LEADING AGILE SELF-ORGANIZING TEAMS: A COLLECTIVE LEARNING PERSPECTIVE

Behnaz Gholami1
bgholami@uni-mannheim.de
Institute for Enterprise Systems
University of Mannheim – Germany

Armin Heinzl
heinzl@uni-mannheim.de
Institute for Enterprise Systems
University of Mannheim – Germany

Abstract

Agile Software Development (ASD) puts a higher emphasis on people as well as social interactions, which require a higher degree of human communication and learning. Self-organizing ASD teams are considered to be the fundamental features of agility boosting and benefitting from learning. Hence, ASD changes the role of the project manager from a project controller to a team facilitator. Nevertheless, according to current research, self-organizing teams do not work in an entirely leaderless or free from management control. Therefore, this study attempts to understand the role of leaders of ASD teams and determines the areas in which leaders can influence collective learning as a group process phenomenon. Adopting the interpretive case study, the results unfold collective learning incidents and extend recent work on the influence of leaders on collective learning in ASD.

Keywords: Leadership, Self-organizing teams, Agile Software Development, Collective learning

1 INTRODUCTION

Recently, Agile Software Development (ASD) methods and practices have been widely adopted in the enterprise software industry. The principles of “The Agile Manifesto”2 focus on self-organizing teams and empowered individuals in order to build software products more effectively. Self-organizing teams are fundamental features of agility in software development (McAvoy and Butler 2009; Cohn 2010; Hoda, Noble, and Marshall 2012). ASD teams are considered to be democratic, without a strict hierarchy. They are essential social networks that interact intensively (Boehm and Turner 2004). Accordingly, ASD changes the role of the project manager from a project controller to a team facilitator (McAvoy and Butler 2009). The process of decision making now resides with the team members rather

---

1 Corresponding author: Address: Schloss, 68131 Mannheim, Germany; E-mail: bgholami@uni.mannheim.de; Phone: +49 621 181-1752; Fax: +49 621 181-3627

2 The Agile Manifesto (http://agilemanifesto.org/), also called the Manifesto for Agile Software Development, is “a formal proclamation of four key values and 12 principles to guide an iterative and people-centric approach to software development” (Rouse 2011 p. 1).
than the team manager. Team members learn how to work together and how to mutually oversee the activities of each other (Barker 1993). ASD puts a higher emphasis on people as well as social interactions, which requires a higher degree of human communication and learning.

Self-organizing teams are not entirely leaderless or free from management control (Takeuchi and Nonaka 1986; Cohn 2010). The issue of leadership in self-organizing teams has been studied in related contexts (see: Barker 1993; Bunderson and Boumgarden 2010). Additionally, scholars tried to elaborate on the role of leaders in ASD teams (McAvoy and Butler 2009; Vidgen and Wang 2009; Hoda et al. 2012). Recently, for instance, Hoda et al. (2012) identified six informal and spontaneous roles that make agile teams self-organizing. Nevertheless, the reliance of ASD on self-organizing teams creates a common misconception about the role and leadership style in agile teams (Cohn 2010). Leaders of agile self-organizing teams are meant to provide subtle feedback and direction (Takeuchi and Nonaka 1986), coordinate members, obtain resources, motivate the teams (Hoda et al. 2012) as well as facilitate collaboration and teamwork instead of command and control. An ASD team should be able to self-organize its challenges and constrains which have been posed by management (Takeuchi and Nonaka 1986; Cockburn and Highsmith 2001; Cohn 2010; Moe, Dingsøyr, and Dybå 2010). Therefore, leaders play an important role in shaping self-organizing teams.

