AGILE & DISTRIBUTED PROJECT MANAGEMENT: A CASE STUDY REVEALING WHY SCRUM IS USEFUL

Lene Pries-Heje, Department of Informatics, Copenhagen Business School, Denmark, lph.inf@cbs.dk

Jan Pries-Heje, Dept. of Communication, Business & IT, Universitetsvej 1, 4000 Roskilde, Denmark, janph@ruc.dk

Abstract

Scrum has gained surprising momentum as an agile IS project management approach. An obvious question is why Scrum is so useful? To answer that question we carried out a longitudinal study of a distributed project using Scrum. We analyzed the data using coding and categorisation and three carefully selected theoretical frameworks. Our conclusion in this paper is that Scrum is so useful because it provides effective communication in the form of boundary objects and boundary spanners, it provides effective social integration by building up social team capital, and it provides much needed control and coordination mechanisms by allowing both local and global articulation of work in the project. That is why Scrum is especially useful for distributed IS project management and teamwork

Keywords: Agile, Virtual, Distributed, Project management; Teamwork; SCRUM; Boundary objects, Social capital, Articulation Theory.

1 Introduction

Future work will be done everywhere, globalization is here now (cf. Friedman, 2006). This means that work is being done by anyone who does it better, cheaper or faster. Thus *distributed project teams* and *distributed projects* will be common in the future.

Agile approaches have received much attention from both the practitioner and researcher community over the last 10-15 years. First as a novelty and later as a development approach that has become widely used in practice (Dybå and Dingsøyr, 2008). Agile approaches are often referred to as high-speed development (Ågerfalk et al., 2009); an approach for dealing with change (Conboy, 2009); and as characterized by iterative processes (Austin and Devin, 2009).

A prominent example of an agile approach is *Scrum* (Rising and Janoff, 2000, Scwaber, 2004, Sutherland and Schwaber, 2010). Based on discussions with Danish project managers in the Danish Project Management Association it may be as much as 1/3 that are using (parts of) Scrum today.

In short Scrum is an iterative approach. One 15-30 day iteration is called a *Sprint*. The customer view is represented in the role as *Product Owner*. The wished-for functionality is written as *User Stories* and prioritized in a *Product Backlog* by the Product Owner. The highest priority functionality is broken down into tasks in a Sprint Planning Meeting. Then the Sprint starts. Every day the project team meets in a *daily stand-up meeting* lasting no more than 15 minutes. The meeting is owned and enforced by a *Scrum Master*. During the meeting every team member answers four questions. (1) What did you do yesterday? (2) What are you doing today? (3) Problems encountered? (4) Innovations? The meeting takes place in front of a *Scrum Board.* This is a board with four columns: Estimated tasks from the Sprint Planning meeting to the left. In the second *in progress* column team members can move tasks that they have taken on. When they are *done* tasks are moved to the third column, and when another team member have quality assured the task is moved to the fourth column *Done Done*. Finally, when a task is finished it is registered on a BurnDown chart where you can see expected versus realized production. After 2-4 weeks the Sprint produces a deliverable of value to the customer. In the concrete the produced functionality is demonstrated to the Product Owner who then recognizes the value of the deliverable. Last but not least the team looks back and does a retrospective where they try to learn from the Sprint just passed: What worked for us? What did not work? Changes in the next Sprint?

Based on the observation that so many are taking Scrum into use, especially in distributed projects, we phrased the research question: *What is it that makes Scrum so useful?*

The remainder of the paper is organised as follows. We describe our case study and our analysis in section 2. Then follows section 3 to 5 on boundary objects, social team capital, and articulation theory; the three areas where our analysis led us to believe that Scrum provides the most value in distributed projects. Thus our conclusion in section 6 is that the reason Scrum is used so widespread is explained by these three existing theories.

2 Research method

The research methodology we adopted was a contextualized, interpretive one, using the technique of case study research (Pettigrew, 1990, Walsham, 2006). Our research can be characterized as being interpretive research in that we attempted to understand the distributed project using Scrum “phenomena” and the problems therein through the meanings that people assigned to the issues we brought up in the interviews. Thus our access to reality is through social constructions, such as language, consciousness, and shared meanings (Myers and Avison, 2002).

