



Building soft skills through a role-play based approach for Requirements Engineering remote education

Gretchen Torres de Macedo   [Federal University of Amazonas, Amazonas Public Prosecution Office | gtmacedo@icomp.ufam.edu.br]

Awdren Fontão  [Federal University of Mato Grosso do Sul | awdren.fontao@ufms.br]

Bruno Gadelha  [Federal University of Amazonas | bruno@icomp.ufam.edu.br]

 Institute of Computing, Universidade Federal do Amazonas, Av. Rodrigo Otávio, nº 6200, Coroado I, Manaus, AM, 69080-900, Brazil.

Received: 20 December 2022 • Accepted: 12 January 2024 • Published: 07 March 2024

Abstract Teaching Requirements Engineering requires adopting pedagogical techniques to develop students' technical skills for identifying users' needs and designing software solutions. Additionally, since requirements engineering involves group work, students must cultivate social skills such as communication, empathy, and conflict resolution. In remote learning scenarios, developing these skills becomes more challenging due to limited interaction. To address these needs, this paper proposes adapting a project-based collaborative learning approach for remote education that combines Role-Play and Send-a-Problem learning techniques. In this approach, students collaborate on software projects in teams, assuming two roles: customers and software developers. We evaluated this approach during two cycles of Action Research, conducted remotely in two disciplines of a Software Engineering undergraduate course involving advanced and beginner students, respectively. In the advanced students' class, we observed that the methodology enhanced communication skills, analytical reasoning, conflict resolution, and empathy. To validate these results, we conducted a new study with beginning students, achieving positive outcomes despite the need for more support in team communication. As our primary contribution, we provide guidelines for implementing this collaborative learning approach online.

Keywords: Requirements Engineering Education, Remote Learning, Collaborative Learning, Role-Play, Send-a-Problem

1 Introduction

Teaching Requirements Engineering aims to enhance students' technical and social skills, enabling them to comprehend real-world problems and formulate precise software solutions for these issues [Memon *et al.*, 2010; Ouhbi *et al.*, 2015]. In terms of technical skills, students must know requirements elicitation techniques, enabling them to choose the most suitable models and tools for each situation. Additionally, they should utilize social skills to analyze a problem and its context from the customer's perspective, understanding their needs and daily activities to propose an effective solution. Soft skills such as analytical reasoning, empathy, communication, conflict resolution, moderation, self-confidence, and persuasion are highly valued for requirements engineers [Pohl, 2010]. Consequently, Requirements Engineering education necessitates the adoption of methodologies and pedagogical strategies that promote the development of both technical and soft skills in students.

In scenarios of social isolation, like the COVID-19 pandemic, students encounter an additional educational challenge as they must carry out all their academic tasks remotely. This situation particularly impacts group activities, including identifying problems, understanding user needs, outlining software requirements, and specifying functionalities. This context accentuates the importance for students to acquire skills in collaborative remote work, which is also crucial

to preparing for a professional career in the software industry, where the number of remote job positions is on the rise [Miller *et al.*, 2021].

Hence, teaching Requirement Engineering remotely poses a challenge in fostering the development of students' technical and social skills. This requires a combination of learning approaches that support the practice of soft skills and the use of mediation technologies to apply those methodologies in remote scenarios.

Existing technologies, such as the Learning Management Systems (LMS) Moodle¹, communication tools like Telegram² or Slack³, and project management tools such as Trello⁴ can support software design during learning activities in both face-to-face and remote learning scenarios.

Regarding learning approaches supporting soft skills development, it is essential to connect the curriculum with practical activities addressing real-life problems, promoting communication, teamwork, and collaboration [Guerra-Báez, 2019]. Active Learning approaches [Johnson and Johnson, 2008; Svensson and Regnell, 2017] motivate students to actively build their knowledge, fostering reflection and critical thinking [Prince, 2004] – crucial skills for requirements engineers. As an active learning methodology, collabora-

¹<https://moodle.org>

²<https://web.telegram.org>

³<https://slack.com>

⁴<https://trello.com>

tive learning allows students to learn through interactions and develop soft skills [Matteson *et al.*, 2016] by working in groups and engaging in practical activities to solve real-world problems [Johnson and Johnson, 2008]. Therefore, collaborative learning is a suitable methodology for teaching Requirements Engineering, focusing on soft skills development, especially as requirements elicitation involves tasks that necessitate interaction between customers and developers. Moreover, effective collaboration among designers, developers, and project managers within a development team is crucial throughout the software development process.

Given the need to adjust teaching-learning methodologies for conducting collaborative activities remotely, we have adapted Zowghi and Paryani's collaborative learning approach [Zowghi and Paryani, 2003] to foster the development of requirements engineering skills in social isolation settings. This approach integrates two collaborative learning techniques, Role-Play and Send-a-Problem [Barkley *et al.*, 2014], within a project-based learning activity [Guo *et al.*, 2020] designed for teaching Requirements Engineering remotely. We opted for this collaborative learning approach because it promotes student collaboration, encouraging interaction and active participation in course activities, which is suitable for cultivating students' soft skills in a remote learning environment.

The research method we employed to assess this learning approach is Action Research [Susman and Evered, 1978; Tripp, 2005]. We conducted two research cycles with students from a Software Engineering undergraduate course at the University of Amazonas – UFAM, and these cycles were conducted remotely. The first cycle consisted of a Requirements Engineering and Systems Analysis module. The researchers assessed the adapted approach by observing the students' performance during the role-play activities, evaluating the software artifacts they designed, and analyzing their responses to the questionnaires after each course.

The results of the first cycle indicated a need for more thorough monitoring of the teams' activities. Additionally, we identified a need to isolate factors that might have influenced the students' performance, such as experience in Requirements Engineering and friendship. To address these issues, the researchers conducted a second Action Research cycle in a Fundamentals of Software Engineering module with novice students and included follow-up meetings on team activities. The learning outcomes achieved in the second cycle were similar to those in the first despite the students' lesser experience with requirements elicitation. This demonstrates that students were able to communicate, interact, and solve problems even in a remote context. Based on the results of the two Action Research cycles, we provide guidelines for applying the approach online as our main contribution.

The remainder of the article is structured as follows: Sections 2 and 3 introduce concepts related to the proposed learning approach and related work. Next, Sections 4 and 5 present this work's collaborative learning approach and research method. Then, Section 6 describes the research cycles of this study. Next, Section 7 discusses the study outcomes. Finally, Section 8 presents this work conclusion.

2 Background

Several collaborative learning techniques can be applied to develop soft skills in RE learning. The approach presented in this study uses Project Based Learning (PjBL) and a combination of Role-Play and Send-a-Problem techniques [Barkley *et al.*, 2014]. Project-Based Learning [Kokotsaki *et al.*, 2016] is a collaborative learning approach where students develop soft skills by developing projects to solve simulations of real-world problems experienced by professionals. In this way, the construction of knowledge occurs by elaborating artifacts that compose a product that solves the problem. This scenario promotes critical thinking, creativity, innovation, and social and cognitive skills development [Musa *et al.*, 2012].

To simulate the development of software projects, we applied two collaborative learning techniques: Role-Play and Send-a-Problem. *Role-Play* is a technique in which an instructor assigns participants different roles [Han and Zhang, 2008]. Students must collaborate, developing social and communication skills. By assuming specific roles, each group member also fosters a sense of responsibility, increasing commitment to achieving the group's goals [Strijbos and De Laat, 2010]. The environment setting for applying the Role-Play technique involves idealizing a scenario of action and an initial story, identifying the different roles and perspectives the participants should assume. It is also essential to define the start and end events, a time limit for performing roles and activities, followed by discussions about the principles used to address problems [Barkley *et al.*, 2014].

Finally, the *Send-a-Problem* technique is most useful for activities with problems lacking a single correct answer, a characteristic of Requirements Engineering activities. In this technique, players solve problems and evaluate solutions. Each group receives a problem, attempts to solve it, and then passes the problem and solution to another group. Without viewing the previous group's solution, the next group also solves the problem. After groups have tackled the problem, they analyze, debate, and synthesize a solution [Barkley *et al.*, 2014]. This way, students learn from each other and develop teamwork and problem-solving skills. To prepare the scenario for the Send-a-Problem technique, instructors must determine the number of problems, the presentation format, and the order in which they will be sent [Barkley *et al.*, 2014]. The following section presents related work that used these collaborative learning techniques for teaching Requirements Engineering.

3 Related Work

Collaborative learning techniques, like PjBL, Role-Play, and Send-a-Problem, have been extensively employed in the Requirements Engineering teaching/learning process. The following studies aim to foster group work and communication skills, allowing students to experience Requirements Engineering in practice.

