Applying Remotely the Challenge Based Learning Methodology in Undergraduate and Postgraduate Disciplines

Andrew Diniz da Costa, Carlos José Pereira de Lucena, Gustavo Robichez de Carvalho
Informatics' Department
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
andrewdiniz@gmail.com,
[lucena, guga]@inf.puc-rio.br

ABSTRACT

Currently, remote learning has been increasingly adopted by different institutions, with the COVID-19 pandemic being one of the reasons to accelerate the use of this approach. Using different tools, different ways have been experimented for offering good experiences from remote classes, providing overall student support infrastructure, applying learning/teaching methodology and so forth. Challenge Based Learning (CBL) is a modern and active learning approach that has been used by different institutions around the world. It provides an efficient and effective framework for learning while solving real-world challenges. Even though it is in use in a number of different places, there is a dearth of reports offering remote CBL usage experiences. In order to contribute to this gap, this paper presents how the CBL was applied in two computing disciplines: one undergraduate and one postgraduate. From these experiences, a discussion ensued that led to the identification of important learned lessons, contributing to future remote classes applying CBL, which are shared.

CCS CONCEPTS

• **Social and professional topics → Computing education → Computing education programs**; Software engineering education • **Applied computing → Education →** Distance learning

KEYWORDS

Challenge Based Learning, remote learning, software engineering.

1 INTRODUCTION

Remote learning [1] has been providing an opportunity for students and teachers to remain connected and engaged with the content while working at a distance. This learning approach was brought into the spotlight by the COVID-19 pandemic, which necessitated changes on the part of institutions, teachers and students.

Several works [2][3][4] deals with the relevance of providing students opportunities to learn and apply the required 21st century skills. Consequently, new learning methodologies [5][6][7] have been proposed, avoiding traditional approaches.

One of the methodologies applied is Challenge Based Learning (CBL), created with the essential principles of the skills expected in the 21st century. According to Mark Nichols et al. [5], CBL provides an efficient and effective framework for learning while solving real-world challenges. The framework fuels collaboration between students, teachers, families and community members to identify big ideas, ask thoughtful questions, and identify, investigate and solve challenges. This approach allows students to understand more about subject area knowledge and develop the important skills needed to thrive in an ever-changing world.

Since 2014, some teachers at the Pontifical Catholic University of Rio de Janeiro have applied CBL to undergraduate students to teach them how to develop mobile solutions from mobile programming courses following the boot camp model. Despite these experiences, remote learning using CBL has not yet been applied.

As of the COVID-19 pandemic, as occurred at other educational institutions, the undergraduate and postgraduate courses moved to remote learning. This scenario underscored the relevance of understating how CBL could be applied remotely, seeking to maintain the engagement of students in their learning journeys. Thus, this paper presents how that methodology was applied remotely for: (i) an undergraduate discipline called Software Quality (SQ), and (ii) a postgraduate discipline called Software System Design (SSD). The main focus of these experiences was to try to understand what worked and what didn't from the students' perspectives and, hence, use the knowledge from the learned lessons to improve future uses of CBL in the remote mode.

This paper is organized as follows. In Section 2, an overview of the CBL methodology is presented. Next, Section 3 shows some related works, and Section 4 explains how CBL was applied from the remote disciplines offered. Next, Section 5 presents a discussion from results gathered from information provided by students. And last, but not least, Section 6 offers the conclusions of the paper and mentions future works.

2 CHALLENGE BASED LEARNING METHODLOGY

According to [8] and [9], CBL builds on the foundation of experiential learning and leans heavily on the wisdom of a long history of progressive education. The framework is informed by

EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)

innovative practices in education, media, technology, entertainment, recreation, the workplace, and society.
CBL is divided into three main phases – Engage, Investigate, and Act – which are explained below.

The first phase, called Engage, is responsible for engaging learners in some challenge. Thus, this phase looks for personally connecting with academic content through the identification, development and ownership of a compelling challenge. To achieve that goal, three main concepts are considered: big idea, essential question, and challenge. Big idea is a theme, area, domain, to which the challenge will be related. Examples of big ideas could be: education, health, games, entertainment, accessibility, sustainability, etc.

Essential question is a question that engages learners to reflect about personal interests and the needs of the community. One example of an essential question could be: “How can I improve the accessibility of people to tourist attractions in my city?” In order to satisfy an essential question, learners can create several questions that help them to reflect, and subsequently one of these questions can be chosen. The process is known as Essential Questioning.

The third concept offered in the Engage phase is challenge, which calls to action the essential question for deeper understanding of the subject considered. One example of challenge could be: “Informing which accessibility is supported at tourist attractions.”

