

Agile ways of working: A team maturity perspective

Lucas Gren¹  | Alfredo Goldman²  | Christian Jacobsson³ 

¹Department of Computer Science and Engineering, Chalmers University of Technology and the University of Gothenburg, Gothenburg, Sweden

²Department of Mathematics and Statistics, The University of São Paulo, São Paulo, Brazil

³Department of Psychology, The University of Gothenburg, Gothenburg, Sweden

Correspondence

Lucas Gren, Chalmers University of Technology and the University of Gothenburg, Gothenburg, Sweden.
Email: lucas.gren@cse.gu.se

Present Address

Lucas Gren, Chalmers Lindholmen, SE-412 92 Gothenburg, Sweden.

Abstract

With the agile approach to managing software development projects comes an increased dependability on well-functioning teams, since many of the practices are built on teamwork. The objective of this study was to investigate if, and how, team development from a group psychological perspective is related to some work practices of agile teams. Data were collected from 34 agile teams (200 individuals) from six software development organizations and one university in both Brazil and Sweden using the Group Development Questionnaire (Scale IV) and the Perceptive Agile Measurement (PAM). The result indicates a strong correlation between levels of group maturity and the two agile practices *iterative development* and *retrospectives*. We, therefore, conclude that agile teams at different group development stages adopt parts of team agility differently, thus confirming previous studies but with more data and by investigating concrete and applied agile practices. We thereby add evidence to the hypothesis that an agile implementation and management of agile projects need to be adapted to the group maturity levels of the agile teams.

KEYWORDS

agile software development, group maturity, iterative development, teams

1 | INTRODUCTION

In the context of software engineering, a set of practices is applied to manage complex and rapidly changing projects in relation to their requirements.¹ In order to succeed and be responsive to such change, research and practice advocate an increased focus on collaboration.²

The basic idea of agile software development is that complex projects need to combine managing projects with the need to be able to respond to change. A core practice of any agile method is therefore to develop the product iteratively, meaning that the projects are divided into shorter iterations so that the requirements can be reprioritized continuously. Compared with other management approaches, this is the key difference in agile software development, since it is the only project management method that assumes the end-goal is unknown in detail throughout the project life cycle.³ Another key idea about how such responsiveness to change should be possible is to focus more on social-psychological aspects both within the teams and with customers. Cohen et al⁴ even go as far as to say that "... what is new about Agile Methods is not the practices they use, but their recognition of people as the primary drivers of project success, coupled with an intense focus on effectiveness and maneuverability."⁴ In a secondary study on behavioral software engineering (ie, human and social aspects of software engineering), Lenberg et al⁵ concluded that studies have mostly focused on a few concepts, which have been applied to a limited number of software engineering areas. Wufka and Ralph⁶ also concluded that agility emerges from teams' reactions to needs for change and that agility needs to be understood from a process (ie, temporal) perspective of teams.

Therefore, the contribution of this study is an investigation of a subset of the agile practices' correlation to team maturity. This then adds the team maturity concept, from a collaborative perspective, to the understanding of the use of these agile practices in industry and how to improve them. The remaining part of the paper is structured as follows; in the next section (Section 2), we present a background in connection to both research on agile teams in software engineering and some studies on teams in the psychology field. We motivate the study and suggest hypotheses in Section 3. Section 4 presents the method of the study. Section 5 shows results, and in Section 6, we discuss the results, in Section 7 threats to validity, and in Section 8, we conclude the paper and suggest future work.

2 | BACKGROUND

2.1 | Research on agile teams in software engineering

In recent years, the empirical studies on agile software development have increased and focus on different aspects of agile software development and management.² The areas that have been covered by secondary studies are adoption, methods, practices, human and social aspects, CMMI (an organizational maturity model), usability, global software engineering (GSE), organizational agility, embedded systems, and software product line engineering.² The three systematic literature reviews covering human and social aspects and organizational agility were about dimensions of organizational agility (organizational structures, workforce, development process, management and leadership, and infrastructure),⁷ the role of communication,⁸ and developers' motivation in agile projects.⁹ From both a research and industry perspective, measuring *agility* is of course of very high value. However, the vagueness and breadth of the agile descriptions make a definition of one single construct of *agility* in its broader sense very difficult to achieve. The research on agile software development has, therefore, naturally become split into different subcategories, like what is meant by *agile requirements engineering*,¹⁰ *agile contracting* (see, eg, Hoda et al¹¹), and so on and so forth.

In a study by McDonald and Edwards¹² in the software engineering domain, the authors conclude that as much is derived from the workgroup's context, as of the people in it. Still, most studies still focus on individual psychology, such as the study of Seger et al¹³ on psychological needs connected to software development teams. As a side note, the term *team* is almost used exclusively for workgroups in software engineering research and practice and denotes a small workgroup at an organization with people that collaborate with each other. As previously mentioned, human factors have gotten more attention in software engineering.¹⁴ Lenberg et al⁵ also believe more focus on these factors in research is needed. Feldt et al¹⁵ also argue for the usage of personality tests to put together teams, even though they state that personality cannot be considered in isolation. An indication of this is a more recent study by Cruz et al¹⁶ showing that 40 years of using personality tests in software engineering does not give any coherent results. Furthermore, studies by Hannay et al¹⁷ and Licorish and MacDonell¹⁸ have shown that personality tests have little predictive value and looking at behavior in context could be a better approach if prediction is the goal. We therefore want to use the agile team as the level of analysis in this current study.

Another study was conducted by Moe et al,¹⁹ where they studied a Scrum project that was implemented in an organization. They concluded that transition to self-organizing teams needs buy-in from both developers and managers. Team orientation, team leadership, and coordination, as well as division of work, were factors to consider when moving toward becoming an agile team. The same authors also concluded that challenges when implementing shared decision making were alignment of strategic product plans with iteration plans, allocation of development resources, and performing development and maintenance tasks in teams.²⁰ Hoda et al²¹ also showed the importance of having defined roles in agile teams, which is something stressed as very important in group development research.²² The aspects of all these three studies are parts of group development and therefore motivate studying the relationship between the agile ways of working and group, or team, maturity.

Relating agile software development to psychological theories, also advocated by Lenberg et al,⁵ could provide a deeper understanding of the psychological processes in the agile workplace. Such understanding would then hopefully lead to better predictability and intervention in relation to human factors in agile project but does also increase our understanding of the high capacity of mature groups in relation to dealing with volatile demand.

