Exploring Norms in Agile Software Teams

Viktoria Stray[1(✉)], Tor Erlend Fægri[2], and Nils Brede Moe[2]

[1] University of Oslo, Oslo, Norway
stray@ifi.uio.no
[2] SINTEF, Trondheim, Norway
{tor.e.fegri,nils.b.moe}@sintef.no

Abstract. The majority of software developers work in teams and are thus influenced by team norms. Norms are shared expectations of how to behave and regulate the interaction between team members. Our aim of this study is to gain more knowledge about team norms in software teams and to increase the understanding of how norms influence teamwork in agile software development projects. We conducted a study of norms in four agile teams located in Norway and Malaysia. The analysis of 22 interviews revealed that we could extract a varied set of both injunctive and descriptive norms. Our results suggest that team norms have an important role in enabling team performance.

Keywords: Group norms · Team values · Collaboration · Self-managing teams · Behavioral software engineering

1 Introduction

Teamwork is an integral part of contemporary software practice. Productive collaboration in software teams requires a certain unity in norms. Team norms are emergent, consensual standards that regulate team members behaviors [1]. Productive teamwork carries with it a set of norms such as listening and responding constructively to views expressed by others, giving others the benefit of the doubt, providing support and recognizing the interests and achievements of others [2]. Such norms are important because they promote individual performance, which boosts team performance, and good team performance boosts the performance of the organization. Understanding and influencing team norms is therefore key to building a productive software team [3].

With the emergence of agile development methods, we have also seen a substantial research interest in team-related topics such as communication [4], coordination [5] and self-managing teams [6], to name a few. Despite the increased interest in teamwork and behavioral aspects in software development research, team norms has been largely ignored [7]. This paper seeks to contribute to our understanding of team norms in software development by drawing on studies of norms in other disciplines [8–10]. Our main contribution is the application of team norm categorizations to a case study of four software teams. Not only do we report on particular norms in the software development context, but we also

P. Abrahamsson et al. (Eds.): PROFES 2016, LNCS 10027, pp. 458–467, 2016.
DOI: 10.1007/978-3-319-49094-6_31

hope this contribution will enable more data-driven empirical research in this area in order to improve software processes.

We may think of team norms as shared expectations of how to behave in the team [11]. Norms have the power to partially explain human behavior by expressing our motivation for doing certain actions [8]. Norms are normative in the sense that they associate value to certain patterns of behavior. Norms thereby discriminate between acceptable and unacceptable behaviors of members in a team [12]. Furthermore, norms are a fundamental element of a team's structure and constitute an important vehicle for team members' identification with the team. When team members identify themselves with a team they will more easily commit themselves to team goals [10,12]. One of the most important characteristics of team norms is that they do not exist if they are not shared with others [13]. Norms may promote adaptive and effective behavior because people feel compelled to act in ways that are consistent with the norms. Norms simplify team processes because they make it possible for members to count on certain things being done and other things not being done [12].

A recent study at Google found that some norms, for example the norm that team members speak roughly the same amount, could raise the teams collective intelligence, while other norms could halt the team [3]. A study by Teh et al. suggested that team norms can be adjusted to promote certain behaviors in software teams [14]. In that particular study, group norms were altered using task priming, whereby team members would complete a pilot task under direct guidance to establish new norms. Agile methodologies require a shift from command-and-control management to leadership-and-collaboration [15]. McHugh [16] found that norms influence behavior in agile teams and argue that since traditional bureaucratic controls are often reduced in agile teams, team norms may be even more of importance than in traditional software teams. Further, Moe [17] argue that a software team, in order to become self-managing, needs to change the operating norms within the team, as well as in the wider environment. While developing productive norms are important in co-located software teams, it is even more important in distributed software teams. Sharp and Ryan [18] noted that a crucial element of virtual team design was the establishment of a shared set of norms. They argue that virtual teams benefit from learning to express explicit norms and role expectations to new members.

