The Journal of Systems and Software 122 (2016) 274–286

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Teamwork quality and project success in software development: A survey of agile development teams

Yngve Lindsjørn[a,*], Dag I.K. Sjøberg[a,b], Torgeir Dingsøyr[b,c], Gunnar R. Bergersen[a], Tore Dybå[a,b]

[a] Department of Informatics, University of Oslo, Norway
[b] SINTEF, Trondheim, Norway
[c] Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim

ARTICLE INFO

Article history:
Received 15 April 2015
Revised 14 July 2016
Accepted 18 September 2016
Available online 21 September 2016

Keywords:
Agile development
Project management
Team performance
Teamwork quality
Work Satisfaction
Learning

ABSTRACT

Small, self-directed teams are central in agile development. This article investigates the effect of teamwork quality on team performance, learning and work satisfaction in agile software teams, and whether this effect differs from that of traditional software teams. A survey was administered to 477 respondents from 71 agile software teams in 26 companies and analyzed using structural equation modeling. A positive effect of teamwork quality on team performance was found when team members and team leaders rated team performance. In contrast, a negligible effect was found when product owners rated team performance. The effect of teamwork quality on team members´learning and work satisfaction was strongly positive, but was only rated by the team members. Despite claims of the importance of teamwork in agile teams, this study did not find teamwork quality to be higher than in a similar survey on traditional teams. The effect of teamwork quality on team performance was only marginally greater for the agile teams than for the traditional teams.

1. Introduction

Agile methods have been widely used in software engineering over the last decade. Even though agile methods emphasize teamwork more than traditional development methods do (Nerur et al., 2005), there is no thorough investigation of the effect of teamwork quality (TWQ) on project success in agile teams.

Agile development methods are used as an umbrella term to describe a number of development methods (Dingsøyr et al., 2012; Dybå and Dingsøyr, 2008). The agile manifesto[1] advocates "working software over comprehensive documentation", "customer collaboration over contract negotiation", and "responding to change over following a plan". Accordingly, to respond with agility to change, team members should work more closely together, have more frequent communication, be aware of other team members' work efforts, and be able to shift workload between persons. More specifically, the agile manifesto states that the best architectures, requirements, and designs emerge from self-organizing teams; the best communication is face-to-face communication; and business people and developers should work together daily. Collaboration and coordination are also central in the agile literature (Sharp and Robinson, 2010; Strode et al., 2012). In the most popular agile method, Scrum, work is organized in small, cross-functional teams with a facilitator and team members. Team members coordinate their work frequently, such as in daily stand-up meetings (Stray et al., 2016). Vinekar et al. (2006) explain that agile development and traditional development have different views on teamwork. *Agile development* is characterized by collaborative work, which requires multidisciplinary skills, pluralist decision making, high customer involvement, and small teams, while *traditional development* focuses on individual work, specialized skills, managerial decision making, low customer involvement, and larger teams.

Several studies have investigated the effect of teamwork quality (TWQ) on project success in traditional software teams (Hoegl and Gemuenden, 2001; Hoegl et al., 2003; Hoegl et al., 2004; Janz, 1999; Li et al., 2010; Ryan and O'Conner, 2009; Vinod et al., 2009). Hoegl and Gemuenden's (2001) frequently cited study, for example, shows the effect of TWQ on team performance and team membersśuccess for a set of traditional software development teams.

* Corresponding author.
E-mail addresses: ynglin@ifi.uio.no (Y. Lindsjørn), dagsj@ifi.uio.no (D.I.K. Sjøberg), torgeir.dingsoyr@sintef.no (T. Dingsøyr), gunnab@ifi.uio.no (G.R. Bergersen), tore.dyba@sintef.no (T. Dybå).

[1] http://www.agilemanifesto.org.

http://dx.doi.org/10.1016/j.jss.2016.09.028

Due to the lack of studies on the effect of TWQ in agile teams, we conducted a survey on this topic by replicating the study of Hoegl and Gemuenden (2001). Our research questions were:

RQ1: What is the effect of TWQ on the performance of agile software teams?

RQ2: What is the effect of TWQ of team members' success in agile software teams?

RQ3: How does the effect of TWQ on team performance and team members' success differ between agile and traditional teams?

The remainder of this article is organized as follows. Section 2 gives an overview of related work and describes the conceptual model of this work. Section 3 outlines the research method. Section 4 reports the results. Section 5 discusses the results, implications, limitations, and future work. Section 6 concludes.

