Exploring Design Thinking as a Software Accessibility Awareness Raising Methodology in Computing Courses

Alan Trindade de Almeida Silva
Universidade Federal do Pará
Belém - PA, Brasil
alantas@outlook.com

Luciano Arruda Teran
Universidade Federal do Pará
Belém - PA, Brasil
luciano.teran@icen.ufpa.br

Thayssa Águila Rocha
Universidade Federal do Pará
Belém - PA, Brasil
thayssa.rocha@icen.ufpa.br

Marcelle Pereira Mota
Universidade Federal do Pará
Belém - PA, Brasil
mpmota@ufpa.br

ABSTRACT

Despite many advances in the area of software accessibility, there are still many problems to be overcome that prevent a greater availability of accessible software. One of these problems is that accessibility is usually not included in the curriculum of computing courses. This educational gap causes software development professionals to leave college or university without understanding how the software they develop can be adapted to guarantee its use by people with different needs and how to implement these adaptations. This work proposes and evaluates an accessibility teaching methodology based on Design Thinking with the main objective of increasing accessibility awareness among students. The results indicate that the proposed methodology was effective and managed to significantly increase students' accessibility awareness, making them more sensitive and empathetic about the topic. The results also showed that the students approved of the methodology and considered it interesting and engaging.

KEYWORDS

Software, Accessibility, Design Thinking, Awareness

1 INTRODUCTION

When software is accessible, it provides people with disabilities or any diminished capability, be it definitive or temporary, with the ability to fully interact and utilize its functionalities. For software to be accessible, it must be purposely developed with accessibility concerns in mind, which is not always the case.

The demand for software developers with accessibility skills has increased over the last few years [1], which indicates a change in the scenario regarding accessibility. Much of this change is associated with the legal factor, with many countries developing legislation that enforces accessibility in some scenarios [2], and the social responsibility factor, which also leads many companies to invest in accessible development [3].

Despite this increase, the availability of accessible software remains low, preventing its use by those who depend on it. According to the WebAIM Million report of 2022 [4], of the top 1 million homepages of the web, 96.8% failed to fully comply with the WCAG's lowest levels of accessibility requirement, and users with accessibility needs would expect to encounter errors on 1 in every 19 home page elements with which they engage.

In previous studies [5, 6], we surveyed and interviewed software development professionals and explored the reasons for the low adoption of accessibility in software projects. In this study, the professional's lack of knowledge about accessibility emerged among other problems, such as the initial increase in development costs and the lack of interest from stakeholders. This study also found that one of the reasons for this lack of knowledge is that accessibility is not taught in undergraduate courses. Professionals indicated that accessibility was not taught or was taught only briefly and not in-depth during their academic careers, usually in the HCI course. Consequently, new professionals leave college without understanding the importance of accessibility and without the knowledge of how to implement accessible software.

Not knowing how to implement accessibility can be readily addressed at any given time when accessibility appears as a requirement for a project, as several guidelines exist and cover various scenarios and accessibility characteristics. Each development platform also offers detailed manuals on how to implement accessibility, with some even including accessibility evaluation tools [7, 8]. However, without software development professionals being aware and empathetic about the needs of people who depend on accessibility, it is much more difficult for accessibility to become a requirement for the software they develop.

In the presented context, the inclusion of accessibility in the academic curriculum can be addressed from a perspective of raising awareness and not necessarily from a perspective of teaching technical skills. A professional with accessibility awareness understands that there are people with different needs and that there are ways in which they can use computing devices. This understanding allows them to express, inside a team, their opinion on not neglecting accessibility. Although turning a suggestion into reality goes beyond the intention of just one professional, having more professionals working in all positions in the software development process understand the different needs of different people can help to include accessibility in the scope of projects.

The teaching of accessibility in computing courses is not a new research subject, and studies in the literature aimed to find the best ways to integrate it into the academic curricula of institutions [9–12], proposing educational resources and methodologies that seek to both raise awareness and teach technical skills.

However, most previous work either relies heavily on traditional teaching methods, such as lectures and project assignments, without providing in-depth information about the material used for teaching, often disclosing only the topics that were addressed, or lacks a more robust evaluation of the efficacy of the proposed methodology. This scenario makes it difficult to reproduce and broadly adopt the proposed methodologies.

In this work, we propose and evaluate an easily reproducible accessibility awareness teaching methodology to be used in computing courses based on Design Thinking (DT). The proposed methodology was applied as a portion of an HCI course offered to undergraduate and graduate students of computing courses at a Brazilian university and had the participation of 33 students.

The work comprises seven sections: Section 2 presents the related work; Section 3 presents the research design; Section 4 presents the proposed methodology; Section 5 presents the results; Section 6 presents a discussion on the results; Section 7 presents final considerations and future work.

2 RELATED WORK

Several studies addressed the problem of including accessibility teaching in the curriculum of computing courses and the question of how to evaluate the effectiveness of the teaching methodologies they proposed. Some studies suggest methodologies that could be employed within existing courses, and some have developed comprehensive course-long accessibility teaching frameworks.