Danske Bank, Group IT have approximately 2500 IT people employed of which 80% is located in Denmark and 20% is located in India. Many projects have both project members from Denmark and from India. We found a project using the agile method Scrum with participants distributed across the two countries; about 15 in a Scrum team in Denmark and 8 in a Scrum team in India. For ease of reference we call the project *DELHI*.

The DELHI project started in June 2010. We interviewed the Project Manager 1st time right from the beginning of the project. As part of setting up the Indian Scrum team the Danish Project Manager planned a one-week visit to India where the purpose was partly to diffuse knowledge and skills on Scrum, partly to do some classic teambuilding. We interviewed the project manager before and after this visit to India. The 'after' interview focused on whether his expectations and plans had been fulfilled. Some months later we interviewed the Project Manager after the first two Sprints had been finished, and again after the first version was released to the customer.

Furthermore, in November 2010, we went to the DELHI site in India for three weeks. Here we interviewed the Indian part of the team (Scrum Master, Task Manager, Business Developer, Tester) for about an hour each. For both Danish and India interviews we used a semi-structured interview guide. We also observed daily Scrum meetings, the use of the Scrum Board, a daily Scrum-of-Scrum, and a so-called 'All Hands' meeting where everybody in the team from both Denmark and India were present in a video conference.

Our data analysis followed the interpretive tradition, using hermeneutics (Myers, 2009). Interview minutes and observation documents were coded and analyzed. First, we found a number of potential categories. Then a subsequent, more careful qualitative data analysis (Miles and Huberman, 1994) uncovered a number of underlying themes.

2.1 Using three existing theories for explaining findings

When analyzing we realised that three existing theories had the highest explanatory power. First, Scrum seems to provide boundary objects that can be used for improving communication between key stakeholders. ”Boundary objects are objects which are both plastic enough to *adapt to local needs* and … their structure is common enough to more than one world to make them recognizable *means of translation*” (Star and Griesemer, 1989). Second, Scrum builds up Social Capital in the team. Social capital can be defined as “the sum of the actual and potential resources embedded within, available through, and derived from the network of relationships …” (Nahapiet and Ghoshal, 1998). Social capital has three dimensions, namely structural (network and number of ties), relational (nature and quality of relationships), and cognitive (shared language and meaning) (Nahapiet and Ghoshal, 1998). Third, for multiple actors within a project to pursue a common goal they have to coordinate activities, and they have to articulate work. In other words the coordination and articulation work (Strauss, 1988) in the projects.

3 Boundary Objects and Boundary Spanners

In the concrete a systems development project will have four basic stakeholders. Users and developers of course, and besides we have user management and development management. The two latter often in the role as buyer and seller, or supplier and customer. Basically communication is needed between all four groups, and often things go wrong because of mis-communication exactly at the boundaries between the four basic stakeholders. Thus we need something plastic and adaptable, and that was exactly what we found in *Boundary Objects* (Star and Griesemer, 1989).

We found that Scrum offers three obvious boundary objects: (1) User stories, which binds together users (who can express their needs in everyday-stories on use) and developers (who can understand the story and transform it easily to design requirements). (2) Product Backlog, where a user representative can prioritize tasks, and thereby easily communicate to developers what is needed first.

(3) Visible Scrum Board and burn-down charts, where the developers easily can express what value has been delivered and where development management easily can see whether the project is on track.

For communication we also found that the two roles as product owner and Scrum master in a way worked as boundary objects. A closer look at the literature found that these two roles can be classified as so-called *Boundary Spanners* (Levina and Vaast, 2005). In the concrete Scrum can be said to have pre-defined roles whereby individuals are nominated for two boundary spanning roles: (1) *Product Owner* role, that provides knowledge from the user-world to the developer world. (2) *Scrum Master* role, that ensures that the daily stand-up meeting runs smoothly and knowledge is shared between developers and users (represented by product owner).

Thus our first conclusion is that Scrum provides five different mechanisms that work as boundary objects and boundary spanners and seems to work extremely well based on observations from the DELHI case. Figure 1 illustrates where the five mechanisms improve communication between stakeholders.