The next and second phase of the learning methodology is Investigate, which requests research for creating the necessary knowledge foundation to propose some type of solution that deals with the challenge indicated in the previous phase. To conduct this investigation, learners should identify guiding questions (GQs), which are questions that must be answered during the research process. In order to answer them, guiding activities (GAs) should be made, i.e., necessary actions that will make it possible to answer GQs. Examples of GAs could be reading a paper or a book and interviewing someone. In addition, resources used per GA are called guiding resources (GR). Examples of GRs could be books, academic papers, web sites visited, people interviewed, etc. At the end of this phase, learners analyze the data and consolidate the knowledge acquired from the research.

The last and third phase is Act, whose learners can propose and develop solution concepts. After the approval of the solution concept, they can develop prototypes, experiments and tests. New guiding questions can emerge during the execution of these actions. Hence, learners can return to the previous phase to complete the research. Next, learners implement the solution defined, measure outcomes and reflect on the work. The refinement of the solution can continue until the learners are satisfied.

The reflection act is important for learners to think in terms of lessons learned, what they could improve in the future, etc. If learners identify the need for more time for reflecting on a same challenge, more reflection milestones can be defined. Some reflection can be made in text, video, audio or any other manner that is effective for the learners.

It is important to emphasize that the CBL phases can be refined at any time. Considering that a given project is evolving, more knowledge is acquired by learners. Thus, for instance, the engage and investigate phases can be refined at any time to better reproduce the learning journey being followed.

3 RELATED WORKS

In the literature, a number of works present experiences applying CBL in different contexts. Two major and initial CBL studies applied were in 2009 [10] and 2011 [11]. The first study had 321 students and 29 teachers in six U.S. high schools. The results identified the clear efficacy of the CBL approach applied. The second study in 2011 involved 19 institutions from primary to undergraduate education, with 65 teachers and 1,239 students. That study demonstrated the higher satisfaction of students related to learning and schoolwork when applying the methodology.

A systematic literature review offered by Leijon et al. [12] presents works that used CBL in different contexts. Furthermore, it mentions that in 2018, there was a significant increase in the number of publications addressing CBL and a broader geographical spread. Thus, we can identify more people using it. Two examples of works mentioned by Leijon et al. [12] and that exemplify how CBL has been applied in different situations was an experience described by Fiore et al. etc. [13] and another presented by Bertori [14].

Fiore et al. [13] identified the need for a multidisciplinary environment while teaching challenge-based entrepreneurship and connects to design-thinking. In that work, CBL is presented as an educational method to change a learning environment to an active and authentic learning process based on problems.

Bertori [14] applies design thinking as a core part of CBL and explores an innovation course in engineering with a focus on work integration social enterprises that is both commercial and social.

Colombari et al. [15] explore the impacts of remote learning on a Challenge Based Innovation project. A survey of 92 students and interviews were carried out to assess the impact on learning outcomes and processes, and four main success factors were identified: informal interaction, time for exploration, asynchronous lecturing, relevant challenges.

Aiming to understand the impact of CBL in a context more related to the remote learning of students, this paper describes two disciplines of computing that applied CBL remotely. These experiences look for contributing to those interested in applying CBL in different ways, with remote classes being the scenario more intensely applied, especially after the beginning of COVID-19 pandemic.

4 APPLYING CBL IN REMOTE DISCIPLINES

In this section, we describe how CBL was applied remotely to different groups of students. Per subsection, details how CBL was applied in different disciplines are presented. Subsection 4.1 presents the CBL application in the software quality (SQ) discipline, for undergraduate students, while subsection 4.2

describes its application in the software system design (SSD) discipline.

4.1 Software Quality (SQ) discipline

In this subsection, the experience applying CBL in the SQ discipline applied in 2020.1 (March to July, 2020) is presented, with the participation of undergraduates from Computing courses. Initially, an overview of the discipline is presented, mentioning the main learning goal, the student profile and learning topics. Next, how the CBL was applied remotely is explained.

Discipline overview: The main goal of the SQ discipline was to teach the relevance of the Software Quality area and also how to guarantee high software quality from good practices. That discipline had 2 teachers and 22 students, of whom 21 were from the Computer Science course, and 1 person from the Information System course. The group's makeup was 17 men and 5 women.

The following list clarifies which learning topics were offered to the students.

1. Challenge Based Learning;
2. Understanding what software quality is and why it is important;
3. Functional and non-functional requirements;
4. Code documentation;
5. How to name artifacts in a project's development;
6. Modularization;
7. Software test;
8. Debugging;
9. Version control;
10. Risks in development projects.

Materials, feedback and tasks were offered from the Moodle tool [16], which is a popular learning management system. In the discipline, each content taught had at least one video class and a pdf file with slides used in the videos. Orientation was sent out almost every week to students to support them in the learning process, such as sharing doubts identified, recommending what artifacts they could consume, etc. The orientation ensued from discussions, private chats and emails. Thus, the approach applied in the discipline by teachers was to focus on asynchronous meetings. To motivate the participation of students even though the discipline was asynchronous, their participation to share knowledge and make exchanges with other students and teachers was taken into consideration in its assessment. This information was shared at the beginning of the semester and reinforced in the middle of the discipline.