2 Study Design

The company in which this study was set is an international telecommunications software company with roughly 700 employees. It was selected as a research site because it was part of a large research project on teamwork in agile software teams. We studied two teams located in Norway (Mercury and Mars) and two teams located in Malaysia (Jupiter and Pluto), as shown in Table 1. One of the team members worked in both Mercury and Mars, but identified more strongly with the latter, and therefore we place her in this team in Table 1. All teams had used Scrum with the recommended practices for more than two years.

Table 1. Sample of teams

Team name	Country	Team members	Team members interviewed
Mercury	Norway	9	1 architect, 1 developer, 1 Scrum Master, 1 tester
Mars	Norway	9	3 developers, 2 Scrum Masters, 1 technical writer
Jupiter	Malaysia	10	5 developers, 1 project manager, 1 Product Owner, 1 Scrum Master, 1 tester
Pluto	Malaysia	6	1 architect, 1 developer, 1 team leader

The 22 interviews were semi-structured and the respondents were asked questions regarding teamwork and meetings. The interview guide was based on a teamwork model [19], which covers the following teamwork components: communication, team orientation, team leadership, monitoring, feedback, backup, and coordination. By understanding the teamwork components in each team, it will be possible to understand the patterns of behavior and the influencing norms. Understanding team orientation is of particular interest in this study. Team orientation is defined as [19]: "the attitudes that team members have toward one another and the team task. It reflects acceptance of team norms, level of group cohesiveness, and importance of team membership.".

The average interview duration was 60 min. All the interviews were audiotaped and fully transcribed. The first author also observed the teams in meetings and during daily work. Statements in the interviews regarding daily stand-up meetings have been used in previous work [20], but the information concerning norms was analyzed and reported on for the first time for the study reported in this paper.

The two first authors studied the interview transcripts and observation notes and identified statements that indicated norms - i.e. patterns of behavior. We looked for statements where team members described a behavior as an "unwritten rule" or "how our team does it". All the interviews were coded in NVivo. We decided to use the categorization by Cialdini et al. [8] to understand and analyze two type of norms: *injunctive* and *descriptive* (explained in Sect. 3). We discussed which of the categories the identified norms belonged to, and whether we believed the norms positively or negatively affected team performance. We also considered the framework proposed by Forsyth [1], but found that the categories were not as clear as those of Cialdini et al.

3 Results and Discussion

Upon analyzing our data we identified both injunctive and descriptive norms. Injunctive norms are concerned with what people *ought* to do or *should* do. Such norms describe approved or disapproved behaviors. Descriptive norms are norms of what *most* people do, how they *typically* act, feel, and think in a given

situation. Because what is approved behavior (injunctive norms) is often the same as what is typically done (descriptive norms), it is easy to confuse these two types, but they are conceptually and motivationally distinct [8].

3.1 Injunctive Norms

We found that injunctive norms were the easiest type of norm to identify because the interviewees often expressed these as ways people ought to behave. For example, one developer in Jupiter described a norm of how to dress for work: *"We have to wear long pants, and we cannot wear slippers."* In Teams Jupiter and Pluto they had the norm that "the Product Owner (PO) is not allowed to attend retrospective meetings." A third example is that all teams had the injunctive norm: "team members have to be on time for meetings", and they tried to counteract the tendency to violate this norm with concrete sanctions, such as having to pay a fine. While talking about allocation of tasks, one developer from Team Jupiter noted: *"We have specialized roles in order to go in depth in solving problems and to be able to solve tasks faster."* The expected behavior in this team was that team members chose tasks according to specialization. This behavior was positively sanctioned because the team members believed that it made them more productive. This norm suggests that the team prioritized role specialization at the expense of agile teamwork norms such as having backup behavior and knowledge redundancy [5].

Another respondent in Jupiter commented on the autonomy level of the team: *"The thing is, the differences from now and the early days of Scrum is that we have full design rights. Previously we did not."* The reference is here to the positively sanctioned design behavior, i.e. team members are allowed to design. Design is a part of the work that sets direction for the subsequent coding. This is a norm that contributes to team performance since it brings decision-making authority to the level of operational problems.