Zowghi and Paryani [2003] present an approach that uses Role-Play to enhance students' Requirements Engineering skills. Their results showed that the activities aided students in developing empathy and gaining multiple perspectives on

the activities involved in requirements elicitation. Al-Ani and Yusop [2004] also reported applying Role-Play with groups of students, where teachers assigned two problems to each group, alternating the roles of customers and developers. The lessons learned indicated that group work contributed to knowledge consolidation, but the results of using Role-Play and peer review techniques were inconclusive.

In his work, Sindre [2005] also employed Role-Play in group work, where pairs of groups defined problems and acted as customers and developers of each other. The results highlighted the benefits of learning from case development. In another study, Portugal *et al.* [2016] introduced a pedagogical strategy in groups to enable students to interact and collect information. Each team played the roles of client, builder, and auditor with two other teams. Teams pairs performed these roles: Client/Builder, Builder/Client, and Builder/Auditor, Auditor/Builder. The results indicated that pedagogical goals related to client and builder roles were largely achieved, while those linked to the auditor role (quality control) were only partially accomplished.

In a Requirements Engineering course, Svensson and Regnell [2017] conducted a study utilizing the Role-Play technique. The study investigated whether students who excelled in the Role-Play activity also performed well on the written exam. The results revealed that students demonstrating strong project development skills through Role-Play attained higher grades on individual written tests than those with lower performance in project development. In another study, Mitri *et al.* [2017] utilized the Role-Play technique to assist in the ideation and planning stages of an information system in a classroom setting. The study was conducted both in-person and through online collaboration tools. The researchers instructed students to simulate a software corporation with three customers, a CEO, and a system analyst, each from a different functional unit. The findings indicated that the students were not entirely comfortable using online collaboration tools despite having prior experience working together on other tasks before the Role-Play activity.

In their work, Maxim *et al.* [2017] also simulated the experience of working in a software industry environment at a game development studio. Additionally, they combined a Role-Play approach with a gamification framework to reward students for correctly applying the process's steps. The results, measured through questionnaires and prizes in the game environment, demonstrated that these tools helped students develop Requirements Engineering skills.

The aforementioned works [Zowghi and Paryani, 2003; Al-Ani and Yusop, 2004; Sindre, 2005; Portugal *et al.*, 2016; Svensson and Regnell, 2017; Mitri *et al.*, 2017; Maxim *et al.*, 2017] indicate that the Role-Play technique is usually generally with better results in developing Requirements Engineering technical and soft skills when combined with strategies reinforcing student engagement, such as using real problems and gamification.

Other studies have integrated PjBL with collaborative learning techniques to simulate software industry environments and foster technical and soft skills development. Nakamura and Tachikawa [2016] introduced a method for teaching Requirements Engineering at universities using the goal-oriented requirements analysis technique KAOS [Van Lam-

sweerde, 2001]. This approach combines Role-Play technique training with an online specialist system that takes on customer and specialist roles. The questionnaires and portfolio analysis results demonstrated that students enhanced their skills using the KAOS method and reported improved requirements elicitation skills.

In another study, Nwokeji *et al.* [2018] investigated how PjBL can affect students' performance in a Requirements Engineering course. In their research, groups of students defined a real-world problem or opportunity for stakeholders willing to contribute to the project. Preliminary results, collected through questionnaires and cross-checked with student assessments, indicated that projects positively influenced students' achievements. They also identified some factors that may affect the results, such as insufficient involvement of stakeholders and team cohesion.

Costain and McKenna [2019] also detailed the results of a study conducted with students who utilized the Role-Play technique in Joint Application Development sessions. They created a use-case diagram for a proposed system. The Role-Play involved active roles (analysts and marketing managers) and support activities (IT and supervisory manager). Open-ended questions in the study's evaluation questionnaire revealed that most students liked the active learning activity and believed they had expanded their knowledge of use cases.

Related work demonstrates that a common challenge in teaching Requirements Engineering is the difficulty of obtaining real-world projects and collaborating with actual stakeholders to conduct practical activities in the classroom. In this context, related work presents active methodologies and, more specifically, the Role-Play technique to simulate these real scenarios, helping students develop soft skills such as teamwork, communication, and conflict resolution. However, these works applied collaborative learning techniques only in face-to-face or hybrid learning contexts. In this research, we adapted a collaborative learning approach to address collaboration challenges in a remote context. We present the details of implementing this approach in a remote environment in Section 6.

4 Collaborative Learning Approach

To enhance students' social skills by simulating professionals' positions in a software development team, we have introduced a collaborative learning approach for teaching Requirements Engineering in a remote learning environment. This approach is built upon the work of Zowghi and Paryani [2003].

In this approach, students work in teams. The initial team takes on the client role, presenting an issue to the second team. The latter team, acting as software developers⁵, crafts a software solution to address the problem. This team then formulates and conveys its problem statement to the subsequent team, and the process repeats. Ultimately, the last team defines and sends a problem statement to the first team, which in turn acts as their developers. Figure 1 illustrates this dynamic.

⁵Hereafter referred to as developers for brevity.

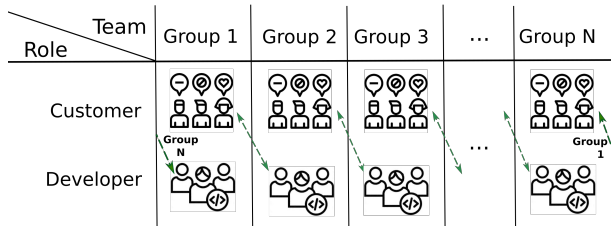


Figure 1. Collaborative learning approach.

In summary, the new approach consists of the following steps:

1. Teachers ask students to organize themselves into groups;
2. Teachers instruct each group to assume the role of a customer from a software development company and propose a problem for the company to solve;
3. The first group sends the problem they created to the second group. Subsequently, the second group takes on the role of a software development team, receiving the problem outlined by the client group. The developer group then elicits requirements to specify a software solution for the client group's problem;
4. The second group, now playing the role of a customer, formulates a problem to send to the third group, and this process repeats successively until the last group receives a problem to solve. The last group then creates a problem to send back to the first group, completing the cycle;
5. Each group develops a requirements specification and user interface prototypes for a software application designed to solve the problem.

Our approach involves crafting a project design encompassing the software application's requirements specification and user interface prototypes. This aids students in envisioning the application even before its full implementation. Additionally, it simplifies the evaluation process for client teams when assessing the products developed by their respective developer teams [Ouhbi *et al.*, 2015]. The following list outlines the primary differences between the original [Zowghi and Paryani, 2003] and the new approaches:

- In our approach, we encourage teams to communicate via videoconference, deviating from the original method where client teams send textual problem specifications to developers;
- Unlike the original approach, our method omits roles within each team and specific activities for each role. The authors from the original approach [Zowghi and Paryani, 2003] reported that constant role-switching confused students. This decision reduces task complexity and enhances students' engagement in building a solution, with the entire team assuming the roles of both customer and developer;
- Our approach does not include instructors' tasks, such as mediating meetings between teams, weekly management, and control of student activities, as stated in the original approach. This significantly reduces instructors' follow-up activities, allowing teams to communi-

cate freely in the manner and frequency they deem necessary;

- Teams have the freedom to manage project execution in any way they prefer. It's noteworthy that project management falls outside the scope of the discipline;
- The original approach recommends that teachers assign problems to the teams to minimize students' time spent specifying problems. In our approach, we follow Sindre [2005] recommendation and let each team create a different problem to simulate a real scenario. Sindre asserts that problem elaboration and customer role performance are most beneficial when performed by students, as they better portray real-world problem characteristics. Otherwise, the teacher would need to exert additional effort to manage the requirements for each assigned problem.

The following section elaborates on the Action Research methodological procedure we followed during the empirical activities.

5 Research Method

The research method adopted in this work is Action Research [Susman and Evered, 1978]. It investigates the effects of an intervention aimed at resolving a previously identified problem within a specific environment under study [Susman and Evered, 1978; Tripp, 2005]. In Action Research, researchers collaborate with participants engaged in practical activities within the investigated environment while gaining insights into the studied setting.

In a study using Action Research, researchers must execute as many intervention cycles as necessary until they successfully address the problem and gain comprehensive knowledge about the environment under investigation [Checkland and Holwell, 1998]. Ultimately, when employing Action Research as a methodological approach, it is essential to predefine the topic of interest and the theoretical framework, serving as a methodological procedure to conduct the study [Checkland and Holwell, 1998].

