The Journal of Systems & Software 143 (2018) 59–70

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

Positive affect through interactions in meetings: The role of proactive and supportive statements

Kurt Schneider*[,a], Jil Klünder[a], Fabian Kortum[a], Lisa Handke[b], Julia Straube[b], Simone Kauffeld[b]

[a] Software Engineering Group, Leibniz Universität Hannover, Welfengarten 1, Hannover, 30167, Germany
[b] Department of Industrial, Organizational and Social Psychology, Technische Universität Braunschweig, Spielmannstrae 19, Braunschweig, 38106, Germany

ARTICLE INFO

Keywords:
Positive group affect tone (PGAT)
Proactive and supportive statements
Empirical study
Statistical analysis of mediation and sequence

ABSTRACT

Software projects are dominated by meetings. For participants, not all meetings are useful and enjoyable. However, interaction within a meeting has an impact on individual and group affects. Group affect influences team performance and project success. Despite frequent yet vague dissatisfaction with some meetings, many software engineers are not aware of the crucial importance of their behavior in those meetings. This can set the tone for the entire project. By influencing group affect, meeting interaction influences success without participants even noticing. Due to this lack of awareness, it depends on good or bad luck whether software teams will adopt a promising meeting style.

In a study of 32 student projects with 155 participants, we coded fine-grained interaction elements during the first internal meeting of each team. The analysis of resulting codes showed that constructive remarks had a positive impact on positive group affect tone (PGAT). However, this effect was only observed when constructive remarks were followed by supportive utterances. We were able to show a complete mediation of this statistically significant effect. Seemingly subtle behavior patterns influence group affect. Software projects could significantly benefit from supportive meeting behavior. We propose practical interventions to improve meeting quality.

1. Introduction

Most employees work in some sort of team setting (Kozlowski and Ilgen, 2006): Most of their experiences and work-related behavior are situated in some kind of social context. In order to achieve a better understanding of organizational processes and their dynamics, organizational behavior should be studied within social contexts (Meinecke and Lehmann-Willenbrock, 2015; Kling, 2007; Kling et al., 1997). It is, therefore, meaningful to consider affect on individual and on group levels.

Occurring in all disciplines and hierarchical levels, meetings present such a context. Moreover, meetings take up a large amount of time in the life of most teams (Cohen et al., 2011; van Vree, 2011). On average, employees spend more than six hours per week in meetings (Rogelberg et al., 2006), senior managers up to 23 h a week, with a tendency to increase in the future (Rogelberg et al., 2007). In the United States alone, more than 25 million meetings take place each day (Newlund, 2012).

Software development often encounters organizational and managerial challenges and problems (Herbsleb et al., 1995) due to communication delays or breakdowns (Kraut and Streeter, 1990). Communication and coordination are crucial for effective software projects (Herbsleb et al., 1995; Ghosh et al., 2004). Collocation of team members simplifies both communication and coordination (Teasley et al., 2000). Working in teams often requires trust. Remotely working teams often lack trust which can complicate team work (Zheng et al., 2002). Meeting face-to-face beforehand can positively influence trust due to the participants visibility, the interaction and the provision of social information (Zheng et al., 2002). This way, a single face-to-face meeting can facilitate remote team work (Zheng et al., 2002). Nonetheless, according to Teasley et al. (2000), working in a collocated setup increases productivity.

Employees do not only perform task-related activities, they also experience more or less intense emotions. Emotions, also referred to as affect, influence individuals' perceptions as well as their performance (Barsade and Knight, 2015; Knight and Eisenkraft, 2015), via processes such as enhanced cooperation, action-taking and creative thinking (Barsade, 2002; Klep et al., 2011; Kuhl and Kazén, 1999). Especially the influence of unhappiness has been studied frequently. Graziotin et al. (2017) found 49 consequences of unhappy developers

* Corresponding author.
E-mail addresses: kurt.schneider@inf.uni-hannover.de (K. Schneider), jil.kluender@inf.uni-hannover.de (J. Klünder), fabian.kortum@inf.uni-hannover.de (F. Kortum), l.handke@tu-braunschweig.de (L. Handke), julia.straube@tu-braunschweig.de (J. Straube), s.kauffeld@tu-braunschweig.de (S. Kauffeld).

https://doi.org/10.1016/j.jss.2018.05.001
Received 29 June 2017; Received in revised form 23 March 2018; Accepted 2 May 2018
Available online 03 May 2018
0164-1212/

such as low code quality, low productivity and a low cognitive performance. As a context in which employees frequently interact, meetings constitute a platform for the conveyance and convergence of emotions.

1.1. Missing awareness for emotions and affects in software team meetings

In software development, there are several types of meetings: During meetings with the customer, requirements or business analysts try to elicit and validate constraints and requirements. Back in the development team, requirements must be communicated to developers, and internal team meetings support communication and coordination within a team. Both traditional and agile development strategies use meetings as a central mechanism for exchanging information and opinions. The type of meeting we investigate below is the debriefing meeting of each team after they had the first face-to-face requirements elicitation session with the customer. Thus, the meeting serves for requirements interpretation and for reflecting on the behavior of and their relationship to the customer. In short, we observe a very early requirements engineering meeting in the process.

There are different intensities and profiles of meetings (Liskin et al., 2013). The working week of most software professionals is structured (and interrupted) by several meetings of different kinds. Not every participant is looking forward to a meeting, and sometimes a meeting seems boring or redundant. There is a wealth of consulting literature on conducting efficient meetings in general (Allen et al., 2015). It is obvious that poorly prepared or overly long meetings do not do the project a favor.