Furthermore, self-organizing teams can foster organizational knowledge creation (Nonaka 1994). Autonomy can enhance team learning in software development teams (Vidgen and Wang 2009) and learning is a continuous process in self-organizing teams. As a result of learning, team members should develop a more profound understanding of customer requirements, how to organize and accomplish the own work processes as well as how to make design and implementation decisions more effectively and efficiently. Zellmer-Bruhn and Gibson (2006) found that the learning is more likely to occur in teams with decision-making autonomy. Continuous self-organization requires the ability for learning that enables norms and rules adapt to organizational change (Moe et al. 2010; Hoda et al. 2012). Self-organizing teams are considered to be cross-functional and have multi-learning skills. They gain, share and maintain knowledge through (a) “multi-learning”: learning across multiple levels (individual, group, and organization) and multiple functions, and (b) “transfer of learning”: learning across other units of the organization (Takeuchi and Nonaka 1986).

Although studies have emphasized on learning in ASD teams and on how agile self-organizing teams can benefit from multi-level learning, a thorough analysis of the role and the influence of leadership in self-organizing ASD teams on learning is deemed to be essential (Spohrer, Gholami, and Heinzl 2012). ASD teams, as mentioned earlier, are not entirely leaderless and as a result the role and leadership style and behaviors of a team leader are likely to affect the success or failure of software development projects. In this study collective learning is defined as a process of group reflecting, collective acting, knowledge disseminating, sharing of understanding, and mutual adjustment (Kolb 1984; Brooks 1994; Kolb, Boyatzis, and Charalambos 2001; Edmondson 2002; Bondarouk 2006). Therefore, this study addresses the question: In which areas can leaders in self-organizing ASD teams influence collective learning?
The remainder of this paper is organized as follows. In the next section we present a brief background of ASD, and particularly Scrum and self-organization. Based on the literature, we briefly go through four areas of improvement that can be influenced by leaders in self-organizing teams. Following the introduction of the research design, we then present the results of a multiple case study of a multinational software company. We present the results according to the four areas we identified in the literature. Finally, the study’s implications and limitations are discussed and the major findings are summarized in the conclusion.

2 BACKGROUND

2.1 Scrum and Self-organization

There are many agile methods and practices for software development. Among all, Scrum is a widely adopted agile method (M. Poppendieck and T. Poppendieck 2003; Sutherland 2004) that includes a project management framework in which development activities such as requirements elicitation, software design, implementation, and testing are conducted together (Highsmith 2010). Scrum is both, an iterative and an incremental process. The iterative process is constituted by the concept of successive refinement. In this context, a self-organizing development team picks work packages from a prioritized list of customer requirements which are called backlog items. The product backlog is decomposed into sprint backlogs which represent the iterative character of the work process (Cohn 2010; Hoda et al. 2012). Team members are supposed to have autonomy over choosing the activities and routines which support them best in developing the required elements of enterprise software. Hence, empowered with collective decision making and cross-functional skills, team members are given significant authority and responsibility for their work (Hoda et al. 2012; Moe et al. 2010). As a consequence, team learning processes are meant to be integral to daily work in Scrum teams. Members of Scrum teams invest a significant amount of time for facilitating and leveraging collective learning among each other. Team members provide their peers with the opportunity to learn about and from them. Self-organizing Scrum teams are supposed to foster the attitude of an open team culture and are expected to experience more psychological safety within their teams.

Scrum defines the following roles as members of a Scrum team:

**Product Owner (PO):** The PO in Scrum represents the needs of the end customers, controls the software development and collaborates with the development team during the entire project. This role combines product and project management work. The PO is responsible for the success of the software project (Pichler 2007 p. 9).

**Development Team:** The development team performs all work needed to convert requirements into product increments. In Scrum, software development is team-based, which means all roles like architect, programmer, tester and others work closely together (Pichler 2007 p. 13).

**Scrum Master (SM):** The SM acts as coach and change agent. S/he supports the team on its way to a high performance team, protects the team from external disturbances and removes impediments (Pichler 2007 p. 19).

Scrum defines the following regular meetings (Pichler 2007):
Planning: In this meeting the estimation of the planned features for the next sprint is done by the Scrum team.

