Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5981
This work is licensed under a Creative Commons Attribution 4.0 International License.

Teaching Leadership in Agile Software Engineering Education: A Case Study using Scrum and Challenge Based Learning

Nicolas Nascimento [**Pontifical Catholic University of Rio Grande do Sul** | ***nicolas.nascimento@pucrs.br***]
Afonso Sales [**Pontifical Catholic University of Rio Grande do Sul** | ***afonso.sales@pucrs.br***]
Rafael Chanin [**Pontifical Catholic University of Rio Grande do Sul** | ***rafael.chanin@pucrs.br***]

Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, RS – 90619-900 – Brazil.

Received: 02 May 2025 • **Accepted:** 02 December 2025 • **Published:** 05 May 2026

Abstract. The increasing adoption of agile methodologies in software engineering has emphasized the relevance of soft skills, particularly leadership. However, little is known about effective strategies to teach and develop these skills in educational settings. In this context, this paper presents an exploratory case study investigating the impact of assigning leadership roles — specifically, the Product Owner role — within agile student teams engaged in a Challenge Based Learning environment. Our main goal in the study was to improve the understanding of role assignments in software engineering education. To achieve this, we have conducted a case study over 10 weeks with 45 undergraduate students across 9 Scrum teams examining six soft skills: communication, coordination, cohesion, teamwork, leadership, and motivation. Data were collected through semi-structured interviews and analyzed qualitatively. Our results provide indications that assigning leadership roles has the potential to enhance students' awareness and development of leadership competencies, promotes team engagement, and supports the alignment of individual and collective goals. The study offers evidence that integrating role-based leadership training in agile learning environments is a valuable pedagogical approach for software engineering education.

Keywords: Active Learning, Leadership Teaching, Software Development

1 Introduction

The shift of agile software development from a novel industry innovation to a widely accepted standard practice has significantly altered interactions among software developers, management, and stakeholders in product creation. This approach, as outlined in the Agile Manifesto [Fowler *et al*., 2001], emphasizes frequent communication, the delivery of small increments of the product, collaborative efforts, and swift adaptation to consumer needs. These practices are designed to meet the market's high expectations for speed, reliability, and adaptability in software development.

This focus on soft skills which the manifesto highlights has required companies to search for software developers that not only possess strong programming skills, but also well developed soft skills, such as good communication and leadership skills [Li *et al*., 2015]. Leadership, in particular, is a soft skill that has recently been studied by the software engineering research community. For instance, some of the topics covered by the community are the relevance of leadership in software development [Faraj and Sambamurthy, 2006; Matturro *et al*., 2015], styles of leadership (transformational, transactional, among others) and their impacts on the software development lifecycle [Athukorala *et al*., 2016; da Silva *et al*., 2016; Faraj and Sambamurthy, 2006; Van Kelle *et al*., 2015], developing leadership in virtual and globally distributed teams [Furumo *et al*., 2012; Sangwan and Ros, 2008; Hidayati *et al*., 2020], and leadership emergence and its antecedents in software engineering [Przybilla *et al*., 2019, 2020].

There are few studies, however, that touch relevant topics of the field, such as leadership teaching and training for software engineering students. Some of these studies, for example, investigate leadership in distance learning [Marquez *et al*., 2022] and teaching responsible leadership [Goyal *et al*., 2022] and, thus, the challenge of understanding and adopting the proper way to teach this skill remains for educators.

In this context, Challenge Based Learning (CBL), an active learning framework that promotes learning by engaging students to solve real world problems [Nichols *et al*., 2016], has demonstrated potential in the teaching of topics such as medicine and design [Gallagher and Savage, 2020], nurturing creative thinking in nursing [Yang *et al*., 2018] and development of mobile applications in agile contexts [Santos *et al*., 2015], even in combination with Lean Startup [Chanin *et al*., 2018a,b] concepts. Given that CBL fosters group collaboration and that these groups often develop real-world projects (*i.e.* mobile apps), exploring the training leadership skills could be a new frontier.

Thus, the goal of this paper is to *investigate leadership teaching in a challenge based learning environment*. Further, we have conducted a 10-week case study with 9 agile software development teams, with 45 individuals in total, as part of a two-year mobile development course that teaches iOS development to college students through CBL. In these teams, we adopted the Scrum agile framework [Schwaber and Sutherland, 2011], assigning the role of Product Owner to one individual. The research question we used as guidance to achieve this goal is: *"What are the impacts to software development student teams of assigning a leadership role (PO) to a team member at the individual level and at the team level?"*.

Overall, our results indicated that assigning a leadership

role to a team member in a software development team tends to be an overall valuable learning experience for new developers. Furthermore, apart from being challenged to be a leader, they also become aware of the duties of leading a team, such as balancing personal goals with project objectives, communicating effectively and sharing leadership.

The remainder of this paper is organized as follows: in Section 2 we explore important concepts and the background for this research. Section 3 depicts the methodology used in this study. In Section 4 we explore our results, followed by Section 5, in which we take away the most important insights. Section 6 presents the limitations of this study. Finally, Section 7 concludes the paper and indicates future work.

2 Background

Since our main study focus revolved around soft skills of study participants, it became important for us to frame the soft skills addressed in the research, *e.g.* (i) **Communication**, (ii) **Coordination**, (iii) **Cohesion**, (iv) **Teamwork**, (v) **Leadership** and (vi) **Motivation**. We propose the following definitions for each one of these:

Communication, as a soft skill, refers to a team and team members ability to communicate effectively. In the context of software development, we define two types of communication: internal, among team members, and external, with external stakeholders. Internal communication refers to a team's information exchanges about the software project (features, bugs, deadlines, *etc.*) and the team itself. External communication refers to communication that is performed among the team (or some of its members) and the stakeholders of a project.

Hoegl and Gemuenden [2001] define **Coordination** as a *"capability of team members to work together to reach project goals"*. In our case, we further define this as a team's ability to manage its effort, by distributing it appropriately among its members and, during the agile development iterations, to deliver business proposed software increments in the appropriate deadlines.

In terms of team **Cohesion**, Wellington *et al*. [2005] defines it as *"the degree to which the team sticks together as they pursue the team's purpose"*. We use a similar definition to this, by defining cohesion as the ability a team and its members have to adapt and distribute effort so that project goals are met, even when faced with project changes (requirements modification).

