WIP: A Preliminary Framework for Fostering Leadership Skills in Software Engineering Students

1st Nicolas Nascimento
School of Technology
PUCRS
Porto Alegre, Brazil
nicolas.nascimento@pucrs.br

2nd Rafael Chanin
School of Technology
PUCRS
Porto Alegre, Brazil
rafael.chanin@pucrs.br

3rd Afonso Sales
School of Technology
PUCRS
Porto Alegre, Brazil
afonso.sales@pucrs.br

***Abstract*—This work in progress innovative practice paper describes a preliminary framework for fostering leadership skills in software engineering students. The software industry is an environment of uncertainty, volatility and constant change. This has shaped the industry components and actors, generating methodologies that are able to cope with market expectations and software development requirements. Among these, agile is the most adopted. Professionals that possess skills to work in these agile teams, often named "soft skills", are challenging for universities to teach. From these skills, leadership, in particular, is a soft skill that has recently been studied by the software engineering education community but remains as an open research topic. A teaching methodology that demonstrated positive outcomes is active learning, as it centers the students in the process, enhances motivation and provides a collaborative approach for learning. In this context, this work aims at proposing a framework to train students to possess leadership skills using challenge based learning. To achieve this goal, we propose a preliminary framework to be used when teaching leadership skills to software engineering students in active learning environments. Our results provide indicatives that the framework could be effective at fostering shared leadership, with the proposed role rotation during the development cycle, but it also presents challenges in the process of generating guiding questions for learning as the process does not seem clear from the perspective of the experts.**

***Index Terms*—Leadership Teaching, Active Learning, Software Development**

I. Introduction

The software industry is immersed in an environment of great uncertainty, high volatility and constant change. This external scenario has shaped the industry, its components and actors, generating methodologies that are able to cope both with market expectations and software development requirements [1]. Agile (via its frameworks, such as Scrum [2] and XP [3]) is the one which has been adopted the most. By providing teams with an overarching set of practices to handle the many assignments a software development project requires while remaining flexible and adaptable enough to incorporate changes that business demands [4].

Professionals that work in agile teams are expected to not only technically excel, but also to possess abilities to handle conflict, perform on-demand adaptations and communicate effectively. These skills are usually denominated "soft skills" (non-technical skills, social skills, etc.) [5].

The quest of developing soft skills in software engineering graduates has been challenging for universities and the academia in general – with the industry demanding them [6], [7] and companies usually having to offer complementary training through, for example, R&D partnerships with universities [8]. From these set of soft skills, leadership, in particular, is a soft skill that has recently been studied by the software engineering research community. For instance, the relevance of leadership in software development [5], [9], styles of leadership (transformational, transactional, among others) and their impacts on the software development lifecycle [9]–[11], developing leadership in virtual and globally distributed teams [12]–[14], and leadership emergence and its antecedents in software engineering [15], [16] are some of the angles currently being investigated by the community.

There are some studies that touch relevant topics on the field, such as leadership teaching and training for software engineering students. Some of this studies, for example, investigate leadership in distance learning and as a component of teamwork [17], [18], teaching responsible leadership [19], using capstone projects [20] and even incorporating entrepreneurship in teaching [21], [22] and, thus, the challenge of understanding and adopting the proper way to teach this skill remains for educators.

Further, Challenge Based Learning (CBL), an active learning framework that promotes learning by engaging students to solve real world problems [23], has demonstrated potential in the teaching topics such as medicine and design [24], nurturing creative thinking in nursing [25] and development of mobile applications in agile contexts [26], even in combination with Lean Startup concepts. [27], [28].

In this context, this work aims at proposing a framework to train students to possess leadership skills using CBL to possess leadership skills. To achieve this goal, we have used an expert panel with a total of 5 active-learning experts to extract insights of potential positive and negative aspects of the proposed framework.

II. Framework

Our proposed framework is based on three concepts: **Scrum**; **Shared leadership**; **CBL**.

