# Fostering Collaboration through Design Thinking: A Study among Software Engineering Students


Dariane Abich
darianeabich@gmail.com
Federal Institute Farroupilha
São Borja, RS, Brazil

Rafael Parizi
rafael.parizi@iffar.edu.br
Federal Institute Farroupilha
São Borja, RS, Brazil

Sabrina Marczak
sabrina.marczak@pucrs.br
PUCRS
Porto Alegre, RS, Brazil



## ABSTRACT

Design Thinking has been recognized for its potential to enhance critical thinking and collaboration skills, particularly in the initial stages of software development. However, despite its benefits, the effectiveness of Design Thinking can be reduced by insufficient engagement from professionals. Thus, it is crucial to foster an integrated practice environment that cultivates both technical competencies and interpersonal skills, such as collaboration, to better prepare students for the complexities of the field. This study investigates the role of Design Thinking in promoting collaborative skills among software engineering students through a case study involving 22 undergraduates engaged in a Design Thinking project over an 18-weeks long semester. Our analysis of interactions, project artifacts, and questionnaires using the 3C collaboration model (Communication, Coordination, Cooperation) pointed out that the iterative, user-centered process of Design Thinking fostered teamwork, idea sharing, and collaborative problem-solving. Our findings indicate that integrating Design Thinking into computing curricula could effectively prepare future software engineers with the dual competencies—technical and collaborative skills—required in professional life. Furthermore, the study offers practical recommendations for educators on implementing Design Thinking to enhance student collaboration. These insights also have broader implications for industry and researchers, providing a framework for applying Design Thinking principles to foster teamwork and problem-solving skills in professional and academic settings. Future studies should explore the long-term impacts of Design Thinking on students' professional performance and investigate additional strategies to enhance coordination within collaborative projects.



## CCS CONCEPTS

• **Human-centered computing → Empirical studies in collaborative and social computing**.

## KEYWORDS

Software Engineering, Collaboration, Design Thinking, Skills, Education


## 1 INTRODUCTION

Education in software engineering has evolved over the years, with an increasing acknowledgment of the importance of developing not only technical skills but also interpersonal abilities [13]. Collaboration is among the most relevant abilities, particularly during the software conception where collective problem understanding and solving are key. Design Thinking (DT) emerges as an innovative approach, offering an iterative, user-centered framework that fosters critical thinking and collaboration [3, 10, 16].

Numerous previous studies have employed DT in teaching software engineering [20, 21]. However, most of these studies primarily focus on teaching the approach and techniques of DT itself, rather than using DT as a tool to facilitate and explore student collaboration [12]. Collaboration is critical in software development, where the ability to work effectively in teams can be as important as technical skills. Thus, this study aims to fill this gap, exploring how DT can be employed not only as part of the syllabus but also as a means to enrich the educational process, especially in terms of developing collaborative skills.

In this study, we posed the following research question: "How does Design Thinking contribute to the development of collaborative skills among Software Engineering Students?". Our investigation is motivated by the need to explore teaching methodologies that develop interpersonal competencies alongside technical skills in computing students. The study focuses on the application of the Double Diamond model of DT [6], comprising the Discover, Define, Develop, and Deliver phases.

We adopted a case study methodology involving 22 undergraduate students from a Brazilian Educational Institution, focusing on a Software Engineering course. The research spanned an 18-weeks long semester, during which we engaged students in a Design Thinking-based project, closely observing their interactions and analyzing various produced artifacts. Our data collection comprised both qualitative analyses of these artifacts and structured questionnaires administered at the beginning and end of the semester. We used the 3C collaboration model (Communication, Coordination, Cooperation) [7] to evaluate how students collaborated throughout the project. This mixed-procedures approach was chosen to gain comprehensive insights into the impact of Design Thinking on the development of collaborative skills in software conception.

By assessing the role of DT on students' collaborative skills in a real learning context, this study seeks to provide valuable insights into how DT can be integrated into the curriculum of computing courses. This way, our research contributes to the body of knowledge on innovative teaching methodologies in the field, highlighting collaboration as an essential competency for future software engineering professionals. Additionally, the study offers practical recommendations for educators on how to implement DT to enhance student collaboration. These findings also have broader implications for industry and researchers, providing a framework for applying DT principles to foster teamwork and problem-solving skills in professional and academic settings.