Salas *et al*. [2015] defines **Teamwork** as *"a group's willingness to pursue a common cause in a climate of mutual aid and spontaneous co-operation"*. Hoegl and Gemuenden [2001] also define that teamwork is composed of *communication, coordination, balance of member contributions, mutual support and effort and cohesion*. Specifically *balance of member contributions, mutual support and effort* are new topics not defined by the previously discussed soft skills. Thus, to better separate teamwork from the other soft skills, we will define teamwork in a team exclusively as the balance a team possesses in the contributions of its members, the level of mutual support among its team members and the amount of effort that is performed by a team.

In agile, **Leadership** and the task of guiding the team, prioritizing user stories and ensuring business goals are met are responsibilities of the Product Owner (or Product Manager) [Schwaber and Sutherland, 2011]. The product owner is a team member that is either fully or partially dedicated to facilitating the interlocution between the development team and stakeholders by managing the product backlog. Leadership can also be performed by a team outsider, such as Project Manager, but an agile software development team should possess autonomy and independency and thus be able to practice some leadership towards the roadmap of the product under development. In our case, drawing from these concepts we will define leadership in a team as the ability of the PO to focus the effort of the team to achieve business goals.

In a broad view, **Motivation** can be defined as a person's (or team's) impetus or inspiration to execute an action (or series of actions) and can be either intrinsic, where the impetus arises from internal reasons or motives, or extrinsic, when the external environment pushes conditions that require action [Ryan and Deci, 2000]. Specifically regarding software engineering, [Beecham *et al*., 2008] have performed a systematic literature review on motivation in software engineering and enumerates that the top five motivators in a software development project were: *necessity of task identification*, *personal interest*, *understanding the task's purpose*, *having job satisfaction* and *working on an identifiable piece of quality work*. All these five elements present a mixture of intrinsic and extrinsic factors. Considering the educational context in which we are focusing, we can reduce the importance of *career path*, as it is not typically expected for an educational environment to promote a career path in the same way a company does. A more adequate motivator would be that of a *learning path*. From this perspective, in our framework, we define motivation as team's ability to possess intrinsic motivators and react to external motivators in a beneficial way for the development of a software project.

3 Methodology

To achieve the goal proposed by this study, we have performed an exploratory case study in a two-year mobile development course that teaches software development focusing on Apple platforms to undergraduate students. The case study was conducted with the participation of 9 agile teams that develop mobile applications. All agile teams used Scrum [Schwaber and Sutherland, 2011] as their agile framework. Since our study was carried out in a Research and Development (R&D) environment, we are not able to make raw data available. The guidelines adopted to the elaboration of the study design were the standard proposed by Runeson and Höst [2009].

As a research method, case studies can be used for software engineering research, as they allow the understanding of a certain phenomena in its natural occurring context [Runeson and Höst, 2009] and are suited to evaluate a method and tool [Kitchenham *et al*., 1995]. As we had few hypothesis being previously established, we have chosen to conduct an exploratory study. As such, case study methodology is suited to be used as it enables the researcher to extract new insights

and ideas, to understand what is happening and to generate hypothesis for other researches [Runeson and Höst, 2009].

We have aimed to investigate the leadership of the individuals assigned as Product Owners (PO) in each agile team. The software teams were our unit of analysis. Hence we have defined the following Research Question (RQ) to guide our study:

> *"What are the impacts to software development student teams of assigning a leadership role (PO) to a team member at the individual level and at the team level?"*

Our goal with this RQ was to assess the impacts to student teams of assigning a leadership role (PO) to a specific team member from the perspective of the assigned individual to himself/herself and from assigned individual to the team.

3.1 Case Study Context

The study was conducted in a R&D environment[1] that develops Apple mobile applications. In the environment, students are introduced to the development ecosystem of Apple platforms and obtain experience by working together in agile teams. The teaching in the environment is done through CBL [Nichols and Cator, 2008]. Students can opt to specialize in software development, known as *"developer"*, or interface design, known as *"designer"*, during their course. However, they receive exposure to both domains throughout their training period. These students are supported by tech, design and business mentors who have solid experience in the software industry. We will refer to these mentors as leadership team in this paper. The teams work on the development of apps that supports users to overcome real-world problems. The environment encourages the team members to cooperate in all software development stages from requirements elicitation to deployment and maintenance steps. All teams use Scrum to manage their work [Schwaber and Sutherland, 2011].

3.1.1 Challenge Based Learning

Challenge Based Learning (CBL) [Nichols *et al*., 2016] is a learning framework based on solving real world challenges. *"The foundations of experiential learning can be found within the history of most cultures, but were formally organized and presented by David Kolb drawing heavily on the works of John Dewey and Jean Piaget"* [Santos *et al*., 2015]. CBL was developed by educators working with *Apple Inc.* [Nichols and Cator, 2008] and has been implemented both in educational and corporate environments. From an education perspective, students acquire knowledge by working on open-ended problems.

In the CBL process, students, tutors, and other stakeholders work together as active collaborators and the focus is not on the final deliverable, but rather on the whole process. In this sense, students must reflect from time to time on their learning evolution.

The CBL process begins with the definition of a *big idea*, which is a broad concept that can be explored in several ways. The big idea has to be engaging and important to students. Once the big idea is chosen and the *essential question* is created, the *challenge* is defined. From this point, students must come up with the *guiding questions* and *guiding activities and resources*, which will guide them to develop a successful solution. The next step is *analysis*, which will set the foundation for the definition of the *solution*. Once the solution is agreed upon, the *implementation* begins. Finally, *evaluation* is carried out in order to examine the whole process and verify if the solution can be refined.

Before our study was conducted, all participants had already been attending the course for more than a year and thus had practical experience with developing software using Scrum in software projects. Furthermore, students also participated in workshops that enhanced soft skills, including leadership.

Considering our study, during the first 4 weeks, teams went through a research phase, in which they proposed solutions to real-world problems and searched for evidence that would validate these proposals. To validate the solutions, teams employed a variety of methods, such interviews with experts, market research, literature reviews and even real-life experience.

After this initial research period, in the following 6 weeks, teams went through a development phase where three 2-week iterations were conducted. In the first iteration, teams employed an adaption of Scrum by applying the concept of a Sprint 0 [Silva *et al*., 2013]. Specifically in Sprint 0, the teams proposed the features of the application to be delivered at the end of the development phase (which we will refer to as **V1**) and conducted technically focused investigation to assess feasibility of the features. These investigations also provided insights to the team regarding user characteristics and needs. Considering these findings, the teams were able to consolidate an initial product backlog. In addition to the product backlog, at the end of Sprint 0, all teams have to designate two team members to have an additional role, a Scrum Master (SM), who would be responsible for facilitating all Scrum ceremonies, and a Product Owner (PO), who would be the team leader and main responsible for the product backlog update and prioritization.