Daily Scrum: The daily Scrum is a short (15 minutes) meeting that supports the self-organization of the team.

Review: The review meeting at the end of the sprint inspects the results and makes the project progress transparent.

Retrospective: The goal of this meeting, as the last activity during a sprint, is to improve the cooperation inside the team and the implementation of the process.

2.2 Leadership Role

Team leadership promotes or inhibits collective learning (Edmondson, Dillon, and Roloff 2007). By involving the members in the decision-making process and outlining the team goals and expectations and having a democratic leadership style, the team leader can positively simulate team learning (Sarin and McDermott 2003). On one hand for instance, central task allocation by a project manager without consultation with team members can inhibit learning in ASD (Vidgen and Wang 2009). On the other hand, a change in team leadership style should also influence team learning (Spohrer et al. 2012). Additionally, how leaders design their teams and the quality of their direct interaction with the team both influence team self-organization and the quality of member relationships (Wageman 2001). Clear direction, skill diversity, task interdependence and challenging task objectives are features of team design in Wageman’s (2001) study. Team members are cross-functional to perform any task the work needs and they also have the authority and responsibility to make the essential decisions necessary to complete the work (Barker 1993). Therefore, to study leadership in self-organizing teams and its influence of collective learning, we focus on four main areas:

2.2.1 Team Empowerment

Team empowerment is defined as “the state in which members share a collective sense that they have the responsibility and the authority to control their work” (Mathieu, Gilson, and Ruddy 2006 p. 102). As discussed, an ASD team should be able to self-organize its challenges and constraints which have been posed by management (Takeuchi and Nonaka 1986; Cockburn and Highsmith 2001; Cohn 2010; Moe et al. 2010). However, Maruping and Magni (2012) argue that through an expanded set of responsibilities and expectations fostered by team empowerment climate, team members may be experiencing work overload, thus reducing their likelihood of exploring and learning. Hence, answering the question of to what extent the team leader should have authority and responsibility and to what extend s/he should give the team autonomy will enhance our understanding of collective learning in self-organizing ASD teams.

2.2.2 Team Climate

Team climate is defined as “distinctive patterns of collective beliefs that are communicated to group members through the socialization process and are further developed through members’ interaction with their physical and social environments” (Lindell and Brandt 2000 p. 331). Agile self-organizing teams should emphasize a common focus, mutual trust, respect and a collaborative decision-making process (Cockburn and Highsmith 2001). In this context, the team leader plays an important role exerting the psychological safety with-
in a team in order to catalyze a process of learning (Edmondson 1999, 2004; Edmondson et al. 2007; Nemhard and Edmondson 2006; Bunderson and Boumgarden 2010). By affecting team members’ perceptions of psychological safety, assessment of risks, and propensity to take independent action, power and status differences can affect the willingness of members to engage in collective learning activities (Bunderson and Reagans 2011). Team learning processes are meant to be integral to daily work in ASD teams and this leads to high transparency among team members. As a consequence, team members are informed about each other’s progress, challenges and knowledge while completing their task.

2.2.3 Team Learning Goal

Learning behaviors are affected by different goals and purposes (Spohrer et al. 2012). Team members may reach different conclusions about how their individual goals are structured, and as a result, adapt their interactions with other team members (Tjosvold, Yu, and Hui 2004). By affecting interactions, information sharing and supporting other members in group challenges, goal setting is likely to affect team learning behaviors (Tjosvold et al. 2004). Also Hirst, Van Knippenberg, and Zhou (2009) note that “the relationship between an individual’s goal orientation and creativity is contingent on team learning behavior” (p. 282). In terms of leadership, Bunderson and Reagans (2011) discuss that the power hierarchy complicates anchoring on shared goals which is the key process of collective learning. Research on ASD team learning should clarify to what degree learning goals can beneficially influence team learning in different configurations (Spohrer et al. 2012).