The remainder of this paper is organized as follows: Section 2 provides the background on DT in software development, studies on DT for SE education, collaboration in software development, and identifies a gap in the literature. Section 3 details our research approach, research questions, participant engagement, execution procedures, team formation, data collection strategies, and data analysis procedures. Section 4 presents the initial results, artifact analysis, and assessment questionnaires. Section 5 discusses these findings and their implications. Section 6 addresses the potential threats to the validity of our study. Finally, Section 7 draws conclusions from the study, highlighting implications for industry and researchers and suggesting avenues for future work.

# 2 BACKGROUND

## 2.1 Collaboration and Design Thinking in Software Development

Software development is an intensive process that involves various stages, each requiring collaborative skills to foster idea exchange, problem-solving, and innovation [2, 23]. Effective collaboration enhances communication and coordination, reducing errors and misunderstandings. Communication is the exchange of information and ideas among participants. Coordination involves managing people, activities, and resources to align with the group's objectives. Cooperation is the combined effort to execute tasks and achieve common goals [8]. These skills enable team members to leverage each other's strengths and collectively contribute to the successful completion of software projects.

Design Thinking has been increasingly used to foster this kind of collaboration by providing a user-centered, iterative framework. DT is often used in a variety of fields, including design, business, and software engineering. In software development, DT can be used to improve the quality of products and services, as well as the development process itself. DT helps software engineers better understand user needs, generate innovative ideas, and collaborate effectively [17].

Studies in literature have investigated the benefits of using DT in software development. For instance, Pereira *et al.* (2021) [17] pointed out that using DT might be of help for: (i) *Better understanding of user needs:* DT emphasizes empathy with users, which can help software engineers better understand the needs and desires of users; (ii) *Generation of innovative ideas:* DT encourages creativity and experimentation, which can lead to the generation of innovative ideas for products and services; (iii) *Improved collaboration:* DT emphasizes collaboration between different disciplines and areas, which can help improve the software development process.

## 2.2 Studies on DT for SE Education

Studies have been conducted on the use of DT for software engineering education. These studies have shown that DT can be an effective tool for developing skills and competencies relevant to software engineering. Table 1 summarizes some of the studies that have investigated the use of DT for software engineering education.

Silva *et al.* (2017) [14] discusses a human-centric software engineering capstone course aimed at balancing course scope and depth to meet the dynamic needs of the industry. Offered to 29 students, the course incorporated DT techniques and agile practices into the project life-cycle. Results indicated that DT was effective in requirement elicitation, software design, and testing, and also encouraged student self-direction, which increased motivation and led to a zero dropout rate. The study concluded that DT is an effective problem-solving tool in SE education, fostering a more hands-on, problem-based curriculum.

Braz *et al.* (2019) [4] investigate the application of DT integrated with Scrum principles in Software Requirements Elicitation at a Brazilian state university. The aim was to develop a system for efficient resource allocation and reservation. Involving a diverse group of stakeholders, the adapted DT approach effectively addressed the challenges of manual resource management processes. The result was the design of solutions that not only met 100% stakeholder satisfaction but also achieved 95% average completeness. This indicates the potential of DT, combined with Agile methodologies, as a powerful tool in software development, warranting further exploration in more complex and extensive applications.

Souza *et al.* (2018) [21] report an experience on teaching DT concepts and techniques in two different Software Engineering courses. In both courses, 15 distinct DT techniques were introduced to support requirements elicitation. In the first course, the DT concepts and techniques were traditionally presented using available materials. In contrast, the second course not only introduced these concepts and techniques but also utilized a tool that recommends DT techniques and a repository with materials on the techniques. Practical assignments in both courses involved using these techniques for eliciting requirements for new systems. Lessons learned from the first course's presentation were applied in the second, such as dividing the presentation into two classes and incorporating more practical activities throughout. The use of the DT technique recommendation tool and the materials repository was a lesson learned to guide students in selecting techniques and reinforcing classroom learning. This approach enabled students to use a broader range of techniques more effectively.

