Understanding work practices of autonomous agile teams: A social-psychological review

Citation for the original published paper (version of record):
Gren, L. (2020). Understanding work practices of autonomous agile teams: A social-psychological review. Lecture Notes in Business Information Processing, 396 LNBIP: 227-235.
http://dx.doi.org/10.1007/978-3-030-58858-8_23

Understanding Work Practices of Autonomous Agile Teams: A Social-psychological Review

Lucas Gren[1,2](✉)

[1] Chalmers | University of Gothenburg, Gothenburg, Sweden
lucas.gren@lucasgren.com
[2] Volvo Cars, Gothenburg, Sweden

Abstract. The purpose of this paper is to suggest additional aspects of social psychology that could help when making sense of autonomous agile teams. To make use of well-tested theories in social psychology and instead see how they replicated and differ in the autonomous agile team context would avoid reinventing the wheel. This was done, as an initial step, through looking at some very common agile practices and relate them to existing findings in social-psychological research. The two theories found that I argue could be more applied to the software engineering context are social identity theory and group socialization theory. The results show that literature provides social-psychological reasons for the popularity of some agile practices, but that scientific studies are needed to gather empirical evidence on these under-researched topics. Understanding deeper psychological theories could provide a better understanding of the psychological processes when building autonomous agile team, which could then lead to better predictability and intervention in relation to human factors.

Keywords: Programming · Social psychology · Agile practices · Teams

1 Introduction

The importance of understanding team autonomy has increased in the last decades due to agile development processes [1]. There have been studies on the barriers of self-organization in agile teams [2], the emerging roles of self-organizing agile teams and how these roles enable agility [3], and the role of senior management [4] to just mention a few. Some authors, like Moe et al. [5], suggest using theories from social psychology to better understand team autonomy. They refer to studies on self-organization in psychology but here are many more theories in that field that would make sense to use in the software development context. The two theories from social psychology, namely Social Identity Theory and Group Socialization Theory where selected from a textbook on social psychology [6]. I do not consider these theories more important than others, they were only selected based on the vast number of studies on them in the last decades. Therefore, they seem to be quite robust and relevant to investigate in the software development context. Some results might replicate, but others might not.

In order to theoretically analyze these two theories from social psychology and their connection to agile teams, I use the study by So [7] in which the author divide common agile practices into *core*, *technical*, *team interaction*, and *customer interaction* practices. Due to saving space, I selected only the *core* and *team interaction* practices (five in total) for this paper. I do not consider these practices to cover all aspects of agility nor to be the most important agile practices, however, the practices chosen are widely used in industry [8].

I will describe the general agile work practices and connect these to existing social, management, and organizational psychology findings, but I will start by presenting the two important psychological theories that are in focus.

2 Important Psychological Theories

I have chosen to focus on two popular theories in social psychology on which none or very few studies exist in software engineering research. The first one is group socialization, which can be defined as the "dynamic relationship between the group and its members that describes the passage of members through a group in terms of commitment and of changing roles" [6]. The idea is that a new team members will go through a certain set of phases through the group's lifespan. The group as a whole will evaluate a new member first by assessing how much a potential new member can contribute to the group's goal-fulfillment. The individual will also assess how much the group can fulfill their personal needs. Step two is commitment, which takes the outcome from the evaluation as input and is an assessment of both parties' beliefs about the rewadingness (i.e. the quality of being rewarding) of the relationship (and other alternative ones). The last phase is role transition in which the commitment reaches a critical level and the relationship thereby changes. These three phases are continuously depending on the result of the assessment. The individual goes through five phases of group socialization: (1) Investigation, (2) Socialization, (3) Maintenance, (4) Resocialization, and (5) Remembrance. These phases have transition steps in-between that are: (1) Entry, (2) Acceptance, (3) Divergence, and (4) Exit [9].