2.2 | Agile work practices

There are many agile approaches to developing software, eg, eXtreme Programming (or XP),³ Kanban,²³ and Scrum,²⁴ which try implement responsiveness to changing requirement in form of concrete practices. These practices are widely used in industry,²⁵ but it is difficult to find scientific studies on their usefulness in relation to concrete benefits for the developed software other than that customers are overall more satisfied with an agile approach.²⁶ Most descriptions are from popular books with no or very little scientific underpinnings (see, eg, previous studies²⁷⁻³⁵), and the scientific studies on scaled agile is still scarce with only five studies in 2018.³⁶ Some of the practices stem from different fields, like Kanban that comes from Lean Production.³⁷ We have only found one study³⁸ that has explicit connections between some very common agile practices²⁵ and direct team benefits. These practices mostly stem from Scrum, and we will not investigate Lean in particular in this paper. The author presents a quite old (2010) but thorough quantitative study on agile practices and goal commitment, social support, adaptation, and knowledge growth.³⁸

We use the eight agile practices suggested by So³⁸ in this current study. However, we do not consider these the only agile practices that are important for agile teams, we used these eight in this study because we wanted as validated surveys as possible. These practices, or similar variants, are also presented as some of the most common ones in Licorish et al.²⁵ The practices are *iterative development*, *continuous integration and testing*, *stand-up meetings*, *customer acceptance tests*, *retrospectives*, *iteration planning*, and *customer access*.

2.2.1 | Iterative development

The idea of delivering software iteratively is to deliver working software comprising some initial functionality that gives value to the customer at a very early point in time in the development life cycle. No big planning is therefore conducted up-front, but the next increment is decided with the customer continuously.³⁹ Two example items from the measurement of this practice by So³⁸ are as follows: (a) when the scope could not be

implemented due to constraints, the team held active discussions on re-prioritization with the customer on what to finish within the iteration. and (b) working software was the primary measure for project progress.

2.2.2 | Continuous integration and testing

Continuous integration and testing prescribe a continuous integration of source code into the software covered by different tests.⁴⁰ Two example items from the measurement of this practice by So³⁸ are as follows: (a) new code was written with unit tests covering its main functionality, and (b) for detecting bugs, test reports from automated unit tests were systematically used to capture the bugs.

2.2.3 | Stand-up meetings

A common practice to give teams the possibility to coordinate is the daily stand-up. This meeting is time-boxed (a rule of thumb is around 15 minutes which is the amount of time people can “stand up”), and the three standard questions to be answered by every team members are as follows: (a) What have you done since we last met? (b) What are you planning to do until we meet again? (c) What, if any, impediments are you encountering that are preventing you from making progress?⁴¹ Two example items from the measurement of this practice by So³⁸ are as follows: (a) stand-up meetings were to the point, focusing only on what had been done and needed to be done on that day, and (b) when people reported problems in the stand-up meetings, team members offered to help instantly.

2.2.4 | Customer acceptance tests

In a customer acceptance test, the customer gets to see and test the latest working version of the product with some functionality of high priority to provide continuous feedback, but also to be given the possibility to change the requirements of the product.³ This is done in order to incorporate responsiveness to change. Two example items from the measurement of this practice by So³⁸ are as follows: (a) a requirement was not regarded as finished until its acceptance tests (with the customer) had passed, and (b) the customer provided a comprehensive set of test criteria for customer acceptance.

2.2.5 | Retrospectives

In a retrospective meeting, the team members are to reflect on how their collaboration and work is going and find improvements for the next iteration.⁴² Two example items from the measurement of this practice by So³⁸ are as follows: (a) the retrospectives helped us become aware of what we should improve in the upcoming iteration(s), and (b) our team followed up intensively on the progress of each improvement point elaborated in a retrospective.

2.2.6 | Iteration planning

The process of planning an iteration is often called the *Planning Game* in the agile context.⁴³ When planning an upcoming iteration, the team (including the customer or a customer representative) prioritizes and conducts effort estimation on user stories.⁴⁴ Two example items from the measurement of this practice by So³⁸ are as follows: (a) all members of the technical team actively participated during iteration planning meetings, and (b) all concerns from team members about reaching the iteration goals were considered.

2.2.7 | Customer access

Customer access is very more important for agile teams since they are to be responsive to changes in the prioritization of what they are to deliver next.⁴⁵ Two example items from the measurement of this practice by So³⁸ are as follows: (a) the developers could contact the customer directly or through a customer contact person without any bureaucratic hurdles, and (b) the feedback from the customer was clear and clarified his requirements or open issues to the developers.

2.3 | Research on teams in psychology

Keyton⁴⁶ defines a group as three or more members that interact with each other to perform a number of tasks and achieve a set of common goals. This means that many large groups are in fact a set of smaller subgroups and should be handled as separate groups. If the group consists of more than eight individuals, they are less productive than smaller groups, and research findings suggest that small workgroups of three to six members have a much better chance of reaching the higher stages of group development than larger groups.⁴⁷ A *workgroup* is a composition of members that is striving to create a shared view of goals and trying to develop a structure to achieve these goals.

The study of the behavior of small groups was launched with the establishment of a research center for group dynamics in 1946. Several research groups proposed different ways of analyzing the behavior of groups.²² Some studies propose group development can be described as states or levels of activity (cyclic models, see, eg, Bion⁴⁸), but an integrative theory of linear and cyclic models was first introduced in 1964 according to Wheelan.²² A large integrative analysis of various group development models was conducted by Tuckman.⁴⁹ The result of his analysis was a conceptual model including four stages of group development, namely, Forming, Storming, Norming, and Performing. The model suggested by Wheelan²² largely overlaps these stages and is presented next.

2.3.1 | The integrated model of group development

Wheelan²² created an integrated model of group development with four different development stages.

- Stage 1: "Dependency and Inclusion." The first stage is categorized by three main areas; concerns about safety and inclusion, member dependency on the designated leader, and a wish for order and structure. The group is supposed to become organized, capable of efficient work, and achieve goals, so the first state must have a purpose in getting there.⁵⁰
- Stage 2: "Counter-Dependency and Fight." The second stage of a group's development is a conflict phase where fight is a must in order to create clear roles to be able to work together in a constructive way. The members have to go through this in order to be able to trust each other and the leader. When the group safely navigated through the first stage, they have gained a sense of loyalty. As people feel more safety, they will dare to speak up and express opinions that might not be shared by all members.⁵⁰
- Stage 3: "Trust and Structure." The third stage is a structuring phase where the roles are based on competence instead of striving for power or safety. Communication will be more open and task oriented. The third stage of group development is characterized by more mature negotiations about roles, organization, and processes.⁵⁰
- Stage 4: "Work and Productivity." The fourth and final stage (excluding the termination phase) is when the group wants to get the task done well at the same time as the group cohesion is maintained over a long period of time. The group also focuses on decision making and encourages task-related conflicts. This is a time of intense productivity and effectiveness, and it is at this stage the group becomes a team.⁵⁰

The largest contribution of Wheelan and Hochberger²² is probably to connect a questionnaire to the suggested model of group development (the Group Development Questionnaire [GDQ]⁵⁰). In doing so, it has become possible to diagnose and pinpoint in what group stage the group focuses its energy and therefore provides a means to move forward in its development.

