coordination to understand why Scrum is useful as a management practice. The paper highlights the role played by social integration, trust, coordination and collaboration in distributed software development. Several issues are highlighted in a single case study as to how the Scrum practices are adapted in a software development team (Schrodl and Wind, 2011). The difficulties faced by the software development team as they transition from Waterfall model to Scrum is attributed to the preparation of the project environment, effective communication and optimal team structure (Flouri and Berger, 2010). In this case study, we identify two teams that are different in several characteristics and examine how APM practices are adapted. We also contribute to the body of knowledge by identifying certain issues that pertain to the entire organization and highlighting several valuable lessons learned.

The research method used in the study is presented in the next section.
RESEARCH METHODOLOGY

A case study research method was used (Yin 2003). This methodology enables the researcher to investigate a phenomenon in its natural setting. Our focus on this study is to answer the “Why” and “How” questions. Our intent in this study is to explore how organizations practice APM and identify issues raised by the transition from PMBOK to APM.

Site Selection

We selected a single organization for the purpose of this study. The study was conducted in November 2011. We will call it Company A, which is a leading provider of insurance services. It is highly regarded in its industry sector. The unit of analysis was the project group. Senior management introduced agile development practices a decade ago and APM practices were introduced in 2009. Scrum (Schwaber, 2009) was chosen as the APM methodology. The development environment included collocated teams in the US office of the company and offshore teams located in India. An onsite coordinator at the local office acts as the liaison between onshore and offshore teams.

Data Collection

Key knowledge management areas in project management were identified (Gray and Larson, 2007) and a semi-structured questionnaire was developed to be used for data collection. This questionnaire was thoroughly vetted before being used. Two leading experts in project management looked at each item in the questionnaire and verified that the questions were relevant to areas of interest. We dropped several questions, included new questions and modified some questions after this process. The areas covered by the questionnaire included knowledge management areas such as budget planning, cost estimation, expense monitoring, schedule planning, monitoring, scope creep and risk management. Data was collected through onsite and phone interviews. While we were guided by the questionnaire, open ended questions were frequently used to tease out peculiarities of project management practices employed by each team. In each team, we interviewed the Project Manager, Scrum Master, Product Owner and several Developers. In addition, we also interviewed the Chief Information Officer (CIO), the Chief Operations Officer (COO), the head of the Project Management Office (PMO), and several business owners. A total of 14 interviewees were interviewed. The interviews were recorded using a voice recorder. The interviewees were clearly told that their responses would be kept in strict confidence, and their explicit permission was obtained before proceeding with the interviews. The data was collected over a period of 6 months. In addition to the interviews, we had the opportunity to observe Scrum meetings of various teams. We also gathered information from published sources.

DATA ANALYSIS AND FINDINGS

The transcripts from the interviews served as the primary source of data. The analysis was carried out in an iterative manner to identify interesting patterns. The analysis revealed that while Scrum was mandated to be implemented in the entire organization, there were distinct differences among development teams regarding the extent of adoption and/or adaptation of Scrum practices. Here we report the experience of two teams, with contrasting characteristics, to highlight how adoption of APM may vary within the same organization. We also discuss organization wide issues that surfaced due to the adoption of APM.
Organization Level Issues in APM adoption
The analysis revealed certain issues that covered the entire organization.

Need to measure and spread word of progress
APM practices were introduced in 2009. We could discern from the interviews of the CIO and CEO, that APM was introduced in order to increase the accountability of the different teams involved in the development of software. However, the interviews with several respondents revealed a sense of restlessness with respect to the success of APM practices. No metrics are used to determine whether the practices worked over time. Also there is no way to determine the relative success of these practices across teams. It is clear that all the interviewees do feel that APM practices has instilled a sense of accountability in the different teams and that it has not been a futile exercise. We argue that surveys of team members can be used to disseminate the details about the benefits realized across the organization.

Role of Leadership
As per the CIO, “We are faced with the danger of falling back on old habits”. In Company A, the introduction of APM practices were helped by certain serendipitous events. In 2009, the new Project Management Office (PMO) head was an ardent champion of the new practices and espoused them wholeheartedly. In addition, the old defunct Project Review Board was revived, giving it additional powers which enabled the company to introduce these new practices. Now, the organization is bringing in another head of PMO who is rather lukewarm to the potential of APM. New practices, such as those envisaged under APM, are often radically different from those that were used before, and therefore their assimilation takes a longer time. It is very important that top management support be extended to these efforts over a long period of time. The governance mechanism should be examined and changed in such a way that, change in top management does not have deleterious effects on the assimilation of a new methodology. Bringing in external consultants can help to some extent, but we contend that consistent top management support is of paramount importance.