Palan et al. [11] evaluated the accessibility awareness and accessibility knowledge of university students before and after a week of receiving accessibility lectures and developed a methodology to measure and compare the students' overall knowledge of these two topics. Results showed that after the lectures, there was an overall increase in these metrics. The methodology proposed for measuring teaching efficacy was reproduced in the literature for other similar studies, and our work extends upon it to evaluate the efficacy of our proposed teaching methodology. Despite the positive results presented, the authors do not describe the lectures' content, only the topics of each lecture, making it irreproducible.

El-Glaly [9] presents the development and refining over three cycles of a dedicated accessibility course offered to graduate students of Software Engineering. The third and final iteration was a course comprising 70% development assignments and 30% design assignments, with three projects being developed over the period of the course. Through a qualitative analysis of the answer to a post-course questionnaire, the author found that the students believed the course was useful to their careers as well as useful in teaching them about accessibility. The study presents the proposed methodology and the activities but lacks a robust analysis of its efficacy.

Shinohara et al. [12] incorporated accessibility in a Technology Design course; the authors investigated the tensions between inclusive design and general design and if teaching accessibility alongside the pre-existing course would introduce barriers to the process of learning Design Thinking. All the students involved in the experiment met the course objective of designing a project incorporating accessible design, and the work thoroughly describes the methodology applied. Still, it lacks a comprehensive evaluation of how students' knowledge and awareness of accessibility changed before and after the course, relying on their self-reported perceptions.

Zhao et al. [10] performed an extensive 4-year study across 29 courses, exploring the use of 4 different methodologies: lectures on accessibility, team projects, direct interaction with someone who uses accessible technologies, and collaborating with a fellow student with a disability. Through an evaluation method based on Palan et al. [11] proposal, the students were evaluated on accessibility knowledge and awareness before and after the course. The authors found lectures, projects, and interactions to be the most effective interventions for increasing accessibility awareness. Based on the authors' results, we explored a way to integrate the effective intervention found into just one easily reproducible methodology.

Table 1 presents a brief comparison between the main contributions of the works presented above in relation to this work. Although several studies have proposed different approaches to teaching accessibility, most of them either rely on traditional teaching methodologies or lack a robust evaluation of their effectiveness. This study aims to fill this gap by presenting a new accessibility teaching methodology based on the Design Thinking framework and a comprehensive evaluation of its effectiveness.

Table 1: Comparison between the contributions of related work.

Contribution	Work				
	[11]	[9]	[12]	[10]	This work
Full description on how to reproduce the methodology		X	X	X	X
Extensive evaluation of the effectiveness of the methodology	X			X	X
Comparison between traditionally applied methodologies				X	
Proposal and evaluation of a new methodology					X

3 RESEARCH DESIGN

For this work, we applied an accessibility teaching methodology based on Design Thinking with the objective of increasing accessibility awareness among students. To measure the effectiveness of the methodology, we employed a technique already applied in literature consisting of administering the same questionnaire before and after the classes, with the questions of this questionnaire being designed to directly explore the participant's intention and understanding regarding accessibility.

In addition to the awareness questionnaires, the participants also answered a demographic questionnaire and a questionnaire that collected their opinions about the applied methodology. Figure 1 shows the steps followed in the development of this work. The Ethics Committee of the Federal University of Pará approved this research under the number 5.222.718.

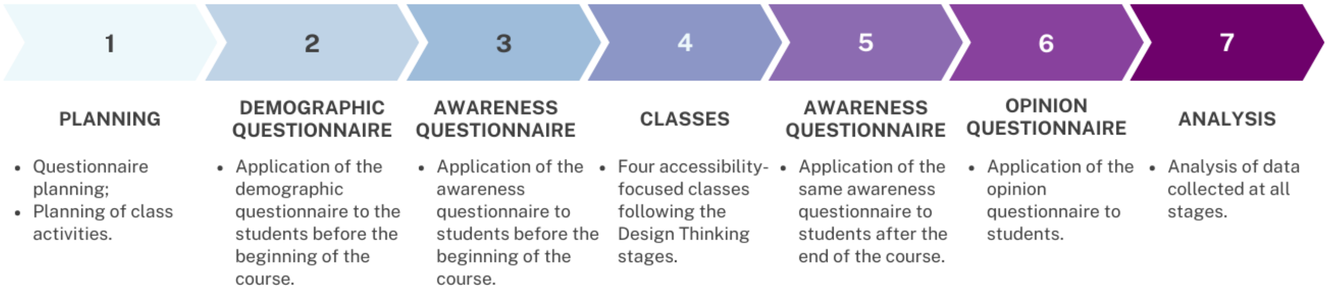

1
PLANNING
• Questionnaire planning;
• Planning of class activities.
2
DEMOGRAPHIC QUESTIONNAIRE
• Application of the demographic questionnaire to the students before the beginning of the course.
3
AWARENESS QUESTIONNAIRE
• Application of the awareness questionnaire to students before the beginning of the course.
4
CLASSES
• Four accessibility-focused classes following the Design Thinking stages.
5
AWARENESS QUESTIONNAIRE
• Application of the same awareness questionnaire to students after the end of the course.
6
OPINION QUESTIONNAIRE
• Application of the opinion questionnaire to students.
7
ANALYSIS
• Analysis of data collected at all stages.

