



Problem reports and team maturity in agile automotive software development

Lucas Gren

Volvo Cars and Chalmers | University of Gothenburg
Gothenburg, Sweden
Blekinge Institute of Technology
Karlskrona, Sweden
lucas.gren@lucasgren.com

Martin Shepperd

Chalmers | University of Gothenburg
Gothenburg, Sweden
Brunel University
London, UK
martin.shepperd@brunel.ac.uk

ABSTRACT

Background: Volvo Cars is pioneering an agile transformation on a large scale in the automotive industry. Social psychological aspects of automotive software development are an under-researched area in general. Few studies on team maturity or group dynamics can be found specifically in the automotive software engineering domain.

Objective: This study is intended as an initial step to fill that gap by investigating the connection between issues and problem reports and team maturity.

Method: We conducted a quantitative study with 84 participants from 14 teams and qualitatively validated the result with the Release Train Engineer having an overview of all the participating teams.

Results: We find that the more mature a team is, the faster they seem to resolve issues as provided through external feedback, at least in the two initial team maturity stages.

Conclusion: This study suggests that working on team dynamics might increase productivity in modern automotive software development departments, but this needs further investigation.

CCS CONCEPTS

• **Software and its engineering** → **Agile software development; Programming teams.**

KEYWORDS

automotive software development, teams, team maturity, problem reports

ACM Reference Format:

Lucas Gren and Martin Shepperd. 2022. Problem reports and team maturity in agile automotive software development. In *15th International Conference on Cooperative and Human Aspects of Software Engineering (CHASE'22)*, May 21–29, 2022, Pittsburgh, PA, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3528579.3529173>



This work is licensed under a Creative Commons Attribution International 4.0 License. *CHASE'22*, May 21–29, 2022, Pittsburgh, PA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9342-3/22/05.
<https://doi.org/10.1145/3528579.3529173>

1 INTRODUCTION

An agile approach to projects has become extremely popular both within [2], and outside software development [17]. In contrast to traditional plan-driven development (a waterfall process), this paradigm has been given the name *agile*, which literally means to be able to move quickly, easily and be flexible [23]. Feasible solutions to customer needs should be developed in close customer collaboration, rather than being subject to the uncertainties of large up-front design and planning [19]. With the explosion of software in the car, this complexity has forced automotive companies to, not just focus on customer value, but to also to rely on teams, and teams of teams, to develop modern cars with around 150 Million lines of code. This implies that the social aspect of group dynamics need to be understood in order to scale the development projects. The car projects also differ in that the agile approaches are used across both software, hybrid, and pure hardware teams. This study, though, focuses on the software teams in this quite different context since the ≈ 800 teams essentially build one single product together.

Research on automotive software engineering has mainly focused on testing and building software models [9]. We have not been able to find many studies on organizational psychology in general except studies on “human factors” associated with individuals (e.g., Maro et al. [14]). We have not been able to find any studies on social psychology in general, or team maturity in particular, in automotive software development.

Some studies have been conducted on self-organization in general agile software development (i.e., not automotive) teams in a broader software engineering context, also referred to as team autonomy. Moe et al. [16] concluded that one of the barriers to agile team autonomy was a lack of system support for teams and reduced external autonomy. More recently, Hodgson and Briand [11] investigated how teams adopt agile practices over time. They conclude that some teams do a “big bang” while other implement agile practices one-by-one. They saw a trend that team members without much previous agile knowledge prefer the latter while experienced agile practitioners prefer the former. Volvo Cars has gone through an agile transformation in the last five years and there is a strong push towards providing teams with the mandate to lead themselves to a larger extent, i.e., a “big bang” on a very large scale impacting close to 800 teams.

Hoda et al. [10] showed that senior management have a key role in enabling team agility. That there is a challenge in

fitting the agile approach into traditional management in a software engineering context was also shown by Hodgson and Briand [11]. In more general leadership research, there are traits that a good leader must have, but on top of that, good leaders in all fields adapt their leadership style dynamically to the situation [12]. Leadership could also be seen as a function of group action [20] and not a role, which makes it possible to share or change over time. This means that there is no best leadership style since many different styles are needed depending on the context, the group, and the people involved. In the agile space, e.g., servant leadership is advocated as the foundation for leading, but the definition of what that is remains vague [18]. Leadership is a difficult construct to research since it happens in complex systems and is influenced by a vast number of factors.