Training needs of the organization
There seem to be many Product Owners and some are often at odds with the members of the development team. As per the Senior Architect and Project Manager of the Team-A - “Product Owner often sees value where there is none and does not see value where there is. The team does not like to work on unimportant issues”. At least one Product Owner took exception to the confusion caused by some of the seemingly irreconcilable differences between the ADM and APM practices. The differences between both should be clearly disseminated across the organization. The CIO of the company too stressed the need for more training for Product Owners, given the pivotal role that they play in the APM practices.

The COO of Company-A was seriously concerned about the attrition of employees from some Agile teams – “Sometimes teams fall apart. You lose a member, you change a member. A member gets a divorce and gets in the dumps. Suddenly the team is broken. There is self-governing, but they are still teams and they are still people. When it starts to fall apart, there is not a boss to go to. I mean, there is a boss but you do not want to rat out your team members you know”. In pure Scrum teams the project manager is the mentor and facilitator. The Project Manager needs to undergo intensive training in order to be comfortable with his new role.
Team Level Issues in APM adoption

Team-A is a web development team that comprises young developers. A few members are located in different cities. Team-B is a data warehousing team of senior Data Base Administrators and developers who have separate cubicles. The differences between the two teams are captured in Table-1.

<table>
<thead>
<tr>
<th></th>
<th>Team A</th>
<th>Team-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Type</td>
<td>New Development</td>
<td>Predominantly Maintenance with some enhancements</td>
</tr>
<tr>
<td>Team Location</td>
<td>Distributed with team members in different sites</td>
<td>Collocated</td>
</tr>
<tr>
<td>Team Size</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Business Requirements</td>
<td>Change Rapidly</td>
<td>Stable</td>
</tr>
<tr>
<td>Team Demographics</td>
<td>Young developers</td>
<td>Middle Aged Senior DBA’s and developers</td>
</tr>
<tr>
<td>Type of Information Systems</td>
<td>Transaction Processing</td>
<td>Decision Support Systems</td>
</tr>
<tr>
<td>Governance Structure</td>
<td>Project Manager and Scrum Master are different</td>
<td>Project Manager is the Scrum Master also</td>
</tr>
<tr>
<td>Software Development Methodology Used</td>
<td>Agile Software Development Methodology</td>
<td>Traditional Software Development Methodology</td>
</tr>
<tr>
<td>Tools Used for Management</td>
<td>Rally</td>
<td>EXCEL</td>
</tr>
<tr>
<td>Language</td>
<td>Java</td>
<td>SQL SERVER, SSIS, SSRS</td>
</tr>
<tr>
<td>Tools Used for Development</td>
<td>Fitness, Selenium</td>
<td></td>
</tr>
<tr>
<td>Workspace</td>
<td>Shared</td>
<td>Separate</td>
</tr>
<tr>
<td>Team Members are</td>
<td>Generalizing Specialists</td>
<td>Specialists</td>
</tr>
<tr>
<td>Sprint duration</td>
<td>Two weeks</td>
<td>1 Month</td>
</tr>
</tbody>
</table>

Team A uses Java for the web development. The QA process is automated. The team has recently started using several tools such as Fitness for testing. The team members start from test cases for the functionality, which forms the basis for the program code. Defects are tracked using a tool called Rally. Rally enables the team integrate the testing and development
activities. Team-B on the other hand uses tools offered in SQL SERVER for their data warehouse needs. QA is performed on a rather ad-hoc basis with very little acceptance criteria.

Both teams also differ in certain key project management areas as shown in Table 2 below.