CBL application: Initially, a video class explaining CBL was offered to students. Next, another video class relating CBL to the software quality area was also offered. In that video, to connect both, the Engage phase shown below was considered.

Big Idea: Software Quality

Essential Question: How can I guarantee a good quality in some software that will be developed using the object-oriented paradigm?
Challenge: Following known recommendations to develop software.

After presenting the Engage phase, a set of GQs was categorized in different learning topics that would be seen in the discipline. Both phases, Engage and Investigate, were useful to motivate students to reflect more about software quality. Below, there are examples of GQs categorized in 5 topics considered.

Learning topic 01: Code documentation

- GQ01.01: What is code documentation?
- GQ01.02: Why is code documentation important?
- GQ01.03: What is code comment?
- GQ01.04: How is it possible to comment some code?
- GQ01.05: Are there recommendations of code documentation?

Learning topic 02: Naming artifacts

- GQ02.01: What is the recommended way to name projects?
- GQ02.02: What is the recommended way to name classes?
- GQ02.03: What is the recommended way to name functions/methods?
- GQ02.04: What is the recommended way to name variables and constants?
- GQ02.05: What is the recommended way to name folders/packages in some project?

Learning topic 03: Modularization

- GQ03.01: What is code modularization?
- GQ03.02: What is code reuse?
- GQ03.03: What are known modularization techniques that can be used in development projects?

Learning topic 04: Software Test

- GQ04.01: What is test of software?
- GQ04.01: When should test be made in software?
- GQ04.02: What is black box test?
- GQ04.03: What is white box test?
- GQ04.04: What is test level?
- GQ04.05: What is test type?

Learning topic 05: Version control

- GQ05.01: What is version control?
- GQ05.02: What are the most known tools of version control used in development projects?
- GQ05.03: What are the necessary steps to deal with versions in development projects from some version control tool?

From the GQs presented, other video classes, which used them to discuss the corresponding learning topics, were also offered to the students. The order in which to watch these video classes was explained in the first class and resources were enumerated to facilitate the order. In addition, to motivate the discussion between participants of the discipline, forums and call meetings were organized by teachers, who acted as the mediators.

A month and a half after the start of the discipline, a challenge was put on the table. It requested the creation of contents related to software quality, to be read by their colleagues in the discipline in order to contribute to their learning journeys. Based on this, the following Engage phase was offered.

Big Idea: Software Quality
Essential Question: How can I share a good practice related to software quality with my discipline colleagues?
Challenge: Offering a content that explains good software quality practices.

From the Engage phase presented, students documented a set of GQs, GAs and GRs used to deal with the challenge proposed. In order to contribute together about how they could document their learning journey using CBL, a template document was shared with all students. It showed a possible structure to be followed that was divided in three parts from a template document: Engage, Investigate, and Act. To the first part, the content of the Engage phase presented previously (big idea, essential question and challenge) was offered for the students. The second part (Investigate phase) presented the structure presented in Table 1. From it, learners could document their GQs, beyond GAs, GRs and results collected per GQ identified.

And last, but not least, for the third part (Act phase) an open space was offered where each learner could write in the Solution proposed.

Table 1: Possible template to document Investigate phase.

Guiding questions	Guiding activities/resources	Result
GQ01	GA01	R01
GQ02	GA02	R02
GQN	GAN	RN

With regard to deliverables, the challenge requested the following artifacts: (i) CBL document describing the learning journey of each student, (ii) content file (e.g., video, article, etc.), teaching some topic.

Considering 22 participants of the discipline, 18 different topics were chosen, validated by the teachers. Below, is the list of the topics chosen by students.

1. Clean code;
2. How to choose test cases;
3. How to name variables from examples identified in the industry;
4. What the control version is and how to use Git in details;
5. Exemplifying a very well-structured algorithm;
6. How, when, and additional tips of modularization;
7. Code coverage criteria in structural tests;
8. Tips related to capability of maintaining automated test scripts from the Selenium IDE;
9. Code documentation using Sphinx for Python;
10. Project documentation;
11. Impact of automation in. relation to software quality practices;
12. Bad smells in code and refactory;
13. Object-oriented applied in games;
14. What Test Driven development is;
15. Interface tests of iOS apps;
16. Unit tests in Javascript;
17. What Gitlab is;
18. Code semantic versioning.

Aiming to motivate student exchanges and contributions to the contents being created, when they were delivered, each content had to receive feedback from at least three students. Each feedback was to inform: (i) positive points, (ii) suggested improvements, and (iii) additional comments. Before beginning this interaction, teachers talked with students, explaining the relevance of giving and receiving feedback.