In the empirical study we conducted to implement the collaborative learning approach presented in Section 4, we adhered to the Action Research canonical process by Davison *et al.* [2004], adapted from Susman and Evered [1978] and Baskerville [1999]. This process comprises five phases carried out in cycles. The phases are:

1. **Diagnosis** – In this phase, researchers should identify a practical problem to seek a solution. In this work, the identified need was for students to interact collaboratively and solve real problems, albeit remotely. This approach aimed to help them develop the necessary skills in Requirements Engineering, encompassing both technical and social aspects;
2. **Planning** – During the planning phase, researchers define action strategies for conducting Action Research. In this work, we developed a methodological approach that integrates collaborative learning techniques, specifically Role-Play and Send-a-Problem, within a project-based learning activity;

3. Intervention – This phase involves executing the activities planned in the action strategy in collaboration with the study participants. In this work, researchers applied the proposed approach when teaching Requirements Engineering to students in a Software Engineering undergraduate course;
4. Evaluation – In this phase, researchers collect data to assess the possible effects of the intervention [Susman and Evered, 1978]. The researchers gathered data through questionnaires, artifacts evaluation, and interviews to assess the results of this study;
5. Reflection – In this phase, researchers reflect on the lessons learned from the research cycles [Davison *et al.*, 2004]. It involves providing solutions to the client's problem and producing information for the scientific community. In this study, the Reflection phase occurred after the qualitative analysis of the data from the two cycles of the empirical research.

The following section describes how the researchers applied the proposed collaborative learning approach using Action Research to teach Requirements Engineering remotely.

6 The Approach in Practice

The researchers implemented the proposed approach to teach Requirements Engineering in the Requirements Engineering and Systems Analysis subject during the academic recess due to the COVID-19 pandemic. The analysis of the results revealed that PjBL, associated with Role-Play and Send-a-Problem techniques, facilitated soft skills development. In addition, they also identified opportunities for improvement in applying the approach. For those reasons, the researchers performed a second study cycle in the Fundamentals of Software Engineering discipline. The following sections describe the two cycles from the empirical study in detail⁶.

6.1 First Cycle

In the first cycle of the empirical study, the researchers implemented the proposed approach to assess whether the modifications to the original method were sufficient to promote the development of soft skills while learning Requirements Engineering. The cycle comprises five steps [Davison *et al.*, 2004], outlined below.

6.1.1 Diagnosis

The diagnostic phase identified the need to cultivate technical and social skills in Requirements Engineering in a remote setting. This realization occurred during the university recess due to the COVID-19 pandemic, when the researchers recognized the need to offer activities to maintain students' motivation and engagement in the Software Engineering undergraduate course.

6.1.2 Planning

The planning phase involved defining mediation tools, organizing the content of the course, and specifying learning assessment instruments. The researchers chose Moodle as a mediation tool to facilitate interactions between instructors and students during the course. They also made available the teaching plan, the schedule for synchronous and asynchronous activities, the course's didactic material, and the link for access to Google Meet⁷ for synchronous meetings on the Learning Management System (LMS). To evaluate the study results, the researchers created online questionnaires using the Google Forms tool⁸.

The discipline was structured into two parts, theoretical and practical, lasting 30 hours. The theoretical part comprised pre-recorded video classes accessible through Moodle. This part also included synchronous videoconference meetings for questions and additional explanations about the content. Additionally, the researchers provided exercises, questionnaires, and support material on Moodle to reinforce learning. The practical part of the discipline involved implementing the proposed approach through hands-on project development activities.

The researchers prepared three questionnaires to assess the study results. Questionnaire 1 focused on evaluating the perception of learning and the pedagogical methodology. In addition, it contained questions about the student's profile (age, semester, experience in the subject), the activities performed while playing the customer and developer roles, and other learning aspects. Table 1 presents the Questionnaire 1.

Questionnaire 2, as presented in Table 2, addressed the satisfaction of teams acting as customers regarding the artifacts delivered by the developers. It utilized the Likert scale, with responses ranging from 1 to 5, where 1 corresponds to Disagree, and 5 corresponds to Agree. The researchers instructed students to complete a satisfaction rating questionnaire after disclosing their grades for the activity. This measure ensured that students wouldn't think a negative review could impact their peers' grades and feel compelled to provide positive evaluations for their classmates' projects.

Finally, Questionnaire 3 (refer to Table 3) assessed students' satisfaction with the remote activity. Additionally, this questionnaire enables the identification of problems and opportunities for improvement in subsequent editions offered remotely. The questions also employed the Likert scale, where responses ranged from 1 to 5, with 1 representing Disagree and 5 representing Agree.

6.1.3 Intervention

The University offered a remote subject as an optional extracurricular activity via email to students enrolled in the Software Engineering course. Thirty-five students enrolled. The researchers conducted the study using the proposed approach during requirements elicitation activities. They did not perform a pilot study to avoid causing learning bias in students.

⁶Research data available at <https://figshare.com/s/9070d722f3adc1d29c47>

⁷<https://meet.google.com>

⁸<https://forms.google.com>

Table 1. Questionnaire 1 – Assessment of learning.

Customer role questions	
Q1-01	How did you define the system you ordered for the developers' team? (text)
Q1-02	Regarding the tools used for your team's communication and collaboration when defining your problem, specify how much each was used (widely used, little used, not used): Videoconference; Email; Instant message; Voice call; Video call.
Q1-03	Rate your team's activities while playing the customer's role concerning the following aspects (weak, moderate, satisfactory, very good, excellent): Communication; Teamwork; Conflict resolution.
Developer role questions	
Q1-04	What strategies did your team use to develop the system artifacts requested by your customer (Requirements Specification and Screen Prototypes)? (text)
Q1-05	Regarding the tools that were used for communication and collaboration of your team as developers with the Customer team and during the solution of the problem by your team, define how much each one was used (widely used, little used, not used): Videoconference; Email; Instant message; Voice call; Video call.
Q1-06	Rate the activities performed among the development team members concerning the following aspects (weak, moderate, satisfactory, very good, excellent): Communication; Teamwork; Conflict resolution.
General learning aspects	
Q1-07	I believe the practical activities were relevant to exercise what I saw theoretically about requirements elicitation. (From 1 to 5, where 1 = not very relevant and 5 = very relevant)
Q1-08	Practicing the concepts learned about requirements elicitation using a role exchange methodology (customer/developer) contributed to my learning. (From 1 to 5, where 1 = little and 5 = a lot)
Q1-09	What were the challenges, difficulties, and limitations of doing the activity remotely? (text)
Q1-10	How did you overcome the challenges, difficulties, and limitations? (text)
Q1-11	Rate how you think the skills and abilities below have influenced the achievement of your results (very, little, nothing): Communication; Analytical reasoning; Empathy; Self-confidence; Conflict resolution; Persuasion; Moderation.
Q1-12	Rate how you consider the skills and abilities below were developed in the discipline (very, little, nothing): Communication; Analytical reasoning; Empathy; Self-confidence; Conflict resolution; Persuasion; Moderation.
Q1-13	Have you had a similar experience before? (Yes / No)
Q1-14	Performing the collaborative learning activity entirely virtually has positive aspects compared to the on-site activity. (From 1 to 5, where 1 = strongly disagree and 5 = strongly agree)
Q1-15	If you agree with the previous statement, cite the positive aspects of the virtual activity compared to face-to-face activity. (text)

During the intervention, the researchers organized a dynamic session where students elicited requirements using Design Thinking (DT) techniques. Students were instructed to utilize at least three DT techniques for eliciting software requirements. The deliverable for this activity was a report on the DT techniques adopted in requirements elicitation, the software's general description, the functional and non-functional requirements, and the software screen prototypes.

Instructors instructed students to organize themselves into groups using the Moodle Group Choice tool. The tool had pre-set groups for students to sign up, and the teacher assigned a group to students who didn't join any team. The students formed ten groups, comprising eight groups of three and two groups of two students, totaling 28 students. It was noted that some students left the discipline. It is important to emphasize that issues related to student dropout are outside the scope of this work.

Subsequently, the groups interacted, assuming the roles of customers and developers, and worked on developing the software project to address the given problem. The students had 11 days to complete the activity. Afterward, each group playing the developer role delivered the requirements specification and the application screen prototypes to their respective customer teams.

The researchers distributed the questionnaires to students via Moodle, configuring all questionnaires to anonymize students' responses. Questionnaire 1, which focused on assessing learning perception, was emailed to students after delivering the artifacts. Following this, the teams playing the customer role evaluated the solution developed for their defined problem. Subsequently, researchers asked the students to answer questionnaires 2 and 3. Questionnaire 1 received 26 responses, Questionnaire 2 received 13 replies, and Questionnaire 3 received 28 responses.

6.1.4 Evaluation

The researchers evaluated the results of the first cycle through the analysis of the problems defined by the students, the artifacts elaborated during the design of the projects, and the questionnaires answered by the students at the end of the activity. Of all the students, 26 (84.6%) are in their fifth semester or higher. Regarding professional experience, 65.4% of the students have previously worked in software development.

Concerning the strategies teams adopted to perform customers' and developers' roles, most groups used brainstorming to specify a problem. They initiated by discussing issues that software could solve in their daily lives. Once they had a

Table 2. Questionnaire 2 – Evaluation of teams as Customers regarding the artifacts received by Developer teams (From 1 to 5, where 1=disagree and 5=agree).