Agile development proposes a set of values that are very connect to recent market requirements and expectations, such as constant increments software releases, high emphasis on informal communication and direct stakeholder involvement. Towards this matter, we were able to identify that specifically the Scrum agile framework is applied when teaching leadership skills (embed with broader concept, such as teamwork) [29]–[32]. Thus, we have chosen to adopt Scrum as our initial pillar to proposed the framework.

Regarding shared leadership, we establish shared leadership as a team characteristic. In order to frame shared leadership, we have used definitions from the literature. Pearce [33] states that shared leadership is *"a simultaneous, ongoing, mutual influence process within a team that is characterized by 'serial emergence' of official as well as unofficial leaders"*. Dickinson and McIntyre [34] in their teamwork model, define *"team leadership"* as the tasks of *"providing direction, structure, and support for other team members. It does not necessarily refer to a single individual with formal authority over others. Team leadership can be shown by several team members,* e.g.*, explaining to other team members exactly what is needed from them during an assignment, listening to the concerns of other team member"*. Freire *et al.* [35] in their literature review, have found shared leadership as being defined simply as *"The authority on the decision making process is shared"*. Gren and Ralph [36], in assessing effective leadership in agile software development, have found that an important component of it is *"dynamic leadership"*, which is composed of three categories: (i) *"Have team members that take initiative and responsibility"*, (ii) *"Let team members lead"* and (iii) *"Step in, if needed"*.

From these definitions, we define shared leadership as an element of teamwork, in which all team members are able to learn how to:

1) Perform shared decision making - A team member's ability to participate in the decision-making process.
2) Set and update a commonly established set of goals - A team's ability to establish and update a common set of goals for a project.
3) Coordinate effort - A team member's ability to coordinate its effort to ensure project goals are achieved.
4) Maintain cohesion - A team member's ability to to work together to reach project goals and, when faced with project changes (requirements modifications), to adapt and distribute effort so that project goals are met.
5) Mutually support other team members - A team member's ability to support other team members whenever challenges for the achievement of the established goals happen.
6) Ensure a balanced contribution of team members - A team member ability to ensure every team member has a similar amount of contribution.
7) Take responsibility for team tasks - A team member's ability to perform team-related activities, such as external presentation, customer interactions, project planning.
8) Start and/or lead initiatives - A team member ability to allow any member to initiate and become responsible for new initiatives. Further, it also encompasses becoming responsible for any team member's initiatives.
9) Resolve internal team conflicts - A team member's ability to resolve internal conflicts which could be diminishing their effectiveness.
10) Provide feedback to team members - A team member's ability to provide feedback to other team members.
11) Learn continuously - A team member's ability to continuously learn and adapt.

In terms of active learning methodology (ALM), based on Santos [37] comparison of methodologies and overall presentation of motivations for the adoption of CBL in contrast to other methodologies, specifically due to its technology adoption, work environment simulation and the fact the teachers stops being at the center of the process. Thus, we choose to adopt CBL as the ALM of our proposal.

From these set of concepts, we conceptualized our framework by understanding CBL combinations. Santos *et al.* [26] propose a model that combines Scrum and CBL, specifically focusing on the development of mobile applications. Overall, results presented are positive and point that the combination of Scrum and CBL is interesting and likely to produce good results in terms of the execution of challenge-related activities and the training of mobile application developers.

In this context, we present an overview of our initial proposal for a framework to teach leadership skills to software engineering students in active learning environments using CBL. Figure 1 presents a visual representation of the framework and then we depict some of core concepts of the framework as follows.