<table>
<thead>
<tr>
<th>Table 2. Key Project Management Characteristics Team A vs. Team B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Management Activity</strong></td>
</tr>
<tr>
<td><strong>Scope Management</strong></td>
</tr>
<tr>
<td><strong>Quality Management</strong></td>
</tr>
<tr>
<td><strong>Managing Customer Involvement</strong></td>
</tr>
<tr>
<td><strong>Human Resource Management</strong></td>
</tr>
</tbody>
</table>

**Incentives**

While developers of the Team-A reported satisfaction over the pace and content of their work, they informed us that they do miss the stability of the requirements before Scrum was introduced. Team-A developer – “*We are constantly changing gears here and often have to deal with lot of variation*”. Team-A faces a dynamic business environment. We suggest that team incentives be modified to reward the teams take care of sudden and perceptible increase in work load.

**Command and Control versus Autonomy**

A key developer in Team A was visibly upset with the fact that the business was giving them solutions rather than requirements – “*Our Scrum Master constantly fights the battle of the business giving us solutions. They are supposed to give us requirements, high level stories. The Scrum master is always trying to extrapolate the underlying business need, but they always try to couch the need in form of a preconceived solution. And they may not even know they are doing it. The solution should come from the developers*”. The head of the PMO confided – “*Building the product specifications is important, the plan is what gets approved by state insurance departments. What we are asking developers is - Do exactly on what is on this piece of paper*”. One business owner pointed out that after the implementation of Scrum, the user group, project management and development need to work together. There appears to be a palpable tension between the user groups which are used to a command and control mode and
Adopting Agile Project Management Practices

the Agile teams which are autonomous by nature. We argue that user groups too need to be trained in APM practices, in order to educate them about the change in the role of developers.

Scrum not implemented in spirit

Team-B seems to have implemented Scrum reluctantly. “We were already Agile many years before Scrum was introduced” – Scrum Master. While the decision to adopt Scrum is an organization wide decision, Team-B works under the condition of stable business requirements. The nature of their work seems to justify the way they have adapted the core practices of Scrum.

Team-B takes care of scope creep by cancelling sprints. This may be attributed to the fact that the sprints are longer (30 days). When there is a spike in work because of scope creep, team members drop everything and try to work on that particular task. The original committed work suffers in the process. We note that this may also be due to the fact that this team has specialists and therefore the original work cannot be transferred to another resource. In addition, the lack of transparency of the burn-down charts and sporadic interaction with the Product Owner precludes any review of the priorities. Normally, in a truly Agile team this task will be moved to another sprint. The team members do not collaborate actively, which may further hinder the transfer of tasks.

LESSONS LEARNED

Based upon the analysis of the interview we learned many lessons, by looking at how the two teams implemented APM practices. It is clear that APM practices have been introduced by the top management with the goal of increasing accountability of the different teams. Now teams deliver software that meets requirements on a timely basis. This is a major change in the organization. While the development teams have become more agile, the user community does not seem to have embraced the change wholeheartedly. It is very important that all the teams affected by the change be aligned with each other. Although adoption of APM was an organization wide initiative, software development teams have adapted them depending upon the team characteristics such as type of information systems developed, software development methodology used and demographics of the team. The role of the Project Manager is more of a mentor or facilitator. However it was seen that team members are not able to approach the Project Manager for help. Since the roles of key players such as Product Owner and Project Manager have changed, there is urgent need for constant monitoring and training of key personnel till such practices are assimilated. There is potential for users to use iterations as an opportunity to change requirements. A well trained Product Owner can prevent such occurrences. It is also apparent that conflicts can arise between the user and developer communities regarding the degree of autonomy. We also learn that there is imminent need for the use of proper change management procedures. Sustained top management support is very crucial, so that these aspects are taken care of and new practices become entrenched.

LIMITATIONS AND FUTURE DIRECTIONS

This research paper is a case study based on a single organization and highlighted certain issues. Although these results may not be generalizable to other organizations, it still is an important step in the right direction. A follow up on this case study can be carried out in the organization in order to tease out many other factors. Future research may address model development and empirical validation. Certain promising research questions that could help...
both academics and practitioners alike include: - Does pure Scrum work for all types of projects? If not, how should they be adapted to projects with different characteristics? What are the critical success factors for APM?

CONCLUSIONS

Agile Project Management practices such as Scrum are becoming increasingly popular in the industry. This research paper takes an exploratory approach at examining the various project management areas in a leading long term insurance care provider. Through in-depth interviews of the various stakeholders in the development process, we have highlighted several important aspects of the Agile Project Management practices as practiced in different teams. The paper discusses some lessons that we learnt as a part of this case study and culminates with certain research questions.

REFERENCES