Teachers read all the feedback submitted by students, and they offered additional comments. Some were directed to each student, while others were shared from a motivational discussion forum Next, one reflection per student was requested, to be written considering the following prompts.

1. My experience in that challenge was …
2. For me, the experience of remote learning has been …
3. The topics that I more learned from the discipline were …
4. The topics that I less learned from the discipline were …
5. I have more desire to learn about…

When the challenge ended, students could watch new content related to risks in development projects, which would be useful for them in the next and final challenge of the discipline.

That last challenge requested that each student provide an example of good strategy related to software quality to deal with some project development issue. Students could describe a real situation they had, or they could offer an example explaining how such a strategy could be applied. To guide the learners about the new challenge, the following Engage phase was presented to them.

Big Idea: Software Quality
Essential Question: How can I apply a strategy that helps to achieve a good quality in some software that I develop?
Challenge: Applying some strategy recommended in development projects understanding its related risks.

Besides the Engage phase presented, a set of GQs were also offered to the students. See below.

1. What was the main goal of the project development chosen?
2. What was the strategy chosen to contribute to the software quality in the project considered?
3. Were there risks associated to the strategy chosen?
4. What was the issue related to the strategy chosen?
5. What was the evidence (e.g. diagram, code, document etc) that shows the success of the strategy applied?

The time defined for the challenge was two weeks, with the following deliverables: (i) CBL document describing the learning journey of each student, and (ii) an additional document (e.g., pdf file) describing in detail the strategy applied along with the evidence.

Following the approach used in the previous challenge, at the end teachers sent individual and shared feedback to the students. In addition, students wrote another reflection considering the same prompts mentioned in the previous challenge.

In order to assess the students, teachers considered: (i) participation of activities offered in forums, such as quality of the feedback offered to their colleagues, and (ii) deliverables per challenge.

4.2 Software system design (SSD) discipline

In this subsection, the CBL application in the Software System Design discipline of pos-graduation offered in 2021.1 (March to July, 2021) is presented. Following the same structure presented in subsection 4.1, initially an overview of the discipline is presented. Next, details on how the CBL methodology was applied remotely are offered.

Discipline overview: The main goal of the SSD discipline was to teach how to design software systems using recommended software engineering practices. This discipline had 2 teachers and 12 Computing postgraduate students (4 women and 8 men).

The learned topics of the discipline were the following:

1. Challenge Based Learning;
2. Relevance of prototyping to software engineering;
3. Object-oriented paradigm;
4. Use case diagram;
5. Class diagram;
6. Sequence diagram;
7. Software agents;
8. Creational design patterns;
9. Structural design patterns;
10. Behavioral design patterns;
11. Framework
12. Software Product Line.

The contents, feedback and tasks were offered from the Moodle tool, and synchronous meetings from the Zoom tool [17] were held once per week (3 hours per class). These meetings represented a change compared to the SQ discipline presented in subsection 4.1, because some of the participants thought that it was important to have synchronous interactions with teachers and other students of the discipline. This turned out to be a good opportunity to recognize if these meetings would have a positive impact on the learning journey of the SSD students.

In addition, per class offered, at least one video class and pdf with slides produced by teachers were shared with students. Before the beginning of each class, students had to watch video classes offered and related to the learning topic defined for the next synchronous meeting. Students were motivated to bring questions related to the learning topic contemplated, such as how to apply them in the challenge in which they were participating.

All remote meetings had an initial overview of the correspondent learning topic. While that overview was offered, students could share their questions identified previously. Next, mentorships of the ongoing challenge were organized from the creation of rooms by Zoom. In each room students could work on the challenge while teachers went through the groups to check the need for some support.

CBL application: SSD began with an initial class about CBL, which introduced the learning methodology and some examples of design thinking techniques [18] (e.g., Brainwriting and Lotus Blossom). These techniques were a way to demonstrate how they could begin a creation process from a given dynamic.

The next classes looked at using CBL to present other learning topics. One example of an approach applied in several classes was to provide an Engagement phase at the beginning of the class with a Learning Topic as the Big Idea, as shown below.

Big Idea: Prototyping
Essential Question: How can I guarantee the creation of a good prototype?
Challenge: Creating a prototype to satisfy the desires of a customer.

Next, a set of guiding questions that considered important learning points and that should be answered were presented. Below, there are some examples of guiding questions used for some learning topics offered in classes.

Learning topic 01: Prototyping

- GQ01.01: What is low prototyping?
- GQ01.02: What is medium prototyping?
- GQ01.03: What is high prototyping?
- GQ01.04: What is the relevance of prototyping to software engineering?
- GQ01.05: How can you create a prototyping to receive early feedbacks of customers?

Learning topic 02: Object Oriented Paradigm

- GQ02.01: What does class mean?
- GQ02.02: What does object mean?
- GQ02.03: What is polymorphism?
- GQ02.04: What is class inheritance?
- GQ02.05: What is the difference of object variable and class variable?