Group socialization is separate from stages that the whole group goes through together. The most famous and used model of group development was introduced in 1965 by Tuckman [10]. He integrated many theories and research findings into a model with the four stages Forming, Storming, Norming, and Performing. The forming stage is when the group is new and need to set the stage and figure our what it is supposed to do and who can do what. Storming is a conflict stage where people now feel safe to question the other team members, which is needed to figure out good group goals and good strategies of work. The Norming phase is when the group starts to set group norms of their collaboration and know how to organize to be productive, and the final stage, Performing, is when the team can focus the most of being productive because they have created a system of good collaboration and effective conflict resolution techniques. These phases are similar to the ones suggested by Agazarian and Gantt [11] in systems centered theory, and what Wheelan [12] also suggested as an integrated model of group development.

Another well-researched approach to explaining many group phenomena is the social identity theory (see e.g. Hewstone et al. [13] or Hogg [6]). Not only has the theory gained empirical evidence in social psychology research but also in quite recent research in social neuroscience (see e.g. van Bavel and Cunningham [14]). To understand that theory, we first need to understand the concepts on which it is based. Social categorization is the classification of people into different social groups, which is a deeply-rooted human trait, and a person's social identity is the part of the self that is derived from the various memberships we have in social groups. Social identity theory is, therefore, the theory of group membership and intergroup relations based on self-categorization, social comparison and a self-definition regarding in-group[1] properties (i.e. a prototype[2]). Self-categorization is how we categorize ourselves and thereby construct a social identity [15]. According to the minimal group paradigm [16], even explicitly random group assignments trigger discriminatory behavior against an out-group[3]. The idea is that a successful intergroup bias creates or protects (high) in-group status, which provides a positive social identity (which in turn satisfies group-members' need for positive self-esteem[4]). Researchers have successfully explained how groups gain positive self-esteem through intergroup bias but have been less successful when explaining intergroup bias motives due to threats or depressed self-esteem [13]. However, Hogg and Williams [15] suggest that competition for positive social identity characterizes intergroup behavior.

3 Agile Practices and Social Psychology

3.1 Iterative Development – A Core Practice of Agile Development

Delivering in short iterations has high face validity, but when broken down, these ideas include a diversity of competences and dynamics needed by the agile team to deliver value in such short iterations. In more general management research, there has been more thorough research on which general work practices contribute to performance (see e.g. Combs et al. [17]) and to successfully implement iterative development, the team must have a high degree and maturity of, for example, staffing, decentralized decision-making, and communication [18]. So to understand the dynamics of iterative development, we should consider these confounding factors before we, as researchers, jump to conclusions about other found effects.

[1] A group that an individual is a member of [6].
[2] Cognitive representation of the typical/ideal defining features of a category [6].
[3] A group that an individual is not a member of [6].
[4] Feelings about and evaluations of oneself [6].

3.2 Iteration Planning – A Teamwork Practice

Obtaining empowered and motivated individuals that have the needed support to solve any given task together with high levels of trust, are all aspects known to be necessary [19] but are not always in place [20]. Creating a shared vision has also been shown in research to be a key for success since the beginning of the 1990s and is one of the main components of transformational leadership [21]. A shared vision is necessary since the team needs an overall goal to break down when planning the upcoming iteration. Regarding the importance of simplicity in agile is somewhat connected to the concept of reducing waste in lean manufacturing, together with the continued avoidance of doing unnecessary activities in the project (or process) life-cycle [22]. To plan in such a way, the team must know the members' real competences and abilities, which also implies maturity in the development process and that the members of the group are committed and fully integrated into the group. With such prerequisites, understanding the group socialization process then becomes paramount when understanding how teams plan in short iterations.

3.3 Stand-Up Meetings – A Teamwork Practice

Developers, but also business people and testers, should be on the same team and collaboratively work together through the whole project life-cycle (i.e. having cross-functional teams). When connecting the popularity of having cross-functional teams in the modern workplace (see e.g. Denison et al. [23]) to social identity theory, it becomes clear that it, in fact, decreases intergroup bias. Having these various organizational functions share their chores and issues often, would be expected to increase cohesion and understanding of the whole project through shared mental models, which have also gained initial empirical support [24]. Having social identity theory and intergroup bias as factors in software engineering research would then probably increase the explained variance.