Figure 1: Sequential activities followed.

3.1 Participants

All students enrolled in the HCI course for the given semester participated in the classes in which the study was carried out. However, students were presented with the option of not having their data collected for the study. The HCI course is a core course for Information Systems undergraduate students, and it is also offered electively to other computing-related undergraduates, masters, and doctoral students. The class had a total of 43 students, composed of 11 graduate students and 32 undergraduate students. However, only 33 students opted to have their data collected and participate in the study. They were four Doctoral students in Computer Science, four Master's students in Computer Science, 15 undergraduate students in Information Systems, five undergraduate students in Computer Engineering, and five undergraduate students in Computer Science who participated.

Of the participants, 17 reported already working in the software development industry. Nevertheless, only 30.3% indicated that they had already participated in developing software that implemented accessibility. Regarding contact with people with disabilities in their personal lives, 63.6% of the participants indicated not having it, 18.2% answered having a friend with a disability, 15.2% indicated having a relative with a disability, and 3.0% (one person) indicated having a disability themselves.

3.2 Measuring Accessibility Awareness

Many studies proposed different approaches to measure the effectiveness of accessibility teaching methods. While some approaches are based on the qualitative analysis of open-ended questions [9] or objective questions about the student's opinions on how the applied teaching methodology has changed their perceptions of accessibility issues [12], Palan et al. [11] proposed an extensive Accessibility Awareness measuring technique consisting of a questionnaire through which the participants received an "accessibility awareness score" according to their answers. Higher composite scores indicated greater accessibility awareness.

For this study, we created an adaptation of the methodology applied by Palan et al. [11], which also consisted of a questionnaire but was composed of six questions categorized into four learning objectives: overall accessibility awareness, technical knowledge, empathy, and potential endeavors. Baker et al. [13] performed a systematic literature review of papers on accessibility in computing education and identified these four categories as the key learning objectives commonly covered. We then created questions for each category that were not aimed at the knowledge that would be directly taught in class but subjectively explored participants' perceptions on the categories.

Table 2 presents the questions and their categories. Most questions could be answered on a five-level Likert scale ranging from Strongly Disagree to Strongly Agree, except for questions 3 and 4. Question 4 was a Yes or No question, and question 3 was a five-level Likert scale ranging from Not Confident to Very Confident. The accessibility characteristics presented on questions 3 and 4 were the same and are also based on what is proposed by Palan et al. [11], with the addition of low literacy: low vision, blindness, deafness, autism, learning disability, low literacy, motor disability, older adults and intellectual disability.

The participants could receive a score from 0 to 4 points for each answer. The Yes or No questions scored 0 for No and 4 for Yes. Likert scale questions started at 0 for the most negative answer and increased by 1 until the most positive answer, worth 4 points. Participants answered questions 3 and 4 multiple times, one for each accessibility category. The possible score range for the questionnaire was 0 to 88 points. This questionnaire was then applied to the students before the start of classes and then reapplied after the end of the cycle of accessibility classes in order to compare how the proposed accessibility teaching methodology influenced their overall accessibility awareness.

Table 2: Awareness questionnaire and their categories.

#	Category	Question
1	Empathy	I consider it important to include accessibility concerns in software development.
2		I think about the accessibility implications of the software projects I develop.
3	Technical Knowledge	I feel confident and capable of developing accessible software for the following accessibility categories.
4	Overall Awareness	Do you know how people with the following accessibility characteristics interact with computers, cell phones, and other computing devices?
5		I believe teaching accessible software development is important in the curriculum of computing courses.
6	Potential Endeavors	I have a personal interest in developing software products that have accessibility concerns.

3.3 Teaching Accessibility Using DT

Design thinking is defined as an analytic and creative process that enables a person or a group of people to experiment, create and prototype models, gather feedback on them, and redesign them if needed [14]. The Design Thinking process defines a problem-solving framework composed of five stages: empathize, define, ideate, prototype, and test. One of the main advantages of solving a problem using Design Thinking is that it is human-centered [14]. It puts human beings, the people we are ultimately solving problems for, at first.

In this context, Design Thinking can also be considered an excellent prospect framework for teaching accessibility, particularly when the main focus is raising students' awareness of accessibility issues. By going through all the stages of Design Thinking to create an accessible software product, the student is encouraged to seek knowledge about the necessities of people who need accessibility, understand the process and adaptations that allow these people to use software, understand how to prototype and implement these adaptations, as well as learn how to evaluate what was prototyped.