Learning topic 03: Creational Design Patterns

- GQ03.01: What are the creational design patterns?
- GQ03.02: What is Singleton pattern?
- GQ03.03: What is Factory Method pattern?

Learning topic 04: Framework

- GQ04.01: What is framework?
- GQ04.02: What are hot-spots of a framework?
- GQ04.03: What are frozen-spots of a framework?

The Engage phase of a new challenge was offered to the students in order to enable them to apply good recommendations in a development project. For that challenge, we set up 3 groups with 3 students, one group with 2 students, and 1 student alone. The formation group was freestyle, but the maximum number of members per group needed to be 3 people.

Big Idea: COVID-19
Essential Question: What if we could offer access and visibility of vital signal data from patients to health professionals remotely?
Challenge: Provide visibility of vital signal data from hospitalized patients.

That challenge was divided in two parts. The first part requested the creation of a solution to the challenge offered. The solution should be a software and the time was 1 month. The deliverables defined were the following artifacts:

1. CBL document describing the learning journey;
2. Prototypes of the solution defined;
3. Class diagram;
4. Video class explaining some content related to software engineering learned from the discipline;
5. Reflection about the experience per student.

The second part of the challenge requested the creation of a framework based on the software proposed in the first part of the challenge. The time of the second part was 2 months, and the deliverables defined were.

1. Evolution of the CBL document describing the learning journey;
2. Evolution of the prototypes;
3. Class diagram with hot- and frozen- spots of the framework and classes of an instance of the framework;
4. A new video explaining some content related to software engineering learned from the discipline;
5. Demo of the solution developed;
6. Reflection about the experience per student.

Both reflections requested to the students from the part 1 and 2 of the challenge had the same prompts used by SQ discipline, which were the following:

1. My experience in that challenge was …
2. For me, the experience of remote learning has been …
3. The topics that I more learned from the discipline were …
4. The topics that I less learned from the discipline were …
5. I have more desire to learn about…

Two health experts participated in the challenge during the entire journey who could contribute questions related to the domain of the challenge. Furthermore, each part of the challenge had a final presentation, with the health experts offering feedback about the domain considered, while teachers provided technical feedback related to software engineering.

5 DISCUSSION

In this section, results gathered from reflections offered in the SQ and SSD disciplines are presented. These reflections functioned as self-assessments about the experiences of each of them in the discipline. Initially, the topics that had been more deeply learned and emphasized by students per discipline, as well as what they desired to learn more about, were presented. Next, what students thought about the experience of having remote classes was discussed. Lastly, some validity concerns of the work performed were addressed.

5.1 Learning topics of students who participated of the SQ discipline.

Of the two reflections requested from the 22 students, 20 of them answered both, being considered in the data collect. Table 3 presents which topics these 20 students most learned during the discipline. All students mentioned at least one of the topics listed in the table, with code documentation, version control and CBL, the topics most mentioned. The students that mentioned CBL never had seen this methodology previously and informed that it was a differential approach that contributed to the learning process.

Analyzing the reflections and talking with each student, teachers identified 5 participants that already had some previous contact with CBL. These students had the first contact with that methodology at a boot camp course offered by one of the teachers. Three of these 5 students mentioned in their reflections that they applied CBL in their in-person classes, but the remote experience was also good, because it supports how to structure the thinking in challenges.

Table 3: Topics most learned and highlighted by students in the Software Quality Discipline.

Topics	Amount
CBL	10
Code Documentation	9
Version control	9
Software Test	5
Debug	5
Good Practices of Development	5
UML Modeling with Design Patterns	5
OOP	5
How to name artifacts	5
Modularization	5
Prototyping	1

Table 4 shows which topics students had more desire to learn. The topics most mentioned were: UML diagrams and design patterns, software test, continuous integration, prototyping and modularization. One of the participants mentioned that the discipline covered all the necessary subjects of the learning topics presented, and thus he was satisfied. Another student did not suggest any additional topics to explore. Thus, the topics mentioned in Table 4 considered the participation of 18 students.

Table 4: Topics that students indicated desire to learn more about the Software Quality Discipline.

Topics	Amount
UML diagrams and design patterns	7
Software Test	3
Continuous Integration	2
Prototyping	2
Modularization	2
Metrics for software quality	1
Documentation of Requirements	1
Version Control	1
Improve quality of work in teams	1

5.2 Learning topics of students who participated in the SSD discipline.

Following the same approach as the SQ discipline, the two reflections offered in the SSD discipline were used to collect information from the 12 participants.

Table 5 shows the topics most learned in the discipline per student. The topics that received more mentions were: CBL, Design Patterns and Framework Concept. All students mentioned that CBL was a positive feature of the discipline, and it helped to deal with the challenge proposed and develop a good solution. Six of these students mentioned that relating CBL to learning topics presented in the discipline was a good approach to facilitate the learning of the methodology.