3 | RESEARCH GAP AND HYPOTHESES

Gren et al⁵¹ analyzed group development aspects qualitatively and concluded that the definitions of agile teams and mature teams overlap. They also used a version of the agile maturity model suggested by Sidky et al⁵² and showed correlations between Wheelan's²² GDQ Scale IV (work and productivity) and agile organizational behavior. A problem with that study is that they used a modified version of the survey of Sidky et al,⁵² which is quite high level with regard to agility and do not accommodate concrete agile practices used. Their study was also conducted on a smaller sample ($N = 66$), which warrants a replication with validated scales and larger data sets.

No studies have been found that specifically connect the temporal perspective of group dynamics (ie, group development) and concrete agile software development team practices. The GDQ Scale IV has been shown to be correlated to a set of performance measurements in many different fields, eg, an ability to finish projects faster,⁵³ better student performance on standardized test (SAT scores) if faculty team is mature,^{54,55} and lower mortality rates in intensive care units performing surgery.⁵⁶

The agile practices described above are explained in a way that they seem to be tailored for quite mature workgroups from a group developmental perspective,^{57,58} ie, it is hard to see how they would be implemented in very immature groups, for example. 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 and the agile approach to projects. In addition, software engineering provides a context where projects are highly complex and rapidly changing, which is somewhat a new context of research for more general small group research. The existing research on group effectiveness has also been criticized for having been built on a relatively small number of groups from specific contexts.⁵⁹ It is therefore important to apply a temporal perspective to the agile team dynamics, since team agility might be dependent on the maturity of workgroups and therefore need different and contextualized implementations. In this present study, we therefore selected to focus on this perspective and opted to use the GDQ, since it is built on the empirically tested idea that groups develop across time.⁵⁸ It is also important to note that a team that has been stable and met over a longer period of time is more likely to have matured,⁵⁸ although group can get stuck in the earlier stages.²² Even though we used a cross-sectional design, the result of the GDQ has a built-in temporal perspective and is a measurement of where a small group is in its development.

We have only found one study that has summarized a set of agile practices that the authors argue are common to most agile methods, created a scale for each practice, and validated these factors by using psychometrics, which is the study by So and Scholl⁴⁰ presented above. In this present study, we therefore investigate the connections between these two measurements. Since we, at this point, do not know the relationship between dependent and independent variables, we chose to conduct a bivariate correlation analysis. We do not imply that these seven agile practices cover the concept of agility by any means, this study is just a first step toward understanding these connections and we only argue that the practices suggested by So and Scholl⁴⁰ at least cover some aspect of agile ways of working.

Hence, we set up the research hypothesis that any of the seven agile practices as defined and operationalized by So and Scholl⁴⁰ (*iterative development, continuous integration and testing, stand-up meetings, customer acceptance tests, retrospectives, iteration planning, and customer access*) are correlated to *group maturity* as defined and operationalized by Wheelan.⁵⁰ The null and the seven alternative hypotheses for each agile practice are as follows:

- The null hypothesis (H_0) in all cases is that the correlation coefficient is zero, ie, that there is no significant correlation between Team Maturity (tm) and the other variable.

$$H_0 : \rho_{tm,id} = \rho_{tm,cit} = \rho_{tm,sum} = \rho_{tm,cat} = \rho_{tm,r} = \rho_{tm,ip} = \rho_{tm,ca} = 0$$

- H_1 . There is a significant correlation between Team Maturity and Iterative Development (id).

$$H_1 : \rho_{tm,id} \neq 0$$

- H_2 . There is a significant correlation between Team Maturity and Continuous Integration and Testing (cit).

$$H_2 : \rho_{tm,cit} \neq 0$$

- H_3 . There is a significant correlation between Team Maturity and Stand-Up Meetings (sum).

$$H_3 : \rho_{tm,sum} \neq 0$$

- H_4 . There is a significant correlation between Team Maturity and Customer Acceptance Tests (cat).

$$H_4 : \rho_{tm,cat} \neq 0$$

- H_5 . There is a significant correlation between Team Maturity and Retrospectives (r).

$$H_5 : \rho_{tm,r} \neq 0$$

- H_6 . There is a significant correlation between Team Maturity and Iteration Planning (ip).

$$H_6 : \rho_{tm,ip} \neq 0$$

- H_7 . There is a significant correlation between Team Maturity and Customer Access (ca).

$$H_7 : \rho_{tm,ca} \neq 0$$

where ρ = correlation coefficient.

4 | METHOD

4.1 | Subject selection and data collection

This study was carried out at seven organizations (see Table 1 for an overview). Three companies and one university from Brazil and three companies from Sweden participated in the study. All the participating companies stated they had been using an agile approach in their software development for at least 3 years but with teams of different maturity levels in that process. The companies were found through our research network, and the second author was teaching the course with the student sample. Since the student sample did not have lower maturity than practitioners on any measurements (see below), we opted to analyze all data in conjunction. The first author did all the data collection across all the participating organizations and did not know any of the contact persons before this research started. The data collection took place between February 2015 and June 2016.

The Brazilian subsample contains data from IT departments at one large online media and social networking enterprise with around 5000 employees, one smaller software life-cycle consultancy company with around 35 employees, and one company that offers programming courses to individuals and companies with around 100 employees. We also collected data from the University of São Paulo including software engineering students enrolled in an agile software development project course where they developed software in small groups using an agile approach. We collected 132 responses from 22 groups in total. One response sheet lacked information of what team that person belonged to. The 22 groups had a total number of 142 group members that received the surveys via their managers but the replies were anonymous. The response rate was 93% for the Brazilian subsample.

The Swedish subsample consisted of 12 groups across three large organizations: one being a multinational networking and telecommunications equipment and services company (around 115 000 employees), the second an aerospace and defense company (around 14 000 employees), and one automotive parts manufacturing company (around 160 000 employees). The 12 participating teams consisted of 68 agile team members, and the survey was initially distributed to 77 people. The response rate for the Swedish subsample was therefore 88%.

In total, the survey was distributed to 219 agile team members and 200 responded, hence an overall response rate of 91%. This high response rate was due to the fact that the surveys were distributed in paper form during a special session on site at each company. A total of 34 agile teams participated in this study, and the team sizes ranged from three to 11 members for the participating teams.

Since we wanted to look at connections between averages on a team level, we did not differentiate between different types of team or roles in the organizations. Nor did we look at where they were placed in the organization or the content of their software development tasks, or at differences between the two countries due to the small Swedish subsample. However, having data from two different continents strengthens the external validity.

4.2 | Operationalizations of group maturity and a subset of agile practices

The Scale IV of the English GDQ⁵⁰ was used to assess maturity of groups in this study. The 15-item Scale IV of the GDQ is rated on a Likert scale from 1 (Never true of this group) to 5 (always true of this group) and measures the work and productivity aspects of a workgroup. Due to copyright reasons, only three example items can be included here: (a) the group gets, gives, and uses feedback about its effectiveness and productivity; (b) the group acts on its decisions; and (c) this group encourages high performance and quality work.