3.4 Retrospectives – A Teamwork Practice

The idea of a retrospective meeting is that the team should reflect on possible improvement points about their teamwork at the end of each iteration [25]. More generally, such reflective meetings are often called *team debriefs*, and have been shown with scientific rigor to increase effectiveness [26]. McHugh et al. [19] found that these types of meeting need work and careful guidance to function in their intended way also in software development. In a recent longitudinal study, Lehtinen et al. [27] showed that, initially, newly formed teams focus more on task progress and task outcome and, as the teams mature, they focus to a larger extent on process and cooperation. Such findings also relate the "agility" of a team to group socialization and group development since members of the group will behave differently depending on how well integrated they are in the team [9], meaning that a well-integrated individual will be more likely to perform retrospectives in the way they are intended. If the socialization process is not a part of understanding the dynamics of retrospective meetings, studies will have difficulty explaining and predicting patterns of behavior.

3.5 Co-location – A Teamwork Practice

Having the team co-located in the same room with requirements as sticky notes on physical boards have been promoted by the agile community in order to, again, increase the velocity of the development in a rapidly changing environment. Many cases have been reported where the communication challenges of distributed teams have been satisfactory dealt with using modern technology and slightly different practices (see e.g. Berczuk [28]). Another study showed that both agile and traditional projects have the same issues regarding co-location [29]. All-in-all, every social aspect of building relationships will become more cumbersome with distance and implies that more effort is needed to mitigate these challenges [30]. Since the social problems are amplified with distance, failing to understand their influence in distributed agile teams will have even larger negative effects on teamwork. And since agile processes are dependent on the team as a working unit, understanding the social aspects of both distributed and co-located teams are a key to building effective agile teams.

4 Discussion and Implications

As we have seen in this review, there is a lot of overlap between existing knowledge of, and research on, the workplace in general and the agile practices. A few internal organizational examples being decreasing inter-group bias through cross-functional teams [23], striving towards self-organization of teams in order to increase responsiveness to change [18], creating organizational citizenship behavior through shared visions [21], empowerment and trust [31], and removing *waste* in the process [22]. All these aspect are of interest to agile software engineering researchers when trying to understand the development of software using agile teams because these theories might add explanatory power to the observed behaviors.

However, the theory could be seen as complex and hard to grasp for people without any behavioral science education, which means researchers must first run experiments to gather empirical evidence in order to eventually build a theory of "agility," and then provide scientifically founded and validated guidelines to practitioners. One large hurdle of achieving this, though, is the fact that an overwhelming majority of human factors research in software engineering is conducted by software engineering researchers interested in psychology and not psychology researchers interested in software engineering, which often means that the research findings have little depth and offer little new insights from a psychological perspective. I will not cite any studies here due to the fact that such studies were conducted with the best of intentions and do have high value in that they have highlighted the importance of looking at psychological factors in the software engineering domain, which was not the case at all before.

Social identity theory could be utterly useful when navigating through the added complexity of the different social relationship surfacing in an agile project. Hogg and Williams [32] explicitly suggest a set of propositions for how social identity and self-categorization relate to the organizational context. One of their propositions is that changes in which out-groups the in-group compares itself to, will change the view of the group's own identity, including the properties of the ideal member (i.e. the prototype). Another proposition is that harmonious relations between different subgroups of the organization is best kept by recognizing both the subgroups (e.g. Quality Assurance Engineer, Software Developers, Software Tester, etc.) and other organizational constellations, including the teams and the company as a whole. This means that the cross-functional agile teams must recognize both the value of the team as a whole but also the different roles and make distinctions between them. All these aspects should be part of agile team measurements in the future in order to fully make sense of the agile team context.