Artifacts evaluation by Customers' team	
Q2-01	The requirements elicitation document portrays the needs that I presented as a customer.
Q2-02	The requirements listed in the elicitation document are complete: they comprise everything I would like the requested system to have.
Q2-03	The screen prototypes received correspond to the requirements presented.
Q2-04	The screen prototypes presented comprise ALL the requirements raised and documented.
Q2-05	The prototypes of screens present solutions to the problems that I presented to the development team.
Q2-06	I was satisfied with the solution presented by the development team.
Q2-07	If you answered "Unsatisfied" to any question, please describe the reason for your dissatisfaction here. (Text)

Table 3. Questionnaire 3 – Remote activity evaluation – Requirements Engineering (From 1 to 5, where 1=disagree and 5=agree).

Students' perceptions about the online discipline	
Q3-01	I enjoyed participating in the special extracurricular activity.
Q3-02	I liked the video classes available.
Q3-03	I liked the activities I performed.
Q3-04	I enjoyed the practical group work activity.
Q3-05	I learn better in the classroom than at home.
Q3-06	I prefer watching video classes to reading handouts and textual material.
Q3-07	I am interested in continuing with studies in the proposed format.
Q3-08	Report your experience in this remote discipline. State your expectations, the difficulties encountered, criticisms, and suggestions for improvements. (text)

list of potential issues, they chose one to focus on. The teams also mentioned having at least two video conferencing meetings with their customers to gather requirements for the developers' role.

Cycle 1-P5 – *"We had team meetings and brainstorming using the needs experienced by team members as a starting point."*

Cycle 1-P14 – *"Through conversations with group members, several ideas emerged, but the one used in the activity came after being suggested by a group member, and all members approved it for being quite different."*

We qualitatively analyzed the students' perceptions of their roles as customers and developers in the software engineering project (Q1-03 and Q1-06). We found that the students reported higher levels of communication, teamwork, and conflict resolution skills when they assumed the developer role than when they acted as customers, as shown in Figure 2. This suggests that the developer role provided more opportunities for students to practice and enhance these skills in a collaborative setting.

Students also mentioned the skills developed during the activity (Q1-12). They indicated communication and analytical reasoning as the most developed skills. Also, Empathy stands out, possibly because Design Thinking techniques were also a topic of the remote class. We present these results in Figure 3.

When asked about challenges and difficulties encountered in carrying out the activity online (Q1-09), students pointed out the low quality of the Internet connection, difficulty reconciling schedules, communication with customers and team members, and the use of modeling tools.

The restrictions of carrying out the course remotely encouraged students to seek new forms of communication (Q1-10). They explored management techniques and used tools that support collaborative work, such as Balsamiq⁹ for screen prototyping, Google Forms for gathering customer information, and Google Docs¹⁰ to prepare the final report. Using tools that support collaborative work allowed the shared elaboration of artifacts. Besides, they used project management resources, such as creating task backlogs, setting goals, and assigning activities.

Cycle 1-P7 – *"Communication via WhatsApp, video conference, division of activities to be developed, shared online documents, personas development, stakeholder map, and questionnaire."*

Cycle 1-P9 – *"Balsamiq allowed us to distribute elaboration of screen prototypes among members, as it was possible to access the same project and make changes without conflicts. It was also possible to share the [personas] questionnaire elaboration in Google Forms. (...) As for the [discipline's final] report, Google Docs allowed its creation in a shared way."*

Regarding the role exchange proposed by the approach (Q1-08), 73.1% of students believed they contributed significantly to their learning. When asked about the advantages of performing activities online (Q1-15), most students pointed out time-saving, reviewing recorded classes or meetings, schedule flexibility, and the convenience of carrying out activities at home. In addition, some students pointed out that communication tends to be more simplified, which

⁹<https://balsamiq.com>

¹⁰<https://docs.google.com>



Figure 2. Activities performed by team members - First Cycle.

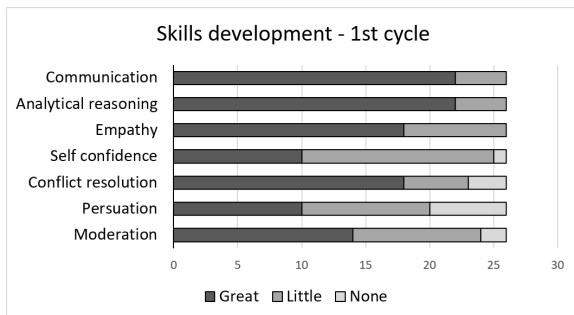


Figure 3. Skills development — First Cycle.

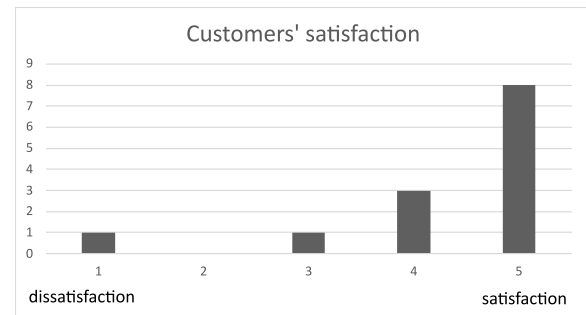


Figure 4. Customers' satisfaction concerning the product received.

has positive points, making meetings more objective. On the other hand, they might overlook discussing details of topics that could impact the project. The following quotes exemplify these findings:

Cycle 1-P13 – *“An advantage of online activity is the more simplified conversation with the team.”*

Cycle 1-P24 – *“A positive point is synthesizing the content of conversations. With virtual means of communication, people try to comment only on the most relevant aspects (...), bringing to light the focus of the problem but also opening up an abyss of uncommented questions for the development of the whole project. Perhaps a virtual activity would be suitable for a first contact, followed by face-to-face meetings with customers.”*

Questionnaire 2 revealed great satisfaction from the client teams concerning the solutions their respective developer teams proposed. Of all questions in Table 2, only one respondent in Q2-07 claimed to have been dissatisfied with the received solution because of low-priority requirements and very simple screen prototypes. Figure 4 shows the evaluation values for this question.

Finally, the instructors evaluated the students' ability to elicit requirements using the proposed approach and found that the teams performed the tasks as expected. In addition, when assessing the problems defined by the students, the researchers found that they had different levels of complexity.

6.1.5 Reflection

The results from the study using the RE learning approach in a remote environment showed that students obtained good results regarding the development of technical and soft skills.

However, the researchers identified opportunities for improvement in the proposed approach.

The customers' satisfaction regarding the software design developers conceived to solve the proposed problem, students' perception of their soft skills improvement, and students' performance assessed by researchers indicate that soft skills were successfully developed despite communication issues. Conducting the learning approach remotely demanded additional efforts to overcome group communication issues. The mitigating actions included using online tools for collaborative work and the individual assignment of tasks.

The researchers identified some factors in applying the collaborative learning approach that may have positively affected the results. Firstly, since most students were already in the fifth semester or higher, they already had some knowledge of Requirements Engineering from previous subjects. Secondly, a significant percentage of students (65.4%) indicated they already have professional experience in systems development. These factors may have facilitated collaborative activities since many students already knew each other.

Regarding the requirements elicitation activities, the researchers let the students conduct them without monitoring because, as fifth-semester students, they were more independent in carrying out assigned tasks. However, at the end of the activity, the researchers verified that follow-up meetings could have identified the differences in the complexity of the proposed problems and interaction problems between the members of each team or between the client and developer teams.

For these reasons, the researchers adapted the approach to better track students' activities. They designed a new study to mitigate factors such as previous experience to enable a more comprehensive approach evaluation.

6.2 Second Cycle

The second cycle of Action Research was conducted to improve the approach based on the observations obtained in the Reflection phase of the first cycle. The following sections detail its steps.

6.2.1 Diagnosis

In the second research cycle, the diagnosed problem remains the same: the need to teach Requirements Engineering with the development of soft skills in remote mode. The aspects of the approach that need improvement are the monitoring of students' activities and the integration between teams while playing the roles of client and developer. Additionally, there is a need to mitigate the influence of students' previous experiences on learning outcomes.

6.2.2 Planning

To enhance the monitoring of students' activities, the approach includes weekly remote meetings with the teams conducted by researchers. This action aims to verify the students' strategies in elaborating the problems they send to other teams and solving the received problems. Furthermore, the monitoring allows verification of each problem's complexity to reduce possible discrepancies in the degree of complexity.

The researchers have added a presentation meeting on teams' software projects to the approach. This meeting is intended for teams in the developer role to present their projects for evaluation by the customers. Finally, the researchers have included an informal interview after the teams' project presentation to understand students' main challenges and difficulties. Figure 5 illustrates the approach activities after the modifications in the second cycle.