Design patterns, framework and UML diagrams were topics that were not a big surprise, because they are mentioned in several moments of the discipline. However, iOS development was a good answer, because considering the 12 participants of the discipline, 9 of them decided to develop using iOS, while 3 students developed using Android. The choice of which platform to use to develop the challenge was unrestricted, but both teachers would provide support. In the classes offered, examples of codes were presented using Java [19] and Swift [20], programming languages used to develop, respectively, apps for Android and Swift. In the cases where students decided to develop using the iOS platform, but if they did not have a Macbook or a remote access to a computer with Xcode [21], IDE to develop iOS apps was offered.

Table 6 shows which were the topics that students informed they were interested in learning more about. Two of the students reported they were satisfied with the contents presented. Thus, they did not mention any additional topic. Moreover, another student did not inform any topic, and did not offer an explanation. Hence, Table 6 considered the other 9 students, who informed what they desired to learn more about. Of these, 6 students informed the desire to learn other design patterns, while 5 students informed the desire to learn other software engineering recommendations for project development.

Table 5: Topics most learned and highlighted by students in the Software System Design Discipline.

Topics	Amount
CBL	12
Design Patterns	11
Framework	11
iOS Development	9
UML diagrams	9

Table 6: Topics that students indicated desire to learn more about in the Software System Design Discipline.

Topics	Amount
Additional design patterns	6
Additional software engineering recommendations	5

5.3 Experiences with Remote Classes

Table 7 presents the reactions of students of both of the two remote learning discipline experiences. In order to simplify the view of the feedback received, Table 7 organizes them according to the following categories: positive, neutral and negative. The first category considered only totally positive feedback provided. The second category is when students identified positive and negative points. Finally, the third category represents the situations where students mentioned only negative points.

Table 7: Experience of remote learning.

Discipline	Positive	Neutral	Negative
Software Quality (SQ)	11	6	3
Software System Design (SSD)	12	0	0

It was possible in the SQ discipline to identify 11 positive, 6 neutral and 3 negative feedbacks. While the SSD discipline identified only positive feedback from the 12 participants.

Aiming to identify which positive and negative points were informed by students per discipline, Table 8 shows the positive points, while Table 9 shows the negative points.

Beginning with Table 8, it is possible to identify the most positive points indicated by the students, which were the following: participating in practical challenges (25 mentions) and using the CBL methodology (22 mentions). In both disciplines, the relevance of having practical activities motivated the students' learning journey. Furthermore, students of both disciplines commented that the CBL methodology contributed to remote learning, such as giving them flexibility to decide upon embarking on different learning journeys, and to organize and identify which questions they should answer to deal with the related challenge. That approach was useful to engage them to learn the concepts presented in the disciplines.

All the students of the SSP discipline mentioned that CBL was a positive point to the remote learning process, whereas in the SQ, 50% of the participants who made the requested reflection highlighted CBL as positive point.

Table 8: Positive points mentioned by students of the remote learning per discipline.

Positive Points	SQ	SSP
Participating in practical challenges that showed how to work remotely and exchange knowledge with colleagues.	14	11
Using CBL was a good approach to remote learning.	10	12
Having free access at any time to good quality files (e.g. videos and documents) offered by the discipline.	11	6
Being a learning in a place process without having to worry about transport, traffic, delays, and eating out.	8	4
Having synchronous meetings, especially to deal with questions identified from some challenge.	0	10
Having different options to contact teachers.	2	3

Other points that also received a good number of mentions were: having free access to quality content at any time (17 mentions), and to learn from any place (12 mentions). Many of these latter mentions related to learn at any place informed that to choose when and where to study made a big difference in their lives.

Next, synchronous meetings received 10 mentions. These mentions were only from participants of the SSP discipline, which ran synchronous meetings. Below, the reasons mentioned for them are presented.

- feeling closer to the teachers and colleagues from the discipline;
- complementing the learning of the available artifacts offered to the students;
- holding short meetings to deal with questions identified in a given moment of the discipline. That was an option offered by teachers them in order to support the learning of them.

On the other hand, some students of the SQ discipline informed they wanted more face-to-face meetings with colleagues and teachers. Table 9 shows the negative points informed per discipline.

The points that received more negative feedback (7 mentions each one) were: (i) maintaining focus and learning engagement and (ii) feeling overwhelmed with other disciplines and other activities. Some students mentioned that the number of emails received, works of different disciplines and activities related to jobs overwhelmed them during their studies. In addition, some of them mentioned that it was hard to decide when they would review some available materials, because they were available since the beginning of the discipline. This lack of self-organization and planning (3 mentions) was an issue that impacted their focus and learning engagement.