After that, both in Sprint 1 and 2, the teams proceeded to the sprint planning and defined a set of user stories (USs) which were prioritized and then moved to the current sprint backlog. Taking into account the sprint backlog, the teams discussed and defined a set of acceptance criteria (AC) for each USs of the sprint backlog. In the environment, the sprint planning lasts for 4 hours on average, in line with the Scrum guide [Schwaber and Sutherland, 2011]. The teams are used to adopting US in their projects and they have 1.5 year experience with US specifications. As the AC of all USs were finished, the teams started the Scrum sprint. Each SM in the team ensured the teams performed the usual Scrum ceremonies during the sprint (*i.e.*, daily meetings, sprint review, sprint retrospective). Sprints were conducted by every team independently, in parallel with others and in sync with the defined 2 week sprints. In total, 3 development sprints were conducted. In our study, each team (N=9) was composed of 4-7 members plus one external project manager (PM). From these 4-7 members, at least one was a dedicated designer

[1]The environment has a restriction on the information that can be made public, but we have published as much as possible.

(see Table 1). The Scrum Master (SM) role was performed by a team member and the Product Owner (POs) role was performed by a different team member. The external project manager was a member of the leadership team and had a supportive role aiming to foster students to learn by themselves as much as possible. Each team worked on developing mobile apps which addressed different scopes (see Table 1). The team members dedicated 20 hours per week to software development activities. For the purpose of our study, the leader of the team was designated as the PO, and this person was expected to develop leadership skills.

Table 1. Team characteristics

Team #	Project Scope	Team Formation		
		Developers	Designers	PM
1	Services	3	1	PM1
2	Competition	4	2	PM1
3	Fashion	4	1	PM2
4	Traveling	4	2	PM2
5	People Interaction	3	1	PM3
6	Sports	3	1	PM4
7	Education	5	2	PM5
8	Volunteering	3	1	PM3
9	Health	4	1	PM4

Every team's PO had a bi-weekly meeting with the leadership team during the development phase of study. During this meeting, each PO presented the following items for their team: (i) Product backlog and the sprint backlog; (ii) Progress towards V1 and burndown chart [Woodward *et al*., 2013] for current sprint; (iii) Sprint highlights and lowlights; (iv) Sprint impediments and respective mitigations; (v) Future risks for V1 delivery and respective mitigations.

The leadership team also asked questions about the team's productivity and discussed strategies that the PO had to bring to his/her team so this metric could be enhanced.

3.2 Sampling

Our study main data source were answers provided by the teams in a semi-structured interview. The sample population (45 students) was composed of undergraduate students in a mobile application development program, who were chosen using the convenience criteria due to the fact that participants were selected for their availability. The sample population size was defined using the higher number of available people to participate in this study.

In order to avoid students from associating the questions provided in the data collection with their individual evaluation in the program, each student was told about the purpose of the research and about the zero-impact their answers had on their evaluation. After this, all participants provided informed consent.

Considering the participants of the case study, the teams were composed of 19 participants that self-identify as female and 26 that self-identify as male. Further, 3 students were up to 20 years old, 32 were between 21 and 25, and 10 were over 25, with the oldest student being 36. Most students were pursuing an undergraduate' degree with a single exception pursuing a master's degree. In general, all 45 students had at least 1.5 year of experience working with mobile application development, with 28 of those having had more than 1.5 year of experience in some level with mobile application development. In general, the teams also had around 1.5 year of experience with User Experience (UX). All the projects considered in the case study involved graphical interface design.

3.2.1 Semi-structured interview

Specifically the semi-structured interview was conducted at the end of the 10-week period in which the software products were delivered. In this interview, excluding demography, we focused our questioning in the soft skills. Specifically, we have chosen to further investigate: (i) Communication, (ii) Coordination, (iii) Cohesion, (iv) Teamwork, (v) Leadership, and (vi) Motivation. The choice of these soft skills was based on improving communication with teams during the interview process regarding leadership and motivation.

Each of these six soft skills has been investigated from the perspective of the PO of each team, focusing on both individual and team levels. To gather insights on these six soft skills, we have created the following questions in which each PO had to answer *"How would you classify"*:

1. *"**Your communication** with your team?"*
2. *"**Your team's communication** in general?"*
3. *"**Your coordination** with your team?"*
4. *"**Your team's coordination** in general?"*
5. *"**Your cohesion** with your team?"*
6. *"**Your team's cohesion** in general?"*
7. *"**Your teamwork**?"*
8. *"**Your team's teamwork** in general?"*
9. *"**Your leadership** as a product owner?"*
10. *"**Your team's leadership** in general?"*
11. *"**Your ability to motivate** your team?"*
12. *"**Your team's motivation** in general?"*

Each one of these open questions were asked by the researcher in an interview to each participants. The interview we have created has been customized to be focused on the soft skills which the authors believed to be the most relevant in the environment in which the study was conducted.

4 Results

In this section, we present the results of our study. It is important to note that we present substantial evidence we collected in the interviews. However, Table 2 presents a summary of the obtained results providing a single example sentence for each justification provided. In this table, we present the results sorted by frequency, which is the number of occurrences of a given justification for a given soft skill regarding either the participant himself/herself or the team.

To perform a qualitative analysis, we have used responses participants provided to the open questions. From these, we have performed thematic analysis, *"a method for identifying, analyzing, and reporting patterns (themes) within data"* [Cruzes and Dyba, 2011]. This technique consists of reading the responses, identifying segments of text, labeling

them, translating codes into themes, and creating categories by grouping them. For coding, we employed an inductive or grounded theory approach [Cruzes and Dyba, 2011]. This process is known as open coding and can be defined as a classification process that requires detailed examination of the data, which helps to identify initial patterns and concepts [Gray, 2022]. It involves constant comparisons, *e.g.*, each case identified must be compared with previous instances to determine similarities and differences. If the new case does not fit the original meaning, the definition must be modified, or a new code must be created. Thus, codification arises from the data, and analysis occurs without pre-definitions towards relevance or expectations of previously assumed themes. We used descriptive codes to capture the central ideas or concepts in the data [Gray, 2022].

During analysis, we have classified the responses into open codes. Thus, we have applied as many codes as possible to capture all aspects in a single response. Next, we have grouped these into categories, intending to present themes and to facilitate establishing analytical lines. One researcher initially conducted this process, and then two others reviewed the proposed coding and themes. In case of disagreement, the researchers discussed so that consensus was reached.