Corral and Fronza (2018) [5] explore the use of DT and Agile methodologies as instructional approaches in undergraduate Software Engineering courses focused on creating innovative software products from scratch. The study analyzes the similarities and differences between DT and traditional software development processes, particularly Agile Practices. It compares the methods, artifacts, and final products produced by students using Agile and DT in two different educational settings. The study found that while both methodologies are effective in managing software projects, they introduce distinct characteristics in software development and change the nature of the final product. Agile is noted for its rigor in strict Software Engineering aspects, whereas DT encourages more innovative and creative outcomes. The study suggests a need for further research to confirm the direct influence of each methodology on the project outcomes and to examine other variables in the two different working environments.

Marques *et al.* (2020) [11] explored how DT provides practical experiences and supports the development of soft skills in SE courses. The authors developed a practice based on the DT phase that simulates practical software project situations. An experimental study with postgraduate students found that DT effectively developed soft skills, linking creativity to Brainstorming and Co-creation Workshops and client expectation management to PATHY and Empathy Maps. As a result, they suggest to be important to investigate the adoption of the dynamic in different contexts, such as undergraduate courses, in both beginner and final year classes, aiming to consolidate the use of DT in software development.

**Table 1: Summary of studies on the application of Design Thinking in Software Engineering Education**

| Work | Application of DT | Results |
|---|---|---|
| [14] | Human-centric software engineering course incorporating DT methods and agile practices | - Effective use in requirement elicitation, design, and testing<br>- Encouraged student self-direction<br>- Zero dropout rate<br>- Fosters hands-on, problem-based curriculum |
| [4] | DT integrated with Scrum for Software Requirements Elicitation | - Efficient resource allocation system developed<br>- 100% stakeholder satisfaction<br>- 95% average completeness<br>- Highlights DT's potential with Agile methodologies |
| [21] | Teaching DT concepts and techniques in Software Engineering courses | - Introduced 15 DT techniques<br>- Use of DT recommendation tool and repository<br>- Effective in eliciting requirements<br>- Enabled broader use of techniques |
| [5] | DT and Agile methodologies in undergraduate Software Engineering courses | - Effective in project management<br>- Agile noted for strict engineering aspects<br>- DT led to more innovative outcomes<br>- Need for further research on methodology impact |
| [11] | DT on developing soft skills among software engineering students | - Enhanced soft skills as problem-solving, teamwork, creativity, and critical thinking<br>- Acted as a valuable instructional tool for fostering essential soft skills in software engineering education |

### 2.3 Gap in Literature

We have studied the use of DT in software development for years [16, 18, 19]. DT aids in understanding user needs and fosters creativity, but it remains challenging, particularly in terms of participant engagement and collaboration [16]. Even facilitators using our recommendation system [15] for appropriate techniques find participant collaboration to be one of the most critical tasks. Therefore, in this study we investigate *"How can Design Thinking contribute to the practice of collaboration between software engineering students?"*. We conducted a more targeted investigation into how Design Thinking can be effectively integrated into software engineering education to not only foster technical knowledge but also to enhance collaborative skills necessary in the software development process.

## 3 METHODOLOGY

### 3.1 Research Approach

Our research involved 22 undergraduate students from a Brazilian university enrolled in a Software Engineering course. We conducted a case study [24] aiming for delving into the complexities and nuances of applying DT in an educational setting to develop collaborative skills.

### 3.2 Research Questions

In this study, we aim to answer the following research question: *"How does Design Thinking contribute to the development of collaborative skills among Software Engineering Students?"*

To do so, we have defined 2 auxiliary research questions:

- AQ1- What are the effects of using Design Thinking on the collaboration among students in terms of their interactions and the quality of the developed products?
- AQ2- What is the participants' perception of practicing collaboration through the use of DT?

### 3.3 Participant Engagement

The participants were students enrolled in a Software Engineering Course that integrated various disciplines. Of the 22 students, 6 work in software development but have no DT experience, ensuring no prior bias. We introduced DT to ensure everyone was on the same page. Thus, we captured a wide range of experiences and insights from these participants throughout the study.

### 3.4 Procedure Execution

Over an 18-weeks long semester, we engaged students in a practical project based on DT principles, in line with Yin's case study methodology (see Figure 1). This involved close monitoring of their interactions, collection of documents and records, and conducting in-depth interviews to gain comprehensive insights into the DT application in their projects.

We adapted the Double Diamond [6], one of the most widely used DT models for software development [16]. The Double Diamond model includes four stages: Discover, where insights are explored and gathered; Define, where the problem is identified and clarified; Develop, where solutions are generated and refined; and Deliver, where the final solution is implemented and tested. In addition, to get all students on the same page about DT in software development, we conducted the first 2 weeks with initial training activities.