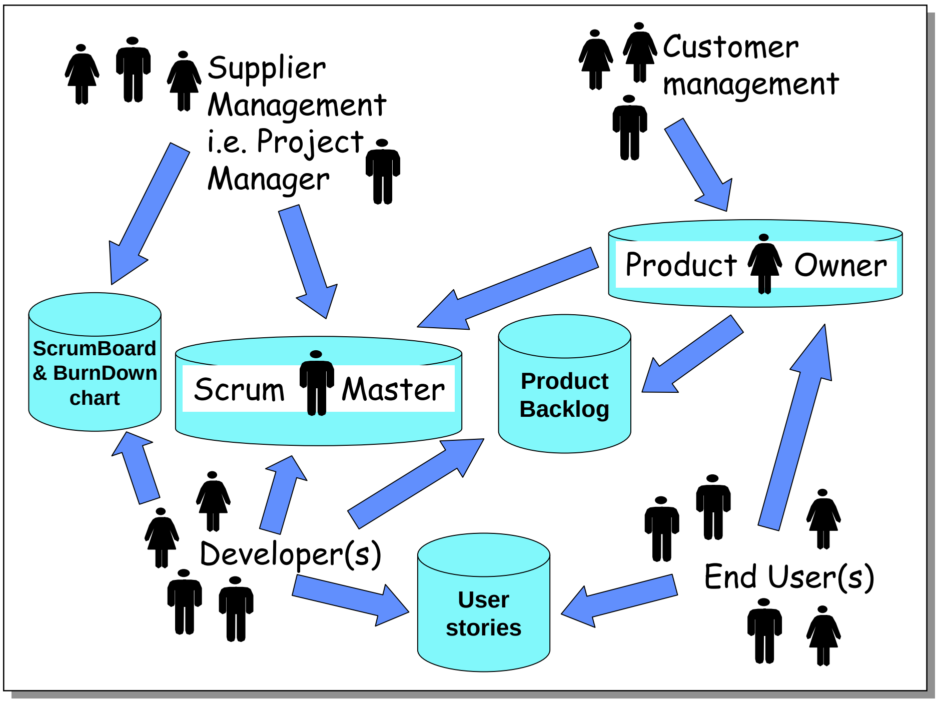

Figure 1. The five buckets represents five boundary spanners and objects identified in Scrum. The arrows pointing to the buckets indicate who communicates through each.

4 Social Team Capital

The concept of Social capital is an attempt to bring together a number of concepts. It can be defined as "*the goodwill available to individuals or groups. Its source lies in the structure and content of the actor's social relations. Its effects flow from the information, influence, and solidarity it makes available to the actor*" (Adler and Kwon, 2002).

The DELHI case used the distributed Scrum-of-Scrum model (Sutherland et al., 2007) hence two separate Scrum teams – one in India and one in Denmark were formed, and they were integrated by a daily Scrum-of-Scrums meeting, and by an *all-hands meetings* at the end of each sprint. What is special about Scrum is that this configuration more or less prescribes an internal network within each of the distributed Scrum team, and ties between the Scrum teams related to roles. Thus within each of the Scrum teams strong and redundant ties between all participants is emphasised, and so is strong ties between the Scrum Master for each team and the project manager / product owner. The *all-hands* *meeting* at the end of each sprint provide opportunity to form at least some sort of (initial) ties between members of the two Scrum teams. By prescribing and emphasising ties within the Scrum teams and between people possessing specific roles within the overall project organization, Scrum creates attention and opportunity for social ties to manifest themselves. Ties however, do not become strong just by being predefined. The quality of any tie depend on the individuals having trust in each other and being motivated to maintain the ties.

To analyse whether Scrum provides opportunity to develop trust we will user four types of trust suggested by Sabherwal (1999) of which we found Scrum to provide three. *Knowledge-based trust* may exist prior to starting a project if participants have a history together. Developing knowledge about each other is supported by teams being relatively small and co-located (providing physical presence) and the daily Scrum meetings allow all team members to interact and get to know what the others are doing. *Identification trust* comes from the negotiation of priority for user stories may contribute here. *Performance-based trust* comes from Scrum providing fast feedback through fast iteration. Between Scrum teams performance trust is supported by demos/deliveries at the end of each Sprint. Finally performance trust is established relatively fast between the Scrum team and the project manager based on the principle of daily Scrum-of-Scrum meetings, the burn-down chart, and deliveries as the result of each sprint.

So we find that Scrum in the DELHI case both supported the building of trust (of all kinds) and the establishment of ties. Thus Scrum is quite effective in building social capital.