To conduct the study with less experienced participants and mitigate the influence of students' prior knowledge on their performance, the second cycle was completed during the Fundamentals of Software Engineering discipline, offered to students in the first semester of the Software Engineering undergraduate course at the Federal University of Amazonas. These students had only one week of face-to-face classes before the social isolation due to the COVID-19 pandemic. Table 4 summarizes the points of the approach that we modified for the execution of the second cycle. The discipline was offered to students online in a Special Extracurricular Activity with 32 registered students. As in the first cycle, the activity had two stages: theoretical, where students learned the fundamentals of Requirements Engineering, and practical, where they developed a project using the approach.

As the discipline content is different, involving the teaching of Use-Case diagrams, Questionnaire 2 has some modifications. Since use-case diagrams are artifacts evaluated by the client, researchers removed questions Q2-01, Q2-03, and Q2-04 and added questions Q2-08 and Q2-09, presented in Table 5. Again, the questions used the Likert scale, with answers ranging from 1 to 5, where 1 corresponds to disagree completely, and 5 corresponds to agree completely.

6.2.3 Intervention

The researchers organized a practical activity for students to elicit requirements. The activity deliverables included a general description of the system, the functional and non-functional requirements, the use-case diagram with an explanation of each use case, and the screen prototypes of the complete system.

Students were divided into ten groups, eight of them consisting of three members and two with four members, totaling 32 students. They had 13 days to complete the activity and deliver the artifacts. After assigning the elicitation activity to students, researchers conducted weekly follow-up interviews with each team, allowing them to present project details and discuss any difficulties.

Following the submission of students' projects, the researchers arranged two videoconferencing meetings with the students. During these meetings, the teams took on the developers' role to present their projects and the customers' role to evaluate them. All project documentation was provided to the teams in advance to facilitate this. After the presentation meetings, researchers conducted an informal interview, posing various questions about students' perceptions of their performance in the activity.

Researchers invited the students to respond to three evaluation questionnaires about the activity, provided through links on Moodle. All questionnaires were configured to anonymize students' responses, with Questionnaire 1 receiving 26 responses, Questionnaire 2 obtaining 16 replies, and Questionnaire 3 receiving 23.

6.2.4 Evaluation

The educational outcomes of the second Action Research cycle were similar to those of the first cycle, even with less experienced students. Concerning participants' professional experience, 96.2% of students who responded to the survey claimed to be in their first college semester, reducing the likelihood of students already knowing each other. Additionally, the experience level in Requirements Engineering and Systems Development decreased to 26.9% and 30.8%, respectively.

The follow-up meetings enabled the assessment of the proposed problems' complexity and assisted in resolving communication issues between teams. While the proposed systems' complexity levels were similar and suitable for the proposed activities, the researchers found that six out of ten teams did not initially meet virtually with their customers. Instead, customers sent developers a text specification of the problem. Additionally, one team encountered difficulties with aspects of the requested system outside the application's scope. In this case, the instructor helped them delimit the problem's scope.

The teams' most adopted interaction strategies (Q1-01 and Q1-04) were brainstorming, mind maps, task division, user interface prototyping, and remote elicitation and validation meetings. According to their perceptions (Q1-03 and Q1-06), the activities where students exercised Communication, Conflict Resolution, and Teamwork the most were in the developer role rather than the customer role, as shown in Figure

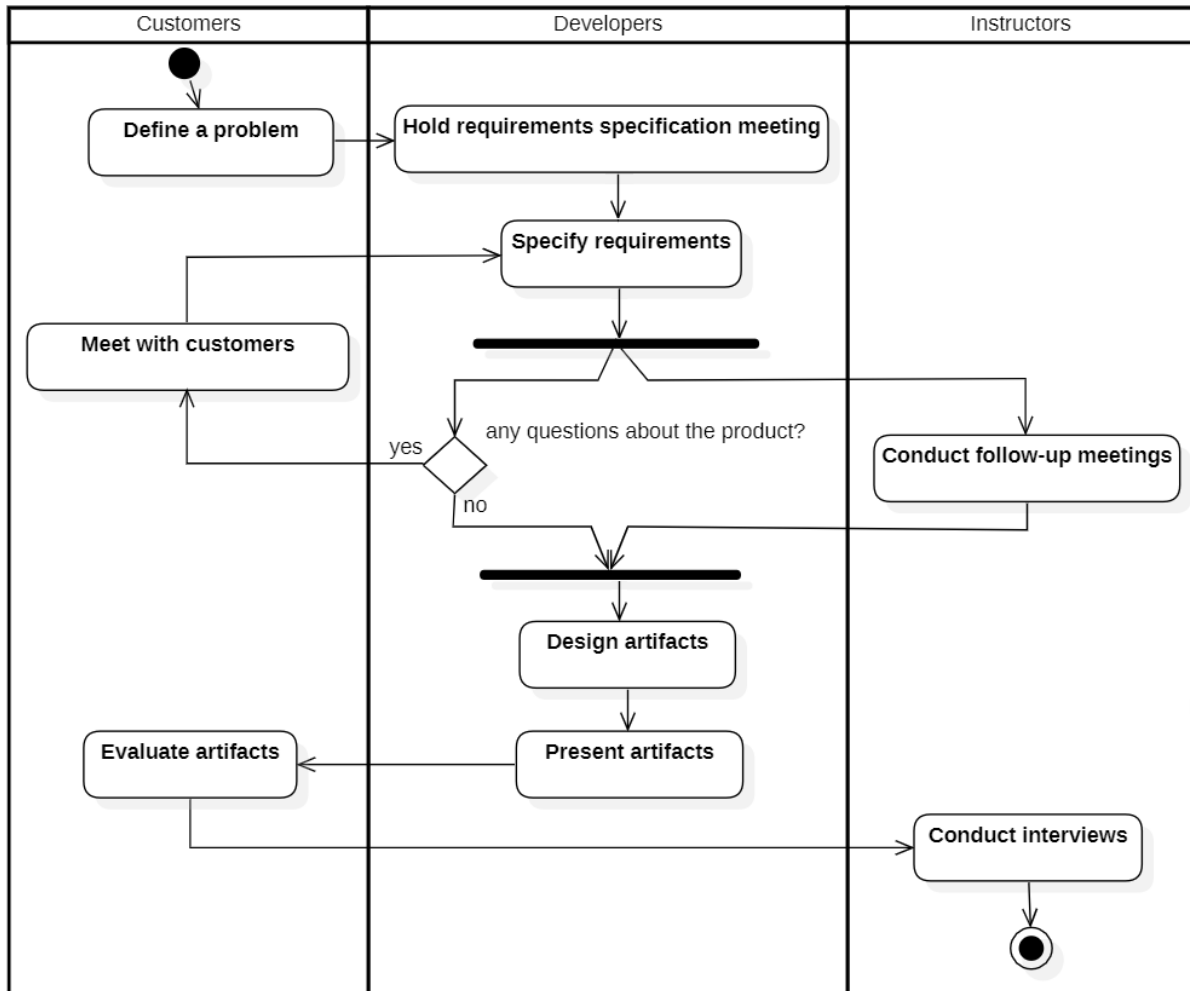


Figure 5. Collaborative learning approach activities.

Table 4. Changes in the learning approach.

First Cycle	Second Cycle
No monitoring of team activities	Meetings for monitoring team activities
Artifacts sent to the client team	Artifacts presented in a virtual meeting
Students who already knew each other	Students who did not know each other
Experienced students	Students without experience

6.

Concerning the competencies students developed during the discipline (Q1-11 and Q1-12), Communication and Analytical Reasoning were again the most indicated (Figure 7). Conflict Resolution also stands out, which can be explained by the students’ increased effort to get to know each other, understand how each team member works, and align ideas.

Regarding the challenges and difficulties encountered in carrying out the activity online (Q1-09), besides well-known problems, researchers noticed a significant difficulty in communication between teams when performing client and developer roles. The teams overcame such communication difficulties by intensifying instant messaging, reducing the duration of meetings, and meeting with only a few team members (Q1-10). Unlike the first cycle, the teams did not report adopting project management tools.

Cycle 2-P2 – “The most significant difficulty was

the developer and customer communication. It was necessary to double the attention to understand what the customer asked. I believe that meeting in person is much better than a video conference. (...)”

Cycle 2-P7 – “The most significant difficulty was communicating with the team that played the client role.”

During the presentation meetings for their projects, the students were able to experience a process similar to that of real-world developers, receiving immediate feedback from customers and gaining a better understanding of the products they had requested. Regarding customer satisfaction with the product, 25% of the answers to questions Q2-02, Q2-08, and Q2-09 indicated some dissatisfaction, for which we could identify some possible reasons when analyzing answers to question Q2-07.