One developer in Pluto reflected on a negative incident with a team member: *"Someone actually decided to take up a user story without informing us and then told us it was done before the story was even groomed. It is not ok that team members take such decisions without informing the rest of the team."* The injunctive norm here suggests that team members should not pick up user stories without informing the other team members. This kind of behavior is often referred to as decision hijacking [21] and is an example of violation of a norm. The injunctive norm that team members should inform each other is an enabler for effective teamwork because agile team members should make decisions together.

3.2 Descriptive Norms

Descriptive norms are concerned with the behavior that generally occurs, and these norms are predominantly based in implicit assumptions. Hence, in order to identify these norms, we had to supplement the analysis of the interviews with observational data to identify the usual behavior of the team members. For example, when investigating how the burndown chart was updated, one

respondent in Team Mars replied: *"In Team Mercury, the team members report and the Scrum Master update it. We have concluded that we do not do it like that. We do it ourselves. Each one of us has the responsibility to update it."* This statement suggests a pattern of behavior that is established in the team. However, we observed that the team members rarely updated the burndown chart. Hence, the descriptive norm in the team was to update the burndown chart rarely, even if the project manager wanted them to update it often.

In Team Jupiter, during planning poker, the team member who estimated the highest or lowest number of hours had to give an explanation of his or hers estimate. This had resulted in a norm that most team members tried to estimate a middle value in order to avoid speaking up and explain their value to the others.

Another observation of a descriptive norm in Team Jupiter was that it was ok to be present in team meetings without paying attention, if the team member said they had something more important to do. For example, some team members coded during planning meetings. A consequence of this norm may be a reduced shared understanding of the work and the teams goals, which negatively affects team performance.

In all of the teams, team members often arrived at work just in time for the daily stand-up meeting, even if company policy stated an earlier time. This illustrates an important aspect of team norms: the informally agreed on guidelines for acceptable behavior in a team may conflict with the organization's expectations of behavior. Team members will then find themselves in a position where they, often unconsciously, choose or negotiate between different norms. In this example, team norms got precedence above organizational norms. This may indicate that the team members identified more strongly with the team than the organization.

3.3 Co-existing Norms

Injunctive and descriptive norms may co-exist in the same behavioral pattern [8]. In the beginning of our data analysis this created some confusion. An example of an injunctive and descriptive norm acting simultaneously is the following statement from a developer: *"When I have a problem, I ask for help immediately, I do not try to sit for days trying to solve the problem myself"*. We often observed that team members asked each other for help, either by going to a person sitting close by or by sharing the problem in the daily stand-up meeting. The behavior of seeking and providing assistance from each other was positively sanctioned in these teams (injunctive norm). At the same time, it was what people usually did (descriptive norm).

Another, more intricate example of co-existing norms is illustrated by the following statement from a manager: *"John is not too harsh on the PO, so the PO would always give him new tasks behind the Scrum Masters back. This is how John approaches stuff, so we can just let him. It is not really wrong by the way, he is just doing his part to improve the product."* It is disapproved (and hence an injunctive norm) to allow the PO to approach team members directly without the Scrum Masters consent. Nevertheless, this often happens (descriptive norm.)

Teams go through a natural process of creating norms to find a comfortable way to operate [12]. They try to operate in such a way that they maximize the chances for success and minimize the chances for failure, and that they also maximize the satisfaction of the team members and minimize interpersonal discomfort [22]. For example, the team as a whole are satisfied when they try not to accept tasks from the PO, but at the same time they accept that some team members solve this type of tasks because it minimizes interpersonal discomfort to let this person say yes. Nevertheless, we believe that the injunctive norm (team members should reject tasks from PO) positively affected team performance, while the acceptance of this being violated (the descriptive norm) negatively affected team performance.

3.4 Psychological Safety

Some norms of communication were described by the interviewees as cultural differences. For example, in Malaysia, one tester noted: *"In Norway, the testers would just go to the developers cubicle and just talk to them whenever there is a problem. In Malaysia, maybe the working culture is different, because most of the time we are communicating through e-mail to have it in black and white."* However, we believe that other factors than culture are also important in explaining norms for communication. Norms of how team members behave towards each other are closely related to the concept of psychological safety, which is a sense of confidence that the team will not embarrass, reject or punish someone for speaking up [23].