Gren et al. [8] found initial indications confirming that the definition of agile teams overlaps with what is meant by a mature group in social psychology [22]. Agile teams are dependent on also being mature in terms of collaborative behaviour plus having navigated through the initial stages of group development. Mature teams are also known to collaborate better with other teams [21], which is of interest when scaling the agile ways of working at Volvo Cars. We define team maturity in this study as the degree to which a team has navigated through the group development stages according to Wheelan [21]. In Wheelan's Integrated Model of Group Development (or IMGD), small groups (or teams) start in the Dependency and Inclusion stage where team members are tentative and polite in their communication and need to build an initial level of psychological safety. Team members do not have a common mental model of the group goal nor do they know the real competencies of the other team members. When higher levels of psychological safety start to emerge, the team transitions into the second phase, Counter-Dependency and Fight. This is a conflict stage but is absolutely necessary to determine who can contribute to what and create a shared mental model of the team's purpose. After the second stage, the team can organize work better and set productive group norms. With higher levels of trust and a continuous improvements of work organization, the team can reach stage four, which is the Work and Productivity stage [21].

Leadership, self-organization, and agility can thus be mapped onto the team development stages [8]. As the development cycle moves along from a psychological perspective, all members of the team (including the managers or leaders) must change their behavior [21]. Team maturity, as measured by the Group Development Questionnaire (GDQ), has been shown to be correlated to measurements of productivity also in software engineering [1]. In this study, we used a recently developed short version of the GDQ comprising 13 items instead of 60 [5], which makes it practically more useful considering the time availability of teams.

If the results on the connection between team maturity and team productivity from social psychology [21] apply to automotive software development, aspects of productivity should also be connected to measurements of team maturity

in that context. One quite objective proxy for productivity for these Volvo Cars teams was their issue management, i.e., how long the teams take to fix issues reported back from their deliveries. This metric is external to the team, which makes it in many ways more appropriate than e.g., flow of backlog items or self-assessed productivity.

2 METHOD

We distributed the short version of the GDQ, (the GDQS [5]) to 14 teams engaged in software-only development at Volvo Cars. The teams also logged the number of issues they have with the delivered software in the same month. Issues could be for example bugs or other types of change requests. The number of issues can accumulate since many of them remain unaddressed so it may not a good measurement of the current team performance. We, therefore, instead looked at the average age of issues in the same month as the teams filled out the GDQS. The average is sensitive to outliers but Volvo Cars chose that metric as a KPI since all issues need to be resolved. 82 team members answered the GDQS from all the 14 teams. Since this is an average of responses per team and the teams are usually from 3 to 12 individuals, we estimate this response rate to be around 80%.

We describe the data and its patterns or lack thereof and avoid using p values for null hypothesis significance testing in accordance with McShane et al. [15]. We instead use an approach based on Bayesian Data Analysis (BDA), i.e., we will plot the re-sampled (using Markov-Chain Monte Carlo) distribution of all our parameters and be explicit about our assumptions (see supplementary material). For a description and tutorial applied to software engineering see Furia et al. [3].

Note all the shared data in this paper, plots and supplementary material were transformed using arbitrary linear transformations. This was done so as to not reveal the absolute values of issues and team maturity, since that was deemed as sensitive information by the company. The average ages of issues were downloaded from the Volvo Cars instance of Jira and the team maturity data was collected through the company's own survey tool.

In the second step, we interviewed the Release Train Engineer (RTE), which is a role that has an overview of this whole set of teams. The interviewee was asked to describe the situation around the data collection and also critically assess the connection between team maturity and the teams' capacity to resolve reported issues.

Based on previous studies of team maturity and aspects of productivity (see Section 1), we hypothesize that the first two stages (Stage 1 and 2) should be positively connected to the average age of issues, i.e., the more collaborative aspects to solve in these stages, the longer it would take to solve reported issues. We would also hypothesize that the latter stages (Stage 3 and 4) would have the opposite relationship to reported issues, since higher values on these two scales imply more maturity (e.g., higher levels of trust etc.).

