Team Effectiveness in Software Development

Human and Cooperative Aspects in Team Effectiveness Models and Priorities for Future Studies

Torgeir Dingsøyr
SINTEF
NO-7465 Trondheim, Norway
torgeird@sintef.no

Tore Dybå
SINTEF
NO-7465 Trondheim, Norway
tored@sintef.no

Abstract—Software development is most often done in teams, where human and cooperative aspects are vital for team effectiveness. This has been the topic of study in several disciplines, and in this article we describe three team effectiveness models from other fields. We discuss priorities for future studies on software teams, and ask: Do we need our own effectiveness model for software teams?

Keywords—Team effectiveness models; software development; agile development; empirical studies; sociotechnical system

I. INTRODUCTION

Software development can be seen as a sociotechnical system comprised of human and technical entities that, when functioning as an integrated, coordinated unit, can address a wide range of problems that are too complex to be addressed by individuals or technologies working alone. However, the design and implementation of complex software systems tend to place primary emphasis on technological innovation without equal consideration for the social component—the teams and groups of software developers—that uses that technology to create the software systems.

Three key issues characterize complex sociotechnical systems. First, such systems adapt to environmental uncertainty through self-managed (viz. “self-regulated”; [1]) processes without a central executive or blueprint guiding that adaptation. Complex systems that are robust to environmental uncertainty, therefore, are adaptive by maintaining a level of autonomy in how they adapt to uncertainty.

Second, technological interventions do not increase sociotechnical system effectiveness if they are not supported by the social (team and group) components of the system. Early sociotechnical research, for example, revealed that isolated “tinkering” with either the technical or social components of the system may not result in the desired improvement of overall system effectiveness [2].

Finally, a third key issue in complex sociotechnical settings is that adaptation emerges from coordinated and collaborative interactions at a team or group level.

Such team interactions are of vital importance in software development, and the recent focus on agile software development [3] has, therefore, led to an increased interest in how to effectively organize small self-managing teams. Members of such self-managed teams are usually responsible for managing and monitoring their own processes and executing tasks. Typically, they also share decision authority, leadership tasks, and responsibilities [4].

Although the majority of studies on self-managing teams report that using such teams has positive effects, some studies offer a more mixed assessment. In addition, research on team performance indicates that the effects of autonomous work groups are highly situational dependent. Further, autonomy on the individual level may conflict with autonomy on the group level. Barker [5], for example, pointed out that self-managing groups may end up controlling group members more rigidly than they do under traditional management styles, while Markham and Markham [6] suggested that it may be difficult to incorporate both individual autonomy and group autonomy in the same work group.

Given these characteristics of software development, and focusing on human and cooperative aspects in software development teams, we would like to discuss the use of team effectiveness models for better theoretical understanding and priorities for future studies of teamwork. The question is: Do we need our own team effectiveness models in software development?

In the remainder of this article, we first describe three models of team effectiveness, which have been developed in other research fields. These are models that we either have used or are planning to use in future studies of software development teams. We describe pros and cons of these models, and finally discuss what we see as priorities for future studies of team effectiveness in software development.

II. MODELS OF TEAM EFFECTIVENESS

There are a number of models of team effectiveness, originating from several disciplines of science [7]. In particular, teamwork has been a much researched topic in management science and in psychology. In the following, we present three models of teamwork that we have used or will be using in future studies on software teams. The two first are taken from psychology, while the last one is taken from management science. These models all focus on team effectiveness, and mainly on internal aspects of the team. There are other models that take into account characteristics of individual team members and the external environment.
In a study which examined how the agile method Scrum facilitates teamwork [8], Moe and Dingsøyr used the model by Salas et al. [9] (Figure 1). This model defines three coordinating mechanisms that are said to exist in all effective teamwork: Shared mental models, mutual trust and closed-loop communication. Given these, five other factors are hypothesized to influence team effectiveness: Team leadership, mutual performance monitoring, team orientation, back-up behavior and adaptability.

A strength of the Salas model is that it originates from a solid literature review. A weakness is that the model is not particularly tailored to self-managing teams, which is expressed as an important goal for agile development teams.

In a further study of the same case by Moe et al. [10] we explained the findings with another model, which is tailored to self-managing teams: The Dickinson and McIntyre model [11], as shown in Figure 2.