We present our main insights of the open ended questions obtained from the analysis as follows. In addition, we have divided our analysis into two slices: *Self*, which encompassed data which the participant provided as an answer to a self-evaluation question, and *Team*, which encompassed data which the participant provided as an answer to a group-evaluation question. Next, for each slice, to better present our results, we chose to focus on the top most representative subjects mention by study participants.

4.1 Communication

In terms of communication participants have provided motivation for their answers and covered a wide range of topics.

Self The most reported impacting factor for the participants' answer to our interview was a **perception of improvement opportunity for themselves**. Participants' #2, #4, #5, #6 have provided answers which make this explicit. Participant #2, for example, stated that *"I knew my role was that of a PO, but I have not performed as I should have"*. Improvement since the beginning of the project was made explicit by #4 *"It has improved significantly since we began, but I think there are things to improve still because sometimes we speak and one person is not listening and then he/she asks the same thing so I think we are lacking a bit."*. Participant #5 stated that *"we can always still improve"* and #6 that *"I feel I'm getting closer to a very good process [...]"*.

Obtaining a sufficient amount of detail and being objective when communicating was also significant impacting factor for participants' #1, #3, #8. For example, participant #1, which rated self communication as *amazing*, stated that *"I have gone after everything to try to get as much communication as possible"*. On the other hand, participant #3, which rated his communication as *average*, stated that *"I questioned and then I was told that 'It's going', and I think my duty at that point should be to say 'No, I need a more detailed answer because I need to create a report and I need to understand'."*. Participant #8 said that communication was very good on his team because on his words *"I was able to pass everything that we have objectively"*.

Another important aspect mentioned by participants was with regard to **being comfortable with the people in our team**. Participant's #7 and #9 have mentioned this aspect in their answers. #7, for example, stated that *"I rate this way because we already had a good communication before getting into the challenge"*. Similarly, #9 stated that *"[...] the time spent with the course colleagues, this allows me to be more comfortable, I end up being more resourceful, more safe to communicate [...]"*.

Team The most important factor mentioned by the participants regarding their team's communication in general was a **perception of improvement**. Participant #1 stated that *"[...] One of the main points we have improved upon was communication [...]"*, participant #4 explains rating average as *"[...] I think that because sometimes we do not understand each other, it is improving steadily [...]"*. Participant #6 expressed that perception of improvement by stating *"I think we are 'good', moving towards 'very good'"*.

Another important factor mentioned was the **team's transparency in communication**. Participant #1, for instance, stated that *"we have plenty of communication problems by not being informed about what was happening with the team members [...]"*. Participant #6 stated *"[...] someone is working in that task, a long time has passed, it should have been verbalized that more time has passed than expected [...]"*. Participant #7 mentions 'no-communication' *"[...] In the first sprint, a 'no-communication' occurred about certain things [...]"*.

4.2 Coordination

In terms of coordination, participants have provided motivation for their answers and covered a wide range of topics.

Self The most relevant impacting factor reported by the participants with regards to their personal coordination abilities towards the team was a **general difficulty to lead/manage the team**. This was reported in multiple forms: Not having experience with leading (Participant #9 stated *"[...] I think this lack of experience in a real project also heavily affects with my coordination [...]"*), not being able to manage the team (#1 stated *"[...] There was a failure that I consider a personal failure of not looking at the presentation before and in this case I trusted this activity to certain members and believed it would have been actually done so I did not manage before and there were big problems with it [...]"*), not performing the PO role satisfactorily (#2 stated that *"I knew my role as PO, I knew what should be done but I ended up not performing the role the PO should have [...]"*), not being able to delegate tasks (#3 stated that *"[...] when we began to delegate and stuff, I did not delegate directly, I did not know who was performing and who was not [...]"*).

The second most important element mentioned when reporting coordination was regarding **managing task**

creation, task/step dependencies and task separation (or atomization). Participant #3 stated that *"I think since the beginning when we established the tasks and the USs, we were not able to perform this [...]"*. Participant #6 exemplified this when she stated *"[...] It would have been more a process of even I myself organizing which step is best in each moment [...]"*. Participant #7 stated *"[...] We still have issues with some tasks not being more in parallel among them [...]"*. Participant #9 stated *"[...] There are things I end up forgetting, messing up when creating dependent tasks, tasks that are too long and having a sprint with too many tasks [...]"*. Participant #8 stated that *"I think it was more problems of the foundation of the US, of the acceptance criteria [...]"*.

Team: The most mentioned aspect with regards to coordination in the group was the sense of **general engagement the team possessed**. Participants #2, #4 and #7 provided the following statements when answering respectively: *"[...] I will say it is 'very good' because everybody is engaged with the product delivery. [...]"*, *"[...] I think everybody is engaged but I think that some people need to go a bit further than they are currently [...]"* and *"[...] I think it is a 5 (very good), I think the team does a lot [...] they always try to do as much as they can [...]"*.

4.3 Cohesion

With regards to cohesion participants have provided motivation for their answers and covered some topics.

Self The most relevant provided influencing factor reported by the participants regarding their personal cohesion was a **focus on the goals the team had set both for the software product they wanted to developed and the team members themselves**. Participants #1, #2 ,#3, #5, #6, #8 and #9 provided statements which are tangent to this. Participant #1 stated *"[...] we have always continued focusing even (on) what had to be developed. [....]"*, #2 stated *"[...] So the part of 'everybody engaged', 'everybody focused on that goal' is very good, at least my part is very good and the part of team is also very good [...]"*. #3 said *"[...] We managed to go through the same path. Aiming at the goal we wanted. [...]"*. In addition, #5, #6 mentioned *"[...] I think we can solve all these conflict things perhaps and similar things in a natural way, then, always envisioning the goal, having the product as focus"* and *"I'm very aligned with regards to this and I am always exposing to the team what is our goal, where we want to arrive and what path we trace towards that, so I'm really focused on this, completely 100%"* respectively. Finally #8 and #9 also said that *"I think this is very aligned also with the expectations of each member from the start [...]"* and *"[...] That happened precisely due to thinking of the main objective, which is precisely having a finished app in the (App) store [...]"*.

The second most important factor mentioned as relevant by the participants was the fact that many of them had **aligned the goals and objectives of the team and product from the start**. This was mentioned by participants #1 and #8, in which they say respectively *"I think from the start we defined what it (the product) would be and we continue always to focus even (on) what has to be developed. [...]"* and *"I think this is very aligned also with the expectations each member had from the start, what we wanted to delivery for the Macro (the development time frame established)"*.