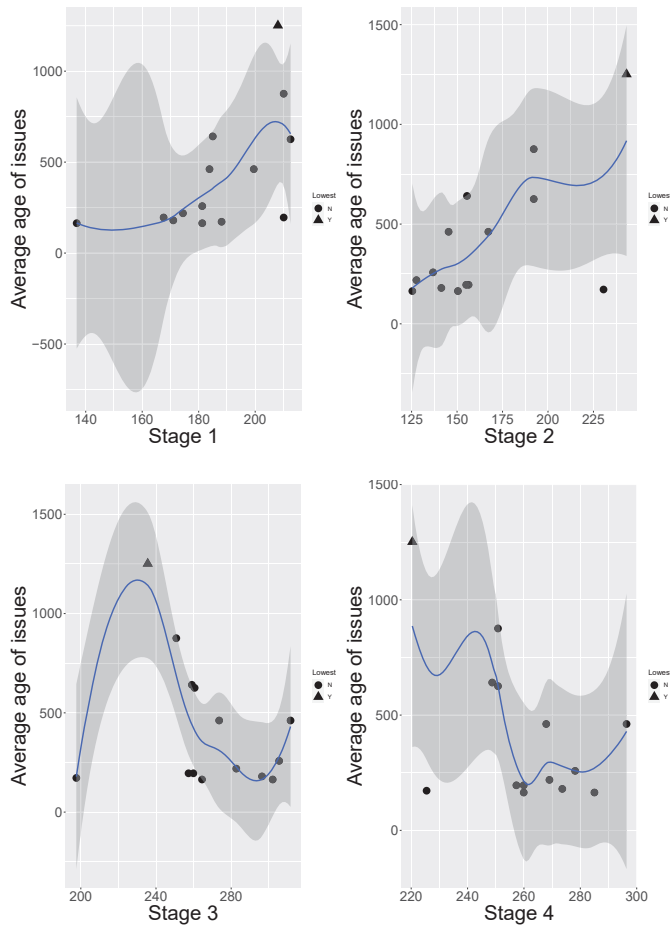


Figure 1: Scatter plots with each with one of the developmental stages of team maturity as independent variable (the x axis) and the average age of issues as dependent variable (the y axis).

3 RESULTS

3.1 Quantitative results

The result shown in Fig. 1 indicates that the direction of the lines are in the hypothesized direction. However, Fig. 2 shows that only the distributions of stages 1 and 2 are distinct from zero with close to a 95% credible interval.

As mentioned, we do not want to use classical NHST inference, which means that we see this quantitative result as indicative, but further studies are needed. We would like to, instead, highlight the importance of the qualitative data below.

In summary, when the teams are newly formed (or less mature) or in the conflict stage of creating shared mental models of the team goals and figuring out who can really contribute with what in order to fulfill these goals, the average age of issues is higher for those teams.