Table 2 shows the 18-weeks activities plan we executed. It includes the week ID, the activities, and the artifact delivered in a certain week, if any.

### 3.5 Team Formation

Participants enrolled in the course came from different levels of the program. They started by attending initial classes to ensure everyone was on the same page about DT in software development.

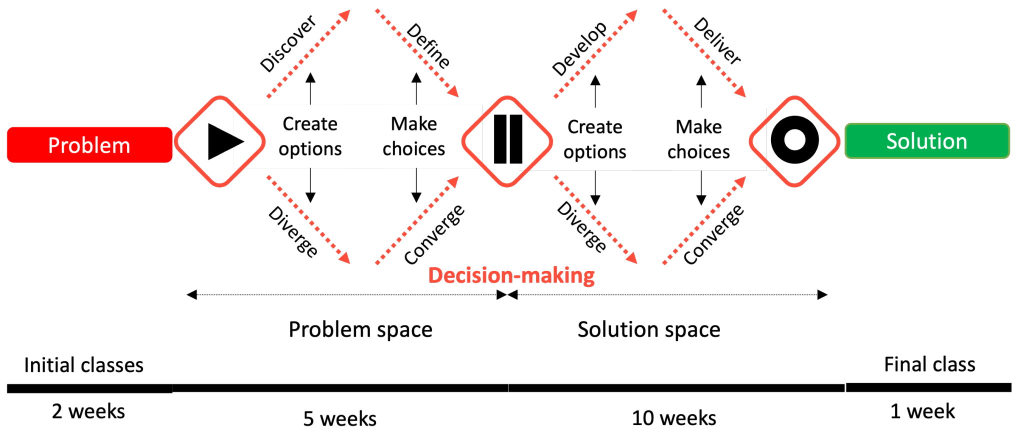


**Figure 1: Activities for a 18-weeks long semester using DT**

**Table 2: Weekly Activity Plan**

| Week | Activity | Deliverable |
|---|---|---|
| 1 | Initial presentation of the course, objectives, concepts of integrated project | - |
| 2 | Introduction to Design Thinking in Software Development | - |
| 3 | Team organization, computational solutions conception, introduction to Design Thinking | Team formation |
| 4-6 | Discovering the problem: Research and information gathering | Information collection |
| 7 | Defining the problem, scope, objectives, initial version of the executive presentation | Initial executive presentation, activity diary |
| 8 | Assessment I: Public presentation of the identified problem | Problem presentation |
| 9-13 | Developing the solution: Brainstorming, prototyping, project progress review | Prototypes, activity diary |
| 14-15 | Delivering the solution: Feedback collection and analysis | Feedback analysis, activity diary |
| 16 | Developing/Revising the solution: Prototyping and review | Prototype revisions, activity diary |
| 17 | Finalizing presentations and documents, creating a customized poster | Final executive presentation, poster |
| 18 | Solution presentation, feedback and discussion, technology suggestions | Evaluation of executive presentation, activity diary, poster |

**Table 3: Teams formation**

| Group | Product Name | Number of Participants |
|---|---|---|
| G1 | ReciclaTech | 5 |
| G2 | Action Food | 5 |
| G3 | Telemed SUS | 6 |
| G4 | TechThinkers | 6 |

The 22 participating students organized themselves into 4 groups based on affinity. These groups are represented in this study as G1 to G4 (see Table 3).

### 3.6 Data Collection Strategy

We collected data in order to answer our research questions using two primary instruments:

- *Artifact Analysis:* We qualitatively analyzed the artifacts produced by students, including reports and prototypes. We invited a Project Manager with 5 years of experience in software development, DT and collaboration (Question AQ1). She validated our collaboration table (Table 2) and assessed the groups based on the produced artifacts (activity log entries in Google Docs[1] and Overleaf[2], prototypes in Figma[3], and student interactions during classroom presentations—knowledge of the subject, presentation style, and peer interaction).
Additionally, we applied the 3C model to further analyze the artifacts, focusing on i) Communication: exchanging information between team members, ii) Coordination: organizing tasks and activities for efficient workflow, and iii) Cooperation: working together to achieve common goals. Based on that, we evaluated the 3C elements as follows:
  – Communication: amount of message exchanges via Google Docs, in-class discussions, and interactions during presentations in Weeks 8 and 18;

[1]https://docs.google.com
[2]https://overleaf.com
[3]https://figma.com

   - Cooperation: Active participation in artifact creation (Google Docs, Figma, Overleaf editing histories);
   - Coordination: Google Docs comments requesting improvements or edits.