We identified several norms that indicated a high degree of psychological safety. For example, in Jupiter, the developers had the norm that they responded positively whenever they were confronted with a bug. One tester explained: *"In my previous job I was afraid that developers would be offended when I filed a bug because basically you are telling them that they have made a mistake. So I had to think a lot of how I would present the bugs I found. But, I do not get that feeling in this team because this team is quite mature. The developers are happy if you find a bug. It makes me feel happy about my job and my team."* This supports the findings in a recent study of norms that stated that productive software teams have norms that fosters a high degree of psychological safety in the team [3].

3.5 Changing Norms

A capacity for learning about norms and how to change them is needed to improve team performance. The results of this study indicate that it is important that teams reflect on the two types of norms associated with how they are operating as a team, and how such norms evolve. Norms are socially developed through interactions among team members. As a consequence, they are not static. An intriguing aspect of norms is that behavior that is found effective can gradually be turned into routine, norm-driven behavior [12,24].

Organizations seek to establish norms in different ways by enforcing process standards, code of conduct etc. Similarly, teams will try to establish norms, for example by agreeing on rules to regulate the team's interaction. One example is the set of rules defined by Team Pluto. Figure 1 is a picture of a working agreement that the team had posted on the wall in the office space. However, agreeing on these rules are not sufficient to designate them as norms, they are merely potential injunctive norms. A rule must also be associated with motivation to behave according to this rule (which, naturally, will most often be the case). Hence, we must find evidence of people's inclination to behave according to a certain pattern of behavior to verify if they are indeed injunctive norms.

By discussing the working agreement in Team Pluto, the team tried to establish their own norms for effective teamwork. Teams that are able to improve their own work methods often achieve a higher level of autonomy than teams that do not make such decisions [11]. One way of changing norms in a conscious manner is by reflection. To enable reflection, agile methods typically establish some form of retrospective meetings. We argue that these should be used as a means to discuss team norms.

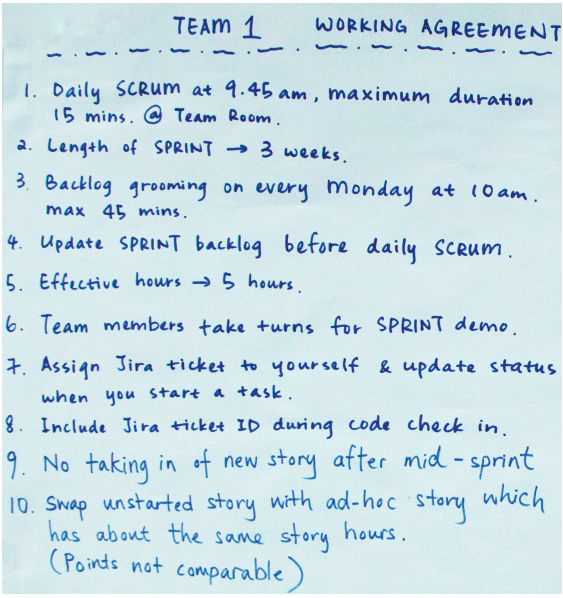

Fig. 1. Working agreement in Team Pluto

In observation of retrospective meetings, we noted several examples of issues related to team norms, for example: (1) How can we make sure people are punctual to grooming meetings? (2) How can we make team members prioritize the retrospective meetings when they are busy preparing for the sprint demo? (3) Should we ban laptops from meetings? and (4) How can we make sure that the

burndown chart is updated more often? By discussing these issues, the teams reflected on descriptive norms and tried to establish injunctive norms that would subsequently be adopted as typical. This shows that rituals and ceremonies such as daily stand-up meetings and retrospective meetings may reinforce acceptable behaviors. Discussing the team's own norms is an example of clan-based control. Often, the team will seek to establish sanctions to uphold these injunctive norms [9]. Clan control is a type of control that operates when the behavior in a team is motivated by shared values and norms [25]. Clan control empowers team members in agile software teams [16].