2.2.4 Task

This category includes the characteristics of the task, the cross-functionality, skill and experience of team members to complete the task, and the time needed to perform a task: There is empirical evidence of the effects of task characteristics on team learning. Edmondson (1999) states that “highly routine repetitive tasks with little need for improvement or modification” may inhibit team efficiency and performance (p.354). On the other hand, uncertain and risky tasks may raise the need for teams to learn continuously in order to understand the environment, as well as customers’ needs. Uncertain tasks may further require team members to coordinate more effectively (Spohrer et al. 2012). Nevertheless, work of cross-functional teams does not always call for team learning and knowledge traverse but team doing and knowledge transcend, especially when the teams are faced with novelty (Majchrzak, More, and Faraj 2012). Vidgen and Wang (2009) mention that multi-skilling is one of the enablers of learning in ASD teams. In contrast with traditional teams in which roles are grouped around functional tasks reflected by their organizational roles, in ASD teams, members are cross-functional and are not limited by their organizational job titles, training or experience(Hoda et al. 2012). Furthermore, one of the goals of ASD is reducing the time from decision to feedback (Cockburn and Highsmith 2001). In an agile team, skill development is important, so that each person can deliver more value with time (Cockburn and Highsmith 2001). Additionally, since ASD teams are small teams, they spend less time coordinating the efforts of team members (Cohn 2010). On the other hand, the team’s perception of time pressure is negatively related to their use of knowledge repositories (Gray and Durcikova 2006). Task characteristics of each ASD team vary, and together with cross-functionality and time pressure can affect collective learning.
3 RESEARCH DESIGN

The goal of this research is to understand the role of leaders in agile self-organizing teams on collective learning.

3.1 Research Context

This study was conducted in a multinational software corporation in which Scrum has been adopted for four years in order to increase the efficiency in developing enterprise software products. This research is embedded in a larger research program aiming at understanding and improving the process of software development within software development teams. The company tries to raise knowledge and awareness of employees about agile software development by investing a lot into training classes, virtual trainings, books and other sources of learning. The company values people empowerment and believes in open environment where people can easily speak up and trust each other.

In the company, not all the teams had to shift to Scrum model at the same time. Even after four years, the transformation from traditional software development to agile software development is still in the progress. Scrum is tailored from team to team. Teams are empowered to selectively implement or change particular techniques according to their own need and situation. Furthermore, apart from two key roles of the SM and the PO which are defined in Scrum, other key roles such as software architecture, solution manager, quality manager and line manager exist in this company. Solution manager, quality manager and software architecture are also parts of the development team and the line manager is a manager who manages a program team. Each program team has a unique scope for a product and is composed of two or more Scrum teams. This study focuses on the PO and SM as leaders of a Scrum team.

3.2 Methodology

This study is based on multiple case design (Yin 2008). We followed interpretive case study guidelines (Sarker and Sarker 2009 pp. 445–446) to examine and make sense of the data. Results proposed by this study are nascent (topics for which little or no previous theory exists) and this approach seems to be particularly appropriate for answering the research questions on how leaders in agile self-organizing teams influence collective learning (Edmondson and Mcmanus 2007) in which ASD attracted little formal theorizing to date.

3.3 Data Collection and Data Analysis

Cases were chosen according to selective sampling (Glaser 1978). The two cases were suggested by the senior staff and one of the managers who was responsible for process development and learning in the company. Team work was observed in the period from November 2012 to January 2013. During the observation period, a research assistant visited team meetings. She participated in some daily Scrum meetings, planning meetings, review meetings and retrospectives. The semi-structured face-to-face or telephone interviews generally lasted between 40 to 70 minutes. The interviews were recorded. In some cases, we took notes instead of recording when the interviewee requested to do so. During the interviewing process, we tried to be flexible, non-direct, specific, and varied how the interviews were
conducted depending on the interviewees’ motivation (Sarker and Sarker 2009 p. 445). The authors were also provided access to relevant internal documents (e.g. internal wiki pages).