This model uses many of the same mechanisms as Salas et al: Team leadership, team orientation, back-up behavior and communication. Monitoring is similar to what Salas et al. describe as mutual performance monitoring, while feed-

Figure 1: Team effectiveness as seen by Salas et al. [9].

Figure 2: Team effectiveness according to Dickinson and McIntyre [11].

back and coordination are other characteristics than what is used in the Salas model.

In management science, Hoegl [12] identified the model described in Figure 3 based on a literature review, which we are currently using for a survey on teamwork in agile development. Teamwork quality is described as consisting of communication, coordination, balance of member contributions, mutual support, effort and cohesion.

The three models were developed for different purposes. The Salas model was developed for the US Army to summarize team research in a practical model inspired by the "big five" personality factors in psychology. The Dickinson and McIntyre model was similarly directed at practical use, but focused on self-managing teams. On the other hand, the Hoegl model was developed for a survey study on the effect of teamwork quality on team performance in innovative projects.

Although the models use different terminology, there are a number of similarities in the mechanisms included. Communication is central in all models; team orientation in the first two models is related to effort (defined as "Do team members exert all efforts to the team's tasks?"). Back-up (behavior) can be found again in mutual support. Hoegl puts more emphasis on the importance of motivation in work effectiveness through the focus on team cohesion. In the other models motivation is embedded in team leadership, which involves "motivating team members" in Salas, and "establishing a positive atmosphere". The Salas model is the only one to explicitly focus on trust and shared mental models as requirements for effective teamwork. Learning and feedback is a characteristic of the Dickinson and McIntyre model.

III. PRIORITIES FOR FUTURE STUDIES OF SOFTWARE TEAMS

Although there are several models of team effectiveness originating from other disciplines, there are many challenges and opportunities for research on software teams, teamwork, and team effectiveness. Building on Salas et al. [13], we see five issues that are fundamental to further our understanding of software team effectiveness, which should be prioritized in future studies.

A. Better Measurement

Although there have been advances in the measurement of team behavior, teamwork quality and team cognition in other disciplines, we still need more robust and valid diagnostic measurement approaches of teamwork and team...
performance in software development that can be practically implemented in the field.

**B. More Rigorous Industrial Case Studies**

Software teams are embedded in organizations and in broader sociotechnical systems. The connections between the team and the other parts of the system definitely affect teamwork and team effectiveness; however, there are few rigorous studies of industrial teams in their fully situated context. Such case studies can enable researchers and practitioners to provide higher-quality context-specific guidance that complement existing theoretical models.

**C. Better Understanding of Dynamic Configurations**

Increasingly, software organizations require dynamic capabilities to build, integrate and reconfigure resources to maintain responsiveness to highly complex and turbulent environments (see e.g., [14]). This entails organizational configurations enabling spatially and temporally distributed resources to be assembled dynamically to meet the changing needs of the software organization. However, merely connecting people with collaborative technology is not sufficient to guarantee effective team performance. Work is needed to better understand the dynamics of distributed or virtual software teams.

**D. Increased Emphasis on Team Cognition**

Although much is generally known about rule-based and automated task performance, far less is known about the complex cognitive tasks typically performed by software teams. Better theories and models of software teamwork and software team effectiveness are therefore needed to better represent and capture this type of collaborative cognition.

**E. Better Understanding of Multicultural Contexts**

As software development globalization progresses, there is an increasing need to better understand the role of culture in software team performance. Currently, the majority of team performance research involves US or Western populations. However, this research raises the question of whether the existing models are representative and useful for software teams with a heterogeneous cultural composition. The degree to which the existing models and frameworks of team effectiveness apply to these multicultural contexts must be assessed, and the models must evolve to include the role of culture in team performance.

In the introduction we asked the question: Do we need our own team effectiveness models in software development?

As with other areas of software development research, we also need more empirical studies and better theoretical grounding in studies of software teams and software team effectiveness.

In this article, we have shown that there exists a large pool of relevant theory in other disciplines, including a number of models of team effectiveness. Research on software development teams should connect better to this pool of research.

With respect to empirical studies of software teams, there is a need to test, extend and update theories from other fields. With the current state-of-the-art in studies of software teams, we cannot say whether we need our own models of team effectiveness in software research, but we need to increase the understanding of the genuine characteristics of software development to know what applies.

**IV. References**