Team The most impactful factor reported by the participants in their answers was the **team's eagerness to deliver the proposed software products**. Participants #5, #6 and #9 stated sentences with this intent. #5, for instance, stated *"I think the team works together a lot, we have not deviated our focus in any moment and I think that this is very important."*. Similarly, #6 stated *"[...] I think everybody wants to deliver the best possible, wants to get there [...]"* and #9 said *"[...] it was something I wanted to do independently from the program [...]"*.

4.4 Teamwork

Teamwork-wise participants have provided reasoning for their answers and covered the topics below.

Self The most mentioned influencing factor reported by the participants regarding their individual teamwork was the **level of commitment they had towards the team**. This was reported by #1, #2, #5 and #8. #1 stated *"I think that from these three things, I have always been motivated in all of them."*. #2 reports *"[...] if we promised to deliver today, we will deliver today."*. In addition, #5 reports *"[...] I feel sometimes I get so focused on getting deliveries done that I try to absorb everything to achieve this and to not let the weight rest on the others [...]"*. Finally #8 stated *"[...] In the first month of the Macro (challenge), I was working almost everyday at night but it was because I wanted to, not because there was demand to do this, just to speed up the project."*.

The second most mentioned factor was **balance in team member contribution**. This was mentioned by participants #3, #4 and #5. #3 stated *"[...] I think there was nobody overloaded [...]"*. #4 reported *"As I have mentioned before, there is a balance there [...]"* and #5 stated *"[...] sometimes it is nice to let activities for other people. [...]"*.

Team The most important influencing factor was **being able (or not) to assist other team members in difficult moments**. Participants #3, #4, #6, #8 and #9 have provided evidences that support this. For instance, participant #3 stated *"[...] Some moments we helped each other, the guys helped each other but it could have been more [...]"*. #4 stated *"[...] sometimes there is a problem and and I take responsibility for it [...]"*. #6 stated *"[...] So I see there is still an issue in letting go of what you're doing, which perhaps is not a priority and (instead) try to help the other person [...]"*. #8 reported *"[...] Most of the time, we manage to help each other nicely [...]"*. Finally #9 reported *"[...] The team did not help, did not make the case [...]"*.

The second most mentioned factor influencing the participants' answer regarding their team's teamwork was **aware-**

ness regarding the project (*i.e.* source code bugs, documentation status, *etc.*). This was mentioned by participants #1, #2 and #3. #1 stated *"[...] sometimes people need to speak out more of the things they perceive [...]"*. #2 reported *"[...] Everybody is conscious of the things that are being done, everything is documented [...]"*. Finally #3 stated *"[...] I think it was good because everybody was on the same page. [...]"*.

4.5 Leadership

In terms of leadership, participants have provided many reasons for the impact it has had in their answers.

Self The most mentioned aspect by the participants in their answer regarding self and leadership revolved around a **perception of inexperience and that of a learning experience**. Participants #4, #5, #7 and #9 mentioned this. Their sentences which illustrate this point are, respectively, *"[...] I think there are things I have to learn, things I am in doubt, if it is the role of the PO or the SM [...]"*, *"[...] It is my first time experiencing this so I think I would not be able to say I am at my best. [...]"*, *"I think I still fail in somethings because I'm still learning. [...]"* and *"[...] As a PO, I don't know if I'm performing well in my role as PO, this is the issue [...]"*.

Team The most mentioned influencing aspect in relation to leadership in the teams of the participants was the **processes of assuming tasks and decision-making**. Participants #1, #4, #5, and #7 provide motivation for their answers and mentioned this aspect. #1 stated *"[...] sometimes it takes a lot for someone to say 'let me handle this' [...]"*. #4 reports *"[...] when there is something that went wrong or we need to go to other direction, most of the group takes charge and says 'let's go this way, let's go that way', I think there is this level of leadership."*. #5 reported *"[...] I think the team can see clearly, having a vision of what is happening and taking the lead."* and #7 stated *"[...] The decisions are made collectively, but I think everybody leads together, despite the fact that I am the PO, everybody has the big picture [...]"*.

The second most mentioned aspect influencing the participants rating of the team's leadership was the fact that **leadership duties were being conducted not exclusively by the POs, but by multiple team members**. Participants #3, #6 and #8 mentioned this when answering this question. For example, participant #3 stated *"[...] The Scrum Master ends up arbitrating the deadlines for some rituals [...]"*. Participant #6 reported *"[...] When the subject is something more about the project's scope, it ends up staying with me [...]"* and participant #8 mentioned *"[...] The designer can, when the task is design-specific, to organize and lead us to show what she is doing, her opinion, (she) can organize a group dynamic more or less."*.

4.6 Motivation

Specifically regarding motivation, participants have provided many reasons for the impact it has had in their answers.

Self The most mentioned aspect by the participants when justifying their report on the ability to motivate their peers was a **need to motivate other team members** with some strategies they had implemented. This is mentioned by #1, #2, #3, #4 and #8. #1 stated *"[...] There was a retrospective that I looked like a 'coach' speaking, to motivate the team. [...]"*. #2 reported *"[...] Keeping people engaged, motivated is something I can do just fine."*. #3 said *"[...] This is why I think the PO must have a personal engagement with the product, if not, this (motivating others) becomes difficult [...]"*, #4 mentioned *"[...] I always attempt to motivate when something goes wrong [...]"* and #8 said *"[...] I am the person who is leading where our product goes, I have to give the example as much as I can in that time [...]"*.

The second most mentioned in this regard was a **perception of low motivation and engagement of the team**. This was mentioned by participants #1, #3, #4, and #7. Participants #1 reported *"There was a moment in the first sprint [...] that people began to lose motivation and 'Come on, we still have all these months ahead' so we could not lose motivation at that time."*. Similarly, #3 stated *"[...] I was not engaged with the product, it wasn't a product I valued. [...]"*. Further, #4 stated *"[...] When I am not motivated, I can't motivate others, so in this matter, I think I could improve."*. Finally, #7 stated *"[...] sometimes it is difficult to stay motivated because we are not only working in this program, we have a life outside, in college [...]"*.