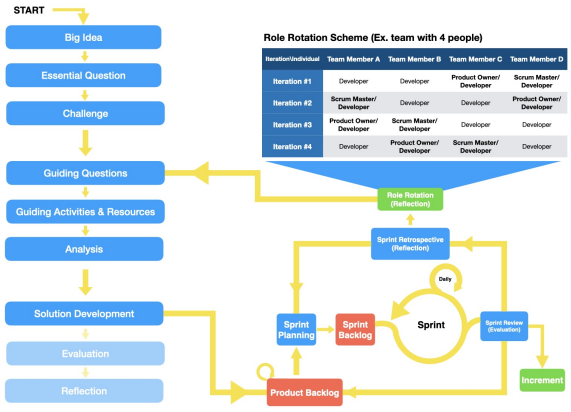

Fig. 1. Initial Proposal

Our proposal is based on the framework combining CBL with Scrum provided by Santos *et al.* [26], but adds modifications to the model so that it incorporates the capacity of enhancing the training capability of leadership skills. These modifications are explained below.

Roles: In the framework, we define role playing as an important component of the Scrum process. This means that every team that undergoes our framework must adopt at least three roles: *Developer*, *Developer/Scrum Master*, and *Developer/Product Owner*. All members of the Scrum team are developers. The "Developer/Scrum Master" role is performed by a team developer who, for a single iteration, will have to perform the activities that the Scrum Master (SM) has to perform. Thus, this member is responsible for the oversight of all Scrum processes. Furthermore, The "Developer/Product Owner" role is also performed by a team developer who, for a single iteration, will have to conduct the activities which are the responsibility of a Product Owner (PO) in a Scrum team. Thus, this member is responsible for the product backlog prioritization, user stories ambiguity resolution *etc*.

Role Rotation: In the framework, we introduce a mandatory ceremony we name *"Role Rotation"* to be conducted after the Sprint Retrospective. In this ceremony, students should switch roles based on a fixed schedule. The roles are attributed either

randomly or by personal interest in a specific role. After a finished iteration, as a follow-up to the Retrospective, students proceed to switch roles, such that both the "Developer/Product Owner" and "Developer/Scrum Master" are not assigned the same role again. Before changing roles, the entire team should discuss the learning from the SM and the PO so that everyone knows the state of the product and processes of the Scrum team. Further, doubts generated become guiding questions to be investigated by team members to learn and improve.

Active learning: In the framework, CBL is always present. During the Sprint Retrospective, a ceremony we name *"Role Rotation"* should take place and the results of this ceremony serve as input for the creation of more guiding questions for the team's investigation and learning.

Combing all these concepts together, we propose that all of these are able to provide opportunities for the student to have the chance of learn by doing and thus are able to achieve the proposed components of shared leadership.

Regarding minimum number of iterations, our proposal establishes that every team member should be a "Developer/Product Owner" and "Developer/Scrum Master" at least once. So we can calculate the minimum number of iterations for a team based on the number of team members. For example, a team with 4 team members requires 4 iterations minimally. After this minimum number of cycle ends, initially, the framework adopts a "keep cycling" policy, where rotations should be maintained until the end of the project life cycle. The effectiveness of such additional cycling towards learning is a topic to be assessed in a future study we will conduct. Moreover, the framework requires the team to be composed of at least two team members as it is necessary for the roles of "Developer/Product Owner" and "Developer/Scrum Master" to be played.

III. Methodology

In order to achieve the goal of this study, we have perform an expert panel. Expert panel research can be used to capture expert judgment [38] and aid in the improvement of hypothesis generated. In addition, experts are usually better at predicting and even preventing [39] errors when performing certain tasks.

Thus, our study has an exploratory approach and is qualitative. To ensure reliability of results, we have followed the guidelines proposed by Maxwell [40]. As such, the goal of this research is *proposing a framework to train students to possess leadership skills using CBL to possess leadership skills*. We have established two research questions: **RQ1** - *What are the positive aspects about the proposed leadership teaching framework?* and **RQ2** - *What are the negative aspects about the proposed leadership teaching framework?*.