When looking at the descriptions of the agile practices overall, many of the internal practices seem to assume full group-membership seen from a group socialization perspective [9]. They also assume the entire work-group to be mature from a developmental perspective [10]. In order to fully understand the social-psychological components of the team-based workplace in general, and the agile context in particular, we also need to investigate the temporal perspective of the interplay between group development, group socialization, and the agile approach to projects by setting up autonomous teams.

As have also been shown in this short review, the prescribed behavior in these agile practices are well-founded in social psychology, which provides social-psychological reasons for their popularity. The reason for this is that if the agile practices enable mechanisms known to work well for people in other contexts, it is likely that they would also be appropriate in some variation in the agile context. One example is the decrease of intergroup bias by having cross-functional teams. Therefore, I argue for that these theories should be applied more to the study of autonomous agile teams. In the review by [33], they also call for more theory-based research since the current status of the field mostly comprises method-specific case studies, which is particularly the case in software engineering studies on human factors. In this present study, I have explained some social-psychological underpinnings in relation to five common agile practices, which contributes to founding agile practices in more general social psychology theories. An understanding of such underpinnings can help abstract the essentials of agile software development as opposed to other approaches, but also guide researchers in conducting experiments using theory from social psychology in the software development context. Many of the agile principles are far from new in relation to human knowledge of work-groups. However, what might be considered as having gotten a stronger acceptance is the implementation of being responsive to change. The reasons for not relating agile software development to any existing science outside of software engineering might be due to lack of research knowledge from practitioners, but it might also reflect the difficulty of interdisciplinary research.

5 Conclusion

Without understanding the psychology of groups, agile maturity survey findings are hard to use in order to improve one's own practices. Relating agile practices to deeper psychological theories, like in this study, could instead provide a deeper understanding of the psychological processes of implementing autonomous agile teams.

References

1. Moe, N.B., Stray, V., Hoda, R.: Trends and updated research agenda for autonomous agile teams: a summary of the second international workshop at XP2019. In: Hoda, R. (ed.) XP 2019. LNBIP, vol. 364, pp. 13–19. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30126-2_2
2. Moe, N.B., Dingsøyr, T., Dybå, T.: Understanding self-organizing teams in agile software development. In: 19th Australian Conference on Software Engineering (ASWEC 2008), pp. 76–85. IEEE (2008)
3. Hoda, R., Noble, J., Marshall, S.: Self-organizing roles on agile software development teams. IEEE Trans. Softw. Eng. **39**(3), 422–444 (2012)
4. Hoda, R., Noble, J., Marshall, S.: Supporting self-organizing agile teams. In: Sillitti, A., Hazzan, O., Bache, E., Albaladejo, X. (eds.) XP 2011. LNBIP, vol. 77, pp. 73–87. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20677-1_6
5. Moe, N.B., Dingsyr, T., Kvangardsnes, O.: Understanding shared leadership in agile development: a case study. In: 2009 42nd Hawaii International Conference on System Sciences, pp. 1–10. IEEE (2009)
6. Hogg, M.A., Vaughan, G.M.: Social Psychology, 7th edn. Pearson, Harlow (2014)
7. So, C.: Making Software Teams Effective: How Agile Practices Lead to Project Success Through Teamwork Mechanisms. Peter Lang, Frankfurt am Main (2010)
8. Licorish, S.A., et al.: Adoption and suitability of software development methods and practices. In: 23rd Asia-Pacific Software Engineering Conference (APSEC), pp. 369–372, December 2016
9. Levine, J.M., Moreland, R.L.: Group socialization: theory and research. Eur. Rev. Soc. Psychol. **5**(1), 305–336 (1994)
10. Tuckman, B.W.: Developmental sequence in small groups. Psychol. Bull. **63**(6), 384–399 (1965)
11. Agazarian, Y., Gantt, S.: Phases of group development: systems-centered hypotheses and their implications for research and practice. Group Dyn. Theory Res. Pract. **7**(3), 238 (2003)
12. Wheelan, S.: Group Processes: A Developmental Perspective, 2nd edn. Allyn and Bacon, Boston (2005)
13. Hewstone, M., Rubin, M., Willis, H.: Intergroup bias. Annu. Rev. Psychol. **53**(1), 575–604 (2002)
14. van Bavel, J.J., Cunningham, W.A.: A social neuroscience approach to self and social categorisation: a new look at an old issue. Eur. Rev. Soc. Psychol. **21**(1), 237–284 (2010)