The data collection and analysis occurred iteratively. As field notes and interviews were transcribed, they were coded. Through this early analysis we formulated new questions for later interviews (Strong and Volkoff 2010). We identified and refined concepts through constant comparison. Data from this study was imported into Nvivo for analyzing the qualitative data. The first author coded the material, using open coding, looking for material related to leadership, empowerment, climate, goal and task. The unit of analysis is the team and particularly the Scrum team. In this study it is preferred to use the term “collective learning” rather than “team learning” since learning occurs not only internally among the members of development teams, but also externally among other stakeholders of the team (see Table 1).

<table>
<thead>
<tr>
<th>Table 1. Number of Interviewees</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Team 1</td>
</tr>
<tr>
<td>Team 2</td>
</tr>
<tr>
<td>Team 3</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

*Includes translator, or Line Manager.

4 FINDINGS AND DISCUSSIONS

4.1 Leadership Role

In team 1, the team seems to be satisfied of the roles. There was no special complaint about roles and responsibilities; however, the SM was also entitled as Project Leader:

SM [1]: “I am both the Project Leader and the SM. From project management perspective, Project Leader is a kind of project manager, but it is much concerned with the shipment of the software. So I am basically doing two tasks.”

Developer [1]: “From the team perspective it is necessary for the SM to take care of the responsibilities of Project Leader.”

The team did not exactly know whether the responsibilities of a Project Leader are defined for the SM. Since this role is not defined by Scrum, the SM was also assigned as Project Leader.

Developer [1]: “I would guess that her two roles are quite similar. The approach is how to bring the product forward based on the viable scope. The SM is responsible to organize the team itself and bring the planning as well as the backlog items for-

---

3 The number represents the team (team 1, team 2, and team 3) which the person belongs to.
ward. The Project Leader takes more the outside-in view that is how to wrap up the product. This role is closer to the customer.”

Developer [1]: “I guess the Project Lead takes also care of organizational matters where I am not sure if the SM would do that… I doubt that it is necessary to have both roles in the team.”

In three cases the SM is assumed to be organizer and facilitator in terms of team interactions, collaborations and knowledge dissemination.

SM [2]: “I am the SM of the team, so I think that is also a part of the SM’s work to at least show the team where the possibilities of gaining knowledge needed are.”

Developer [2]: “For example the agile software engineering training was initiated by our SM. I guess, everybody and every development team in the organization will get this training and it is part of the SM’s duties to organize that.”

Developer [1]: “Our SM is the facilitator of the process, with her experience, she knows exactly what needs to be done, until when and which decisions needs to be made to bring the whole piece forward.”

In team 1, although it was not explicitly mentioned, there seems to be confusion between responsibilities of the PO and the SM with other roles.

SM [1]: “In some cases, the information we need is not properly transferred to the team by the PO.”

Developer [1]: “In the team, the PO does not clearly inform us about requirements. I mentioned it indirectly several times, but the problem still exists.”

PO [1]: “I am quite new in this role; it is only 2.5 years that I have this role. The SM has more experience than me. In team we lack the role of Solution Manager or anyone to be informed and experienced about the product.”

Developer [1]: “I would say for functionality-related topics, what the software should do, the PO is the person in charge. When it goes more to the organizational point of view or maybe helping how something should be implemented the Project Leader [is the person in charge].”

In team 3, the SM responsibilities have wider range than two other cases:

Line Manager [3]: “in the team, the SM has a wide range of responsibilities. Sometimes he also acts in place of the Line Manager. In the company there is strict guideline that the SM should not be the Line Manager. I don’t share that opinion that you should have a strict distinction between the SM and the Line Manager.”