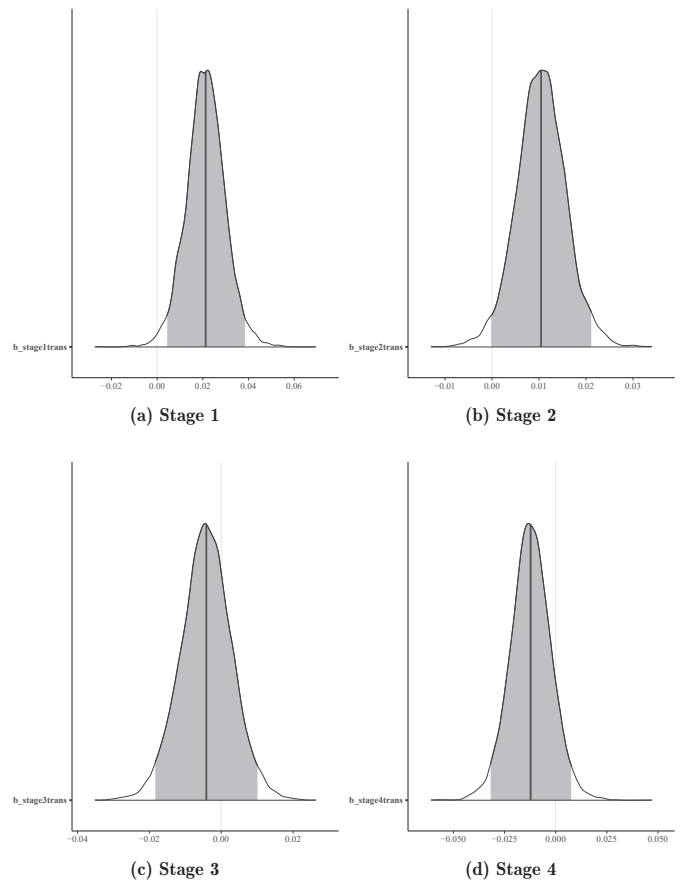


Figure 2: Re-sample distributions of four separate models, each with one of the developmental stages of team maturity as independent variable and the average age of issues as dependent variable.

3.2 Qualitative validation

Below is a summary of what the RTE commented upon in the interview.

During the time of the data collection, the teams were focusing on removing issues that were never to be dealt with. The team developed their software in-house and wanted to remove issues they had inherited from suppliers. The teams were working on two different platforms, one old and one new. Different teams were working on the different platforms. The teams working on the old platforms were described as having lower motivation for their work compared to the teams working on the new platform. The teams that worked on the old platform would have preferred to work on the new platform instead. The teams working on the new platform used to even be located at a different municipality.

The performance and collaboration of the teams were described as very different. Regarding their reported issues, some teams could solve those internally while other teams

were highly dependent on suppliers. For some teams, solving issues was a complex task with many steps.

The teams working on the old platform had closed a high number of issues, other teams received reports of defects that were incorrectly perceived as being errors.

The teams working on the old platform were assessed as having lower team maturity compared to the ones working on the new platform. They closed a lot of issues that they realized would never be dealt with. Fixing issues from problem reports was a part of all the teams' daily work.

As a form of validation, the team with the highest values on the y axis in Fig. 1, was the team the interviewee identified as the least mature team (denoted as a triangle).

4 DISCUSSION

The results of this study suggest that the automotive software development having transformed into agile ways of working, might be dependent on team maturity in order to deliver value in the shape of fixing issues in their delivery. Since the team as a unit of analysis lies at the heart of an agile organization, research on automotive agile software development should also use theories and models from social psychology in order to understand the mechanisms at play.

On the necessary journey towards being more responsive to change, Volvo Cars has been on an agile transformation for the last five years. Some parts of the company has gotten farther than others at the time of this research, but the vehicle software and electronics department were the forerunner in the transformation.

Now that the automotive industry has software as a key to gaining a competitive advantage, research on software engineering adapted to the automotive industry is needed and conducted to understand this somewhat specific context. The psychological aspects of automotive software engineering has, though, not been much in focus and this study is a tiny step in beginning to fill that gap. Research so far has focused on testing and building software models [9] and the human factors investigated have been associated with individual psychology (e.g., Maro et al. [14]). Since the aspects of agile team autonomy now also surfaces in the automotive software development, we need to investigate this context further also with regards to self-organization. Moe et al. [16] concluded, among other things, that one of the barriers of agile team autonomy was a lack of system for team support. This is an aspect identified in the agile transformation at Volvo Cars as well and the participating teams (and many others) were trained for at least three hours in Wheelan's Integrated Model of Group Development. They were also given access to a toolbox connected to the different stages after having filled out the GDQS, which hopefully helped them maturing further as a team, but this remains to be investigated.

4.1 Practical implications

We have assumed that the team maturity would be connected to aspects of productivity in the automotive software development at Volvo Cars, but this current study is the first one

showing data on this actually being the case. If the results of this study would hold for automotive software engineering in general, it has a multitude of implications for developing such software. Many of the challenges go hand-in-hand with the agile transformation, but also provides way of getting teams to mature faster. One of these implications is to focus even more on training leaders and managers (in addition to training teams, of course) in adapting their leadership style dynamically to the team maturity level [12]. A team will not self-organize on day one because the team members are unlikely to have met before. Also there are the leadership challenges of designing teams well and assuring the emergence of good work principles and norms [6], instead of micromanaging technical details. Agile leadership has been described as "servant leadership," which remains vague [18]. But a key aspect of thinking about servant leadership is that it challenges many managers' mental model of leadership being based on hierarchical power, however, agile leadership need to be a servant in a way that serves the teams with what they actually need right now. That could also be to confront behavior that does not build psychological safety, as an example [7].