Team The most mentioned justification for the rating provided by the participants when answering about team motivation was a **perception of an elevated team motivation**. This was reported by participants #2, #3, #6, #7, #8 and #9. #2 demonstrated this by stating *"[...] as everybody is engaged with the product and wants to deliver it well developed, everybody kind of ends up self motivating. [...]"*. #3 said *"[...] (motivation-wise) I am doing not as 'high', not 'loose', but it is average, roughly"*. #6 reported *"[...] There are days of motivation, where we understand everything is going to work and it ends up not working but I see that even then it isn't something that you get into a dark tunnel and never leaves. I see that the entire team is motivated to deliver the best possible [...]"*. #7 said *"[...] I think that despite of everything I said (that) we have our moments of sadness, get a little bad, we always have this engagement [...]"*. Finally #8 and #9 reported, respectively, *"[...] I think everybody is working a lot and is motivated with our product, even too much, there were two weeks in which we did not even have breaks from work for example [...]"* and *"[...] People wanting to work, 'Let's go, let's make the app, make the website', we put the website online, we bought the domain. We are investing our own money, we bought the domain."*.

The second most mentioned justification for rating provided by the participants was a **perception of low motivation moments in the team**. This was reported by participants #1, #3, #4, #5 and #7. Beginning with #1, who said *"[...] It is mostly because there was a moment in sprint 1 in which it (motivation) felt, but it is fine."*. Next #3 reported *"[...] Because there were moments in which we were able to be more motivated, including me. [...]"*. In addition, #4 reported *"We*

lack a few motivation, I think there are peaks of motivation. [...]". #5 stated *"I think there are moments that we end up like 'Darn, that didn't work' and then we have to an even larger effort to become motivated again [...]"* and #7 reported *"[...] So if someone is in a bad vibe, another one comes and 'No, let's go, I'll take you by the hand and we will make it' [...]"*.

5 Discussion

After gathering our results, we have proceeded to conduct our analysis over the prism of our research question. To better frame our analysis, we have used the soft skills as anchors for our answer. In this sense, we present this discussion for our **RQ: What are the impacts to software development student teams of assigning a leadership role (PO) to a team member at the individual level and at the team level?**

1. *Communication* - Communication's reports mostly revolved around the participants perceiving that the POs and their teams were in a learning experience and had room for improvement and to learn. In addition, team transparency, being comfortable around team members and communicating effectively were relevant for the POs when rating communication towards their team and inside their team.
   Our results suggest that assigning leadership roles enhances both the leader's and the team's perception of communication as a learning process, improving transparency and effectiveness.
2. *Coordination* - Coordination's reports had three axis around which they revolved: (i) project management, (ii) difficulty in leading and (iii) a positive team engagement. These results are interesting as they present a mixture of difficulty in performing leadership tasks, which is coherent with a first time experience, a difficulty in leading, again coherent given that most of them had no previous experience with leading, and a positively engaged team in general, an aspect that could interpreted as being multifaceted, as for example they could be identifying with the idea, being intellectually stimulated, identifying with the group, *etc.*
   From this, we can provide indications that assigning a leadership role to a team member in a software development team with regards to coordination tends to bring the relevance of project management skills and tasks to the students, alongside an improved understanding of the the challenges of leading a software development team. Further, it brings to team the importance of team engagement (or motivation).
3. *Cohesion* - Cohesion's rating reasoning for the participants could be summarized in a general focus on delivering the software product and its features but remaining aligned with personal goals.
   Findings such as this present different analysis prism, committing to the development of the group-defined software features and remaining aligned with personal goals are sometimes contradictory actions and this tension was relevant for participants and brought by them in their answers.
   In this sense, our results provided indications that assigning a leadership role to a team member in a software development team with regards to cohesion tends to make leaders have to deal with the need to align the team's objectives for the software product (relating to schedule, feature development, performance, *etc.*) and personal goals that students could have (*e.g.*, exploring a specific technology or skill, improving development skills, creating a personal portfolio, *etc.*).
4. *Teamwork* - Teamwork rating reasoning provided by participants in their answer did not converge to more encompassing terms than the ones presented in the results. This indicates that teamwork reasoning for participants had many aspects for consideration such as mutual assistance, commitment, personal contribution and project awareness.
   This results provide indications that assigning a leadership role to a team member in a software development team concerning teamwork tends to impact many aspects and enhance the awareness of the team and the leader towards these aspects. Namely mutual assistance, commitment, personal contribution and project awareness.
5. *Leadership* - In terms of leadership, participants most cited justification revolved around a sense of a learning experience and somewhat sharing this experience with team members.
   These results indicate that participants were mostly aware that they were learning and thus felt in a safe space to do this, understanding that they would perform as effectively, and also to share leadership tasks with teammates.
   Thus, our study has presented indications that assigning a leadership role to a team member in a software development team concerning leadership tends to be mostly positive as leaders felt safe to fail and share duties with other team mates and view the experience as a learning opportunity.
6. *Motivation* - Regarding motivation, although the most individually cited justification was a perception of high motivation (engagement) of the team, a perception of low motivation and the need to motivate others was also reported.
   This result is interesting as it makes it clear that among the tasks that the participants felt they had to do, was to improve team engagement with the project and to ensure this sense is maintained throughout the project.
   In this sense, our study has presented indications that assigning a leadership role to a team member in a software development team concerning motivation tends to bring to awareness of the leader of importance of motivation and the need to act to either maintain it or raise it.

Overall, considering all answers from the soft skills and looking from the prism of leadership, our results indicated that assigning a leadership role to a team member in a software development team tends to be an overall valuable learning experience for new developers. Furthermore, apart from being challenged to be a leader, they also become aware of the needs that leading a team can have, such as balancing personal goals with project objectives, communicating effec-

Table 2. Summary of the answers to the interview (sorted by frequency)