Furthermore, some Scrum mentors or coaches gave the impression that Scrum development teams are better than traditional development teams in terms of learning. This phenomenon
is called “impression management” (Moe et al. 2010; Morgan 2006) and is related to a particular challenge to team leadership and barrier for learning and improving work practices (Moe et al. 2010). Evidences indicate that a phenomenon like “impression management” can be observed within three teams.

Developer [1]: “They (managers) told us to do something, but they never told us why? Why if we do those instructions [of Scrum] we will get better results. They did not show us the real value.”

Developer [2]: “I participated in the Scrum training, but I didn’t feel that it was a huge difference as what we have been told [before]… In theory, Scrum is about working on small modules, then test it and if the module is not good enough you can change it. To me, it was not really the case. It is something like Waterfall (traditional model), but in smaller chunks. We cannot change these small chunks. It all depends on the first plan and just like the traditional way, if the initial plan is not good enough the process is not successful enough… The initial plan completely depends on managers.”

Findings show that despite of long introduction of Scrum to software development teams, the role and responsibility of team leaders varies substantially from team to team and may create confusion and dissatisfaction among team members. Team members do not often reach consensus upon who should be the team leader and what responsibilities s/he should have.

4.1.1 Team Empowerment

Moe, Dingsøyr, and Dybå (2008) divided autonomy in an agile project into individual, internal and external autonomy. According to them, individual autonomy refers to the amount of freedom an individual has in carrying out assigned tasks. Internal autonomy refers to the degree to which all team members jointly share decision authority, rather than a centralized decision structure. External autonomy is defined as the influence of management and other individuals outside the team on the team’s activities (Moe et al. 2008 p. 78). In team 1 and 3, although the team feels internal autonomy, there is also a sign of lacking individual autonomy.

Developer [1]: “The decisions are just made by the PO, the SM and maybe the architect and we just should do it. For my previous role [in traditional development model] I had more autonomy over my tasks and decisions. In this mode of development, one should have a powerful person by her side to have the chance that her decision is taken into account.”

Developer [1]: “We really don’t have autonomy over choosing a task. In such a small team the skills of each person is known to others and if the task is related to your expertise it is automatically yours.”

PO [3]: “The team is quite self-empowered and I don’t need to tell them [what to do] that much.”
In three cases external autonomy does not seem to be very high when it comes to customer needs:

Line Manager [3]: “[If something urgent needs to be implemented] First of all, what we try to do is to make clear why the solution of the problem is so urgent. So we need to put ourselves in the customer’s shoe […] and usually the team should handle that.”

SM [1]: “If the PO asks the team to do something, even it is in the middle of their work, they accept to do it.”

Contrasting individual and internal autonomy in the teams is not of a surprise. Scholars find that in teams with a high level of individual autonomy, internal autonomy is low (Uhl-Bien and Graen 1998). This is because of shared decision making and diffused responsibility rather than single decision making and individual responsibility (Kirkman and Rosen 1999). The PO’s role is to exert external autonomy and the SM’s role is to facilitate internal autonomy. The findings show that all three teams are self-organizing. Both self-managing teams and empowered teams are autonomous, but the members of empowered teams also share a sense of doing meaningful team work (Kirkman and Rosen 1999) that advances collective learning. Although there is evidence of leaders being successful in exerting autonomy in Scrum teams, the team empowerment as the antecedence of learning calls for improvement.

4.1.2 Team Learning Goal

Since teams are self-organized and team members are empowered, the team learning goal is often not in line with individual learning goals. In three cases, the teams were aware of the team task, but learning and bringing the knowledge to the team level is not something that all team members agree on. In the company, having a special expertise, for instance, is a core competency for individuals and gaining new knowledge is mostly personal. Introduction of Scrum facilitated collaboration, but learning new knowledge is still missing in teams’ attitudes.

Developer [2]: “When we stop learning, then after one or two years we are out of business, especially in the IT industry.”

