Investigating the Perception of Success in Software Projects Among Developers from a Brazilian Software Company

Claudio Nascimento[1], **Rafael de Mello**[2]

[1]Centro Federal Tecnológico Celso Suckow da Fonseca, Rio de Janeiro, Brazil

[2]Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

claudio.nascimento@aluno.cefet-rj.br, rafaelmello@ic.ufrj.br

Abstract. *Understanding and properly applying agile practices in software development projects is a common challenge for developers. Consequently, the expected benefits of these practices may not be perceived, hampering developers from reaching critical success factors of software projects. This work aims to investigate the perception of success in software projects in the context of a Brazilian software company whose development process is based on Scrum. Besides, the development teams from this company follow several technical and non-technical agile practices. For this purpose, we conducted an opinion survey with 17 developers distributed among four development teams of this company. The survey findings revealed prevailing gaps in the teams' perception of critical success factors in their projects despite following a common subset of agile practices addressing them. Among others, the gaps include a lack of perceiving realistic schedules, clear requirements, and good quality management. Aiming at improving this scenario in the company, we plan to conduct a collaborative action research with one of the development teams investigated.*

1. Introduction

The agile concepts and methods have been widely disseminated and inserted into the software development practice since the Agile Manifesto [Fowler et al. 2001]. Currently, it is difficult to identify teams applying development processes without any agile inspiration. Consequently, it is common to observe development teams and companies following agile methods. In this way, there are several factors to be considered when choosing a particular method, such as the project scope and the capacity for adaptation of the team members [Dybå and Dingsøyr 2008]. Such factors typically lead development teams to conclude that some agile practices deserve more attention than others. Consequently, it is common to observe development teams or even entire companies depicting and employing customized approaches [Leal and Santos 2015, de Mello et al. 2021]. Not rarely, these approaches include the combination of ceremonies and practices derived from different agile methods [Leal and Santos 2015], such as lean development [Poppendieck and Cusumano 2012], Scrum [Schwaber and Beedle 2002], and Kanban [Ahmad et al. 2013].

Development teams should pursue success in their projects, which may be considerably influenced by the development process employed [Chow and Cao 2008]. The concern on tailoring the software development process is often motivated by a collective and somehow unconscious search for success. Reaching success in software projects goes

beyond meeting deadlines and budgets, being directly connected to the organizational climate and the developers' motivation [Dutra et al. 2020]. Therefore, it is important to evaluate the perception of success from a holistic perspective, which includes different critical success factors [Chow and Cao 2008]. In this sense, particular agile practices [de Mello et al. 2014] would contribute for reaching different success factors in software projects [Pimenta and Santos 2016]. For this purpose, it is expected to assure that the agile practices to be followed should be feasible for the project team and properly applied by its members, who should carefully reflect on the benefits and potential drawbacks of these practices in their development contexts [Leal and Santos 2015].

Despite the common concern of developers in tailoring the development process, it is also common to observe a lack of proper following the preconized agile practices. For instance, a development team would argue by following daily meetings despite not following key recommendations such as restricting the discussions about activities and obstacles reported during the meeting time [Schwaber and Beedle 2002]. Besides, several project team members may not be ready to employ different agile practices requiring intense communication and knowledge sharing [Kamei et al. 2017]. Consequently, the effect of these practices on the success of software projects may be minimized or even nullified.

This paper report an investigation on the perception of success in software projects by developers from a Brazilian software company in which developers were fully working remotely due to the COVID-19 pandemics. The company has several development teams specialized in evolving specific modules from an information system for health services. We conducted an opinion survey with 17 developers from this company, distributed among four teams. Both teams reported following several relevant agile practices in common, including daily meetings, product backlog, and whole team. Despite this, the survey results indicate that reaching some critical success factors in software projects tends to be challenging for the developers. In general, we found a trend among developers in not perceiving the feasibility of the projects' schedules and the effectiveness of change management activities. Besides, several developers have negative perceptions about the effectiveness of quality management and risk management activities, among other issues. The survey findings contrast with the expected benefits of the agile practices followed by the teams [Pimenta and Santos 2016].

The replication of the survey protocol introduced in this paper may be employed as a first research step for diagnosing how developers from other companies perceive the success of their projects. Based on the survey findings, alternative research strategies may be applied to overcome the challenging success factors. We opted to plan the conduction of a collaborative action research [Thiollent 1996] with one of these teams aiming to identify, prioritize, and apply interventions for improving the perception of success among its members. The plan of the action research is also introduced in this paper.