To answer these questions, we have sought the opinion of active-learning teaching experts. These experts were selected due to their expertise teaching in active learning environments and due to convenience. Experts who were interested in participating in the study were presented to this questionnaire used in the research. The questions used are presented below:

1) *How many years of experience do you have teaching Scrum?*
2) *How many years of experience do you have teaching CBL?*
3) *How many years of experience do you have teaching software engineering?*
4) *In your opinion, what are the positive aspects of the proposed leadership teaching framework?*
5) *In your opinion, what are the negative aspects of the proposed leadership teaching framework?*

The first three questions were used to assess the familiarity of participants with the concepts of the proposed framework, Scrum, CBL and Software Engineering. Our goal with these questions was to extract the expertise level of each participant. Questions four and five would directly aid us to answer our research questions. These questions were personal and open-ended. Finally, a space was provided for additional comments to be written.

All the data obtained from the questionnaire was achieved via an interview with the five study participants. All interviews were conducted in person with one of the researchers facilitating the interaction. All interviews were recorded so that transcription could be performed afterwards. All participants were presented to the research terms and agreed to participate before the answer was provided. After collecting answers, we have manually transcripted all the interviews and organized them using a spreadsheet. These spreadsheets contained all answers from participants in a textual manner.

From this point, we began a qualitative analysis of data, following the principles of Maxwell [40]. The analysis was done using two strategies: clustering and categorization. We first clustered answers which presented similar results, counting the number of occurrences of a particular topic. In addition to this, as a qualitative study, we have collected interesting insights from participants. Afterwards, the generated clusters were categorized (which could be one or more) regarding the general topic they addressed. As a final step, we have grouped similar categories. Finally, using the available data and generated analysis, we were able to extract insights from the data. The results of our analysis are presented in the following section.

IV. Preliminary Results

Demography - Our questionnaire had 5 valid responses. From a demography standpoint, all participants had 6.2 years of experience with Scrum, with the maximum year of experience being 11 years and the minimum 3 years. In terms of CBL, participants had at least three years of experience, and an average of 5.4 years of experience in such environments, with a maximum of nine years. Regarding teaching, participants had at least one year of experience teaching, with an average of 5.8 years and a maximum of 13 years.

RQ1 - What are the positive aspects about the proposed leadership teaching framework? - Study participants have reported in two instances that the visual representation proposed in the framework was decent, as it incorporates both CBL, for the learning facet, and Scrum, for the development facet (P1 - *"[...] we can see it, the illustration, the diagram. It is structured in a way that it is possible to understand well what are the steps and what happens in each one."*).

Experts have reported in three instances that they felt they need to try the framework in practice to be able to see its

effects (P3 - *"Because I have not tried this framework/model I don't know how much of an impact it will have regarding time having this role switching."*). Further, they proposed to try the framework in other context, such as companies which already have something similar to this (P2 - *"I remembered the Rotation Program from company XYZ in which they provide you with the opportunity to go to different places in the company to found out where you fit better and, wanting or not, with this you will get to know various places [...]"*).

Experts also pointed out in six occurrences that they saw the framework as positive as it "forces" every team member to try some leadership roles (P3 - *"The first think I notice is that when we have this rotation, we kind of obligate people to take a leadership position and thus get everyone to have the same experience, which tends to level the experience level regarding these leadership roles – PO and SM."*). They saw it as a learning opportunity both at being a leader and also at teaching others (P4 - *"I think this case is works very well and I see a lot that the role rotation ceremonies serves as this point of... this opportunity of stopping and say "OK, let me pass it to you (the role and) some knowledge and impressions that occurred to me during this week I performed this role" [...]"*). Furthermore, experts thought that the shared decision process could be enhance given that everybody had to chance at being a PO and a SM. (P1 - *"[...] you were talking of shared decision making, making decisions in group, maybe keeping something like "SM knows what has to be done and the PO knows what has to be done" but maybe these people they don't do these things but they are responsible for stimulating others to do them together with this person [...]"*).

Further, in two occurrences, experts have reported that the process of generating guiding questions after the role rotation ceremony was a nice element (P5 - *"The thing I liked the most in you talk was "In this new ritual, we will generate more guiding questions"."*).