TABLE 1 Participant information

Organizations	Business	Number of Employees	Participating Teams	Participating Individuals	Country
Company 1	Online media	5,000	3	22	Brazil
Company 2	Software consultancy	35	8	57	Brazil
Company 3	Online training platform	100	3	11	Brazil
University students	Agile course	43 (enrolled students attending)	7	43	Brazil
Company 4	Telecommunications	115 000	4	21	Sweden
Company 5	Defense	14 000	4	24	Sweden
Company 6	Parts manufacturing	160 000	5	22	Sweden
Total			34	200	

In order to get a full profile of the group development, the whole 60-item survey for the four different stages needs to be used. However, immature groups will get lower scores, and mature groups will get higher scores on Scale IV. Also, previous group development studies have shown the mean of Scale IV to be significantly correlated to other external measurements of effectiveness (eg, previous studies^{53-56,60}). The full GDQ was also used in a subsample of the data and tested the hypothesis that Stage II (counter-dependency and fight) would be negatively connected to the agile practices, which was confirmed.⁶¹

To measure agile practices we used the survey developed by So and Scholl,⁴⁰ which was also in English. As mentioned, it is the only survey we have found that is validated through a factor and reliability analysis ($N = 227$). The measurement comprises items and factors for the seven agile practices *iterative development* (seven items), *continuous integration and testing* (nine items), *stand-up meetings* (five items), *customer acceptance tests* (five items), *retrospectives* (six items), *iteration planning* (seven items), and *customer access* (four items). All items for each scale can be found in So and Scholl,⁴⁰ and in their study, the scales had Cronbach α s ranging from .82 (good) to .95 (excellent). In our present study, the α s were between .72 (acceptable) and .97 (excellent), and the lowest value was for the practice stand-up meetings in both cases. The scale for assessing Cronbach α s is that less than .5 is unacceptable, between .5 and .6 is poor, between .6 and .7 is questionable, between .7 and .8 is acceptable, between .8 and .9 is good, and more than .9 is redundant and not preferred, ie, too many similar questions.⁶² The seven scales are replied to on a Likert scale from 1 (never) to 7 (always). We have already presented two example items from each practice in Section 2.2.

4.3 | Data analysis

In order to analyze the connections between the agile practices and group maturity, we ran separate and bivariate correlation analyses on all seven agile variables and GDQ IV, using the mean of all the sums for each scale. For each scale (set of items), like GDQ IV and iterative development, instead of taking the mean value for all the items in the scale as a measurement for that individual's score, we sum up the item answers instead. The mean of the sums is then all the individuals' sums in each team divided by the number of individuals. To clarify, for each team, we calculate $\frac{\sum_{j=1}^m (\sum_{i=1}^n item_i)}{m}$, where n is the number of items in a scale (eg, team maturity), and m is the numbers of team members. The reason for using sums of item responses for each scale was due to the fact that all the previous studies on the GDQ uses the sums instead of only the means, which allows a comparison between studies. The Pearson product-moment correlation coefficient was used as a measurement of correlations between mean values of the 34 groups on all scales, except *iteration planning* and *customer access*, since the Shapiro-Wilk tests of normality in data were significant for these variables ($Test\ Statistic = .850, P < .000$, and $Test\ Statistic = .924, P = .021$). For these two variables, we instead ran Spearman ρ correlation analyses. Since we did not know the direction of the correlations, we opted to use two-tailed tests. To assess the size of the effects, we compared the coefficients to the guidelines suggested by Cohen.⁶³

We also ran Mann-Whitney U tests to see if there were any differences between countries and between students and practitioners. The results from the Mann-Whitney U tests to see if there were any differences between countries and between students and practitioners, we found a significant difference between Sweden and Brazil on the *customer access* practice ($W = 215, P = .003$). The Swedish teams had less customer access which was most likely due to the sizes of the participating Swedish companies. Regarding differences between students and practitioners, we found that students had significantly higher on *customer access* ($W = 26, P = .004$) and *iteration planning* ($W = 29, P = .004$) than practitioners. There were no significant differences on any other agile practices nor in team maturity across countries or type of respondent (student or practitioner teams). The fact that students scored higher on two of the agile practices led us to assume that they were at least as mature in using them as the practitioners, hence we kept them in the analysis.

5 | RESULTS

The mean of all the mean sums and their standard deviations for the 34 groups on the used scales are shown in Table 2. As can be seen in Table 3, more mature teams make the intended use of the following practices ($P < .01$): (2) *iterative development* (.647**) which is a strong correlation (or a large effect), and (6) *retrospectives* (.490**), which is a medium correlation/effect but very close to a large effect of .5. Thus, we reject the null hypothesis in favor of H_1 and H_5 and fail to reject the other hypotheses. The size of the effects were compared with the effect size guidelines as suggested by Cohen.⁶³ This means that a higher group maturity is connected to a better implementation of *iterative development*

TABLE 2 Descriptive statistics for the included variables

Measure	Mean of the Teams' Mean Sums	Standard Deviation of the Teams' Mean Sums
1. GDQ Scale IV	60.0	4.4
2. Iterative development	37.4	4.1
3. Continuous integration and testing	39.0	7.0
4. Stand-up meetings	26.7	3.5
5. Customer acceptance tests	19.2	5.5
6. Retrospectives	29.6	6.0
7. Iteration planning	38.4	5.1
8. Customer access	19.7	5.3

Measure	1	2	3	4	5	6
1. GDQ Scale IV	1					
2. Iterative development	.647**	1				
3. Continuous integration and testing	.314	.569**	1			
4. Stand-up meetings	.155	.450**	.184	1		
5. Customer acceptance tests	-.040	.166	.310	.367*	1	
6. Retrospectives	.490**	.511**	.517**	.234	-.316	1

* $P < .05$.** $P < .01$.**TABLE 3** Pearson correlations of variables (N = 34 teams)**TABLE 4** Spearman correlations of variables (N = 34 teams)

Measure	1	2	3
1. GDQ Scale IV	1		
7. Iteration planning	.324	1	
8. Customer access	.012	.263	1

and *retrospectives*. It also means that if the group is less mature, ie, obtained lower values on GDQ Scale IV, these two agile practices will probably not work as intended. However, since a bivariate correlation analysis was conducted, we do not know the direction of the effect, and these two agile practices might instead lead the group to mature.

The result of the two non-parametric correlation analyses are shown in Table 4. The results show that none of the variables were significantly correlated to GDQ Scale IV. Another interesting aspect of the correlations is that all we can confirm So's³⁸ result that *iterative development* seems to be at the core of agile software development since *continuous integration and testing*, *stand-up meetings*, and *retrospective* were all highly correlated to this variable.