Developer [1]: “There is no culture of sharing within the team. OK! If somebody asks everybody is willing to share. But nobody thinks it is necessary to share when nobody asks for it.”

Developer [2]: “Learning is an ego boost, when you learn something new and complicated and you try it, there is a personal positive experience.”

The collective learning goal is not clarified in the teams, neither by the SM nor the PO. The learning goals are defined individually and independent from the team learning goal. As mentioned earlier, in team 2, the SM finds himself as the facilitator of collective learning. He encourages team members to share their knowledge with their teams. The PO is responsible for bringing the knowledge and requirements from customers to the team. To have
cooperative learning goals, both the SM and the PO play an important role to motivate members and raise the need for collective learning.

4.1.3 Team Climate

Despite of being self-organized, team members in all three cases still believe that the team leader is responsible to infuse both transparency and psychological safety to the team. In three cases the leader is the key person to manage the conflicts if the problem is not solved within the self-organized team.

Developer [3]: “We easily discuss the different points of view in the team, but usually the PO or the Line Manager is the person who decides if the problem persists.”

Scrum as a project management framework is likely to increase the transparency within a team, but creating transparency without psychological safety poses new challenges for teamwork and collective learning, threatening the positive attitude of openness of teams. As stated earlier, on the one hand, the company believes in an open environment where people can easily speak up and trust each other. People easily share their knowledge with others and feel free to ask technical questions. On the other hand, team members talk about trust and transparency as very separate issues. They believe that Scrum creates transparency and raises awareness about each other’s task, but they also believe that trust and psychological safety are not very much concerned with this transparency.

Developer [1]: “Transparency is much much higher [in comparison with the traditional model], to me it is not a bad point because you exactly know where you are and what you are doing.”

Developer [2]: “We can talk and express ourselves, no problem, but I think it is not because of the transparency which Scrum creates, but because of the culture of the company that is very open.”

Even if team members feel free to express themselves, over-reliance on Scrum meetings may inhibit the open climate of the team. Vidgen and Wang (2009) discussed that over-reliance on informal communication and collaboration and over-communication between team members are inhibitors of team learning in ASD teams. This study also confirms the phenomenon of over-communication among team members.

Developer [2]: “We have so much freedom over our decisions. But sometimes meetings are too much... Sometimes we even have meetings to decide whether we need meetings or not!”

Developer [3]: “I told my manager that I don’t come to the meetings, I want to work!”

Developer [1]: “Meetings about everything are too much… In our team for example there is a person who does not like to participate in the meetings… this is not good for the climate of the team that we do not have him in the meetings”
4.1.4 Task

Task Characteristics: The characteristics of the tasks determine whether team members need to gain new knowledge or not. Highly routine and repetitive tasks obviously need less learning, and innovative and novel tasks, on the other hand, need more. Team 1 and 2 introduced their tasks as mainly routine task. Team 1 has to do both development and maintenance tasks, and maintenance tasks do not motivate them to learn.

Developer [1]: “Maybe if the team task is changed the situation will change and people are more motivated for gaining new knowledge. Especially our maintenance task does not motivate us to learn and share.”

Developer [1]: “My task quite a routine one, I do not need to learn that much”

Multi-functionality and multi-skilling of the developers, which means no separation of functional roles (Vidgen and Wang 2009), is one of the characteristics of Scrum teams. Leaders support team members not to be specialized only on specific tasks but gain new skills and functionalities and have the opportunity to do different tasks. In this study, in all three teams others mostly refer to the most experienced developers to seek for the knowledge and ask technical questions. In cases 1 and 3, there are more than one expertise on each topic so in case somebody leaves the team temporarily or permanently, the other can manage to handle the task and maintain knowledge.

SM [1]: “We have more than one person on each topic. When someone leaves we have back-up sessions which compensate the lost.”

Line Manager [3]: “We do share knowledge on every level; we have Monday schools, so we have a common knowledge. Basically everyone can always do everything; that is ideal.”