Section 2 presents related work on critical success factors in software projects and agile practices. Section 3 describes the settings of the opinion survey conducted with the company's developers. Section 4 presents the survey results. In Section 4.4, we discuss the main threats to validity. Section 5 discusses the main findings of our study. Section 6 introduce the action research to be conducted. Finally, Section 7 concludes the paper and indicates future work.

2. Related Work

Chow and Cao [Chow and Cao 2008] conducted an empirical study on the relationship of critical success factors in software projects following agile practices. Based on the analysis of 408 respondents and 109 projects, the authors identified six critical success factors and corresponding attributes resulting from a proper management process of agile projects. The factors include *delivering strategy*, *agile software engineering techniques*, *team capability*, *project management process*, *team environment*, and *customer involvement*.

Nasir et al. [Nasir and Sahibuddin 2011] analysed the content of 43 research papers aiming at characterizing critical success factors for software projects. The authors found 26 critical success factors, considerably varying in frequency. Most of these factors address non-technical issues such as budget, schedule, and leadership, just to name a few. However, the lack of clear and stable requirements also appeared among the most frequent factors.

One common challenge for development teams addresses the selection of an agile method that best fits their characteristics and needs [Dybå and Dingsøyr 2008]. Regardless of the agile method followed, it is notable that agile principles and practices have been incorporated into the routine of development teams in the last decade [Leal and Santos 2015], which could be considered one great disruption in the field. In this sense, Abrantes and Travassos [Abrantes and Travassos 2013], conducted an extensive systematic literature review for characterizing the principles and practices of agile development. Based on the findings of this review, they carried out an opinion survey with the authors of the relevant papers identified in the review. The survey findings revealed that 16 characteristics of agility and 15 agile practices were perceived as relevant by most researchers. The first column of Table 1 shows an excerpt of these agile practices.

The protocol of the survey conducted by Abrantes and Travassos [Abrantes and Travassos 2013] was then replicated by de Mello et al. [de Mello et al. 2014] over a larger and diverse sample of software professionals. These professionals were identified, stratified, and recruited by following a systematic sampling plan for identifying representative samples [de Mello et al. 2015]. After the analysis of the 272 responses from this replication, the authors found that the perception of software professionals about the relevance of agile practices tends to be predominantly similar to those observed among researchers. Besides, the survey findings allowed expanding the original set of agile practices investigated.

Based on previous research, Pimenta and Santos [Pimenta and Santos 2016] proposed an association between 16 critical success factors in software projects and the relevant agile practices mapped by de Mello et al. [de Mello et al. 2014]. This work may be used by development teams may identify opportunities for introducing/evolving certain agile practices that would enhance the perception of success in their projects. Table 1 show an excerpt of these associations.

It is important to note that following some agile practice in isolation cannot assure reaching some critical success factor satisfactorily. Possible issues on applying certain agile practices may mitigate or even nullify the contributions of these practices to the success of software projects. Besides, other issues related to each agile practice may also

Table 1. Excerpt of the associations between agile critical success factors and agile practices proposed in [Pimenta and Santos 2016]

Agile Practice	Critical Success Factor
product backlog	clear requirements
	clear goals/ objectives
	effective communication/ feedback
	adequate planning
	good risk management
whole team	clear requirements
	realistic schedules
	good leadership
daily meetings	effective communication/ feedback
	updated progress report
	good leadership
small releases	clear requirements
	client-user involvement
	technology familiarity
	committed/ motivated team
project visibility	effective communication/feedback
	updated progress report
	good leadership
continuous integration	good quality management

affect the perception of expected benefits. For instance, a certain development team may build a software system through continuous integration, contributing to good quality management. However, the development team may perceive their configuration management as unsuccessful due to the also continuous introduction of technical debt elements, such as code smells [Zazworka et al. 2014, Iammarino et al. 2019] in the new releases. The incidence of these code smells may be mitigated and combated through the agile practices of pair programming [Hannay et al. 2009] and refactoring [Zazworka et al. 2014].

3. Survey Design

The opinion survey reported in this paper is inserted in the context of a broader investigation aiming at *characterizing the main obstacles perceived by developers for the success of software projects*. In this survey, we want to answer the following research questions:

How are critical success factors of software projects perceived by software developers from teams following agile development approaches?