4 Methodological Implications and Future Work

As far as we are aware, this is the first study of norms in agile software teams. Studying norms is a challenging undertaking because most people do not reflect on how norms guide their behavior. Additionally, they may not be aware of which norms that regulate their actions. We have come to understand norms better through our analysis according to the framework by Cialdini et al. [8]. Perhaps symptomatic for many soft topics in software engineering, we need to be vigilant to opportunities for using theory from other disciplines that explain the practice of software development.

While it seems clear to us that norms are an integral aspect of working in a team, they may be difficult to uncover because of their degree of visibility, as shown in Fig. 2. In order to understand norms, researchers have to uncover assumptions. One may start with identifying artifacts and behaviors in the teams to decipher the underlying sources of motivations, such as norms. Our position

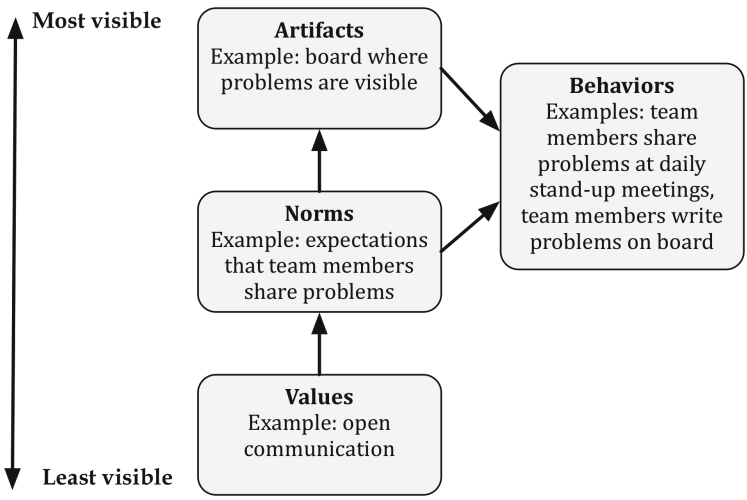

Fig. 2. Visibility of values and norms. Figure adapted from [25], examples from Team Mercury

is that it is not enough to just interview project members to uncover norms, one should supplement this data collection method with field observations to see what people actually do. Research in social psychology can serve as useful examples for future research in software teams [10].

Future research might also explore the concept of team values. Team values guide behavior and decision making in the team and they underlie norms [26,27]. However, team values may be even more difficult to identify because they are even less visible than norms.

5 Conclusion

Productive teams, where team members act in a collaborative manner to achieve project goals, are important for successful software projects. In such teams, team members often exhibit a strong sense of commitment to the team, and members are influenced by shared norms. The purpose of our study was to evaluate the presence of norms in four software teams. The results support the idea that some norms enable team performance, while others hinder. In order to encourage productive team member behaviors, we suggest that teams regularly reflect on both their injunctive norms (what is approved/disapproved behavior) and descriptive norms (what is commonly done). Our contribution can serve as an initial basis to guide and integrate research findings about norms in software teams.

Acknowledgments. We thank Yngve Lindsjørn and Øystein Ingebrigtsen for assisting with data collection. We also thank the participants for sharing their experiences with us. This work was supported by the Smiglo project, which is partly funded by the Research Council of Norway under the grant 235359/O30.