This current study is in line with the result by Al-Sabbagh and Gren [1] that team maturity as measured by the GDQ is connected to productivity aspect of software development, but extends it into the automotive software development domain. A very practical implication for the automotive industry is to measure team maturity in a short validated survey, such as the GDQS [5], train both managers, agile leaders, and team members in team development, by for example using the Integrated Model of Group Development [22] in order to increase the speed of fixing issues in the teams' deliveries. We infer this causality by logic and the qualitative data, since it does not make sense to hypothesize that fixing issues fast would lead to the team maturing, however, the mediating role of team maturity between team building interventions and the speed of fixing issues, of course needs much more research.

4.2 Threats to validity

We use relevant parts of the checklist created by Gren [4] for behavioral software engineering research.

Reliability. Both the internal consistency and stability have been checked multiple times in the development of the GDQ [22] and GDQS [5]. The lack of a formal definition of what becomes an issue in the problem reports at Volvo Cars is a threat to the reliability of us actually measuring the average age of issues the teams have. Further studies should explicitly ask for such a definition, or help define it, if it does not exist. We do not know how extreme values might have affected this average that Volvo Cars chose to use as a KPI. Perhaps the median age of issues might have been more reliable.

Construct validity. The constructs under investigation were the four team development scales and the average age of issues. The GDQS is a well-validated scale for measuring

team maturity and we only selected teams in a part of Volvo Cars that we know reported and maintained their problem reports continuously. The validity of the used constructs are, therefore, assessed as high.

Internal validity. We tried to argue for a theoretical reason for causality between team maturity and average age of issues, but we did not measure any other confounding factors. Potential confounds could be type of work, team composition, conformance to processes, individual competencies, etc., and in a social complex adaptive system there could be many more.

External validity. The small sample of 14 teams and a collection of data from only one part of Volvo Cars are other large threats to this study. We do not know if these teams and the way they work is representative to any teams outside this context. A random sample was not possible in this case since it was hard to find a set of teams that were in a position to continuously report on, and receive feedback from their deliveries. This is an ongoing transformation at Volvo Cars, i.e., to become more data-driven. However, it does seem plausible that they could be applicable to other similar safety-critical software development environments.

5 CONCLUSION AND FUTURE WORK

We conclude that the average ages of issues in automotive software development teams were associated with teams having higher values on the first two stages of team development. This could mean that mature teams fix issues faster. However, this is just an initial study with a small sample and future studies should sample randomly and also use other productivity measurements. Another part of the automotive software development that could be studied in the future is extending the work of Lenberg et al. [13] and further look at how teams deal with safety-critical systems.

SUPPLEMENTARY MATERIAL

The Supplementary Material is available on Zenodo¹. It includes an anonymized raw data CSV file and all the R scripts used.

ACKNOWLEDGMENTS

Both authors would like to thank the Swedish Research Council funding this work under the auspices of the 2022 Tage Erlander research professorship. Both authors thank Volvo for the provision of the data that made this work possible. This project is an associated project under the Software Center, Gothenburg, Sweden.