5 Articulation Theory for Coordination

In order for multiple actors within a project to pursue a common goal, they have to perform activities, which single actors pursuing the same goals would not have to do, these extra ordinary activities we call coordination. Thus coordination can be defined as "*the additional information processing performed when multiple, connected actors pursue goals that a single actor pursuing the same goals would not perform*"(Malone, 1988). One way to understand how coordination comes about could be using the notion of *articulation* (Strauss, 1988). Thus, in order for the actors within a project to be able to collaborate and coordinate their effort the tasks involved in pursuing the common goal need to be articulated and it should be established who is doing what, when they do it, and how and when to coordinate/align their work.

When interviewing project managers and members of the DELHI Scrum team they point out that what they like about working with Scrum is that Scrum helps them to understand very clearly what work needs to be done within the whole project and the specific Sprint; what they are expected to do themselves, what others are doing, and how to coordinate work.

Analysing the DELHI case with articulation theory indicate that articulation work including coordination is performed in a very constructive manner when using Scrum. What is important when performing project work is to break down project work into tasks, sequencing the tasks, assigning the tasks to specific individuals, deciding how to perform the tasks and recognizing the need for coordinating/aligning tasks. In Scrum abstract tasks (general user stories) are defined as part of the product backlog, without considering *who* or *how* to perform the tasks. Second, somewhat more detailed tasks are defined when establishing a Sprint Backlog. Thus establishing the product backlog and the sprint backlog allow reconciliation about the *what* part (tasks) of the project work (between the product owner and the project team) to take place without complicating it with the *who*, *when* and *how* part. Third, tasks are defined in more details when moving tasks from the Sprint Backlog into the *in progress* column, and at the same time it is established *who* (in the team) is actually doing the work. Thus the need to coordinate with other tasks/people is addressed at a more detailed level. Finally Daily Scrum meetings and the Scrum Board provides a simple structure supporting the team finalizing the articulation work.

It seems that Scrum allow articulation to take place just-in-time involving the actual participants in the work. Thus using Scrum the articulation work takes place at a more and more detailed level as the process progresses, and reconciliation of the articulation work takes place when the people actually performing the tasks get involved. Summing up Scrum provides a framework that support all parts of articulation work, and yet spent very little time trying to foresee and negotiate the work flow and coordination mechanisms prior to actually conducting the work. Especially four aspects of Scrum provide coordination: the product backlog, the sprint backlog, the Scrum board and the daily Scrum meetings.

Having established an analytical understanding of how Scrum support articulation work, we now turn our interest toward understanding why Scrum also gain momentum as a paradigm used in virtual teams although it was originally targeted at collocated project teams.

By nature distributed project teams dependent heavily on a developing method which can support communicating and coordinating their daily work tasks (Sauer, 2006). The challenge is to be able to establish good communication and coordination through some kind of electronic communication-channels, a collaboration method that has proven less successful than working in collocated projects (Olson et al., 2002). A suggestion is to use practices such as synchronizing head milestones, frequent deliveries, use of peer-to-peer communication links, problem solving practices, information- and monitoring practices and client/vendor-relationship building practices (Kussmaul et al., 2004, Paasivaara and Lassenius, 2003).

The main challenge that distance between team members creates in virtual teams are related to communication, coordination and control (Carmel and Agarwal, 2001), and Scrum seems to be able to address these issues with some very simple yet powerful principles: Communication, Coordination, and Control.

In relation to *Communication* especially the principle of daily Scrum meetings ensures an open channel for communication. In the DELHI case separate daily Scrum meetings were conducted on location in India and Denmark followed later in the day by a Scrum-of-Scrums meeting. At the end of each sprint an all-hands meeting was conducted as a video conference plus shared presentation of documents, PowerPoint slides, product demo and more. In addition the communication within the Scrum team is supported by boundary objects and boundary spanners (see above). Although meetings and boundary objects provide an opportunity to communicate it may not result in open and honest communication as social capital (see above) obviously is more difficult to develop in distributed teams. However, we have shown that some of the principles used in Scrum are very powerful and helps develop trust more or less for free.

In relation to *Coordination* the principle of making tasks, their sequence and their status visible through the spring backlog and the Scrum board allow all team members to get an overview of the tasks to be performed and already performed tasks, and understand who is/has been working on what tasks. This knowledge enables each team member to approach other team members directly if a common issue has to be resolved. The daily Scrum meetings further allow team members to realize when coordination is needed.