Time pressure: Time is one of the most important inhibitors of learning in the company. In the internal corporate portal poll employees confirmed that. Time pressure is also the dominant inhibitor of learning in three teams.

Developer [2]: “From the project management point of the view Scrum is very good. But what is bad about Scrum is that it takes your time. Managers know your capacity and want you to dedicate that capacity to the task!”

Developer [1]: “With our team often the topics go from sprint to sprint and they are not getting finished. But I know that there are other teams who try and finish everything in each sprint. Do everything by time. Perhaps for some types of software a month might be too short for a cycle.”

This is a significant responsibility of the PO to control that backlog items do not take place more than a sprint. The PO is supposed to offer backlog items which fit into one sprint. Moreover, there is a time slot called slack time, which was assigned by some managers to some teams. Slack time is the time that employees are free to do whatever they like during
working hours. In case 3, it is called Monday schools. Monday schools in team 3 are meetings in which team members share their knowledge and do very short workshops of about half an hour to half a day. The idea of Monday schools in team 3 was proposed by the Line Manager. There is also the impression that Scrum is a management tool which divided the task into smaller pieces and does not give room to learn and share.

5 IMPLICATIONS

Research in the IS domain is characterized by a significant lack of empirical research in ASD research (Dybå and Dingsöyr 2008; Conboy 2009; Lee and Xia 2010). However, during the last few years, an increasing amount of empirical effort has been made to examine and explore ASD through various theoretical lenses. Additional to this effort, the initial findings of this study can first extend our understanding of the role of leaders in ASD self-organizing teams and second in which areas they can influence collective learning in teams. Depending on various factors such as the characteristics of team task, leaders of ASD self-organizing might adopt different roles to support learning. Moreover, scholars have accentuated the need for team learning theories that take the specifics of disparate tasks and industry domains into account (Edmondson et al. 2007). This study makes a contribution to existing research regarding team learning and ASD literature as a topic of high multidisciplinary interest.

It is important for managers to know that implementing agility into software development does not necessarily enable collective learning in agile self-organizing teams. This study of collective learning delivers additional insight into the aspects, new opportunities and behaviors of learning in ASD which are totally or partially ignored or taken for granted. This study enables managers to locate and maintain the knowledge within their agile organization and foster learning aspects according to the team type and task.

6 LIMITATIONS

The results of this study have to be generalized in other Scrum teams. During the interviews, the important role of Line Managers emerged. The role of Line Managers was not the scope of this study from the beginning. For team 1 and 2, it was not possible to conduct interviews with the Line Managers and also more developers from team 2 and 3. This study tries to shed light on the areas of improvement in Scrum teams which the PO and the SM can influence regarding collective learning, but the question of how exactly these roles and other key roles can influence collective learning remains to be answered.

7 CONCLUSION

Adopting the interpretive case study design, this study attempts to understand the role of leaders of ASD teams and determines the areas in which leaders can influence collective learning as a group process phenomenon. These areas of improvement are leadership role clarity, team empowerment, team climate, team learning goals and task. Evidence showed that the role and responsibility of the team leader varies substantially from team to team and may create confusion and dissatisfaction among team members. This may also affect
collective learning. Team empowerment is twofold: although Scrum was successful creating self-organizing teams, individual, internal and external autonomy within a Scrum team need improvement. Sometimes, leaders do not transfer knowledge the team needs assuming the team is self-organized and responsible for its own needs. Team climate is perceived to be more open compared to the traditional model. People can easily speak up and express their opinions; also the transparency is high but this does not necessarily lead to psychological safety. The amount of learning depends on the nature and characteristics of the task. Multi-functionality works well to maintain knowledge within a team but time pressure is still the main inhibitor of learning. Team members expect their leaders to give them time to learn and share knowledge. These results try to unfold collective learning incidents and extend recent works on influence of leaders on collective learning in ASD.

REFERENCES