For this research, we considered the set of relevant agile practices identified by Abrantes and Travassos [Abrantes and Travassos 2013] and de Mello et al.

[de Mello et al. 2014]. The critical success factors considered in this study are those compiled by Pimenta and Santos [Pimenta and Santos 2016]. For answering the research question, we conducted an opinion survey [Linaker et al. 2015] with developers from a Brazilian software company where the first author work. The characteristics of this company and the challenges perceived in the daily work motivated this study with their developers.

3.1. Population and Sample

The survey population is composed of software development professionals working in development teams following agile practices. The sample recruited was established from the developers working for different teams from a Brazilian software company. At the time of the survey execution, the company had around 150 employees, including trainees, effective positions, and temporary ones. The company has two operational headquarters located in the two biggest Brazilian cities (Rio de Janeiro and São Paulo), attending to customers from different parts of the country. Due to the COVID-19 pandemics, all developers were working remotely during the execution of the study.

The company established the adoption of Scrum in 2019 to all development activities. For this purpose, the company hired external consultants to train all developers on the Scrum process and ceremonies. We recruited developers working in four teams from this company to participate. The four teams are responsible for conducting evolutive maintenance projects addressing different modules of the same information system. The system support health companies, such as health insurance operators. The customers frequently requires system evolution addressing new business needs and regulation in the field. These improvements typically result in one-month to three-month projects.

Before performing the recruitment, we mapped the agile practices followed by each team. Besides working in the same software system, we found that all teams surveyed tend to follow similar subsets of relevant agile practices. All teams work with *daily meetings*, *product backlog*, *whole team*, *on-site costumer*, and *pair programming*. Besides, three of the four teams also follow the agile practices of *small releases*, *project visibility*, *continuous integration*, and *simple design*. Therefore, based on the associations presented in Table 1, one could expect that reaching some critical success factors would be less challenging for these teams.

3.2. Instrumentation

The survey questionnaire is composed of 26 items distributed among three blocks. The first block (Table 2) has questions for characterizing the experience of developers in software engineering, based on the schema proposed in [de Mello and Coelho 2021].

The items composing the second block include 16 statements designed for gathering the developers' level of agreement on reaching the critical success factors in their software projects (see Table 3. This level of agreement was gathered through applying the same Likert scale [Joshi et al. 2015] to each item: *totally disagree*, *partially disagree*, *partially agree* and *totally agree*.

The third block of the survey questionnaire is composed of the following two open questions. Through these questions, we aim to gather complementary considerations of the developers about broader points addressing their beliefs about success in software

Table 2. Questionnaire items for characterizing the professional experience of the survey participants.

Questionnaire Item	Answer Type
Higher Academic Degree	Nominal
Years of experience in software development	Number
Years of experience in software development in the current company	Number
Years of experience in software development in the current team	Number
Number of software development projects participated	Number
Number of software development projects participated in the current company	Number
Number of software development projects participated in the current team	Number

projects and the practices that contribute to the success of the projects in their teams. Based on these answers, it will be possible to perform a deeper analysis of the results obtained in the second block.

- *What do you understand by a successful software project?*
- *Which practices followed by the team collaborate to the agility of the development process?*

4. Execution and Results

We conducted the opinion survey in April 2021. Except for the first author, the other 32 developers composing the four selected teams were able to participate in the survey. The developers had five days to answer the survey questionnaire, individually sent by e-mail. After the recruitment, the survey questionnaire were answered by 17 developers: six from the *Billing* module, four from the *Contracts* module, four from the *Operations* module, and three from the *Accounts* module. Most of the 17 survey respondents (88%) declared having at least a Bachelor's degree. Table 4 shows the average working experience (mean/median) of these developers in different contexts.

On average, one may see that the developers have high levels of experience in software development, including at their current working company. These results indicate that the survey participants tend to be experienced and sufficiently immersed in the organizational culture of their working company. Besides, it indicates that most of the developers had participated in the activities for implementing Scrum in the company. Indeed, we identified that the company hired only four of the 17 developers more recently. The mean and median values found for the experience in the team suggest some concern and preference for preserving the team composition. The general characteristics of the developers' experience suggest their opinions will be relevant and well-grounded considering the survey context.