Two additional points related to focus and engagement was the absence of live in-person classes (5 mentions) and more face-to-face meetings offered (4 mentions). The number of mentions of both was not high, but deserves attention in order to maintain the engagement of learners, because not everyone self-adapted to the remote class model.

Table 9: Negative points mentioned by students to the remote learning.

Negative Points	SQ	SSP
Maintaining focus and learning engagement.	7	0
Feeling overwhelmed with other disciplines and other activities that they must deal with daily.	5	2
Missing in-person classes.	5	0
Missing interactions face-to-face with teachers and colleagues.	4	0
Organizing and planning him/herself for completing the activities requested.	3	0

Considering the information collected from the reflections sent by students, a set of important learned lessons were identified, as follows.

1. Deciding to use CBL in remote disciplines work, in view of the fact it is an active methodology that engages with the learning of students. There was no negative feedback about it, and CBL contributed to their learning journeys, as mentioned by 22 students;
2. Presenting learning topics using CBL, and offering practical challenges were good ways to reinforce the use of the methodology;
3. Providing additional quality resources that can be accessed by students anytime makes a difference. Several synchronous SSD classes were used to discuss topics presented in previously available videos, as was how to use them correctly in development projects. That approach was more mentoring time than traditional teaching with someone explaining slides over the course of various hours. This format stimulated interaction between teachers and students;
4. Sharing tips that can contribute to the planning and organization of challenges can help deal with other activities that students have. For instance, the use by both disciplines of the possibility of using a Kanban board [22] to support the view of which guiding questions were and were not answered. At the end of the disciplines, one student of each discipline (SQ and SSD) mentioned that they began to use Kanban to organize their personal activities, being useful to deal with and balance these activities.
5. Offering synchronous meetings contributed to the engagement of SSD students, and some SQ participants mentioned that also could have happened. Thus, providing synchronous meetings seems to be important to some students. Some possible approaches that can be offered in these meetings and that were used in SSD are the following:

   a. Engaging students to bring questions from some learning topic. To offer some artifact (e.g., paper or video) that needs to be consumed before the meeting, it is a possible way to motivate them;
   b. Presenting complementary contents that supports the student learning related to a given topic. Creativity techniques [18] called Brainwriting and Lotus Blossom were presented to the SSD students. These techniques were useful, for instance, to support the creative process of the students to identify new guiding questions in the challenge in which they were participating;
   c. Organizing dynamics that allow interactions among them. One example offered for SSD was to establish rooms through the Zoom tool for students to create low fidelity prototype based on requirements explained by teachers;
   d. Defining remote presentation of solutions developed by students. The SSD students presented to the results achieved and they received feedbacks from the teachers.

5.4 Validity concerns

In this subsection, we have listed some limitations related to the assessment approach offered. See below.

First, as is typical for studies of educational settings, it is possible that our results are context dependent. For example, per discipline offered we had students with different profiles and experiences. For SQ, we had undergraduates (20 to 28 years old), while SSP had postgraduate students (22 to 40 years old). Moreover, some differences between the disciplines contributed to the answers offered, the disciplines contributed for answers offered, such as, having more synchronous meetings in SSD than SQ, and offering the SQ discipline in the beginning of the pandemic. When SSD was offered, the participants were more used to remote classes. However, even with these differences mentioned, these experiences made it possible to identify relevant points learned.

Second, we did not use the same group of students to offer in-person classes, avoiding a comparison with remote classes. It was not possible, because during the pandemic, the university was closed. Even with that restriction, we understood the relevance of sharing learned lessons using CBL in remote classes.

Third, two SQ students did not send in the requested reflection. These answers could identify new information identified. However, considering interactions that teachers had with the students, the most important feedbacks were identified.

Fourth, reflections offered by students were used to collect data. These reflections worked as self-assessments of the experiences that each student had in a given discipline mentioned in the paper. Hence, differences could be identified if an effective and more formal assessment was applied in order to understand the student learning process.

6 CONCLUSIONS AND FUTURE WORKS

This paper presented how the application of CBL was carried out in two remote learning disciplines related to computing. From student reflections, teachers could identify what they thought of remote disciplines applying CBL. After collecting and analyzing the related data, teachers of the disciplines understood that a number of them considered its application useful and important.

The teachers tried to talk individually with the 12 students of the SQ discipline who had not mentioned CBL as a highlight of the course, after they had sent their reflections, in order to map what they thought about the learning methodology. Teachers spoke with 8 of these students, and they mentioned that applying CBL had been a good experience, reinforcing its contribution to the disciplines.

Important learned lessons were identified, contributing to some future steps, as follows.

- Applying CBL in hybrid classes, i.e., a subset of students will be remote, while the others will be physically in the same environment along with the teacher. This class approach is a possibility with the return of live classes in the university;
- Experimenting different dynamics, challenges, and group formations in remote and hybrid disciplines aiming to engage the interaction of the students;
- Applying CBL in research with different student profiles working on a same team: PhD candidates, Masters candidates, undergraduates, and teachers from different fields.