RQ2 - What are the negative aspects about the proposed leadership teaching framework? Experts pointed out in eight occurrences that the need to return to the CBL process, by generating new guiding questions that have to be investigated, could be a time consuming activity (P1 - *"The thing of returning from the role rotation to the guiding questions I don't know if it ends up... While I learning about Scrum, one thing that bothered me a little was the bureaucracy or too many processes [...]"*).

Experts have also added five times that being a leader could be challenging for different types of team members, such as designers (P5 - *"Have you thought of other roles besides developer? For example, designer? that could benefit from this [...]"*). This was further mentioned when experts reported that passing information when leaving a role could be challenging (P1 - *"[...] This communication bridge with the old and new (PO and SM) maybe could be something that is mapped or I don't know, an one-on-one really, with the old PO and the new."*). Moreover, they have also pointed out that some team members might not want to be leaders (P2 - *"[...] There are people who are like "I never want to be a SM", but this person kind will have to take this role [...]"*). Finally, they have reported that rotation in a industry setting is challenging (P1 - *"[...] But in a context of a real company with a client I think the approach would not work, or it would have to be re-thought. Like the PO would not change for example, and I don't know how the SM would work."*).

Experts have reported in two instances that a product owner in a software development context could lead to a behavior of not having strict requirements and deadlines (P4 - *"[...] Given that the POs they are the project owners and the developers, everybody is the same thing meaning there is not an external entity generating a "tension", it is very easy for them to automatically say "OK, this is high priority, but this will be a lot of work, let's lower its priority? Let's do it. Let's remove this feature?" [...]"*). Some even proposed an external PO (P2 - *"[...] One thing I saw someone commenting on is that usually the PO, ideally, should be someone from outside the development team, from outside this."*).

Experts have pointed out six times that the CBL and Scrum interaction is itself challenging, with Scrum possessing a learning curve having a different focus (*e.g.*, development) in comparison to CBL (P5 - *"[...] When it comes to teaching, CBL is the strongest element. When it comes to software development, maybe Scrum is the strongest element, right?[...]"*). Some experts even challenged the framework in the sense that maybe Scrum should be at the center, not CBL (P5 - *"[...] CBL is the base to Scrum begin. Have thought of the opposite? Scrum is the base and you fit CBL in the sprints* [1] *or in the entire process [...]"*).

Experts also stressed in two occurrences that the number of iterations should be equal to N+1, when N is the number of people in the group and that the last iteration could serve to answer guiding questions raised during the N-th iteration (P5 - *"[...] Wouldn't it be the case that you have 5 cycles for a group of 4 people? Wouldn't it be the case that you always have 1 cycle more than the group size? I imagine that the last cycle will be problematic as well, not necessarily problematic, but it will raise questions, but raising questions only makes sense if you have one more cycle to solve them."*).

V. Conclusion

This paper has presented a proposal of a framework to train students to possess leadership skills using CBL. To test the framework, we have conducted an expert panel research with 5 interviews with active learning specialists. Overall, our results provide indicatives that the framework could be effective at fostering shared leadership, by providing a learning opportunity to team members with the proposed role rotation during the development cycle, but it also presents challenges in the process of generating guiding questions for learning as this process does not seem clear from the perspective of the experts. In the future, we intent to expand the work by proposing a new version of the framework and evaluating it in a case study in an active learning environment.

[1] In Scrum, a sprint is a fixed length event of one month or less [2].

ACKNOWLEDGMENTS

This paper was supported by the Ministry of Science, Technology, and Innovations, with resources from Law No. 8.248, dated October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex.

REFERENCES

[1] M. Fowler, J. Highsmith *et al.*, "The agile manifesto," *Journal of Software Development*, vol. 9, no. 8, pp. 28–35, August 2001.

[2] K. Schwaber and J. Sutherland, *The scrum guide*. Scrum Alliance, 2011.

[3] K. Beck, *Extreme programming explained: embrace change*. Addison-Wesley Longman, 2000.

[4] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," pp. 1213–1221, 2012.