  We monitored the students' performance and indicated high values for communication, cooperation and coordination for the groups that interacted the most, serving as a baseline. In this study, we did not analyze the specific techniques used, as this is the subject of a separate investigation. Our findings indicate that selecting DT techniques is a gap in the literature [16, 19]. Although we developed a recommendation system to aid in decision-making, this process remains challenging. Therefore, the groups in our study used techniques suggested by our system [15].
- *Assessment Questionnaire:* To assess students' perspectives on collaboration and the influence of DT on their skills (Question AQ2), we invited the teams to collaboratively respond to 4 initial questions in a questionnaire (Table 4).

## 3.7 Data Analysis procedure

Following a qualitative approach, we employed Content Analysis [9] to evaluate the gathered information, allowing for an in-depth understanding of the dynamics of student collaboration under the influence of DT.

## 3.8 Rationale Behind Method Selection

Adopting a mixed-methods approach, we combined artifact analysis and questionnaires to gain a comprehensive view of the impact of DT on student collaboration. This approach uncovered tangible collaboration outcomes through artifact analysis and gather subjective experiences and perceptions through questionnaires.

# 4 RESULTS

This section presents the results of applying DT as a means to explore collaboration among students in software development. Initially, Section 4.1 shows general outcome data, including the formed groups and participating students. In Section 4.2, we present results aimed at answering AQ1, which relates to the evaluation of the produced artifacts. Finally, in Section 4.3, we present the evaluation of students' perceptions regarding the effect of DT on collaboration through a questionnaire to answer AQ2.

## 4.1 Initial Results Data

Table 5 shows the problems identified by each team, the proposed solutions, and the DT techniques used over the 18-weeks long semester. The groups aimed to develop computational solutions that meet the collective needs of the population. The participants considered addressing the United Nations' Sustainable Development Goals (SDGs) [22]. According to the United Nations, the SDGs are *"hand-in-hand with strategies that improve health and education, reduce inequality, and spur economic growth – all while tackling climate change and working to preserve our oceans and forests"*.

As a result, the groups addressed 4 main collective needs:

G1) Recycling of technological products: focusing on developing innovative solutions to enhance the recycling processes of technological products, aiming to reduce electronic waste and promote environmental sustainability;
G2) Combating food waste: dedicated to implementing strategies that can effectively minimize food waste throughout the supply chain, from production to consumption;
G3) *Optimization of urgent and emergency care*: aiming to optimize urgent and emergency care services through advanced systems and technologies, ensuring faster and more efficient responses to critical medical situations;
G4) Combating digital crimes: focusing on developing comprehensive measures and tools to combat digital crimes, enhancing cybersecurity and protecting individuals and organizations from cyber threats.

## 4.2 Artifact Analysis

**Auxiliary Question AQ1**

What are the effects of using Design Thinking on the collaboration among students in terms of their interactions and the quality of the developed products?

Figure 2 illustrates some software prototypes that the groups delivered as part of the artifacts produced. In addition to the prototypes, the groups produced posters for executive presentations and reports of interviews and questionnaires conducted for data collection on the problem space (in DT) and for collecting feedback (in DT's solution space)[4].

The produced artifacts were assessed in 2 stages, according to the schedule of the course presented in Table 2. An industry professional contributed to the evaluation. The evaluation consisted of an analysis of the products in terms of elements of collaboration such as Communication, Cooperation, and Coordination [7]. She considered message exchanges via Google Docs, in-class meetings, and presentations (communication); participation in artifact construction – Google Docs, Figma, Overleaf (cooperation); and task organization via Google Docs comments (coordination).

Table 6 shows the assessment matrix of the collaboration elements, evaluating Communication, Coordination, and Cooperation. Each element is assessed qualitatively, using the indices Low, Medium, and High. The descriptions within each cell provide the meaning of each index in the context of the respective collaboration element. In summary, the assessment used the indices Low, Medium, and High, where:

- Low: Indicates poor or insufficient levels of collaboration;
- Medium: Indicates moderate or satisfactory levels of collaboration;
- High: Indicates strong or effective levels of collaboration.