As for *Control* the best distributed teams uses common milestones, frequent delivery, quick feedback, frequent meetings, and frequent progress reports (Paasivaara and Lassenius, 2003). All these practices can be found in Scrum as we saw it practiced in DELHI; Sprints serve as common milestones, each Sprint (which is relatively short) results in a delivery. Feedback on individual tasks is given when moving a task from *Done* to *Done Done*. Frequent meetings are implemented as daily Scrum meetings, Scrum of Scrum meetings. Progress can be read directly of the daily update of the scrum Board and the Burn-down chart.

6 Conclusion

In this paper we phrased the research question: *What is it that makes Scrum so useful?* To answer this we carried out a case study. In the study we obtained an in-depth understanding of reasons why. We analysed the data gathered using an interpretive hermeneutic approach. Our conclusion and answer to the research question phrased is that: Scrum is so useful because it provides effective communication in the form of boundary objects and boundary spanners, it provides effective social integration by building up social team capital, and it provides much needed control and coordination mechanisms by allowing both local and global articulation. Therefore Scrum works so well for a distributed project team.

References

Adler, P. S. and Kwon, S.-w. (2002) Academy of Management Review, 27, 17-40.

Austin, R. D. and Devin, L. (2009) Information Systems Research, 20, 462-a-477.

Carmel, E. and Agarwal, R. (2001) IEEE Software.

Conboy, K. (2009) Information Systems Research, 20, 329-354.

Dybå, T. and Dingsøyr, T. (2008) Information & Software Technology, 50, 833-859.

Friedman, T. L. (2006) The World is Flat: The Globalized World in the Twenty-First century Penguin Books, London.

Kussmaul, C., Jack, R. and Sponsler, B. (2004) Extreme Programming and Agile Methods-XP/Agile Universe 2004, 147-154.

Levina, N. and Vaast, E. (2005) MIS Quarterly, 29, 335-363.

Malone, T. W. (1988) National Science Foundation Coordination Theory Workshop, Cambridge, Massachusetts.

Miles, M. B. and Huberman, A. M. (1994) Qualitative Data Analysis: An Expanded Sourcebook, Sage Publications Inc., Thousand Oaks.

Myers, M. D. (2009) Qualitative Research in Business & Management, Sage Publications, London.

Myers, M. D. and Avison, D. (2002) In Qualitative Research in Information Systems - A Reader(Eds, Myers, M. D. and Avison, D.) Sage Publications, London, pp. 3-12.

Nahapiet, J. and Ghoshal, S. (1998) Academy of Management Review, 23, 242-266.

Olson, J. S., Teasly, S., Covi, L. and Olson, G. (2002) In Distributed work(Eds, Hinds, P. and Kiesler, S.) The MIT Press, pp. 113-136.

Pettigrew, A. M. (1990) Organization Science, 1, 267-292.

Paasivaara, M. and Lassenius, C. (2003) Software Process Improvement and Practice, 8, 183-199.

Rising, L. and Janoff, N. S. (2000) IEEE Software, 26-32.

Sabberwal, R. (1999) Communication of ACM, 42, 80-86.

Sauer, J. (2006) In Proceedings of the 29th Information Systems Research Seminar in Scandinavia, Helsingborg, Denmark, dated: Aug-12-15-2006.

Scwaber, K. (2004) Agile Project Management with Scrum, Microsoft Press, Redmond, WA.

Star, S. L. and Griesemer, J. R. (1989) Social Studies of Science, 19, 387-420.

Strauss, A. (1988) The Sociological Quarterly, 29, 163-178.

Sutherland, J. and Schwaber, K. (2010), Vol. Draft - 14 May 2010 SCRUM Training Institute.

Sutherland, J., Viktorov, A., Blount, J. and Puntikov, N. (2007) In 40th Hawaii International Conference on System ScienceHawaii.

Walsham, G. (2006) European Journal of Information Systems, 15, 320-330.

Ågerfalk, P. J., Fitzgerald, B. and Slaughter, S. (2009) Information Systems Research, 20, 317-318.

Acknowledgements

The authors wish to acknowledge the valuable help of Danske Bank Group IT, DCI, ITC Infotech, Linda Olsen, Søren Gildsig and the many project team members who allowed access to their project..