Another advantage of its application as an accessibility learning methodology is that the knowledge about accessibility is not presented to students but constructed by themselves during each stage of Design Thinking. This process removes the need to share lecture material, an approach used by some authors [15, 16], and allows flexibility in terms of the duration of its application. For short periods it is possible to define that only one iteration of the process will be performed, as well as to define lower fidelity prototyping approaches and faster testing alternatives.

In our experiment, students were responsible, over four classes, for developing the Design Thinking process for an accessible software project that would meet the needs of a pre-defined accessibility category. The nature of the developed project was a free choice, and the accessibility category could also be freely chosen as long as it was not repeated. Students attended one class per week, and each class lasted for about 3h50m. The course had 11 graduate students, and 11 groups were formed for the development of activities, each with one graduate student and 32 undergraduate students divided between them. In the first class, the groups explored the first two stages of design thinking, and one stage per class was explored in the following three classes.

All the Design Thinking stages were developed in the classroom, and students were encouraged to bring computers, tablets, cell phones, sketchbooks, and any other resource they thought they would need to use to develop the activities. After the development of each stage, the groups were responsible for presenting to the class their learning and what they had developed for that stage. At the same time, the other students were encouraged to ask questions about the development of the presented stage. In the test stage, there was no presentation to the class; however, each group tested their project with a member of another team. This process aimed to disseminate knowledge about each of the accessibility characteristics attributed to each group to all groups.

3.4 Opinions of the Participants

In addition to answering the awareness questionnaire for a second time, the students also answered, after the classes, a questionnaire exploring their opinions. In this questionnaire, students could share their opinions on the teaching methodology and its impact on their intentions regarding accessible software development. Table 3 presents the questions that made up this questionnaire. Questions 1 to 8 were to be answered on a 5-point Likert scale ranging from Strongly Disagree to Strongly Agree. Questions 9 and 10 were open and non-required questions.

Table 3: Opinion questionnaire questions.

#	Question
1	The teaching methodology used aroused my curiosity about the subject of accessibility in software.
2	The teaching approach used was interesting and held my attention.
3	After the classes, I believe I know more about the subject of accessibility in software.
4	After classes, I am more likely to consider accessibility in the projects I develop.
5	After classes, I am more likely to consider accessibility as a field of research that I would develop.
6	After the classes, I am interested in deepening my knowledge on the subject of accessibility in software.
7	After classes, I feel more sensitive to the difficulties faced by people who need accessibility.
8	I believe that accessibility in software should be a separate subject offered in the curriculum of computing courses.
9	What do you think could have been added to the classes to improve them?
10	Use this space to add criticism, compliment, observations, or any other comments you want about the accessibility classes taught.

The answers to this questionnaire, while not making up the final student accessibility awareness score, provide an excellent mechanism to explore the student's side of participating in the experiment, how they felt exploring Design Thinking as a teaching methodology, and how they judge its effectiveness.

4 PROPOSED METHODOLOGY

In this section, the activities carried out in each class will be described in detail, as well as observations on what happened in the classroom during the experiment. This section aims to make it easier for other educators to replicate the method by providing a step-by-step guide as well as insights into what to expect while developing the activities.

4.1 First Class

The main objective the students had in executing the Design Thinking process and going through all its predefined stages was the creation of one accessible software. In the first class, students performed the empathizing and defining stages of Design Thinking. Students were asked to divide into groups, each containing 1 of the 11 graduate students and an equal division of the 32 undergraduate students. Afterward, the groups were asked to choose one accessibility characteristic (such as blindness, low vision, and low literacy). We also explained to the students that each group would be responsible for developing one software aimed at the needs of users with the accessibility characteristic chosen by them.

Once the accessibility characteristic of the group was defined, we moved on to the empathizing stage. First, we gave the groups 10 minutes to search the internet for as much information as possible about their chosen accessibility characteristic. The intention was for the students to discover what difficulties people with those accessibility characteristics may face and what adaptations they need, both in the real and digital worlds.

After that, the groups had another 10-minute session to focus their attention on how people with these accessibility characteristics interact with computing devices and how they perform tasks they could not perform without accessible software design. For accessibility characteristics that are easily simulated (such as blindness or deafness), we suggested that groups try to perform a software interaction task by simulating them. For accessibility characteristics that are more difficult to simulate (such as low vision or low literacy), we suggested that they search for videos showing these people interacting with computing devices.

Finally, each group had 5 minutes to share the knowledge they obtained with everyone. The groups shared with each other the particularities of each accessibility characteristic, the main difficulties faced, how these people interact with computing devices, and how software developers can help these people.