6 | DISCUSSION

The results of this study show that there are strong connections between a group's maturity, as represented by the *GDQ Scale IV* measurement and two concrete agile practices. The significant correlations found between *iterative development* and *retrospectives* and *GDQ Scale IV* give us insight into what agile practices are connected to the group maturity level in order to function in the intended way. In addition, the agile practice of developing software iteratively is the core of agile software development, according to So,³⁸ which we have confirmed in our result. This also means that the behavior connected to the other practices *continuous integration and testing*, *stand-up meetings*, *customer acceptance tests*, *iteration planning*, and *customer access* could work well even in the early phases of new agile teams, since we fail to reject these null hypotheses, but they are still linked to the iterative development of software. We need more studies on these variables to conclude on their seemingly quite complex intercorrelations and how they interplay causally. Even though we want to highlight the fact that failing to reject the null hypothesis does not lead to the conclusion that it is true. Regardless of this fact, the items measured in the practices *iterative development* and *retrospectives*, as suggested by So and Scholl⁴⁰ are in connection to higher requirements regarding the group maturity as compared with the other practices measured, by asking about, for example, the degree of active discussion, areas of improvement, and evaluation of solutions instead of more technical or process-related practices. The variable with almost a significant effect was *iteration planning*, which also includes items more related to how active group members are in the team. In addition to the *retrospectives*, the *stand-up meetings* are also a synchronization of team members, and we would not have been surprised if this practice was also correlated to group maturity. However, the way this practice is measured in the survey by So and Scholl,⁴⁰ we see that the items regarding *stand-up meetings* are shallower than for the *retrospectives*, for example, with question on if the meeting was "to the point" without details in the expected behavior. Maybe these items do not fully reflect the intended behavior in a stand-up meeting, but this remains to be explored.

Since a bivariate correlation analysis was applied, we do not know the direction of the effect, ie, which variables are the cause and effect, and therefore, team agility might be dependent on group maturity or the agile practices might lead to the group maturing. In order to investigate

the causality between these measurements, longitudinal studies are needed where the effect is preceded by the cause. The results must also be consistent between studies, and eventually, a theory is needed that describes the connections between the constructs, which then can be tested empirically.

The results of this study therefore contribute to our understanding of why a transition to a more agile approach is more difficult for some teams at certain points in time. One could argue that most suggested processes of effective teamwork assume mature teams,⁶⁴ but it is evident that there is more of a team focus in the agile approach compared with others,⁶⁵ ie, groups dynamics is more important to understand in agile team than traditional ones. In summary, this provides an answer to the question (the second research objective) of which (if any) of the selected agile practices are connected to team maturity in order to function as intended.

6.1 | Implications for psychology

This study links a more well-known concept within organizational psychology, namely, group development, to new work processes that evolved in practice in a new type of industry, namely, software development. Because the group development measurement was linked to two core practices in agile development, this means that a group of higher maturity can then be more responsive to change, which is not explicitly described in group development theory.²²

6.2 | Practical implications

Our results imply that including aspects of group maturity could increase the understanding and predictability of agile implementations. We would like to highlight here that a low team maturity could as well be due to external factors in the company ecosystem outside of the team's control. This aspect is also highlighted below under Organizational Support. The result above entails that leaders and managers could be helped by preparing for the disparity of agile practices in use and accept this as a natural part of the group's development. Practitioners should be aware of that the group could make different use of the agile practices when less mature and only leverage these practices when the group has developed further, which is something we have already seen indications of in software engineering research.⁶⁶ Even if some practices will be used differently in the first compared with the latter stages of group development, this does not imply that these practices should not be implemented in the beginning. Rather, such practices are most likely needed for the group to develop since, eg, *retrospectives* give the group a forum to reflect on group work, which is needed in all stages. However, the content will differ since the group has different issues to work on from a group developmental perspective. To clarify, the correlations shown in this study connects the two constructs of agility and team maturity, which implies that agile teams can use the group development model and its appended interventions in order to become more responsive to change in their software development.

6.2.1 | How to create an effective team

In order to give more concrete recommendations of what agile teams can do if wanting to improve their team maturity, we have summarized the 12 keys to productivity for groups as presented by Wheelan.⁶⁷ These points apply to teams in general, but based on the results of this current study, working on these aspects could increase the agility of teams (even if the causality aspect needs further investigation). For more details and references, see Wheelan.⁶⁷

- Goals
- Roles
- Interdependence
- Leadership
- Communication and feedback
- Discussion, decision making, and planning
- Implementation and evaluation
- Norms and individual differences
- Structure
- Cooperation and conflict management
- The shared responsibility
- Organizational support

These aspects are very similar to the theory of Process Consultation created by Schein⁶⁸ and backed up by an extensive study by Senior and Swales⁶⁹ on what team building activities actually work. Working with any of these aspects as an external resource can be defined as an intervention by a consultant, and in the consultancy theory, these aspects have been known for decades.⁷⁰ We will now go through them one by one and briefly explain what needs to be done.

Goals

That members are clear about team goals is the single most important part of high-performing teams. Teams often seem to know their groups but team members usually have different ideas about what accomplishment acquires, and even if they use the same words, they can have a different meaning to different people. The only way to make sure members understand the goal is to discuss it thoroughly. Without being clear about group goals, there is no meaning trying to accomplish them. However, knowing the goal is not enough; in high-performance teams, members actually agree with the team goals. That highlights the importance of goals being relevant and preferably important, reasonable, and attainable to benefit both the team and the organization.⁶⁷

Roles

After the goal is defined, the group can get organized and decide what needs to be done and who does what. The most important thing is that each member really knows what their role is, independently of if they volunteered for the role or not, ie, both the expectations and the process need to be clear. This clarity of roles and process refer to clarity within the teams and does not imply the same roles and processes across the organization. Secondly, the person taking on the role must have the ability and skills that is needed for the tasks. Thirdly, the whole group must agree on all the different roles since they are a part of a whole solution.⁶⁷

Interdependence

In high-performance teams, the tasks demand that members work together as a unit or in subgroups to reach the goal. This means that the group should not be too big.⁴⁷ The goal should also not be possible to reach without team work.⁶⁷

Leadership

In a successful team, the leader changes her leadership style according to emerging group needs. The leadership style also adapts to the group development stage. One important aspect of leadership is that focus needs to be drawn from the leader to the group. It is important to connect the aspect of culture as being the other side of the same coin as leadership, with the success of a group. All members of a group are an essential part of its success.⁶⁷

Communication and feedback

One of the main characteristics of a high-performance team is that it has an open communication structure, which allows all members to actively participate. This means that no matter which race, sex, ethnicity, profession, etc a person has, their idea is still heard. This is easier said than done because of a cultural legacy in society which promotes and sustains inequality to keep the already existing power structure.⁷¹ High-performing teams also get constant feedback on their productivity and effectiveness both internally and from external resources and use this feedback to make improvements to the group work. That is, it is not enough getting the feedback, but utilizing it to make changes is essential. Constructive feedback is also a key aspect, since it promotes improvement and individual development. This type of feedback needs to be strictly connected to making the group even more effective. If the feedback is unrelated to improving group work, it should not be given.⁶⁷

Discussion, decision making, and planning

The best teams spend time planning how they will solve problems and make decisions. This means that they know how decisions are supposed to be made, before they make them. It does not matter if the decision-making strategy is consensus or majority, etc, it is only important that the rules of engagement are defined beforehand. The strategy itself also needs to be generally accepted and effective. Another aspect is that these teams spend time defining and discussing the problems they must solve. These teams actually spend more time preparing for future work compared with other teams. This takes more time in the beginning, but this time is easily made up for when tiresome strategic disagreement can be avoided every time a decision needs to be made.⁶⁷