Table 3. Statements composing the second block of the survey questionnaire and the corresponding critical success factor.

Statement	Critical Factor
The project goal remains clear and concise throughout development	clear goals/objectives
The deadlines given for development are realistic	realistic schedule
The project requirements are properly defined from the beginning	clear requirements
The involvement of users and customers in the project is effective	client/user involvement
The stakeholders' communication generates good feedback	effective communication/feedback
Project members are able to apply the tools and the process followed by the company	familiarity with technology/dev. method
The delivering plans are properly defined	adequate planning
It is employed effective ways to share the level of progress of the activities	updated progress report
General details about the development activities are clear and available	effective control/monitoring
Team leadership is well performed	good leadership
The identification of risks in the project and how to deal with them is carried out effectively	risk management
Available work support resources are suited to the complexity of the project	project complexity
The changes performed over development items are properly managed	effective change and config. management
The team is committed to all activities	committed/ motivated team
There is good quality management, following practices that demonstrate and verify the quality of the artifacts created	quality management
Expectations addressing third-party activities are met	good third-party performance

Table 4. Professional experience of the survey participants.

Metric	SW Industry Exp.	Company Exp.	Team Exp.
Years	14.25/13.50	6.90/6.00	4.44/2.00
#Projects	22.00/11.00	7.00/4.00	6.20/4.00

4.1. Perception of Critical Success Factors in Software Projects

From the 272 answers given by the 17 developers to the 16 statements positively addressing the perception of critical success factors in software projects, we found 32,72% answers indicating some level of disagreement with these statements. This overall number *per se* may be interpreted as worrisome once developers are giving opinions about the projects and teams they work on. Besides, we identified that developers perceive the reaching of some critical success factors considerably less than the other factors. Table 5 presents the eight critical success factors worst evaluated by the survey participants. One

may see that more than half of the developers somehow disagree that the schedules are realistic and the change management activities are effective.

Table 5. The eight critical success factors worse evaluated by the survey developers (CD-Completely Disagre, PD- Partially Disagree, PA- Partially Agree, CA- Completely Agree).

Factor	CD	PD	PA	CA
realistic schedule	-	9	6	2
effective change/configuration management	1	8	6	2
clear requirements	2	6	9	-
good risk management	3	5	6	3
good quality management	1	7	8	1
adequate planning	1	6	7	3
updated progress report	2	5	9	1
effective control/monitoring	2	5	9	1

Regarding the eight critical success factors better evaluated (Table 6), it is important to point out that *Effective Communication and Feedback*, and *Commited and Motivated Team* were the only factors in which the frequency of total agreement is higher than the incidence of partial agreement. Besides, there is a trend among the developers agreeing (82% or higher) about the effectiveness of *user involvement*, *communication with stakeholders*, *leadership*, and *third-party work*.

Table 6. The eight critical success factors better evaluated by the survey developers (CD-Completely Disagre, PD- Partially Disagree, PA- Partially Agree, CA- Completely Agree).

Factor	CD	PD	PA	CA
commited/motivated team	1	-	6	10
good third-party performance	-	2	13	2
client/user involvement	-	2	10	5
good leadership	1	2	7	7
effective communication/ feedback	-	3	5	9
project complexity	-	4	10	3
familiarity with technolog/dev. method	1	4	6	6
clear goals/objectives	-	6	6	5

4.2. Understanding Successful Software Projects

We also asked what the developers understand about a successful software project. After coding the answers, we found eleven distinct characteristics of successful software projects being evoked (see Table 7). Seven of these characteristics address critical success factors already categorized. From the other four characteristics raised by the developers, one addresses possible consequences of successful projects (customer satisfaction) rather than factors.

On the other hand, *qualified and sufficient team*, *following code standards*, and *realistic budget* may be considered additional critical success factors. One may see that seven of the eleven characteristics coded were spontaneously evoked at least by one-third of the developers, which may be considered a high frequency for open questions. These findings suggest a cohesion in the values and beliefs shared among the development teams investigated, reflecting their experience and the influence of the particular community in which they are inserted.

Table 7. Characteristics of successful software projects evoked by developers.