After this activity, we advanced to the defining stage of the Design thinking. This stage is where a meaningful and actionable problem statement is defined, and in a way, our empathizing stage initiated this definition. In order to continue it, we understand that it was essential to define some things about accessible software design with the students. At this stage, we introduced students to the formal and legal concepts of accessibility and disability and explained that accessibility goes beyond helping people with disabilities and that it also serves anyone with a reduced capability. We presented the main accessibility guidelines to the students, with greater attention to explaining WCAG and its main definitions. To finish the defining stage, we showed software interfaces to the students, and we could together give opinions on what design problems could exist on those interfaces and what accessibility characteristics would or would not be met by them. We also encouraged the students to reflect on how the guidelines and the problems on the interfaces we presented related to their group's chosen accessibility characteristic.

The objective of the process developed in the first class and with the first two stages of Design Thinking was to generate, in the students, a general understanding of the problem caused by neglecting accessibility in software, whom it affects, and what tools software development professionals have to help solve them.

4.2 Second Class

In the second class, students developed the ideating stage of Design Thinking. However, before starting the development of group activities, the students received a short lecture on how accessibility should be regarded in the software development process. In this lecture, we presented an outline of how the W3C recommends that software accessibility be addressed in organizations, being always addressed from the very beginning and at all stages of development [17]. We also presented to the students an overview of how accessibility can be embedded in each of the main stages of the software development process: analysis, design, implementation, testing, deploying, and maintenance.

After that, we returned to the current stage of Design Thinking to be developed. Ideation is the stage of Design Thinking to concentrate on idea generation. It represents a process of brainstorming concepts and outcomes, providing the source material for building the prototypes in the next stage. As explained in the first class, each group would be responsible for developing one accessible software, and each group could freely choose the software's nature and functionalities. For this stage, the groups had 45 minutes to brainstorm ideas and define the following topics about the accessible software they would develop: main purpose of the software, for which platform it would be developed, the technological necessities of people with the group's chosen accessibility characteristics, design decisions to avoid when considering this accessibility characteristic, and the technological preferences of people who need this accessibility characteristic.

We explained to the students that the development of software focused on only one accessibility characteristic is not ideal for real-world software (except in specific cases) and that the restriction we proposed in which each software only meets the needs of one accessibility characteristic was only to reduce the scope for the development of activities within just four classes. After the informed time, each group should present their ideas and definitions to the whole class to disseminate knowledge about each accessibility characteristic to all students.

The objective of the second class was to allow students to better understand how the accessibility concerns they learned in the previous stages relate to the requirements of software development. This process elucidates how to translate the needs of different people into software development definitions.

4.3 Third Class

In the third class, we advanced to the prototyping stage of Design Thinking. This stage's objective is the development of a mock-up of what is to be implemented in the future; prototypes are built so that the designers can think about their solutions in a tangible way, allowing for quick and cheap fixes for any mistakes that may appear. Again, some concepts were presented to the students before starting the activities. Through a short lecture, we introduce students to the concept and importance of prototyping and the different ways software can be prototyped.

After that, the students were instructed to implement a low-fidelity prototype of the main screens and functionalities of the software they had ideated in the previous class. Students had one hour to develop the prototyping activity, which could be done on paper or using a computer. For groups that chose paper prototyping, we offered paper and colored pens.

At the end of the time given, each group presented their prototypes to the whole class and could comment on the decisions they made for their software. Figure 2 shows some of the prototypes developed by the groups. Students in the other groups were encouraged to ask questions about the design defined by the group presenting. As in the other stages, the sharing at the end of the activities aimed to disseminate knowledge about all the accessibility characteristics chosen by the groups with all students.

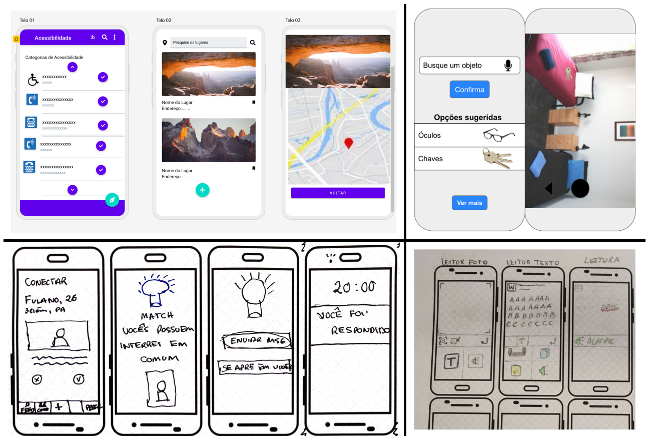


Figure 2: Examples of prototypes developed by the groups.

In this stage, students were able to explore more how accessibility concepts are translated from the conceptual to the interface itself and what considerations should be taken to design interfaces with accessibility concerns.

4.4 Fourth Class

The last stage of Design Thinking is the testing stage. This stage's objective is to understand the product and its users as deeply as possible by testing what was developed with prospective users of the design. Before starting the testing activity, students received a short lecture on testing. In this lecture, students were presented with the different kinds of tests that can be performed in software, the concept of personas, how to define a test scenario, and how to perform the tests.