References

1. Forsyth, D.R.: Group Dynamics. Wadsworth, Cengage Learning, Belmont (2010)
2. Katzenbach, J., Smith, D.: The discipline of teams. Harvard Bus. Rev. **71**, 1–11 (1993)
3. Duhigg, C.: What Google Learned From Its Quest to Build the Perfect Team. http://goo.gl/6md7HQ
4. Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., Still, J.: The impact of agile practices on communication in software development. Empir. Softw. Eng. **13**, 303–337 (2008)
5. Moe, N.B., Dingsøyr, T., Dybå, T.: A teamwork model for understanding an agile team: a case study of a scrum project. Inf. Softw. Technol. **52**, 480–491 (2010)
6. Moe, N.B., Dingsøyr, T., Dybå, T.: Overcoming barriers to self-management in software teams. IEEE Softw. **26**, 20–26 (2009)
7. Lenberg, P., Feldt, R., Wallgren, L.G.R.: Behavioral software engineering: a definition and systematic literature review. J. Syst. Softw. **107**, 15–37 (2015)
8. Cialdini, R.B., Reno, R.R., Kallgren, C.A.: A focus theory of normative conduct: recycling the concept of norms to reduce littering in public places. J. Pers. Soc. Psychol. **58**, 1015–1026 (1990)

9. Barker, J.R.: Tightening the iron cage: concertive control in self-managing teams. Adm. Sci. Q. **38**, 408–437 (1993)
10. Terry, D.J., Hogg, M.A.: Group norms and the attitude-behavior relationship: a role for group identification. Pers. Soc. Psychol. Bull. **22**, 776–793 (1996)
11. Levine, J.M., Moreland, R.L.: Progress in small group research. Ann. Rev. Psychol. **41**, 585–634 (1990)
12. Hackman, J.R.: The design of work teams. In: Handbook of Organizational Behavior, pp. 315–342. Prentice-Hall, Englewood Cliffs (1987)
13. Cialdini, R.B., Trost, M.R.: Social influence: social norms, conformity and compliance. In: Handbook of Social Psychology, pp. 151–193. McGraw-Hill, New York (1998)
14. Teh, A., Baniassad, E., Van Rooy, D., Boughton, C.: Social psychology and software teams: establishing task-effective group norms. IEEE Softw. **29**, 53–58 (2012)
15. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies. Commun. ACM **48**, 72–78 (2005)
16. McHugh, O.: A study of clan control in agile software development teams. Ph.D. thesis, NUI Galway (2011)
17. Moe, N.B.: Key challenges of improving agile teamwork. In: Baumeister, H., Weber, B. (eds.) XP 2013. LNBIP, vol. 149, pp. 76–90. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38314-4_6
18. Sharp, J.H., Ryan, S.D.: A preliminary conceptual model for exploring global agile teams. In: Abrahamsson, P., Baskerville, R., Conboy, K., Fitzgerald, B., Morgan, L., Wang, X. (eds.) XP 2008. LNBIP, vol. 9, pp. 147–160. Springer, Heidelberg (2008). doi:10.1007/978-3-540-68255-4_15
19. Dickinson, T.L., McIntyre, R.M.: A conceptual framework of teamwork measurement. In: Team Performance Assessment and Measurement: Theory, Methods, and Applications, pp. 19–43. Psychology Press, NJ (1997)
20. Stray, V., Sjøberg, D.I.K., Dybå, T.: The daily stand-up meeting: a grounded theory study. J. Syst. Softw. **114**, 101–124 (2016)
21. Aurum, A., Wohlin, C., Porter, A.: Aligning software project decisions: a case study. Intl. J. Softw. Eng. Knowl. Eng. **16**, 795–818 (2006)
22. Feldman, D.C.: The development and enforcement of group norms. Acad. Manage. Rev. **9**, 47–53 (1984)
23. Edmondson, A.C.: Psychological safety and learning behavior in work teams. Adm. Sci. Q. **44**, 350–383 (1999)
24. Benner, M.J., Tushman, M.L.: Exploitation, exploration, and process management: the productivity dilemma revisited. Acad. Manage. Rev. **28**, 238–256 (2003)
25. Kirsch, L.J., Ko, D.-G., Haney, M.H.: Investigating the antecedents of team-based clan control: adding social capital as a predictor. Organ. Sci. **21**, 469–489 (2010)
26. Hogan, S.J., Coote, L.V.: Organizational culture, innovation, and performance: a test of Schein's model. J. Bus. Res. **67**, 1609–1621 (2014)
27. Schein, E.H.: Coming to a new awareness of organizational culture. Sloan Manage. Review. **25**, 3–16 (1984)