Based on the data provided in the table and the definitions of the 3Cs (Communication, Coordination, and Cooperation) [7]:

- *G1 (High Communication, Low Coordination, High Cooperation)*

[4]Some artifacts feature text in Portuguese, as they were developed by Brazilian students.

**Table 4: Questionnaire on DT application for collaboration**

| # | Question | Type | Options/Notes |
|---|---|---|---|
| Q1) | How would you assess the use of Design Thinking to foster collaboration? | Closed | Likert Scale: Excellent, Good, Average, Poor |
| Q2) | What are the positive aspects of Design Thinking for exploring collaboration? | Open | - |
| Q3) | What difficulties did you perceive during the application of Design Thinking? | Open | - |
| Q4) | What improvements would you suggest for the next application? | Open | - |

**Table 5: Group Challenges, Solutions, and Techniques**

| Group | Problem | Solution | Techniques Used |
|---|---|---|---|
| G1 - ReciclaTech | Electronic waste is improperly discarded with regular trash due to lack of knowledge about electronic recycling and misinformation about disposal sites | App to inform about waste collection points, how to collect, and home collection alert. | Solution matrix, questionnaires, interviews, CSD matrix, prototyping, brainstorming [16] |
| G2 - Action Food | Food waste encompasses losses in agricultural production due to adverse conditions and standards, and inefficiencies in retail and dining establishments through overpurchasing and poor management | An app that maps places in a city where people can find or offer food at lower prices, addressing food waste and affordability | Interviews, Desk Research, Prototyping [16] |
| G3 - TelemedSUS | The excessive wait times in Emergency Rooms, where may cases are identified as non-urgent, leading to inefficient use of resources and potentially delaying care for those with genuine emergencies | An app that enables quick online connection with specialist doctors, providing guidance on how and where to seek specialized medical help | Prioritization matrix, Value hypothesis, Idea frame, Storyboard, and User journey [16] |
| G4 - TechThinkers | Digital scams harm society due to low computer literacy. Furthermore, scams replicate websites to gain trust with promises of quick returns, prizes, or significant discounts on products | An app and browser extension that assesses the credibility of websites by indicating whether they are trustworthy or not, based on reviews from other users | Interviews, Desk Research, Prototyping [16] |

**Table 6: Assessment matrix of collaboration elements**

| Element | Low | Medium | High |
|---|---|---|---|
| Communication | Students rarely shared updates or ideas with each other, leading to misunderstandings and a lack of alignment on project goals | Students communicated moderately, holding weekly meetings and using a group chat for occasional updates, ensuring basic information exchange | Students maintained constant communication through stand-up meetings and a dedicated communication channel, regularly sharing progress and brainstorming solutions |
| Coordination | The group struggled to organize tasks and often missed deadlines, with no clear division of responsibilities, causing delays and inefficiencies | The group had a basic level of coordination with some assigned roles and a general timeline, but occasional overlaps or gaps in task management | The group effectively used project management tools to assign tasks, set deadlines, and track progress, holding regular check-ins to ensure everyone was on track |
| Cooperation | The group members worked largely independently and rarely collaborated on tasks, resulting in a disjointed final product | The group cooperated satisfactorily, working together on key tasks and supporting each other when needed, with periodic collaboration sessions to integrate their work | The group exhibited strong cooperation by working closely together throughout the project, frequently pairing up for tasks and helping each other solve problems, leading to a cohesive final product |

**Table 7: Collaboration Matrix**

| | Communication | Coordination | Cooperation |
|---|---|---|---|
| G1 | High | Low | High |
| G2 | High | Low | High |
| G3 | Medium | High | Medium |
| G4 | Medium | Medium | High |

- Communication: G1 was very effective at exchanging messages and ideas, suggesting a highly communicative environment where information flows freely.
- Coordination: The group struggled with organizing people, activities, and resources, indicating potential issues with aligning efforts or synchronizing tasks.
- Cooperation: Despite the coordination issues, the group was highly cooperative, working effectively together in a shared space to execute tasks.

• *G2 (High Communication, Low Coordination, High Cooperation)*
  - Communication: Similar to G1, G2 excelled in communication, indicating strong dialogue and information sharing within the group.