2. Related work and conceptual model

2.1. Teamwork in software development

Teamwork is obviously important in software development. In traditional development, the study by Faraj and Sproull (2000) showed a strong relationship between management of expertise and team performance. Another study demonstrated the importance of cooperative learning on project success for software development teams (Janz, 1999). In agile development, a few studies analyzed teamwork using team performance models, such as the one found in Moe et al. (2010). Sharp and Robinson (2010) described how agile development teams enable collaboration, co-ordination, and communication. Another study Pikkarainen et al. (2008), focused on how agile development methods improve communication, and claimed that Scrum and XP practices improve both formal and informal communication. Maruping et al. (2009) demonstrated that XP practices of collective code ownership and coding standards could lead to increased technical quality of software products. A survey of success factors of agile development found that team capability was one of the factors (Chow and Cao, 2008).

Detailed models that show relationships between various aspects of teamwork quality and team performance have been used in studies of software teams; for example, those described in Hoegl and Gemuenden (2001), Salas et al. (2005), Dickinson and McIntyre (1997) and Janz (1999). In this work, we focus on the factors described by Hoegl and Gemuenden (2001).

2.2. Teamwork quality (TWQ)

We use the construct of teamwork quality conceived by Hoegl and Gemuenden (2001), which refers only to the quality of interactions. Measures of the task process, the task strategy, and the quality of the performance of the task activities performed by the individual team members are not the subject of this TWQ construct, nor are management activities such as task planning, allocation of resources, or management by objectives.

TWQ is conceptualized as a higher order construct and is based on Hackman's input-process-output model on group behaviour and effectiveness (Hackman, 1987) and derived from McGrath (1964). The six subconstructs of *communication, coordination, balance of member contribution, mutual support, effort*, and *cohesion* cover performance-relevant measures of internal interaction in teams; see Table 1. A more detailed description follows below.

Table 1
The TWQ construct with subconstructs.

Subconstruct	Description
Communication	Frequency, formalization, and openness of the information exchange.
Coordination	Common understanding when working on parallel subtasks, and agreement on common work-down structures, schedules, budgets, and deliverables.
Balance of member contribution	The ability to employ the team members' expertise to its full potential. Contributions should reflect the team member's specific knowledge and experience.
Mutual support	Team members' ability and willingness to help and support each other in carrying out their tasks.
Effort	Team members' ability and willingness to share workload and prioritize the teams' task over other obligations.
Cohesion	Team members' motivation to maintain the team and accept that team goals are more important than individual goals.

2.2.1. Communication

Pinto and Pinto (1990) describe quality of communication within a team in terms of frequency and formalization of the information exchange. Frequency refers to how often communication occurs among team members and how much time is spent on it. Formalization refers to the degree of spontaneity in the communication. Communication that requires much planning and includes written status reports, etc., is considered formal, while spontaneous communication, such as talking in the doorway, chatting, talking in front of the screen, etc., is considered informal. Ideas and contributions are usually shared, discussed, and evaluated with other team members more quickly and efficiently in informal communication than in formal communication. It is also critical for the quality of communication that team members share their information openly with each other (Gladstein, 1984). Lack of open communication may hinder sharing of knowledge and experience that may be relevant for common tasks. In agile teams, the team members are often placed close together in open-plan offices to stimulate informal and open communication.

2.2.2. Coordination

Malone and Crowston (1994) describe coordination as "managing dependencies between activities." Such dependencies include shared resources, task assignments, and task/subtask relationships. Many activities in task processes are delegated to individual members. Harmonization and synchronization of these individual activities are important for the TWQ and project success (Tannenbaum et al., 1992; Brannick et al. 1995). Teams need to agree on common structures for breaking down work, schedules, and effort needed for the tasks. Coordination means that the teams must develop and agree upon a common task-related goal structure that has sufficiently clear subgoals for each team member. In agile teams, tasks are often selected or delegated when planning a new iteration. In a given iteration, some of the "user stories" (requirements) in the backlog are prioritized. A user story is often divided into several tasks. The workload for the tasks is estimated and each task is designed for or selected by one or more of the team members.

2.2.3. Balance of member contribution

The contribution of the task-relevant knowledge and experience of all members to the decision-making processes of the team may benefit the team (Hackman, 1987; Seers et al., 1995). Balanced contribution is critical in software teams with members who have expertise in different areas (core development, GUI development, system architecture, testing, etc.). If only one or even just a few team members dominate the discussions, the others may become less motivated for the work, which in turn may hamper overall team performance. The daily meetings (Stray et al., 2016) in agile teams support such a balance of member contribution.

2.2.4. Mutual support

In software teams (as well as other teams working with innovative projects), the many inter-dependent tasks and the tight col-