For this stage, students should define a persona, a testing scenario, and a test script to carry out an evaluation with paper prototyping. In order to perform a paper prototype evaluation, the test subject is presented with an interface drawn on paper and simulates interactions with the prototype by speaking, making gestures, or writing their intentions to interact with the system. The evaluator acts by simulating the execution of the system, expressing reactions to the user's actions, and directing him to the next step. This technique was chosen to be developed by the students at this stage to accommodate the planning and execution of tests in just one class and allow the software to be tested without actually being implemented.

Each group should perform three tests, and the test subjects were members of other groups. For each test performed, the groups should formally document the problems encountered by users during the tests and a list of the lessons learned after each test. Upon completion of all tests, students should turn in the test planning, the problems encountered, and the lessons learned to the teacher. For this stage, there was no presentation of results to the whole class since the students already had to go through other groups as testers during the activity.

This class aimed to show students the importance of testing the interfaces they developed and how the design of interfaces is an iterative process. In the lessons learned the groups turned in, it was common for them to perceive how the external vision can be different from the internal vision of those who participated in the development. At the end of the class, we explained to the students that when developing for accessibility, it is imperative that a user that needs that adaptation test the software, as only they can attest to the quality of what was developed.

5 RESULTS

A total of 33 students participated in the experiment and answered all four questionnaires: Demographic, Awareness before and after classes, and Opinion. We analyzed the data from the Awareness and Opinion questionnaires.

5.1 Awareness Questionnaire

Responses to the awareness questionnaires were analyzed, and each participant received an awareness score. The lowest possible score was 0, and the highest was 88. The student's awareness scores were considerably higher in the questionnaire applied after classes ($Mean = 49.7, Median = 50, StDev = 14.7$) than in the questionnaire applied before classes ($Mean = 35, Median = 33, StDev = 11.4$). A Box Plot comparing the students' awareness scores before and after the classes is shown in Figure 3.

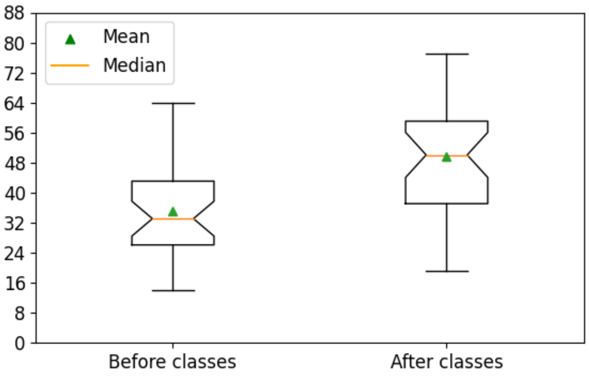


Figure 3: Box Plot showing the distribution of students' awareness scores before and after classes.

To analyze the statistical significance of the results, we first tested the normality of the two sets of samples utilizing the Shapiro-Wilk test. The p-values obtained were p=0.57 for the questionnaire applied before classes and p=0.70 for the questionnaire applied after classes. Since both $p - values > \alpha$ (for $\alpha = 0.05$), we accepted Shapiro-Wilk's null hypothesis, and together with the visual confirmation provided by their Normal Q-Q plots, the data is assumed to be normally distributed.

The Student's t-test for dependent samples was applied based on that analysis, given that our sets met all other assumptions for this test. This parametric test is used when comparing repeated measurements on the same subjects before and after an intervention, and it shows how significant the differences between the sets' means are. The results ($t = 5.31, p = 0.000008$) allowed for the rejection of the test's null hypothesis, as $p < \alpha$ for $\alpha = 0.05$, which shows that the increase in the student's awareness scores was statistically significant and that the teaching methodology applied was successful in substantially increasing the students' accessibility awareness scores.

5.2 Opinion Questionnaire

The participants also answered, after classes, a questionnaire that sought to explore their opinions about the applied teaching methodology. This questionnaire had eight questions on a 5-point Likert scale, where students could answer each question from 1 to 5, for Strongly Disagree and Strongly Agree, respectively, and two open-ended questions.

In the first question, the students answered how much they agreed that the applied teaching methodology aroused their curiosity about the topic of accessibility. Of the 33 respondents, 69.7% (23) answered 5 on the Likert scale, indicating a strong agreement with the statement. Moreover, 24.2% answered 4, and 6.1% answered 3. The second question explored whether students agreed that the applied teaching methodology was interesting and held their attention. Of the 33 respondents, 51.5% (17) answered 5 on the Likert scale, indicating they strongly agree with the statement, and 42.4% (14) answered 4. The rest of the students (6.1%) answered 3.

We explored with the third question whether students believed they knew more about accessibility after receiving the classes. Of the 33 respondents, 51.5% (17) answered 5 on the Likert scale, indicating they strongly agree with the statement, and 36.4% (12) answered 4. The rest of the students (12.1%) answered 3. When asked in the fourth question if they were more likely to consider accessibility in the software projects they develop in the future, 72.7% (24) of the students answered 5 on the Likert scale, indicating they strongly agree with the statement. Moreover, 21.2% responded 4, and 6.1% responded 3.