[5] G. Matturro, F. Raschetti, and C. Fontán, "Soft skills in software development teams: A survey of the points of view of team leaders and team members," in *International Workshop on Cooperative and Human Aspects of Software Engineering*. IEEE, 2015, pp. 101–104.

[6] M. Stevens and R. Norman, "Industry expectations of soft skills in it graduates: a regional survey," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2016, pp. 1–9.

[7] F. Bona., R. Chanin., N. Nascimento., and A. Sales., "Understanding the gaps in software engineering education from the perspective of it leaders: A field study," in *Proceedings of the International Conference on Computer Supported Education (CSEDU)*, INSTICC. SciTePress, 2023, pp. 511–518.

[8] F. R. Monteiro, P. A. Pereira, L. C. Cordeiro, C. F. Costa Filho, and M. G. Costa, "Complementary training programme for electrical and computer engineering students through an industrial-academic collaboration," in *Frontiers in Education Conference*. IEEE, 2016, pp. 1–9.

[9] S. Faraj and V. Sambamurthy, "Leadership of information systems development projects," *IEEE Transactions on engineering management*, vol. 53, no. 2, pp. 238–249, 2006.

[10] C. Athukorala, I. Perera, and D. Meedeniya, "The impact of transformational and transactional leadership styles on knowledge creation in sri lankan software industry," in *2016 Moratuwa Engineering Research Conference (MERCon)*. IEEE, 2016, pp. 309–314.

[11] F. Q. da Silva, C. V. Monteiro, I. E. dos Santos, and L. F. Capretz, "How software development group leaders influence team members' innovative behavior," *IEEE Software*, vol. 33, no. 5, pp. 106–109, 2016.

[12] K. Furumo, E. de Pillis, and M. Buxton, "The impact of leadership on participation and trust in virtual teams," in *Proceedings of Annual conference on Computers and People Research*, 2012, pp. 123–126.

[13] R. S. Sangwan and J. Ros, "Architecture leadership and management in globally distributed software development," in *Proceedings of the first international workshop on Leadership and management in software architecture*, 2008, pp. 17–22.

[14] A. Hidayati, E. K. Budiardjo, and B. Purwandari, "Hard and soft skills for scrum global software development teams," in *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, 2020, pp. 110–114.

[15] L. Przybilla, M. Wiesche, and H. Krcmar, "Emergent leadership in agile teams–an initial exploration," in *Proceedings of the Computers and People Research Conference*, 2019, pp. 176–179.

[16] L. Przybilla, A. Präg, M. Wiesche, and H. Krcmar, "A conceptual model of antecedents of emergent leadership in agile teams," in *Proceedings of the on Computers and People Research Conference*, 2020, pp. 164–165.

[17] J. M. D. Marquez, S. E. Aguja, and M. S. Prudente, "Evaluation of student leadership development amidst online distance learning set-up," in *Proceedings of the International Conference on E-Education, E-Business, E-Management, and E-Learning*, 2022, pp. 118–123.

[18] R. Vivian, K. Falkner, and N. Falkner, "Analysing computer science students' teamwork role adoption in an online self-organised teamwork activity," in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, 2013, pp. 105–114.

[19] D. Goyal, R. Cortinovis, and L. F. Capretz, "A framework for class activities to cultivate responsible leadership in software engineering students," in *Proceedings of International Conference on Cooperative and Human Aspects of Software Engineering*, 2022, pp. 96–101.

[20] J.-G. Schneider, P. W. Eklund, K. Lee, F. Chen, A. Cain, and M. Abdelrazek, "Adopting industry agile practices in large-scale capstone education," in *Proceedings of the International Conference on Software Engineering: Software Engineering Education and Training*, 2020, pp. 119–129.

[21] E. D. Moreno, J. M. Fernandes, V. Alves, M. E. Leon Olave, and P. Afonso, "Transforming ideas and developing entrepreneurship skills in computing sciences and informatics engineering courses," in *Proceedings of the 11th Euro American Conference on Telematics and Information Systems*, 2022, pp. 1–6.