Characteristics	Frequency
clear requirements	9
customer satisfaction*	8
realistic schedule	7
clear goals/objectives	7
updated progress report	6
user/client involvement	6
qualified and sufficient team	6
familiarity with technology/dev. method	5
following code standards*	4
realistic budget*	3
adequate planning	3

4.3. Contributors for the Agility of Software Development

We also asked developers to spontaneously indicate which practices they understand are contributors to the agility of software development. The distribution of the coded answers is presented in Table 8. One can see that nine practices evoked by developers address agile practices already mapped in our research[de Mello et al. 2014]. Of these, six are agile practices largely applied among the development teams (*small releases*, *project visibility*, *whole team*, *daily meetings*, *product backlog*, and *pair programming*). The other two agile practices (*coding standards*,*collective code ownership*, and *software review*) are followed only by two teams. These findings suggest that developers are aware of the relevance of the practices they follow. However, coding standards seems to be a key concern, being also identified as a critical success factor.

4.4. Threats to Validity

Regarding the construct validity, we strongly grounded our survey questionnaire on results from previous work [de Mello et al. 2014, Pimenta and Santos 2016]. In this sense, we carefully verified whether the statements composed for the second block clearly address the 16 critical success factors mapped in [Pimenta and Santos 2016]. The second block of the survey questionnaire is composed of 16 statements. Long lists of survey items may be exhaustive, negatively influence on the quality of the developers' answers. We mitigate this threat by adopting a standardized model combining short statements with an intuitive four-scale Likert scale.

Table 8. Practices for software development agility evoked by software developers.

Practices	Frequency
daily meetings	7
collective code ownership	6
pair programming	6
coding standardization	5
project visibility	5
whole team	4
small releases	4
software review	4
product backlog	2

The first author of the paper work in the company investigated and also conducted the survey with his colleagues. This scenario is positive to allow performing a deeper analysis of the study findings and we will continue to explore it in the next research steps. However, it may also influence in the answers of the survey participants. In this way, we recruited developers from teams that the first author does not work with to participate. Besides, we applied in the third block open questions that would help to identify possible inconsistencies in the developers answers to the closed questions. We conducted a opinion survey in the particular context of a Brazilian Software Company. Therefore, the generalization of the findings reported in this study may be considerably limited. the general findings points to the contributions of replicating this survey in other companies with similar settings.

5. Discussion

The study involved developers distributed between four development teams from the same company. These teams share a common subset of agile practices identified as relevant in previous investigation [de Mello et al. 2014]. As presented in Section 4.1, most of the survey participants are experienced with software development. Besides, a great part of this experience comes from working in their current company. At a first glance, these overall characteristics may be considered benefic for reaching success in software projects. However, our study revealed that 40% to 53% of the survey participants have difficulty realizing that their projects achieve half of the critical success factors investigated. Despite this unfavourable numbers, we identified that all these factors may be enhanced by agile practices already followed by three or all the four teams involved in the survey sample (see Table 9.

In some cases, contrasting results are more clearly evident. For instance, several developers disagree that the progress reports of the projects are continually updated. However, all teams declared performing daily meetings and three teams declared applying practices for promoting the projects' visibility. Besides, several developers highlighted both agile practices in the open questions. This finding suggest the need of investigating in-depth how these practices have been applied and adapted for the context of remote

work. The study findings reveal the need for the teams to find ways to improve how they share their progress during development activities, which may be also harmed by the environmental changing caused by the remote work.

Another example of notable contrasting results address the lack of perception of clear requirements, the only success factor spontaneously evoked by most of developers. It is expected that the combined practices of small releases and product backlog would help the teams to manage the system requirements during the evaluations performed in the source code. One possible reason for this behaviour would address the facility for accessing the customers. Although it is considered an agile practice, teams with this resource should be careful to avoid excessively rely on the informal communication. In this way, we identified that with the managers that the developers are used to frequently keep in touch with the costumers to clarify the system requirements.

The lack of clear requirements harms not only the definition of reliable product backlog and reliable plans, but also establishing realistic schedules despite of the availability of whole teams composed of experienced developers. Without clear requirements, it is hard to estimate the development efforts and consequently feasible deadlines for each development task.

Table 5 also shows that technical agile practices associated with quality management and configuration management are fairly disseminated. These practices include development activities applied for identifying and combating the incidence of technical debt in the source code (*software review*, *refactoring*) and preventing the incidence of defects (*software review*, *TDD*). Here, it is important to note that code review and refactoring activities are important tasks inserted into the context of *pair programming*, which all teams claim to follow. Therefore, this finding point to concrete opportunities for the teams improving the pair programming practice.