Table 5. Questionnaire 2 additional questions.

Q2-08	Do the presented use cases meet all the specified requirements?
Q2-09	Do the screen prototypes correspond to all the use cases surveyed and documented?

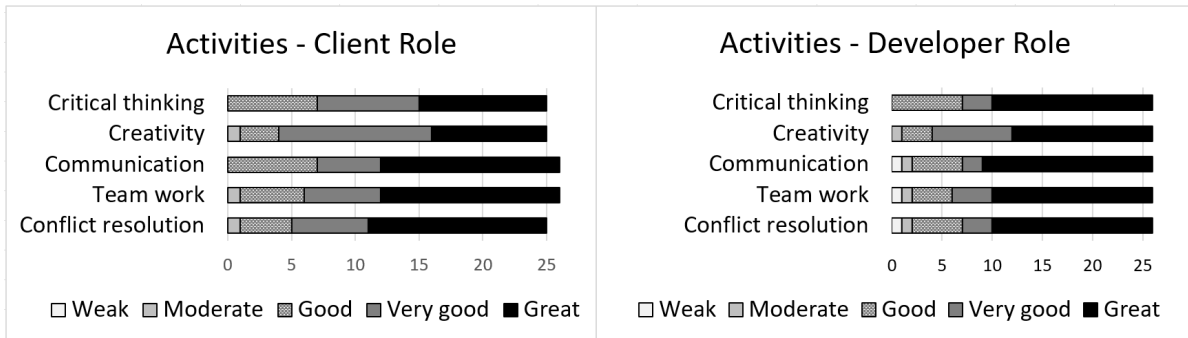


Figure 6. Activities performed by team members - Second Cycle.

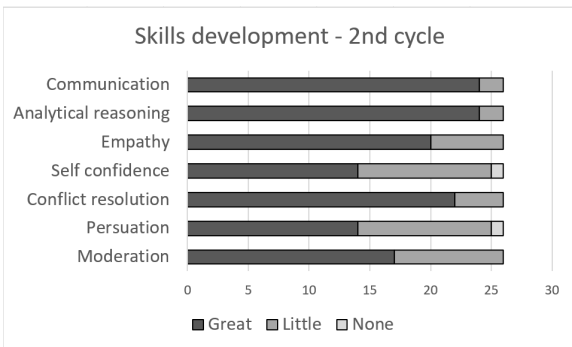


Figure 7. Skills and abilities developed during the activity – Second cycle.

Cycle 2-P4 – “The developers missed the requested colors and the site name.”

Cycle 2-P12 – “Although it was not a very elaborate design, it was functional. So it made me satisfied. However, I feel that it has not fully met my expectations. But they solved the problem.”

Also, students positively evaluated the video lessons and the theoretical and practical activities (Q3-01 to Q3-04, Q3-08):

Cycle 2-P6 – “(...) the suggestion is to record the practical classes. Those involving examples could be synchronous, containing, if possible, two examples (one simple and one complex).”

Cycle 2-P17 – “Activity 2 was very useful. The work was much more dynamic than the first and made the group, which little was known, relate. The work allowed us to assume two different roles, analyzing two views. This experience will bring much knowledge to our professional future.”

The instructors’ assessment of the students’ performance in implementing the practical project found that students accomplished the tasks correctly, exercising communication, teamwork, conflict resolution, critical thinking, and creativity. In addition, students practiced essential skills for requirements engineers, such as empathy and analytical reasoning.

The survey data obtained from questionnaires also revealed that some teams, acting as customers, sent a written and detailed problem specification to the developers’ team, bypassing the role-play performance.

Finally, during the interview conducted by the researchers after the projects’ presentation meeting, the students demonstrated motivation and pride in presenting the developed systems. They were happy to see that their ideas gained a visual representation in prototypes. The presentation also facilitated the project assessment by the teams playing the customer role. In addition, the students talked about their relationships with other colleagues. Many students reported developing good relationships during the roles played within each team, praising colleagues’ proactivity and organization.

6.2.5 Reflection

Following the second cycle, the researchers observed various aspects regarding the modifications implemented in the approach after the first cycle. Firstly, introducing a follow-up meeting a few days after assigning the activity enabled the researchers to identify communication problems between teams and make necessary adjustments without compromising the desired results.

Regarding communication issues, students found a workaround by intensifying the use of instant messages. This mirrors real-world projects where development teams often rely heavily on instant messages, sometimes even replacing email [Alkadhi et al., 2018]. According to Lima et al. [2019], instant messages have become a rich source of information, containing decisions and discussions about project-related issues.

The researchers noted that, unlike the teams in the first cycle, students did not utilize project management resources during project development, even though it was a complex task. This could be attributed to the fact that first-semester students had not yet received instruction on project management. Additionally, the researchers observed more communication problems among the participants compared to the first cycle. These issues occurred within and between groups, in both the customer and developer roles. The lack of personal

relationships due to minimal contact before the suspension of classes might have contributed to this difficulty. This lack of communication may have also led the teams to prefer forwarding textual specifications of the problem to the development teams instead of holding virtual meetings and presenting the problem as actual customers. Students claimed to have done this to free themselves from the role of customers and focus on elaborating the artifacts performed by the developer role.

The slight increase in customer dissatisfaction with developers' projects may have occurred due to communication issues, impacting the correct elicitation of requirements and, consequently, the final product. Another hypothesis is that students felt more comfortable criticizing their colleagues' work as they had not developed closer personal relationships.

7 Results and Discussion

After completing the two Action Research cycles, the researchers compared the results through both statistical and qualitative analysis. The qualitative analysis delved into the impact of the approach modifications on students' learning outcomes. Meanwhile, the statistical analysis compared students' perceptions of developing their soft skills in the first and second cycles, measured by Question Q1-12.

The statistical analysis started with a normality test on data, indicating whether to adopt a parametric or non-parametric statistical test. The Shapiro-Wilk normality test [Shapiro and Francia, 1972] yielded a *p-value* < 0.05 for all the measured skills, as presented in Table 6. This result indicates that the data do not follow a normal distribution, necessitating the adoption of a non-parametric statistical test [Wohlin et al., 2012]. Thus, we performed a Mann-Whitney non-parametric statistical test, and the results are shown in Table 7.

Table 6. Shapiro-Wilk normality test for soft skills outcomes.

		W	p
Communication	First Cycle	0.436	< .001
	Second Cycle	0.301	< .001
Analytical Reasoning	First Cycle	0.436	< .001
	Second Cycle	0.301	< .001
Empathy	First Cycle	0.583	< .001
	Second Cycle	0.524	< .001
Self-confidence	First Cycle	0.724	< .001
	Second Cycle	0.715	< .001
Conflict Resolution	First Cycle	0.633	< .001
	Second Cycle	0.436	< .001
Persuasion	First Cycle	0.800	< .001
	Second Cycle	0.715	< .001
Moderation	First Cycle	0.735	< .001
	Second Cycle	0.604	< .001

The *p-value* for the seven skills in the Mann-Whitney test indicated values greater than 0.05. This implies no significant statistical difference between the first and second cycles regarding students' perception of soft skills development; both results are equally satisfactory.

Table 7. Mann-Whitney test.

	W	p
Communication	312.000	0.200
Analytical Reasoning	312.000	0.200
Empathy	312.000	0.272
Self-confidence	288.000	0.152
Conflict Resolution	280.000	0.076
Persuasion	258.000	0.056
Moderation	290.000	0.156

We can explain the similarity between the two-cycle values by the nature of changes made in the second cycle. On one hand, follow-up meetings with teams, project presentations, and interviews at the end of the course tend to produce better results. On the other hand, changes made in the target audience, using beginner students who did not know each other, aimed to reduce biases that the study with experienced students might have caused in the results obtained in the first cycle. Thus, these two modifications tend to neutralize the differences in the values obtained in the second cycle.

Despite that, analyzing the *p-value* index in Table 7, we observe that the *p-values* most closely approximating significant differences are the Persuasion (0.056) and Conflict Resolution (0.076) skills. We can explain the Persuasion skill because the students had to develop the ability to persuade when working in groups with unfamiliar people. Regarding the Conflict Resolution skill, it can be seen as a natural step in the group formation cycle, as shown by the Tuckman model [Tuckman and Jensen, 1977]. Thus, as groups were formed by people who did not know each other, they could experience different situations to resolve conflicts.

The development of communication skills yielded results similar to those of the first cycle, contrary to our expectation of improvement. Despite students being unfamiliar with each other, communication skills for eliciting requirements were not effectively practiced. Most teams submitted pre-prepared problem specifications to client teams, bypassing the opportunity for role-playing exercises that could have fostered more direct interaction. Regarding communication tools, instant messaging was the most utilized, followed by videoconferencing, indicating a slight preference for asynchronous over synchronous communication.