[22] S. Tenhunen, T. Männistö, P. Ihantola, J. Kousa, and M. Luukkainen, "Software startup within a university–producing industry-ready graduates," *arXiv preprint arXiv:2301.07020*, 2023.

[23] M. Nichols, K. Cator, and M. Torres, *Challenge based learning guide*. Digital Promise, 2016.

[24] S. E. Gallagher and T. Savage, "Challenge-based learning in higher education: an exploratory literature review," *Teaching in Higher Education*, pp. 1–23, 2020.

[25] Z. Yang, Y. Zhou, J. W. Chung, Q. Tang, L. Jiang, and T. K. Wong, "Challenge based learning nurtures creative thinking: An evaluative study," *Nurse education today*, vol. 71, pp. 40–47, 2018.

[26] A. R. Santos, A. Sales, P. Fernandes, and M. Nichols, "Combining challenge-based learning and scrum framework for mobile application development," in *Proceedings of the 20th Conference on Innovation and Technology in Computer Science Education*, 2015, pp. 189–194.

[27] R. Chanin, A. Sales, L. B. Pompermaier, and R. Prikladnicki, "Challenge based startup learning: a framework to teach software startup," in *Proceedings of the Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2018, pp. 266–271. [Online]. Available: https://doi.org/10.1145/3197091.3197122

[28] R. Chanin, A. Sales, A. R. Santos, L. Pompermaier, and R. Prikladnicki, "A collaborative approach to teaching software startups: findings from a study using challenge based learning," in *Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering*, 2018, pp. 9–12.

[29] M. Paasivaara, "Teaching the scrum master role using professional agile coaches and communities of practice," in *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. IEEE, 2021, pp. 30–39.

[30] A. Fontão, B. Gadelha, and A. C. Júnior, "Balancing theory and practice in software engineering education–a pbl, toolset based approach," in *IEEE Frontiers in Education Conference*. IEEE, 2019, pp. 1–8.

[31] Z. S. Li, N. N. Arony, K. Devathasan, and D. Damian, "" software is the easy part of software engineering"–lessons and experiences from a large-scale, multi-team capstone course," *preprint arXiv:2302.05536*, 2023.

[32] J. Libreros, I. Viveros, M. Trujillo, M. Gaona, and D. Cuadrado, "Improving soft skills in agile software development by team leader rotation," in *International Congress on Information and Communication Technology*. Springer, 2020, pp. 186–194.

[33] C. L. Pearce, "The future of leadership: Combining vertical and shared leadership to transform knowledge work," *Academy of Management Perspectives*, vol. 18, no. 1, pp. 47–57, 2004.

[34] T. L. Dickinson and R. M. McIntyre, "A conceptual framework for teamwork measurement," *Team performance assessment and measurement*, pp. 19–43, 1997.

[35] A. Freire, M. Perkusich, R. Saraiva, H. Almeida, and A. Perkusich, "A bayesian networks-based approach to assess and improve the teamwork quality of agile teams," *Information and Software Technology*, vol. 100, pp. 119–132, 2018.

[36] L. Gren and P. Ralph, "What makes effective leadership in agile software development teams?" in *Proceedings of the 44th international conference on software engineering*, 2022, pp. 2402–2414.

[37] A. R. d. Santos, "Um método de aprendizagem baseada em desafios: um estudo de caso em ambientes de desenvolvimento de aplicativos," PhD Thesis, Pontifícia Universidade Católica do Rio Grande do Sul, 2016.

[38] T. Rosqvist, M. Koskela, and H. Harju, "Software quality evaluation based on expert judgement," *Journal of Software Quality*, vol. 11, no. 1, pp. 39–55, May 2003.

[39] S. Lauesen and O. Vinter, "Preventing requirement defects: An experiment in process improvement," *Journal of Requirements Engineering*, vol. 6, no. 1, pp. 37–50, February 2001.

[40] J. Maxwell, *Designing a qualitative study*. Sage Publications, 2008, ch. 7, pp. 214–253.