Responses to the fifth question, which explored whether students agreed that after the classes, they were more likely to consider accessibility as a field of research, were more widespread, as shown in Figure 4a, with the majority of participants (36.4%) answering 3 on the Likert scale. The sixth question explored whether students agreed that they were interested in deepening their accessibility knowledge after classes. Figure 4b shows the distribution of answers, with the majority of participants (36.4%) answering 5 on the Likert scale.

For the seventh question, the students answered how much they agreed they are more sensitive to the difficulties faced by people who need accessibility after classes. Of the 33 respondents, 75.8% (25) answered 5 on the Likert scale, indicating a strong agreement with the statement. Moreover, 18.2% answered 4, and 6.1% answered 3. The eighth question explored whether participants agreed that software accessibility should be a separate subject offered in the curriculum of computing courses. Most participants (66.7%) answered 5 on the Likert scale, indicating a strong agreement with the statement. However, the rest of the answers were widespread, as shown in Figure 4c.

The ninth question was an open and non-mandatory question where students could expose what they thought could have been added to the classes to improve them. Only ten students answered this question, and five indicated that they missed direct contact with a person with a disability who could report more about their first-hand experiences to the class. Two students answered that they would have liked to have a moment in a laboratory to test accessibility software, such as screen readers, to better understand how it works. The other three students answered that they would like the lecture moments to explain in more depth about accessibility guidelines, prototyping techniques, and more examples of designs that meet or violate accessibility guidelines.

Finally, the tenth question was also an open and non-mandatory question where students could present criticism, compliments, observations, or any other comment they wanted about the accessibility classes. Only eight students answered this question. All responses complimented the initiative and the methodology employed, recognizing the importance of teaching accessibility. One of the students also added, "[...] (the classes) brought me more empathy about the topic addressed, now every time I start prototyping something I already think about how I can make it more accessible".

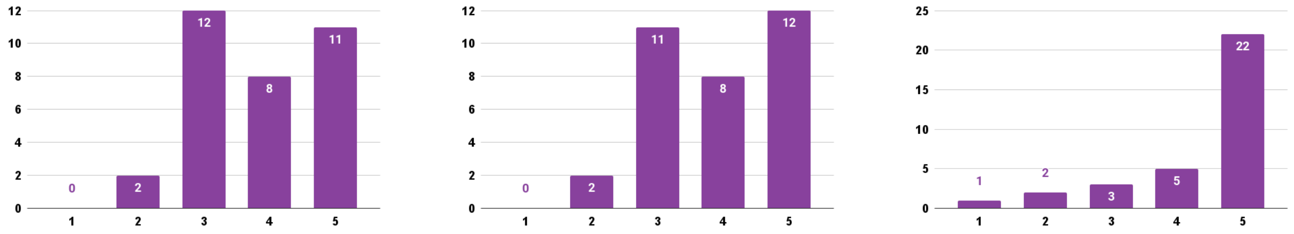
6 DISCUSSION

The main objective of this work was to develop an easily reproducible accessibility teaching methodology based on Design Thinking to be utilized in college and university courses and to evaluate its effectiveness in raising awareness of accessibility in students.

The results of the analysis of the awareness questionnaires applied before and after the classes indicated that there was a significant increase in the awareness of the students who participated. This increase shows that the proposed methodology achieved the intended objective and that after the classes, the students became more sensitive to accessibility issues than before the classes.

The students' perspectives were explored through a questionnaire that collected their opinions on the classes. Regarding the applied teaching methodology, most students indicated that it successfully increased their knowledge of accessibility, awakened their curiosity about accessibility, and held their attention to the content during classes. Most students also answered that after the classes, they are more likely to consider accessibility in the projects they will develop, that they feel more sensitive to the issue of accessibility and that they think accessibility in software should be a dedicated subject in the curriculum. This questionnaire was essential to explore the students' receptivity to the proposed teaching methodology and how it changed their future intentions regarding accessibility, as the methodology's effectiveness goes beyond just confirming the increase in the class's awareness. It also involves knowing if the students are satisfied with the classes, if the classes are successfully engaging their attention, and if their intentions regarding accessible software development evolved positively.

In this same questionnaire, some students indicated that they would have liked to have received more in-depth lectures on accessibility guidelines and prototyping methods and would have liked to explore, hands-on, the use of accessible software. The content presented was limited by the number of classes available for its realization, as the accessibility classes were just a portion of an HCI course. The flexibility of using Design Thinking allows each of its stages to be extended to the desired duration by those who apply the proposed methodology. In a reproduction of the methodology



(a) Responses to whether participants agreed to be more likely to consider accessibility as a field of research.

(b) Responses to whether participants agreed that they were interested in deepening their accessibility knowledge.

(c) Responses to whether participants agreed that accessibility should be a separate subject in the computing courses.