Qualitative analysis revealed that the online setting significantly constrained students' communication despite the use of modern tools like videoconferencing and instant messaging. This limitation may be attributed to online meetings tending to be more focused and objective due to time constraints. In contrast, students tend to fully engage in face-to-face meetings. In online learning, interpersonal relationships tend to develop more superficially and be limited. In our study, this limitation may have impacted the efficiency of requirements elicitation activities, capturing the main functionalities but omitting essential details that could significantly influence the future stages of project development.

Similar studies have also assessed the development of communication skills in online teaching activities. Setiarini and Wulan [2021] analyzed soft skills development using Project-Oriented Problem-Based Learning (POPBL) in the online Software Project discipline. The results indicated a significant improvement in responsibility and hard work

skills, while Problem-Solving and Communication skills decreased. Another study by Zhang *et al.* [2020] reported a case study on adapting the Distributed Software Development course to the online modality. In this course, teams of students from different countries collaborated on a software project. The soft skills assessment showed that communication and cooperation scores decreased among students from the same university who had previously communicated in person. However, these skills improved among students from different universities despite language and cultural background limitations. The authors attribute this outcome to guidelines they followed to facilitate communication, such as using simple, direct language and confirming agreements and design decisions by email to avoid misunderstandings.

In our approach, students defined problems and took on the customer role, following Sindre's recommendation [Sindre, 2005]. The results revealed that using problems defined by students stimulated debates in the modeling process, leading to the emergence of new requirements and fostering the development of communication skills. In contrast to our approach, Arwatchananukul *et al.* [2022] utilized problems defined by stakeholders in an Introduction to Software Engineering course using PjBL. Their development process involved interaction with stakeholders in five out of seven stages (Planning, Requirements Analysis, Presentation Part 1, Design Validation, and Presentation Part 2). Their results highlighted collaboration as the most developed soft skill. However, they required more time to complete requirements elicitation-related phases due to difficulties coordinating meeting times between teams and stakeholders. Similarly to our work, they found feedback meetings throughout the project contributed to learning.

The original approach also suggests a warm-up session to help students integrate their assigned roles. In the proposed approach, conducting a warm-up videoconference remotely would require more complex communication tools to manage student groups within a larger virtual space. Since a warm-up session wasn't held, additional instructor interventions were necessary to emphasize to the students the importance of collaborative learning techniques for their learning. Similarly, related work also had to implement additional control measures to ensure the proper performance of activities related to the software project. In the case study by Zhang *et al.* [2020], professors from the involved faculties conducted weekly meetings to discuss progress and challenges and plan future steps. Additionally, they introduced an extra role in the development team (liaison) responsible for promoting communication between teams and facilitating the integration of parts of the developed software. In Setiarini and Wulan [2021] study, it was necessary to prepare monitoring and evaluation sheets with specific criteria for each soft skill. Due to the complexity and frequency of weekly assessments, teachers reported needing more time to assess students and suggest personalized activities.

The issues we identified after implementing the proposed approach in the two cycles of Action Research have led to the following recommendations for achieving better results when using this approach online.

- Basics of project management: As a project-based activity, students require skills in project management, such as creating requirement lists, setting goals, developing schedules, and dividing tasks. Therefore, before initiating the learning dynamic, the instructor should include a theoretical component encompassing these fundamental concepts to assist in executing the activity.
- Interaction between client and developer teams – To fully explore the benefits of the Role-Play technique, teams should be explicitly directed to communicate with each other, simulating the roles of client and developer in the early stages of system requirements elicitation. Sending the ready specification of the system to the development team should be avoided. Zowghi and Paryani [2003] noted that students perceived role-playing as a joke. The lessons learned suggested a warm-up session where instructors and students discuss team roles, profiles, and operations before starting the dynamic.
- Monitoring of project activities – The instructor should have a follow-up meeting a few days after the activity assignment to check if the teams managed to play the roles of customer and developer and the respective activities involved in each role. Tracking allows instructors to check whether teams have specified a problem a software project can solve and adjust the problem complexity level. It also enhances the group's engagement with the project and contributes to developing students' communication skills.
- Anonymization and evaluation – The instructor should make it clear to the students that the criticisms made of the system received by them while playing the role of a client are part of the role exchange dynamics and will not affect the students' performance evaluation in the discipline. Evaluation will focus on the consistency between requirements, screen prototypes, and other developed artifacts.
- Project presentation virtual meetings – The project presentation at the end of the course is another opportunity for the teams to assume customers' and developers' roles. During the product delivery meeting, each team presents the product project to the customer team, and customer teams share their impressions of the project.

One potential improvement to the approach is to have students address any missing or misunderstood requirements and their corresponding artifacts and prototypes. This would reinforce the importance of proper Requirements Elicitation and prevent costly rework. Additionally, the teacher can develop a list of objective evaluation criteria that students, in the role of customers, can use to evaluate the project created by developers. This would contribute to developing analytical reasoning skills and learning about defining acceptance criteria.

7.1 Limitations and Threats to Validity

This study has some limitations and aspects that may threaten its validity. First, since it was the initial application of the proposed approach, we cannot compare the results of this em-

pirical study with its implementation in a face-to-face setting. The evaluation of the approach relied on analyzing artifacts produced by students and a qualitative analysis of their responses to questionnaires.

Second, the teacher had no involvement in organizing students into groups; they were free to arrange themselves. This decision may have resulted in groups with varying levels of subject knowledge, leading to differences in difficulty levels when choosing their projects. To mitigate this problem in the first cycle, we assessed outcomes by examining artifacts created during project design. Adhering to Sindre's recommendation [Sindre, 2005], additional effort from the teacher was required in the second cycle to handle the requirements for each allocated problem. Monitoring in the second cycle enabled the evaluation of the complexity of each problem, addressing potential variations in complexity levels.

In the first cycle, we recognized the risk of students' professional experience influencing the study results. To mitigate this threat, researchers conducted a study with first-semester students who had little contact before the suspension of face-to-face classes due to the pandemic. Additionally, as the proposed problems in the first cycle had different complexity levels, researchers introduced follow-up meetings with the teams in the second cycle to analyze the difficulty levels of the problems and address students' questions.

Another identified risk was that students might feel intimidated when critically answering questionnaires or uncomfortable criticizing their colleagues' work. To mitigate this risk, researchers adopted anonymous responses and assured students that their performance as customers would not impact their evaluation in the discipline. These measures to mitigate threats are part of the guidelines for implementing this learning approach, presented in Section 7.

Furthermore, the absence of professor support during group sessions hindered the acquisition of additional information regarding the development of soft skills within the groups. Assessing soft skills proved highly subjective, relying on individual responses from each student. This method lacks feasibility as a reliable measure for gauging improvements in soft skills.

8 Conclusion

This study explored methods for teaching Requirements Engineering and fostering soft skills in remote education. Additionally, we aimed to identify the difficulties students face when limiting group activities to the remote modality. To this end, we introduced a collaborative learning approach adapted from Zowghi and Paryani [2003] that combines the techniques of Role-Play and Send-a-Problem for teaching Requirements Engineering remotely. The approach enables students to develop soft skills by collaboratively working in groups to tackle real-world problems through software projects.

We assessed the proposed approach in two cycles of Action Research, where students honed their teamwork skills on two occasions: during problem definition, where team members presented and debated ideas, and in the development of the solution project, where they deliberated on alternatives

to meet the problem's requirements.

The results indicated that students responded positively to the adopted approach. The role exchange facilitated the practice of social aspects associated with Requirements Engineering and soft skills development, including communication, collaboration, empathy, and conflict resolution. Despite communication challenges in online activities, participants actively sought ways to overcome them using dedicated computational applications.

Furthermore, the results highlighted that the approach facilitated knowledge exchange among team members while working on the problem solution, allowing them to apply theoretical knowledge about Requirements Engineering in a practical context. Knowledge exchange also occurred during project presentations, where customers and developers evaluated the proposed solutions to the presented problems.

Based on the study results, we formulated guidelines for implementing the approach in a remote setting. These guidelines addressed challenges from online courses, emphasizing the importance of project management concepts, warm-up sessions between teams, and follow-up meetings for team activities, among other factors. In future work, additional studies could assess the approach's performance in blended learning scenarios.

Declarations

Acknowledgements

We thank the Amazonas Public Prosecution Office – MPE/AM for the incentive to this research provided under the Program for the Functional Improvement of Employees. We thank all the participants in the empirical study and USES Research Group members for their support.

Funding

The present work is the result of the Research and Development (R&D) project 001/2020, signed with Federal University of Amazonas and FAEPI, Brazil, which has funding from Samsung, using resources from the Informatics Law for the Western Amazon (Federal Law n° 8.387/1991), and its disclosure is in accordance with article 39 of Decree No. 10.521/2020.