15. Hogg, M.A., Williams, K.D.: From I to we: social identity and the collective self. Group Dyn. Theory Res. Pract. **4**(1), 81 (2000)
16. Tajfel, H., Billig, M.G., Bundy, R.P., Flament, C.: Social categorization and intergroup behaviour. Eur. J. Soc. Psychol. **1**(2), 149–178 (1971)
17. Combs, J., Liu, Y., Hall, A., Ketchen, D.: How much do high-performance work practices matter? A meta-analysis of their effects on organizational performance. Pers. Psychol. **59**(3), 501–528 (2006)
18. Evans, W.R., Davis, W.D.: High-performance work systems and organizational performance: the mediating role of internal social structure. J. Manag. **31**(5), 758–775 (2005)
19. McHugh, O., Conboy, K., Lang, M.: Agile practices: the impact on trust in software project teams. IEEE Softw. **29**(3), 71–76 (2012)
20. Buchanan, D.A.: You stab my back, I'll stab yours: management experience and perceptions of organization political behaviour. Br. J. Manag. **19**(1), 49–64 (2008)
21. Bass, B.M.: From transactional to transformational leadership: learning to share the vision. Organ. Dyn. **18**(3), 19–31 (1990)
22. Hicks, B.J.: Lean information management: understanding and eliminating waste. Int. J. Inf. Manag. **27**(4), 233–249 (2007)
23. Denison, D.R., Hart, S.L., Kahn, J.A.: From chimneys to cross-functional teams: developing and validating a diagnostic model. Acad. Manag. J. **39**(4), 1005–1023 (1996)
24. Stray, V., Sjøberg, D.I., Dybå, T.: The daily stand-up meeting: a grounded theory study. J. Syst. Softw. **114**, 101–124 (2016)
25. Derby, E., Larsen, D.: Agile Retrospectives: Making Good Teams Great. Pragmatic Bookshelf, Raleigh (2006)
26. Tannenbaum, S.I., Cerasoli, C.P.: Do team and individual debriefs enhance performance? A meta-analysis. Hum. Factors **55**(1), 231–245 (2013)
27. Lehtinen, T.O., Itkonen, J., Lassenius, C.: Recurring opinions or productive improvements – what agile teams actually discuss in retrospectives. Empir. Softw. Eng. **22**(5), 2409–2452 (2017)
28. Berczuk, S.: Back to basics: the role of agile principles in success with a distributed scrum team. In: Agile Conference (AGILE), 2007, pp. 382–388. IEEE (2007)
29. Noll, J., Beecham, S., Richardson, I.: Global software development and collaboration: barriers and solutions. ACM Inroads **1**(3), 66–78 (2010)
30. Alzoubi, Y.I., Gill, A.Q., Al-Ani, A.: Empirical studies of geographically distributed agile development communication challenges: a systematic review. Inf. Manag. **53**(1), 22–37 (2016)
31. Wat, D., Shaffer, M.A.: Equity and relationship quality influences on organizational citizenship behaviors: the mediating role of trust in the supervisor and empowerment. Pers. Rev. **34**(4), 406–422 (2005)
32. Hogg, M.A., Terry, D.I.: Social identity and self-categorization processes in organizational contexts. Acad. Manag. Rev. **25**(1), 121–140 (2000)
33. Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: towards explaining agile software development. J. Syst. Softw. **85**, 1213–1221 (2012)