Implementation and evaluation

To get a well-functioning team, you need to implement the solutions and decisions made. This means that it is necessary to follow up on decisions and evaluate their success. Decisions can be unfavorable, but in high-performance teams, they are altered when this information is available to the team.⁶⁷

Norms and individual differences

Group norms are utterly important for effectiveness. This result has also been replicated in software engineering research.⁷² High-performance teams have norms that encourage high performance, quality, and success. The aspect of what is expected of the team is key. If the group is not

expected to perform, the probability of success is lower. Successful teams also accept differences in people as long as their behavior helps task accomplishment.⁷³

Structure

The structure of the team is important for three reasons. The team members must all contribute, and therefore, a successful team only consists of the smallest number of members necessary to reach the group goal. The group must also allow subgroups to form to work on smaller chores. These subgroups are not seen as a threat to the group, but as necessary and valued for their contribution to the team. One aspect often not mentioned are the resources available to the team. A high-performing team is organized but also given enough time together in order to develop and maintain this high performance.⁶⁷

Cooperation and conflict management

A high-performing team consists of cooperative members and has great cohesion. However, groups can cooperate in order to not work. High-performing teams have conflict frequently, but these periods are short and task oriented. The key is that these teams have well-working conflict management strategies.⁶⁷ How to manage conflict effectively has been known for quite some time.⁷⁴

The shared responsibility

A leader (informal or formal) can affect a group's development vastly but is not solely responsible for the success of the group. That is why it is so important to recruit people with professional goals that are aligned with the organizational goals. The individual also needs room for self-fulfillment within the scope of the employment.⁷⁵ Every team member is responsible and can make a difference in how well-functioning a group becomes by doing what they can to help the group develop.⁶⁷

Organizational support

In order to get high-performing teams in an organization, the organization itself needs to give the teams the right support. Groups have difficulty in performing well if not placed in a favorable organizational culture. The organization needs to clarify the mission, support innovation, expect success, value superior quality and service, pay attention to detail, value team recommendations, set clear expectations for group output, quality, timing, and pacing and reward teamwork rather than individual performance.⁶⁷

It is also necessary to give the groups what they need to in order for them to do their best. Some of these aspects have been mentioned before and have to do with goals and tasks and continuous learning. It is also important that the groups get the human resources required to deliver, as well as technical. Groups should have a defined work area.⁶⁷

As mentioned before, the effectiveness of a team is dependent on all the group members' joint effort. Members need to be recruited to the project based on their skills needed for group success. It is also important to train members in group participation. However, all training needs to be based on research evidence and proven to work. There are plenty of companies offering group development that has no proven effect.⁷⁶ According to Sundstrom et al⁷⁷ and Guzzo et al,⁷⁸ the most effective interventions are goal setting and feedback that includes attention to group development issues adapted to the current group stage they are in.

Another interesting aspect of organizational support is that it should not be too extensive. If a group has too many helpers, the group members might not learn to be effective and productive on their own. Groups need feedback so that they can help themselves. The groups need enough autonomy to get the work done but at the same time stay connected to the organization and regularly have reviews on the organizational support.⁶⁷

In summary, the above recommendations from group psychology and, specifically, group development theory and practice are not dissimilar to the descriptions of agile teams. This is no surprise, but to work on team maturity to get more agile teams is not always well known to agile practitioners.

Another direct consequence of focusing on team maturity when building agile teams is to not see software developers as technical resources only that can be moved between different teams without any negative psychological effects, especially since stable teams over time are more likely to have matured.⁵⁸ Instead, it is important to take the group development into account when building and maintaining software development teams.

7 | VALIDITY THREATS

In this study, we used the most validated agility measurement survey we could find, but of course, the operationalization of agility as a construct is a great threat to construct validity of the measurement used. We believe the measurement captures the important aspects of what is meant by agile behavior, but we cannot be entirely sure of this without further studies on the content validity. Earlier research has also pointed out the difficulty of measuring agility (eg, Gren et al⁷⁹). Another threat is that we assumed that the participating organizations were actually using agile

practices correctly, which might not have been the case. We did not observe the teams using the practices and only confirmed their use through our company contacts. However, the surveys used to measure the agile practices do not make sense without some introduction of the practices to the terms, and we also tried to capture the degree of their implementation, which makes the output relevant for a correlation analysis.

Another threat is that we only looked at factors internal to the teams. Team maturity is, as we have seen, also dependent on external factors to the teams themselves that then are reflected in less maturity. We only know how these agile practices are perceived in connection to a self-assessment of the team members' own teams; hence, we do not know what external factors that could affect both factors.

We did not look closely at the boundaries of the teams or the different roles often included in software development teams (like testers, developers, architects, and leaders). It would be interesting to look at role composition in the software development context in connection to group development. Another threat to the construct validity is the fact that all questionnaires were filled out in English and not in the native languages of the participants, ie, Portuguese or Swedish. This could mean that the participants consistently misinterpreted some words or expressions. However, since we collected data from two different countries with two very different native languages, that potential systematic error is decreased.

We would like to state, again, that broad generalizations of our result should be done with care, due to our smaller sample size. In addition, we employed a cross-sectional research design, which gives us difficulty in drawing any conclusions on causal relationships between our two measurements.

8 | CONCLUSION AND FUTURE WORK

In conclusion, the purpose of this study was to investigate the relationship between group maturity and agile software development practices. The result shows that two of the core agile practices were correlated to group development maturity, ie, higher levels of group maturity were connected to better use of the agile practices *iterative development* and *retrospectives*. Despite our small sample size of 34 groups, we believe our study shows an indication of how some agile practices are connected to group maturity, and through these results, contributes to the emerging new research field of connecting social psychology to software engineering.

For future work, we particularly suggest more studies on possible mediators and moderator in the connection between agility and team maturity. There seem to be many confounding factors that need carefully planned survey studies or experiments. We also suggest the use of qualitative data since such data are very rich and can unveil both causal relationships and more confounding factors.

In this study, we found that students obtained higher scores on the practices *customer access* and *iteration planning*. The first one could be due to more of the participating companies being larger, but an investigation into why students scored higher on these practices would be interesting to conduct.

Future research should also include larger sample size in order to draw conclusions for the general population of agile software development teams and investigate these concepts over time in longitudinal studies, as mentioned above. For example, could we increase agility by helping the group in its group dynamic development? Or does an agile software development process lead to the group maturing? The previously mentioned confounding factors could also be on an organizational level, which also need to be a part of the data in order to fully understand the dynamics of agile teams and what it is affected by.

ACKNOWLEDGMENTS

We would like to thank all the participants and their companies who helped making the data collection possible for this research. We would also like to thank Christoph Stettina for interesting discussions during the execution and analysis of this study.