REFERENCES

- [1] K. Al-Sabbagh and L. Gren. 2018. The connections between group maturity, software development velocity, and planning effectiveness. *Journal of Software: Evolution and Process* 30, 1 (2018), e1896.
- [2] T. Dingsøy, S. Nerur, V. Balijepally, and N. Moe. 2012. A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems and Software* 85 (2012), 1213–1221.
- [3] C. Furia, R. Feldt, and R. Torkar. 2019. Bayesian data analysis in empirical software engineering research. *IEEE Transactions on Software Engineering* 47, 9 (2019), 1786–1810.
- [4] L. Gren. 2018. Standards of validity and the validity of standards in behavioral software engineering research: The perspective of psychological test theory. In *International Symposium on Empirical Software Engineering and Measurement (ESEM)*. 1–4.
- [5] L. Gren, C. Jacobsson, N. Rydbo, and P. Lenberg. 2020. The Group Development Questionnaire Short (GDQS) Scales: Tiny-Yet-Effective Measures of Team/Small Group Development. (Feb 2020). DOI: <http://dx.doi.org/10.31234/osf.io/u3p8c>
- [6] L. Gren and M. Lindman. 2020. What an Agile Leader Does: The Group Dynamics Perspective. In *International Conference on Agile Software Development (XP)*. Springer, 178–194.
- [7] L. Gren and P. Ralph. 2022. What Makes Effective Leadership in Agile Software Development Teams?. In *The 44th International Conference on Software Engineering*. ACM, New York.
- [8] L. Gren, R. Torkar, and R. Feldt. 2017. Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies. *The Journal of Systems and Software* 124 (2017), 104–119.
- [9] A. Haghghatkhah, A. Banijamali, O-P. Pakanen, M. Oivo, and P. Kuvaja. 2017. Automotive software engineering: A systematic mapping study. *Journal of Systems & Software* 128 (2017), 25–55.
- [10] R. Hoda, J. Noble, and S. Marshall. 2011. Supporting self-organizing agile teams. In *International Conference on Agile Software Development*. Springer, 73–87.
- [11] D. Hodgson and L. Briand. 2013. Controlling the uncontrollable: Agile teams and illusions of autonomy in creative work. *Work, employment and society* 27, 2 (2013), 308–325.
- [12] S. Kozlowski, D. Watola, J. Jensen, B. Kim, and I. Botero. 2009. Developing adaptive teams: A theory of dynamic team leadership. In *Team effectiveness in complex organizations: Cross-disciplinary perspectives and approaches*, E. Salas, G. Goodwin, and S. Burke (Eds.). Routledge, New York, 113–155.
- [13] P. Lenberg, R. Feldt, L. Wallgren Tengberg, and L. Gren. 2020. Behavioral Aspects of Safety-Critical Software Development. In *Co-operative and Human Aspects of Software Engineering (CHASE)*. IEEE.
- [14] S. Maro, M. Staron, and J. Steghöfer. 2017. Challenges of establishing traceability in the automotive domain. In *International Conference on Software Quality*. Springer, 153–172.
- [15] B. McShane, D. Gal, A. Gelman, C. Robert, and J. Tackett. 2019. Abandon statistical significance. *The American Statistician* 73, sup1 (2019), 235–245.
- [16] N. Moe, T. Dingsøy, and T. Dybå. 2008. Understanding self-organizing teams in agile software development. In *19th Australian Conference on Software Engineering (ASWEC'08)*. 76–85.
- [17] J. Pajares, D. Poza, F. Villafañez, and A. López-Paredes. 2017. Project Management Methodologies in the Fourth Technological Revolution. In *Advances in Management Engineering*, H. Cesáreo (Ed.). Springer, 121–144.
- [18] D. Parris and J. Peachey. 2013. A systematic literature review of servant leadership theory in organizational contexts. *Journal of Business Ethics* 113, 3 (2013), 377–393.
- [19] K. Petersen. 2011. Is lean agile and agile lean? A comparison between two software development paradigms. In *Modern software engineering concepts and practices: Advanced approaches*, A. Dogru and V. Bicer (Eds.). IGI Global, 19–46.
- [20] M. Von Cranach. 1986. Leadership as a function of group action. In *Changing conceptions of leadership*, C. Graumann and S. Moscovici (Eds.). Springer, New York, 115–134.
- [21] S. Wheelan. 2005. *Group processes: A developmental perspective* (2 ed.). Allyn and Bacon, Boston.
- [22] S. Wheelan and J. Hochberger. 1996. Validation studies of the group development questionnaire. *Small Group Research* 27, 1 (1996), 143–170.
- [23] L. Williams. 2012. What agile teams think of agile principles. *CACM* 55, 4 (2012), 71–76.

¹<https://www.doi.org/10.5281/zenodo.6400809>