The question addressed in this paper goes beyond this general insight: We investigate the social impact and the influence different meetings have on group affect. Group affect is related to project success (Schneider et al., 2015); any influence on group affect can be relevant for the project. Every meeting can have an influence, not only the long or boring ones. As there are several assumptions and findings on certain behavioral patterns within a meeting can influence group affect (Lehmann-Willenbrock et al., 2011), we aim to test a number of these hypotheses by studying a series of student projects within a longitudinal study. In general, meetings have been linked to proximal (e.g. meeting satisfaction) as well as distal outcomes, such as team and organizational success (Kauffeld and Lehmann-Willenbrock, 2012).

1.2. Recurring interaction patterns in meetings

Team members use meetings to exchange information, generate new ideas, and manage relationships (Horan, 2002; McComas et al., 2007; Kauffeld and Lehmann-Willenbrock, 2012). Most importantly, team members interact in order to accomplish teamwork (cf. Schwartzman, 1989). Accordingly, it is this interaction and interdependence between team members that forges them into a team, instead of remaining a group of co-present individuals (Bonito and Sanders, 2011; Meinecke and Lehmann-Willenbrock, 2015).

According to interpersonal theory (Timothy, 1957), social interactions are characterized by reciprocal dependencies between individuals. Behavior shown by one individual limits the options of possible behaviors for other individuals, as certain behavior increases or decreases the probability of certain reactions (Timothy, 1957). Based on these assumptions, statements by one team member are likely to foster or hinder reactions of other team members (Becker-Beck, 2013). We can identify predictable sequences of interaction behavior in groups.

In many software team meetings, recurring patterns of interaction occur: One person supports another; someone makes ironic remarks; others adopt a skeptical attitude, while yet another subgroup remains silent. In the context of meeting effectiveness, dysfunctional communication, such as complaining (e.g. “nothing can be done” or “this has never worked”), has been shown to correlate negatively with meeting satisfaction and organizational success (Kauffeld and Lehmann-Willenbrock, 2012). Unfortunately, negative effects of complaining are not limited to the complainer. Other team members may also experience negative effects, such as general dissatisfaction and negative mood contagion. Lastly, listening to others complain evokes feelings of anger and resentment, thus creating relational conflict (Kowalski, 2002). Previous research (Kauffeld and Meyers, 2009; Lehmann-Willenbrock et al., 2011) has also identified the phenomenon of “complaining cycles” for a number of people who support each other in critical or skeptical remarks. These specific cycles also have clearly detrimental effects on teams functioning. Accordingly, complaining cycles have been linked to a negative group mood (Lehmann-Willenbrock et al., 2011).

As most team members do not know these kinds of phenomena, they will not recognize complaining cycles or other patterns; nevertheless, negative impact will occur. On the other hand, there are patterns that may have a significant positive effect, such as the ones investigated in this paper. For example, functional communication, such as proactive statements (e.g. taking an interest in change or proactivity) or positive procedural statements: those structuring the meeting, such as delegating tasks can also be observed in meetings. These types of statements have been linked to increased meeting satisfaction and team productivity (Kauffeld and Lehmann-Willenbrock, 2012). Moreover, they have also been shown to occur in reinforcing cycles, such as solution-focused cycles (Kauffeld and Meyers, 2009) or interest-in-change-cycles (Lehmann-Willenbrock et al., 2011).

Software engineers and managers often do not know about the importance of emotional awareness and the relevance for project progress (Fountaine and Sharif, 2017). Furthermore, they are usually not aware of positive or negative patterns in meetings. Due to this ignorance, teams may inadvertently run into negative patterns, or they may miss the opportunity to encourage positive patterns. It is the intention of this paper to raise the awareness for seemingly subtle meeting interactions that tend to be relevant for entire projects.

As outlined above, the debriefing meeting during requirements elicitation deals with important requirements information. At the same time, meeting the customer for the first time sets an emotional tone and may cause a reaction in the investigated debriefing meeting. This combination of contents and emotions is typical for early requirements meetings.

Assuming that “lasting patterns of team interaction can appear as early as the first few seconds of a group’s life” (Gersick, 1988), these should be identifiable during situations of early team interaction. This notion has been supported by studies showing that team effectiveness can be determined based on very early team interactions (Zijlstra et al., 2012), potentially even prior to actual work activities (Ericksen and Dyer, 2004). A situation in which these early interaction patterns occur is during a project team’s very first meeting (Ericksen and Dyer, 2004; Gersick, 1988).

1.3. Research gap and contribution

Teamwork is an essential aspect of current software development. Collaboration has been studied by several authors (cf. Storey et al., 2014; Klünder et al., 2017; Hinds et al., 2015; Herbsleb and Mockus, 2003; Olson et al., 1992). The importance and impact of seemingly subtle aspects, such as statements in a meeting, however, was investigated by Olson et al. (1992) from a software engineering perspective before. However, software engineering and psychological viewpoints have been poorly combined before. Graziotin et al. (2014) combined psychological insights and measurements to prove the advantages of happy software developers. We are not aware of any previous work considering psychological aspects in software engineering team meetings. It is the core contribution of this paper to make software professionals and software engineering researchers aware of the measurable influence of proactive and supportive statements. In particular, proactive statements are instrumental but cannot create positive group