Soft Skill	Perspective	Justification	Freq.	Example Quote
Cohesion	Self	Focus on the goals the team had set both for the software product and the team members themselves	7	*[...] we have always continued focusing even (on) what had to be developed. [....]*
Motivation	Team	Perception of an elevated team motivation	6	*[...] as everybody is engaged with the product and wants to deliver it well developed, everybody kind of ends up self motivating. [...]*
Coordination	Self	Managing task creation, task/step dependencies and task separation (or atomization)	5	*I think since the beginning when we established the tasks and the USs, we were not able to perform this [...]*
Motivation	Self	Need to motivate other team members	5	*[...] There was a retrospective that I looked like a coach speaking, to motivate the team. [...]*
Teamwork	Team	Being able (or not) to assist other team members in moments of necessity	5	*[...] Some moments we helped each other, the guys helped each other but it could have been more [...]*
Motivation	Team	Perception of low motivation moments in the team	5	*[...] It is mostly because there was a moment in sprint 1 in which it (motivation) felt, but it is fine.*
Communication	Self	Perception of improvement opportunity for themselves	4	*I knew my role was that of a PO, but I have not performed as I should have*
Coordination	Self	General difficulty to lead/manage the team	4	*[...] I think this lack of experience in a real project also heavily affects with my coordination [...]*
Teamwork	Self	Level of commitment they had towards the team	4	*I think that from these three things, I have always been motivated in all of them.*
Leadership	Self	Perception of inexperience and that of a learning experience	4	*[...] I think there are things I have to learn, things I am in doubt, if it is the role of the PO or the SM [...]*
Leadership	Team	Processes of assuming tasks and decision-making	4	*[...] sometimes it takes a lot for someone to say 'let me handle this' [...]*
Motivation	Self	Perception of low motivation and engagement of the team	4	*There was a moment in the first sprint [...] that people began to lose motivation and 'Come on, we still have all these months ahead' so we could not lose motivation at that time.*
Communication	Self	Obtaining a sufficient amount of detail and being objective when communicating	3	*I have gone after everything to try to get as much communication as possible.*
Communication	Team	Perception of improvement	3	*[...] I think that because sometimes we do not understand each other, it is improving steadily [...]*
Communication	Team	Team's transparency in communication	3	*We have plenty of communication problems by not being informed about what was happening with the team members [...]*
Coordination	Team	General engagement the team possessed	3	*[...] I will say it is 'very good' because everybody is engaged with the product delivery. [...]*
Cohesion	Team	Team's eagerness to deliver the proposed software products	3	*I think the team works together a lot, we have not deviated our focus in any moment and I think that is very important*
Teamwork	Self	Team member's contribution balance	3	*[...] I think there was nobody overloaded [...]*
Teamwork	Team	Awareness regarding the project (i.e. source code bugs, documentation status, *etc.*)	3	*[...] sometimes people need to speak out more of the things they perceive [...]*
Leadership	Team	Leadership duties were being conducted not exclusively by the POs, but by multiple team members	3	*[...] The SM ends up arbitrating the deadlines of some rituals [...]*
Communication	Self	Being comfortable with the people in the team	2	*I rate this way because we already had a good communication before starting the challenge*
Cohesion	Self	Aligned the goals and objectives of the team and product from the start	2	*I think from the start we defined what it (the product) would be and we continue always to focus even (on) what has to be developed. [...]*

tively and sharing leadership. These results are interesting as they provide indications that assigning a leadership role can be an enriching experience for new software developers and provides a manner in which leadership could be developed.

6 Limitations

Even though the researchers actively engaged to reduce biases and external influences, we discussed the construct, internal, external and reliability validity based on Runeson and Höst [2009].

Regarding construct validity, the presentation and explanation of the questions to the participants could have been misinterpreted by study participants. We minimize this by conducting all interviews in-person and in which the researcher is open for clarification and further explanations as questions are presented to study participants.

In terms of internal validity, there is a possibility that researchers might have been influenced to find leadership concepts in the analyzed interview transcripts. To mitigate this aspect, initially one of the researchers conducted the analysis steps separately and independently. After, results were obtained, an alignment meeting was conducted where doubts were discussed and thus reached consensus.

External validity-wise, our results reflect a context where real-world mobile apps are developed and where majority of the teams are composed of students. In this sense, our results are not directly generalizable to real software development teams. Furthermore, since our interviews were conducted only with the students who were assigned the PO role, our results might not be generalizable for the team. However, considering contexts which are similar, such as learning environment that use real-world settings to train software developers (with agile roles, development iterations, *etc.*), then results are somewhat generalizable.

Finally, reliability in the study was addressed by discussing and reaching consensus towards the steps and analysis to be conducted. Every analysis was performed independently and separately by one researcher. After that, an alignment meeting was held to ensure all findings and conclusions are agreed upon by the researchers.

7 Conclusion

This paper focused on the exploration of leadership teaching within a Challenge-Based Learning (CBL) environment. It specifically examined the impacts of assigning leadership roles to team members in agile software development teams. The research aimed to understand how such roles impact individual and team development. The study used six soft skills as guides for the investigation: (i) Communication, (ii) Coordination, (iii) Cohesion, (iv) Teamwork, (v) Leadership, and (vi) Motivation.

The results of the study revealed key indications. First, assigning leadership roles, such as the Product Owner in Scrum teams, was found to be a significant learning experience for new developers. This experience provided POs with an opportunity to improve their abilities to balance personal goals with project objectives and fostered essential skills such as effective communication and leadership sharing. These roles also brought to light the challenges and opportunities in team coordination, emphasizing the importance of project management skills and team engagement. Furthermore, the study highlighted how leadership roles impacted in the cohesion within teams, aligning individual goals with broader project objectives, and enhancing overall teamwork. The study also presented indications that providing a leadership learning opportunity for students is positive for individual growth of team members.

In terms of future work, there is potential to explore the long-term impacts of such leadership roles on career development and professional growth in software development. Future studies could also examine the scalability of these findings in larger teams or different agile methodologies and using more sophisticated qualitative analysis such as thematic analysis and inter-coder agreement techniques. Additionally, investigating the impact of diverse team compositions and different educational backgrounds on the effectiveness of leadership roles in CBL environments would provide more comprehensive insights. Lastly, further research could focus on developing more refined and targeted educational strategies to integrate leadership skills training more effectively into software development curricula.

Declarations

Authors' Contributions

All authors have contributed equally to this research.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

We acknowledge the supported by the Ministry of Science, Technology, and Innovations from Brazil.

Funding

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

Availability of data and materials

Data for this research can not be published due to restricts of the R&D environment in which the research took place.

References

Athukorala, C., Perera, I., and Meedeniya, D. (2016). The impact of transformational and transactional leadership styles on knowledge creation in sri lankan software industry. In *2016 Moratuwa Engineering Research Conference*

(MERCon), pages 309–314. IEEE. DOI: 10.1109/MERCon.2016.7480159.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. (2008). Motivation in software engineering: A systematic literature review. *Information and software technology*, 50(9-10):860–878. DOI: 10.1016/j.infsof.2007.09.004.

Chanin, R., Sales, A., Pompermaier, L. B., and Prikladnicki, R. (2018a). Challenge based startup learning: a framework to teach software startup. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE 2018, Larnaca, Cyprus, July 02-04, 2018*, pages 266–271. DOI: 10.1145/3197091.3197122.

Chanin, R., Sales, A., Santos, A. R., Pompermaier, L., and Prikladnicki, R. (2018b). A collaborative approach to teaching software startups: Findings from a study using challenge based learning. In *Proc. of the 11th Inter. Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '18, pages 9–12, New York, NY, USA. ACM. DOI: 10.1145/3195836.3195844.