ORCID

Lucas Gren  <https://orcid.org/0000-0001-5488-9894>

Alfredo Goldman  <https://orcid.org/0000-0001-5746-4154>

Christian Jacobsson  <https://orcid.org/0000-0003-3941-5935>

REFERENCES

1. Dingsøyr T, Nerur S, Baliyepally V, Moe NB. A decade of agile methodologies: towards explaining agile software development. *The Journal of Systems and Software*. 2012;85:1213-1221. <https://doi.org/10.1016/j.jss.2015.05.008>
2. Hoda R, Salleh N, Grundy J, Tee HM. Systematic literature reviews in agile software development: a tertiary study. *Information and Software Technology*. 2017;85:60-70.
3. Beck K. Embracing change with extreme programming. *IEEE Computer*. 1999;32(10):70-77.
4. Cohen D, Lindvall M, Costa P. An introduction to agile methods. *Advances in Computers*. 2004;62:1-66.
5. Lenberg P, Feldt R, Wallgren LG. Behavioral software engineering: a definition and systematic literature review. *Journal of Systems and Software*. 2015;107:15-37. <https://doi.org/10.1016/j.jss.2015.04.084>
6. Wufka M, Ralph P. Explaining agility with a process theory of change. In: Lassenius C, Prikladnicki R, eds. *Proceedings of the Agile Conference* IEEE. ; 2015: 60-64

7. Tapanainen T, Hallanoro M, Päiväranta J, Salmela H. Towards an agile IT organisation: a review of prior literature. In: Remenyi D., ed. *Proceedings of the 2nd European Conference on Information Management and Evaluation* IEEE. ; 2008: 425–432.
8. Hummel M, Rosenkranz C, Holten R. The role of communication in agile systems development. *Business & Information Systems Engineering*. 2013;5(5):343–355.
9. Melo CdO, Santana C, Kon F. Developers motivation in agile teams. In: Cortellessa V, Muccini H, Demirors O., eds. *38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)* IEEE. ; 2012: 376–383.
10. Inayat I, Salim SS, Marczak S, Daneva M, Shamshirband S. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*. 2015;51:915–929.
11. Hoda R, Noble J, Marshall S. Negotiating contracts for agile projects: a practical perspective. In: Abrahamsson P, Marchesi M, Maurer F., eds. *Proceedings of the 10th International Conference on Agile Processes in Software Engineering and Extreme Programming*. Lecture Notes in Business Information Processing. Springer. ; 2009: 186–191.
12. McDonald S, Edwards H. Who should test whom? *Communications of the ACM*. 2007;50(1):66–71.
13. Seger T, Hazzan O, Bar-Nahor R. Agile orientation and psychological needs, self-efficacy, and perceived support: a two job-level comparison. In: Melnik G, Kruchten P, Poppendieck M., eds. *Agile Conference (AGILE)* IEEE. ; 2008: 3–14.
14. Balijepally V, Mahapatra R, Nerur S. Assessing personality profiles of software developers in agile development teams. *Communications of the Association for Information Systems*. 2006;18(1):4.
15. Feldt R, Angelis L, Torkar R, Samuelsson M. Links between the personalities, views and attitudes of software engineers. *Information and Software Technology*. 2010;52(6):611–624.
16. Cruz S, Silva DF, Capretz L. Forty years of research on personality in software engineering: a mapping study. *Computers in Human Behavior*. 2015;46:94–113.
17. Hannay JE, Arisholm E, Engvik H, Sjöberg DI. Effects of personality on pair programming. *IEEE Transactions on Software Engineering*. 2010;36(1):61–80.
18. Licorish SA, MacDonell SG. Communication and personality profiles of global software developers. *Information and Software Technology*. 2015;64:113–131.
19. Moe N, Aurum A, Dybå T. A teamwork model for understanding an agile team: a case study of a Scrum project. *Information and Software Technology*. 2010;52(5):480–491.
20. Moe N, Aurum A, Dybå T. Challenges of shared decision-making: a multiple case study of agile software development. *Information and Software Technology*. 2012;54(8):853–865.
21. Hoda R, Noble J, Marshall S. Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*. 2013;39(3):422–444.
22. Wheelan S. *Group Processes: A Developmental Perspective*. 2nd ed. Boston: Allyn and Bacon; 2005.
23. Ahmad MO, Markkula J, Oivo M. Kanban in software development: a systematic literature review. In: Demirors O, Turetken O., eds. *39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)* IEEE. ; 2013: 9–16.
24. Rising L, Janoff NS. The Scrum software development process for small teams. *IEEE Software*. 2000;17(4):26–32.
25. Licorish SA, Holvitie J, Hyrynsalmi S, et al. Adoption and suitability of software development methods and practices. In: Potanin A, Murphy G, Reeves S, Dietrich J., eds. *23rd Asia-Pacific Software Engineering Conference (APSEC)* IEEE. ; 2016: 369–372.
26. Serrador P, Pinto JK. Does Agile work?—A quantitative analysis of agile project success. *International Journal of Project Management*. 2015;33(5):1040–1051.
27. Robson S. *Agile SAP: Introducing Flexibility, Transparency and Speed to SAP Implementations*. Cambridgeshire: IT Governance Publishing; 2013.
28. Cooke J. *Agile: An Executive Guide. Real Results from IT Budgets*. Cambridgeshire: IT Governance Publishing; 2011.
29. Schwaber K. *Agile Project Management with Scrum*. Redmond, Wash: Microsoft Press; 2004.
30. Weinberg G. *The Psychology of Computer Programming*. New York: Dorset House Pub. Silver Anniversary Ed. ed; 1998.
31. Kim G, Behr K. In: Spafford G, ed. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*. Portland, OR, USA: IT Revolution Press; 2013.
32. Cobb C. *Making Sense of Agile Project Management: Balancing Control and Agility*. Hoboken: John Wiley & Sons, Inc.; 2011.
33. Chin G. *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*. New York: AMACOM; 2004.
34. Cockburn A. *Agile Software Development: The Cooperative Game*. 2nd ed. Upper Saddle River, NJ: Addison-Wesley; 2007.
35. Oosterhout VM. *Business Agility and Information Technology in Service Organizations*. Rotterdam: Erasmus Research Institute of Management. 2010.
36. Putta A, Paasivaara M, Lassenius C. Benefits and challenges of adopting the Scaled Agile Framework (SAFe): preliminary results from a multivocal literature review. In: Kuhrmann M, Schneider K, Pfahl D, et al., eds. *Proceedings of the International Conference on Product-Focused Software Process Improvement* Springer International Publishing. ; 2018: 334–351.
37. Poth A, Sasabe S, Mas A, Mesquida AL. Lean and agile software process improvement in traditional and agile environments. *Journal of Software: Evolution and Process*. 2019;31(1):e1986.
38. So C. *Making Software Teams Effective: How Agile Practices Lead to Project Success Through Teamwork Mechanisms*. Frankfurt am Main: Peter Lang; 2010.
39. Jalote P, Palit A, Kurien P, Peethamber V. Timeboxing: a process model for iterative software development. *Journal of Systems and Software*. 2004;70(1-2):117–127.
40. So C, Scholl W. Perceptive agile measurement: new instruments for quantitative studies in the pursuit of the social-psychological effect of agile practices. In: Abrahamsson P, Marchesi M, Maurer F., eds. *Proceedings of the 10th International Conference on Agile Processes in Software Engineering and Extreme Programming* Lecture Notes in Business Information Processing. Springer; 2009: 83–93.
41. Stray V, Sjøberg DI, Dybå T. The daily stand-up meeting: a grounded theory study. *Journal of Systems and Software*. 2016;114:101–124.
42. Derby E, Larsen D. *Agile Retrospectives: Making Good Teams Great*. Raleigh, NC: Pragmatic Bookshelf; 2006.
43. Karlsson L, Thelin T, Regnell B, Berander P, Wohlin C. Pair-wise comparisons versus planning game partitioning—experiments on requirements prioritisation techniques. *Empirical Software Engineering*. 2007;12(1):3–33.
44. Mahnič V, Hovelja T. On using planning poker for estimating user stories. *Journal of Systems and Software*. 2012;85(9):2086–2095.