Table 9. Mapping of the followed and non-followed agile practices addressing the worse evaluated critcial succes factors.

Critical Success Factors	**More Followed Practices**	**Less Followed Practices**
realistic schedule	whole team	-
clear requirements	whole team, product backlog, small releases	-
updated progress report	daily meetings, project visibility	-
good risk management	product backlog	-
good quality management	continuous integration, pair programming	refactoring, TDD, sw review
adequate planning	product backlog	-
effective control/monitoring	daily meetings	-
effective change/config. management	small releases, on site costumer	refactoring

Among the positive results, it is notable the positive effects of the company invest-

ing in whole teams predominantly composed of experience developers familiar with the development technologies and methods employed. The internal and external communication (customers/third-party) tend to be positively perceived by the developers. The teams are committed and have experienced leaders available. The adoption of coding standards contributes to the third-party performance.the survey findings also suggest that the overall experience of the developers and the low frequency of rotation among teams may contribute to their positive perception of communication with customers and third-party services providers. In this sense, the availability of the customers on-site is a positive contributor, which was successfully adapted to the remote work.

6. Planning the Action Research

As previously discussed, the survey results indicated gaps and some opportunities for improving the development practice of the teams investigated. This section presents the plan of a collaborative action research [Thiollent 1996] aiming at improving the perception of success in the team the first author works as a developer. The expectation is to run at least one complete action research cycle during a mid-term project. Despite the dynamic nature of action research, it is important to reflect on the main activities to be performed during the first cycle steps.

6.1. Problem Definition

We plan to conduct an open meeting with the participants to start the action research. In this meeting, we will first present the main results of the survey, with a focus on four critical success factors among those worst evaluated by 17 developers. They are *realistic schedule*, *clear requirements*, *good quality management*, and *effective change/configuration management*. We selected this subset once we previously identified with the manager that developers would have more autonomy for applying interventions addressing these factors. After this introduction, the first author will conduct a brainstorming with the team members to map the challenges observed by the developers to apply the agile practices already applied by the team addressing these factors. At the end of the meeting, the team members will be invited to reflect until the next meeting on feasible alternatives to overcome the raised challenges.

6.2. Action Planning

For this step, the first author will moderate a focus group session [Kontio et al. 2004] with the team members to identify and discuss possible interventions to overcome each challenge reported in the problem definition. Therefore, we plan to address a single challenge by focus group section. After the focus group, the validated interventions will be prioritized by the team (vote counting).

6.3. Action

We will identify a feasible subset of the higher prioritized interventions to be applied during the cycle, considering the additional effort and the resources required. The intervention and eventual guidelines/resources will be presented to all team members in a meeting. Then the developers will be oriented to apply the interventions from the beginning of a development project until its end.

6.4. Evaluation and Learning

The first author will work on the projects but he also will be continuously available to follow and orient the team members about the proposed interventions. Besides, he will observe the team dynamics for identifying possible gaps and new challenges. After the end of the project, we will invite the developers to answer an opinion survey on the perceived benefits and challenges of the interventions applied. Depending on the nature of the difficulties observed, we may change the data collection procedure to individual interviews. After that, we intend to conduct a briefing with the whole team to get feedback about the action research experience and gather opportunities for improving the following cycles, if needed.

7. Conclusion and Future Work

Reaching success in software projects is more than achieving budgets and schedules. It is part of a human need, enabled by the unconscious need to feel part of a group that can collaborate and overcome challenges together. As a human feeling, success is a complex thing composed of different and complementary facets.

We may say that the values raised by the agile manifesto two decades ago led the software engineering community to rethink (or remember) the values surrounding success from a holistic perspective. Different studies evidence that several agile practices may significantly contribute to this success when properly applied. However, as part of a complex socio-technical construct, these practices cannot be observed in isolation nor limited to specific methods. Developers should have in mind that agile practices are important resources for a higher purpose: promoting a holistic and collective perception of success among stakeholders.

The study reported in this paper exemplifies how even experienced software teams from companies promoting agile practices may have difficulties in being successful in their projects. The survey protocol reported in this paper may be employed in other software companies as a first step in diagnosing their bottlenecks for reaching success in their projects. This diagnosis should be preceded by an in-depth analysis of the context of each company and the composing teams. In the next step, we will run the action research introduced in this paper. Future work also includes replicating this survey in other companies.