Cruzes, D. S. and Dyba, T. (2011). Recommended Steps for Thematic Synthesis in Software Engineering. In *2011 International Symposium on Empirical Software Engineering and Measurement*, number 7491, pages 275–284, Alberta, CA. IEEE. DOI: 10.1109/ESEM.2011.36.

da Silva, F. Q., Monteiro, C. V., dos Santos, I. E., and Capretz, L. F. (2016). How software development group leaders influence team members' innovative behavior. *IEEE Software*, 33(5):106–109. DOI: 10.1109/MS.2016.120.

Faraj, S. and Sambamurthy, V. (2006). Leadership of information systems development projects. *IEEE Transactions on engineering management*, 53(2):238–249. DOI: 10.1109/TEM.2006.872245.

Fowler, M., Highsmith, J., *et al*. (2001). The agile manifesto. *Software development*, 9(8):28–35. Available at: `https://agilemanifesto.org/`.

Furumo, K., de Pillis, E., and Buxton, M. (2012). The impact of leadership on participation and trust in virtual teams. In *Proc. of the 50th annual conference on Computers and People Research*, pages 123–126. DOI: 10.1145/2214091.2214125.

Gallagher, S. E. and Savage, T. (2020). Challenge-based learning in higher education: an exploratory literature review. *Teaching in Higher Education*, pages 1–23. DOI: 10.1080/13562517.2020.1863354.

Goyal, D., Cortinovis, R., and Capretz, L. F. (2022). A framework for class activities to cultivate responsible leadership in software engineering students. In *Proc. of the 15th International Conference on Cooperative and Human Aspects of Software Engineering*, pages 96–101. DOI: 10.1145/3528579.3529167.

Gray, D. E. (2022). *Doing Research in the Real World*. SAGE Publications, London, UK, 5th edition. Book.

Hidayati, A., Budiardjo, E. K., and Purwandari, B. (2020). Hard and soft skills for scrum global software development teams. In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, pages 110–114. DOI: 10.1145/3378936.3378966.

Hoegl, M. and Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization science*, 12(4):435–449. DOI: 10.1287/orsc.12.4.435.10635.

Kitchenham, B., Pickard, L., and Pfleeger, S. L. (1995). Case studies for method and tool evaluation. *IEEE software*, 12(4):52–62. DOI: 10.1109/52.391832.

Li, P. L., Ko, A. J., and Zhu, J. (2015). What makes a great software engineer? In *Proceedings of the 37th International Conference on Software Engineering-Volume 1*, pages 700–710. IEEE Press. DOI: 10.1109/ICSE.2015.335.

Marquez, J. M. D., Aguja, S. E., and Prudente, M. S. (2022). Evaluation of student leadership development amidst online distance learning set-up. In *Proc. of the 2022 13th International Conference on E-Education, E-Business, E-Management, and E-Learning*, pages 118–123. DOI: 10.1145/3514262.3514302.

Matturro, G., Raschetti, F., and Fontán, C. (2015). Soft skills in software development teams: A survey of the points of view of team leaders and team members. In *2015 IEEE/ACM 8th Int. Workshop on Cooperative and Human Aspects of Software Engineering*, pages 101–104. DOI: 10.1109/CHASE.2015.30.

Nichols, M. and Cator, K. (2008). Challenge Based Learning White Paper. Technical report, Apple Inc., Cupertino, CA, USA. Available at: `https://www.challengebasedlearning.org/wp-content/uploads/2019/03/CBL_Paper_2008.pdf`.

Nichols, M., Cator, K., and Torres, M. (2016). *Challenge Based Learning Guide*. Digital Promise, Redwood City, CA, USA. Available at: `https://www.challengebasedlearning.org/wp-content/uploads/2019/02/CBL_Guide2016.pdf`.

Przybilla, L., Präg, A., Wiesche, M., and Krcmar, H. (2020). A conceptual model of antecedents of emergent leadership in agile teams. In *Proc. of the 2020 on Computers and People Research Conf.*, pages 164–165. DOI: 10.1145/3378539.3393844.

Przybilla, L., Wiesche, M., and Krcmar, H. (2019). Emergent leadership in agile teams–an initial exploration. In *Proceedings of the 2019 on Computers and People Research Conference*, pages 176–179. DOI: 10.1145/3322385.3322423.

Runeson, P. and Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131. DOI: 10.1007/s10664-008-9102-8.

Ryan, R. M. and Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67. DOI: 10.1006/ceps.1999.1020.

Salas, E., Shuffler, M. L., Thayer, A. L., Bedwell, W. L., and Lazzara, E. H. (2015). Understanding and improving teamwork in organizations: A scientifically based practical guide. *Human resource management*, 54(4):599–622. DOI: 10.1002/hrm.21628.

Sangwan, R. S. and Ros, J. (2008). Architecture leadership and management in globally distributed software development. In *Proceedings of the first international workshop on Leadership and management in software architecture*, pages 17–22. DOI: 10.1145/1373307.1373313.

Santos, A., Sales, A., Fernandes, P., and Nichols, M. (2015). Combining Challenge-Based Learning and Scrum Framework for Mobile Application Development. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE'15)*, pages 189–194, Vilnius, Lithuania. DOI: 10.1145/2729094.2742602.

Schwaber, K. and Sutherland, J. (2011). The scrum guide. *Scrum Alliance*, 21. Available at: `https://www.scrum.org/resources/scrum-guide`.

Silva, T. S. d., Silveira, M. S., and Maurer, F. (2013). Ten lessons learned from integrating interaction design and agile development. In *Agile Conference*, pages 42–49, Nashville, TN, USA. DOI: 10.1109/AGILE.2013.11.

Van Kelle, E., Visser, J., Plaat, A., and van der Wijst, P. (2015). An empirical study into social success factors for agile software development. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 77–80. IEEE. DOI: 10.1109/CHASE.2015.24.

Wellington, C. A., Briggs, T., and Girard, C. D. (2005). Examining team cohesion as an effect of software engineering methodology. In *Proceedings of the 2005 workshop on Human and social factors of software engineering*, pages 1–5. DOI: 10.1145/1083106.1083122.

Woodward, C. J., Cain, A., Pace, S., Jones, A., and Kupper, J. F. (2013). Helping students track learning progress using burn down charts. In *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 104–109. IEEE. DOI: 10.1109/TALE.2013.6654409.

Yang, Z., Zhou, Y., Chung, J. W., Tang, Q., Jiang, L., and Wong, T. K. (2018). Challenge based learning nurtures creative thinking: An evaluative study. *Nurse education today*, 71:40–47. DOI: 10.1016/j.nedt.2018.09.004.