This research was also supported by CAPES-PROEX – Financing Code 001 and Amazonas State Research Support Foundation – FAPEAM – through the POSGRAD 22-23 project.

Authors' Contributions

BG contributed to the conception of this research approach. BG and GM planned and performed the studies. GM and AF analyzed the results. GM is the main contributor and writer of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets generated and/or analyzed during the current study are available in <https://figshare.com/s/9070d722f3adc1d29c47>.

References

- Al-Ani, B. and Yusop, N. (2004). Role-playing, group work and other ambitious teaching methods in a large requirements engineering course. In *Proceedings. 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2004.*, pages 299–306. IEEE. DOI: 10.1109/ECBS.2004.1316712.
- Alkadhi, R., Nonnenmacher, M., Guzman, E., and Bruegge, B. (2018). How do developers discuss rationale? In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 357–369. IEEE. DOI: 10.1109/SANER.2018.8330223.
- Arwathananukul, S., Singpant, P., Chondamrongkul, N., and Aunsri, N. (2022). Developing 21st century skills with project-based learning: An experience report in the introductory course of software engineering. In *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, pages 451–455. DOI: 10.1109/ECTIDAMTNCN53731.2022.9720408.
- Barkley, E. F., Major, C. H., and Cross, K. P. (2014). *Collaborative learning techniques: A handbook for college faculty*. Jossey-Bass - A Wiley Brand, San Francisco, CA, USA. Book.
- Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of the association for information systems*, 2(1):19. DOI: 10.17705/1CAIS.00219.
- Checkland, P. and Holwell, S. (1998). Action research: its nature and validity. *Systemic practice and action research*, 11(1):9–21. DOI: 10.1023/A:1022908820784.
- Costain, G. and McKenna, B. (2019). Experiencing the elicitation of user requirements and recording them in use case diagrams through role-play. *Journal of Information Systems Education*, 22(4):8.
- Davison, R., Martinsons, M. G., and Kock, N. (2004). Principles of canonical action research. *Information systems journal*, 14(1):65–86. DOI: 10.1111/j.1365-2575.2004.00162.x.
- Guerra-Báez, S. P. (2019). A panoramic review of soft skills training in university students. *Psicologia Escolar e Educacional*, 23. DOI: 10.1590/2175-35392019016464.
- Guo, P., Saab, N., Post, L. S., and Admiraal, W. (2020). A review of project-based learning in higher education: Student outcomes and measures. *International Journal of Educational Research*, 102:101586. DOI: 10.1016/j.ijer.2020.101586.
- Han, Z. and Zhang, Z. (2008). Integration of game elements with role play in collaborative learning — a case study of quasi-gbl in chinese higher education. In Pan, Z., Zhang, X., El Rhalibi, A., Woo, W., and Li, Y., editors, *Technologies for E-Learning and Digital Entertainment*, pages 427–435, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-69736-7_46.
- Johnson, R. T. and Johnson, D. W. (2008). Active learning: Cooperation in the classroom. *The annual report of educational psychology in Japan*, 47:29–30. DOI: 10.5926/arepj1962.47.0_29.
- Kokotsaki, D., Menzies, V., and Wiggins, A. (2016). Project-based learning: A review of the literature. *Improving schools*, 19(3):267–277. DOI: 10.1177/1365480216659733.
- Lima, M., Ahmed, I., Conte, T., Nascimento, E., Oliveira, E., and Gadelha, B. (2019). Land of lost knowledge: An initial investigation into projects lost knowledge. In *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–6. IEEE. DOI: 10.1109/ESEM.2019.8870171.
- Matteson, M. L., Anderson, L., and Boyden, C. (2016). ”soft skills”: A phrase in search of meaning. *portal: Libraries and the Academy*, 16(1):71–88. DOI: 10.1353/pla.2016.0009.
- Maxim, B. R., Brunvand, S., and Decker, A. (2017). Use of role-play and gamification in a software project course. In *2017 IEEE frontiers in education conference (FIE)*, pages 1–5. IEEE. DOI: 10.1109/FIE.2017.8190501.
- Memon, R. N., Ahmad, R., and Salim, S. S. (2010). Problems in requirements engineering education: A survey. In *Proceedings of the 8th International Conference on Frontiers of Information Technology, FIT ’10*, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/1943628.1943633.
- Miller, C., Rodeghero, P., Storey, M.-A., Ford, D., and Zimmermann, T. (2021). ”how was your weekend?” software development teams working from home during covid-19. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 624–636. DOI: 10.1109/ICSE43902.2021.00064.
- Mitri, M., Cole, C., and Atkins, L. (2017). Teaching case: A systems analysis role-play exercise and assignment. *Journal of Information Systems Education*, 28(1):1. Available at:<https://jise.org/Volume28/n1/JISEv28n1p1.html>.
- Musa, F., Mufti, N., Latiff, R. A., and Amin, M. M. (2012). Project-based learning (pbl): Inculcating soft skills in 21st century workplace. *Procedia - Social and Behavioral Sciences*, 59:565–573. Universiti Kebangsaan Malaysia Teaching and Learning Congress 2011, Volume I, December 17 – 20 2011, Pulau Pinang MALAYSIA. DOI: 10.1016/j.sbspro.2012.09.315.
- Nakamura, T. and Tachikawa, Y. (2016). Requirements engineering education using role-play training. In *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 231–238. IEEE. DOI: 10.1109/TALE.2016.7851799.
- Nwokeji, J. C., Aqlan, F., Olagunju, A., Holmes, T., and Okolie, N. C. (2018). Wip: Implementing project based learning: Some challenges from a requirements engineering perspective. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE. DOI: 10.1109/FIE.2018.8659307.

- Ouhbi, S., Idri, A., Fernández-Alemán, J. L., and Toval, A. (2015). Requirements engineering education: a systematic mapping study. *Requirements Engineering*, 20(2):119–138. DOI: 10.1007/s00766-013-0192-5.
- Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer Publishing Company, Incorporated, Germany, 1st edition. Available at: <https://dl.acm.org/doi/abs/10.5555/1869735>.
- Portugal, R. L. Q., Engiel, P., Pivatelli, J., and do Prado Leite, J. C. S. (2016). Facing the challenges of teaching requirements engineering. In *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, pages 461–470. DOI: 10.1145/2889160.2889200.
- Prince, M. (2004). Does active learning work? a review of the research. *Journal of engineering education*, 93(3):223–231. DOI: 10.1002/j.2168-9830.2004.tb00809.x.
- Setiarini, S. D. and Wulan, S. R. (2021). Analysis software engineering team's soft skills learning using online learning platform with project-oriented problem-based learning (popbl). *Inform*, 6(2). DOI: 10.25139/inform.v6i2.3986.
- Shapiro, S. S. and Francia, R. (1972). An approximate analysis of variance test for normality. *Journal of the American statistical Association*, 67(337):215–216. DOI: 10.1080/01621459.1972.10481232.
- Sindre, G. (2005). Teaching oral communication techniques in re by student-student role play: Initial experiences. In *18th Conference on Software Engineering Education & Training (cseet'05)*, pages 85–92. IEEE. DOI: 10.1109/CSEET.2005.36.
- Strijbos, J.-W. and De Laat, M. F. (2010). Developing the role concept for computer-supported collaborative learning: An explorative synthesis. *Computers in human behavior*, 26(4):495–505. DOI: 10.1016/j.chb.2009.08.014.
- Susman, G. I. and Evered, R. D. (1978). An assessment of the scientific merits of action research. *Administrative science quarterly*, pages 582–603. DOI: 10.2307/2392581.
- Svensson, R. B. and Regnell, B. (2017). Is role playing in requirements engineering education increasing learning outcome? *Requirements Engineering*, 22(4):475–489. DOI: 10.1007/s00766-016-0248-4.
- Tripp, D. (2005). Action research: a methodological introduction. *Educacao e pesquisa*, 31:443–466. DOI: 10.1590/S1517-97022005000300009.
- Tuckman, B. W. and Jensen, M. A. C. (1977). Stages of small-group development revisited. *Group & organization studies*, 2(4):419–427. DOI: 10.1177/105960117700200404.
- Van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. In *Proceedings fifth ieee international symposium on requirements engineering*, pages 249–262. IEEE. DOI: 10.1109/ISRE.2001.948567.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media. DOI: 10.1007/978-3-642-29044-2.
- Zhang, R., Mestre, P., Serodio, C., Prada, M., and GAO, W. (2020). Web-based teamwork: Distributed software development course under covid-19. In *2020 15th International Conference on Computer Science & Education (ICCSE)*, pages 45–50. DOI: 10.1109/ICCSE49874.2020.9201661.
- Zowghi, D. and Paryani, S. (2003). Teaching requirements engineering through role playing: lessons learnt. In *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.*, pages 233–241. DOI: 10.1109/ICRE.2003.1232754.