8. Acknowledgements

The authors thank to the study participants for their valuable contributions.

References

Abrantes, J. F. and Travassos, G. H. (2013). Towards pertinent characteristics of agility and agile practices for software processes. *CLEI Electronic Journal*, 16(1):6–6.

Ahmad, M. O., Markkula, J., and Oivo, M. (2013). Kanban in software development: A systematic literature review. In *2013 39th Euromicro conference on software engineering and advanced applications*, pages 9–16. IEEE.

Chow, T. and Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of systems and software*, 81(6):961–971.

de Mello, R. and Coelho, M. (2021). Characterizing the experience of subjects in software engineering studies. *Proceedings of the XXIV Iberoamerican Conference on Software Engineering (CIbSE)*, pages 42–55.

de Mello, R., da Costa, J. A., de Oliveira, B., Ribeiro, M., Fonseca, B., Gheyi, R., Garcia, A., and Tiengo, W. (2021). Decoding confusing code: Social representations among developers. In *2021 IEEE/ACM 13th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 11–20. IEEE.

de Mello, R., da Silva, P., and Travassos, G. (2014). Agilidade em processos de software: Evidências sobre características de agilidade e práticas Ágeis. In *Anais do XIII Simpósio Brasileiro de Qualidade de Software*, pages 151–165, Porto Alegre, RS, Brasil. SBC.

de Mello, R. M., Da Silva, P. C., and Travassos, G. H. (2015). Investigating probabilistic sampling approaches for large-scale surveys in software engineering. *Journal of Software Engineering Research and Development*, 3(1):1–26.

Dutra, E., Lima, P., and Santos, G. (2020). An instrument to assess the organizational climate of agile teams-a preliminary study. In *19th Brazilian Symposium on Software Quality*, pages 1–10.

Dybå, T. and Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9-10):833–859.

Fowler, M., Highsmith, J., et al. (2001). The agile manifesto. *Software development*, 9(8):28–35.

Hannay, J. E., Dybå, T., Arisholm, E., and Sjøberg, D. I. (2009). The effectiveness of pair programming: A meta-analysis. *Information and software technology*, 51(7):1110–1122.

Iammarino, M., Zampetti, F., Aversano, L., and Di Penta, M. (2019). Self-admitted technical debt removal and refactoring actions: Co-occurrence or more? In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 186–190. IEEE.

Joshi, A., Kale, S., Chandel, S., and Pal, D. K. (2015). Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396.

Kamei, F., Pinto, G., Cartaxo, B., and Vasconcelos, A. (2017). On the benefits/limitations of agile software development: an interview study with brazilian companies. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software engineering*, pages 154–159.

Kontio, J., Lehtola, L., and Bragge, J. (2004). Using the focus group method in software engineering: obtaining practitioner and user experiences. In *Proceedings. 2004 International Symposium on Empirical Software Engineering, 2004. ISESE'04.*, pages 271–280. IEEE.

Leal, T. and Santos, G. (2015). Um survey sobre métodos ágeis e o pós-agilismo. In *Congresso Ibero-Americano de Engenharia de Software-CIBSE*.

Linaker, J., Sulaman, S. M., Höst, M., and de Mello, R. M. (2015). Guidelines for conducting surveys in software engineering v. 1.1. *Lund University*.

Nasir, M. H. N. and Sahibuddin, S. (2011). Critical success factors for software projects: A comparative study. *Scientific research and essays*, 6(10):2174–2186.

Pimenta, D. and Santos, G. (2016). Análise de práticas ágeis no apoio a fatores críticos de sucesso em projetos de software. *RelaTe-DIA*, 9(1).

Poppendieck, M. and Cusumano, M. A. (2012). Lean software development: A tutorial. *IEEE software*, 29(5):26–32.

Schwaber, K. and Beedle, M. (2002). *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River.

Thiollent, M. (1996). *Metodologia da pesquisa–ação (7ª edição)*. São Paulo-SP.

Zazworka, N., Vetro, A., Izurieta, C., Wong, S., Cai, Y., Seaman, C., and Shull, F. (2014). Comparing four approaches for technical debt identification. *Software Quality Journal*, 22(3):403–426.