45. Krebs W. Turning the knobs: a coaching pattern for XP through agile metrics. In: Wells D, Williams L., eds. *Proceedings of the Second XP Universe and First Agile Universe Conference*. 2418 of Lecture Notes in Computer Science. Springer. ; 2002: 60–69
46. Keyton J. *Communicating in Groups: Building Relationships for Group Effectiveness*. New York: McGraw-Hill; 2002.
47. Wheelan S. Group size, group development, and group productivity. *Small Group Research*. 2009;40(2):247–262.
48. Bion W. *Experiences in Groups: And Other Papers*. London: Routledge; 1992.
49. Tuckman BW. Developmental sequence in small groups. *Psychological Bulletin*. 1965;63(6):384–399.
50. Wheelan S, Hochberger J. Validation studies of the group development questionnaire. *Small Group Research*. 1996;27(1):143–170.
51. Gren L, Torkar R, Feldt R. Group development and group maturity when building agile teams: a qualitative and quantitative investigation at eight large companies. *The Journal of Systems and Software*. 2017;124:104–119. <https://doi.org/10.1016/j.jss.2016.11.024>
52. Sidky A, Arthur J, Bohner S. A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in Systems and Software Engineering*. 2007;3(3):203–216.
53. Wheelan S, Murphy D, Tsumura E, Kline SF. Member perceptions of internal group dynamics and productivity. *Small Group Research*. 1998;29(3):371–393.
54. Wheelan S, Tilin F. The relationship between faculty group development and school productivity. *Small Group Research*. 1999;30(1):59–81.
55. Wheelan S, Kesseling J. Link between faculty group: development and elementary student performance on standardized tests. *The Journal of Educational Research*. 2005;98(6):323–330.
56. Wheelan S, Burchill CN, Tilin F. The link between teamwork and patients' outcomes in intensive care units. *American Journal of Critical Care*. 2003;12(6):527–534.
57. Tuckman B, Jensen M. Stages of small-group development revisited. *Group & Organization Management*. 1977;2(4):419–427.
58. Wheelan S, Davidson B, Tilin F. Group development across time: reality or illusion? *Small Group Research*. 2003;34(2):223–245.
59. Kozlowski SW, Ilgen DR. Enhancing the effectiveness of work groups and teams. *Psychological Science in the Public Interest*. 2006;7(3):77–124.
60. Jacobsson C, Rydbo N, Börresen JE. The links between group development and health in manufacturing industry. *Small Group Research*. 2014;45(4):400–415.
61. Gren L. The links between agile practices, interpersonal conflict, and perceived productivity. In: Mendes E., ed. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software EngineeringEASE'17*. ACM. ; 2017: 292–297
62. Streiner DL. Starting at the beginning: an introduction to coefficient alpha and internal consistency. *Journal of Personality Assessment*. 2003;80(1):99–103.
63. Cohen J. Quantitative methods in psychology—a power primer. *Psychological Bulletin*. 1992;112(1):155–159.
64. 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. *Personnel Psychology*. 2006;59(3):501–528.
65. Melnik G, Maurer F. Direct verbal communication as a catalyst of agile knowledge sharing. In: Little T., ed. *Proceedings of the Agile Development ConferenceADC'04*. IEEE. ; 2004: 21–31
66. Lehtinen TO, Itkonen J, Lassenius C. Recurring opinions or productive improvements—what agile teams actually discuss in retrospectives. *Empirical Software Engineering*. 2017;22(5):2409–2452.
67. Wheelan S. *Creating Effective Teams: A Guide for Members and Leaders*. 4th ed. Thousand Oaks: SAGE; 2013.
68. Schein E. *Process Consultation Vol. 1: Its Role in Organisation Development*. Reading, Mass. 2nd ed. Addison-Wesley; 1988.
69. Senior B, Swailes S. The dimensions of management team performance: a repertory grid study. *International Journal of Productivity and Performance Management*. 2004;53(4):317–333.
70. Lowman R. *The California School of Organizational Studies' Handbook of Organizational Consulting Psychology: A Comprehensive Guide to Theory, Skills, and Techniques*. San Francisco, Calif: Jossey-Bass; 2002.
71. Acker J. Inequality regimes: gender, class, and race in organizations. *Gender & Society*. 2006;20(4):441–464.
72. Teh A, Baniassad E, Van Rooy D, Boughton C. Social psychology and software teams: establishing task-effective group norms. *IEEE Software*. 2012;29(4):53–58.
73. Roberge M, Dick VR. Recognizing the benefits of diversity: when and how does diversity increase group performance? *Human Resource Management Review*. 2010;20(4):295–308.
74. Thomas K. Conflict and conflict management: reflections and update. *Journal of Organizational Behavior*. 2006;13(3):265–274.
75. Wahba MA, Bridwell LG. Maslow reconsidered: A review of research on the need hierarchy theory. *Organizational Behavior and Human Performance*. 1976;15(2):212–240.
76. Klein C, Díaz Granados D, Salas E, et al. Does team building work? *Small Group Research*. 2009;40(2):181–222.
77. Sundström E, De Meuse K, Futrell D. Work teams: Applications and effectiveness. *American Psychologist*. 1990;45(2):120–133.
78. Guzzo R, Jette R, Katzell R. The effects of psychologically based intervention programs on worker productivity: a meta-analysis. *Personnel Psychology*. 1985;38(2):275–291.
79. Gren L, Torkar R, Feldt R. The prospects of a quantitative measurement of agility: a validation study on an agile maturity model. *The Journal of Systems and Software*. 2015;107:38–49. <https://doi.org/10.1016/j.jss.2015.05.008>

How to cite this article: Gren L, Goldman A, Jacobsson C. Agile ways of working: A team maturity perspective. *J Softw Evol Proc*. 2020;32:e2244. <https://doi.org/10.1002/smr.2244>