Figure 4: Answers to the fifth, sixth, and eighth questions of the opinion questionnaire after the classes. Likert scale, 1 for Strongly Disagree, 5 for Strongly Agree.

with a more extensive duration, the matters raised by the students may be better explored.

7 FINAL CONSIDERATIONS

This study presented and evaluated an easily reproducible accessibility teaching methodology based on Design Thinking that can be used to raise accessibility awareness among students. The results showed that the methodology was effective and that the students considered it to be engaging and enjoyable. The objective of training more professionals to be sensitive to accessibility issues is that their participation in the software development process might help improve the number of accessible software available.

Many people depend on software to provide accessibility options to fully use all of its features. Furthermore, accessibility should be treated as an essential requirement in the development of any software product. However, much progress is still needed to achieve this, as many challenges still exist in the software development industry. Making more accessible software is a multi-faceted challenge that involves changes in many fields, and previous studies found that these changes range from educational to corporate and even legal. Finding ways to mitigate these problems is essential to increase the availability of accessible software.

One of the biggest threats to the validity of this study is that the results were evaluated soon after the end of the classes. An evaluation with more time after the end of the classes could attest better to the long-term retention of the knowledge that the students obtained. Another limitation of this work is that it did not have a control group. In similar conditions, studies [10] found that even without receiving classes directly focused on accessibility, students could become more knowledgeable about accessibility issues simply by receiving the HCI classes that already teach usability heuristics and general concerns regarding software design.

For future work, we intend to accompany the students who participated in the first cycle of application of the methodology and reapply the awareness questionnaire again to attest to long-term knowledge retention. Also, we intend to carry out is to use the same proposed methodology but for accessibility training within organizations to reach professionals who have already graduated.

REFERENCES

[1] Peatworks. The accessible technology skills gap - peatworks, December 2017. URL https://www.peatworks.org/policy-workforce-development/the-accessible-technology-skills-gap/.

[2] Andre P. Freire, Cibele M. Russo, and Renata P. M. Fortes. A survey on the accessibility awareness of people involved in web development projects in brazil. In *Proceedings of the 2008 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, W4A '08, page 87–96, 2008. doi: 10.1145/1368044.1368064.

[3] Cara Peters and David A Bradbard. Web accessibility: An introduction and implications for a corporate social responsibility marketing strategy. *Journal of Internet Commerce*, 6(4):27–54, 2007.

[4] WebAIM Million. The 2022 report on the accessibility of the top 1,000,000 home pages, 2022.

[5] Alan Trindade de Almeida Silva, Luciano Arruda Teran, Thayssa Águila da Rocha, and Marcelle Pereira Mota. A study on accessibility aspects of software development in brazil. 2022. doi: 10.5748/19CONTECSI/PSE/ESD/6924.

[6] Luciano Arruda Teran, Alan Trindade de Almeida Silva, Giselle Lorrane Nobre Melo, and Marcelle Pereira Mota. A study on discovering accessibility issues in the software development process. 2021. doi: 10.5748/18CONTECSI/PSE/ESD/6849.

[7] Apple. Accessibility programming guide for os x: Testing for accessibility on os x. https://developer.apple.com/library/archive/documentation/Accessibility/Conceptual/AccessibilityMacOSX/OSXAXTestingApps.html, 2018. (Accessed on 10/12/2022).

[8] Google. Test your app's accessibility | android developers. https://developer.android.com/guide/topics/ui/accessibility/testing, 2022. (Accessed on 10/12/2022).

[9] Yasmine N El-Glaly. Teaching accessibility to software engineering students. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 121–127, 2020.

[10] Qiwen Zhao, Vaishnavi Mande, Paula Conn, Sedeeq Al-khazraji, Kristen Shinohara, Stephanie Ludi, and Matt Huenerfauth. Comparison of methods for teaching accessibility in university computing courses. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–12, 2020.

[11] Nidhi Rajendra Palan, Vicki L Hanson, Matt Huenerfauth, and Stephanie Ludi. Teaching inclusive thinking in undergraduate computing. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 399–400, 2017.

[12] Kristen Shinohara, Cynthia L Bennett, Jacob O Wobbrock, and Wanda Pratt. Teaching accessibility in a technology design course. Philadelphia, PA: International Society of the Learning Sciences., 2017.

[13] Catherine M Baker, Yasmine N El-Glaly, and Kristen Shinohara. A systematic analysis of accessibility in computing education research. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 107–113, 2020.

[14] Rim Razzouk and Valerie Shute. What is design thinking and why is it important? *Review of educational research*, 82(3):330–348, 2012.

[15] Helen Petrie and Alistair Edwards. Inclusive design and assistive technology as part of the hci curriculum. In *Proceedings of HCI Educators Workshop*, volume 2006, pages 23–24, 2006.

[16] Brian J Rosmaita. Accessibility first! a new approach to web design. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 270–274, 2006.

[17] W3C. Start with accessibility - education and outreach. https://www.w3.org/WAI/EO/wiki/Start_with_Accessibility, 2014. (Accessed on 10/07/2022).