Thus, this paper showed how CBL was applied in remote disciplines aiming to exemplify its use and, hence, contribute to those who have an interest in applying modern and active approaches of learning.

7 ACKNOWLEDGMENT

This work was supported by the PPI Softex program, Agreement No. 0200-10/2021/SOFTEX/PUCRio/Residence at TIC, financed by the Ministry of Science, Technology and Innovation with resources from Law 8.248 of October 23, 1991. In addition, this work was also supported by State Institute of Engineering and Architecture (IEEA) linked to the State Department of Infrastructure of the State of Rio de Janeiro, through Contract 001/2021 with funds from the Government of the Rio de Janeiro State.

REFERENCES

[1] Tech & Learning: Tools & Ideas to Transform Education. 2021. Retrieved from: https://www.techlearning.com/how-to/what-is-remote-learning

[2] Peggy A. Ertmer and Anne T. Otternbreit-Leftwich. 2019. Teacher technology change: How knowledge, confidence, beliefs, and culture intersect. Journal of Research on Technology in Education, 42, 255-284. https://doi.org/10.1080/15391523.2010.10782551

[3] Michael Fullan. 2013. Stratosphere: Integrating Technology, Pedagogy and Change Knowledge (1st ed.). Toronto: Perason.

[4] Sølvi, Lilllejord, Kristin, Børte, and Erik Ruud, E. 2018. Teaching and Learning with Technology in Higher Education. Knowledge Centre for Education, The Research Council of Norway, Oslo.

[5] Mark Nichols, Karen Cator, and Marco Torres. 2016. Challenge Based Learner User Guide. Redwood City, CA: Digital Promise. Retrieved from https://cbl.digitalpromise.org/wp-content/uploads/sites/7/2016/10/CBL_Guide2016.pdf.

[6] Woei Hung, David H. Jonassen, and Rude Liu. 2008. Problem-based learning. In the Handbook of research on educational communications and technology (3rd ed.), Routledge, 485-506.

[7] Pawan Kumar Jha. 2006. Modern Methods of Teaching and Learning (1st ed.). Publisher: Rajat Publications.

[8] Apple Inc. 2022. Challenge Based Learning. Retrieved from http://challengebasedlearning.org/.

[9] Apple Inc. 2022. Challenge Based Learning. Retrieved from https://www.challengebasedlearning.org/framework/

[10] The New Media Consortium. 2009. Challenge Based Learning: An Approach for Our Time. Retrieved from http://rewiringeducation.com/wp-content/uploads/2018/02/Challenge-Based_Learning-An_Approach_for_Our_Time.pdf.

[11] The New Media Consortium. 2011. Challenge Based Learning: The Report from the Implementation Project. Retrieved from http://cbl.digitalpromise.org/wp-content/uploads/sites/7/2016/06/NMC_CBLi_Report_Oct_2011.pdf.

[12] Marie Leijon, Petri, Gudmundsson, Patricia Staaf, and Cecilia Christersson. 2021.Challenge based learning in higher education – A systematic literature review. Journal of Innovations in Education and Teaching International. https://doi.org/10.1080/14703297.2021.1892503.

[13] Eleonora Fiore, Giuliano Sansone, and E. Paolucci. 2019. Entrepreneurship education in a multidisciplinary environment: Evidence from an entrepreneurship programme held in Turin. Administrative Sciences, 9(1), 28. https://doi.org/10.3390/admsci9010028.

[14] Marco Bertoni. 2019. Work integration social enterprises. A testbed for challenge based learning? In 15th international CDIO Conference. Aarhus.

[15] Ruggero Colombari, Elettra D’Amico, and Emilio Paolucci. 2021. Can challenge-based learning be effective online? A case study using experiential learning theory. CERN IdeaSquare Journal of Experimental Innovation, 5(1), 40–48. https://doi.org/10.23726/cij.2021.1287.

[16] Moodle. 2022. Moodle – Open-source learning platform. Retrieved from https://moodle.org.

[17] Zoom. 2022. Zoom - Video conferences, Cloud phone, webinars, chat, virtual events. Retrieved from https://zoom.us.

[18] Michael Michalko. 2006. Thinkertoys: A Handbook of Creative-Thinking Techniques. ISBN 9781580087735. Ten Speed Press, 2nd ed.

[19] Oracle. 2022. JavaTM Programming Language: Java Documentation. Retrieved from https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html.

[20] Apple Inc 2022. Swift – Apple Developer. Retrieved from https://developer.apple.com/swift/.

[21] Apple Inc. 2020. Xcode IDE. Retrieved from https://developer.apple.com/xcode/.

[22] Leopold, K. (2017, December 29). Practical Kanban: From Team Focus to Creating Value. ISBN 3903205001. Leanability Press.