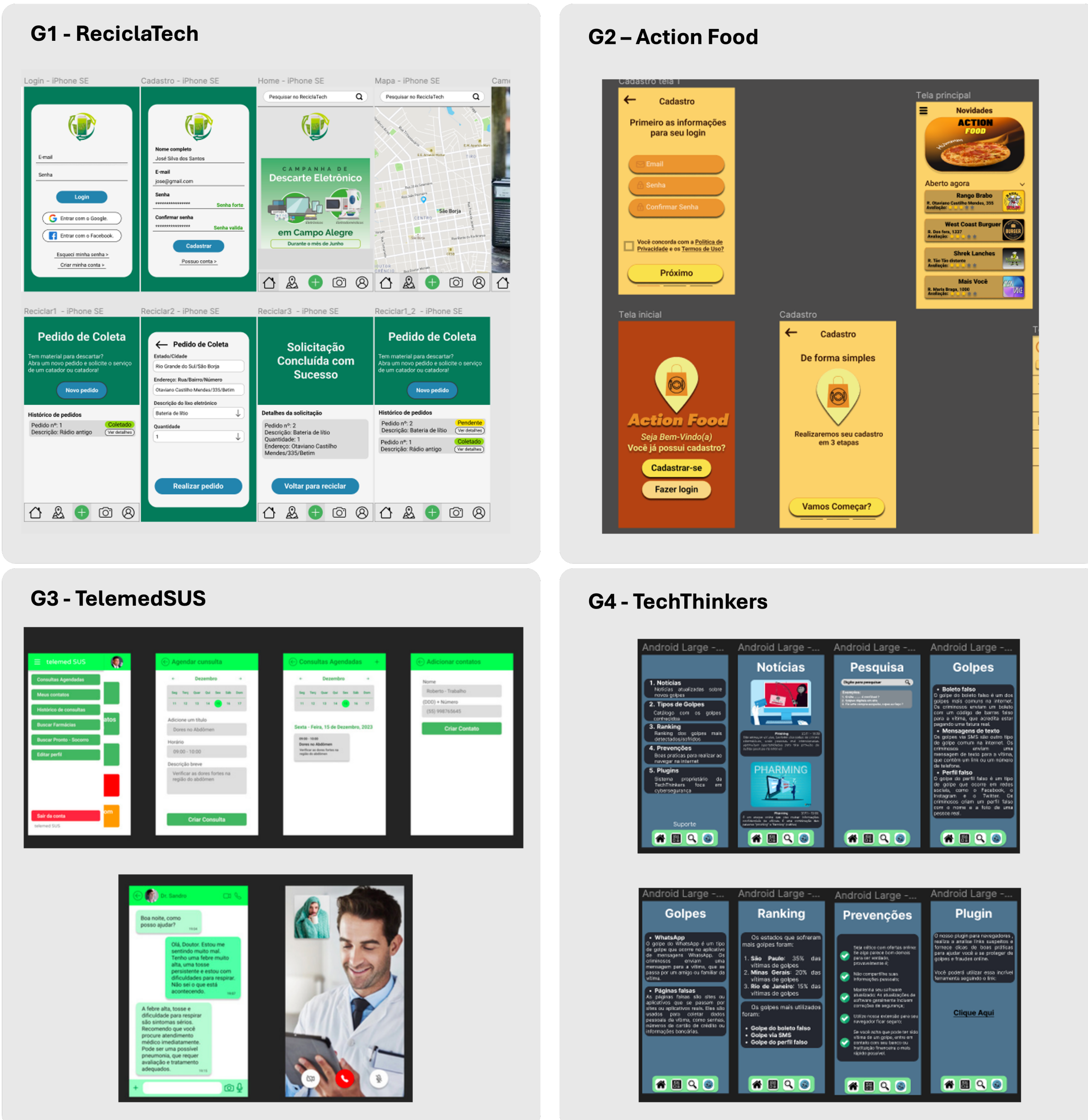


**Figure 2: Teams' Prototypes - artifacts**

  - Coordination: Again, like G1, G2 showed low coordination, which might lead to inefficiencies in managing and aligning the group's activities.
  - Cooperation: The group's ability to work together toward common goals remained high, suggesting that once a direction is set, the group collaborates well.
- *G3 (Medium Communication, High Coordination, Medium Cooperation)*
  - Communication: G3 had a moderated level of communication, indicating a fair amount of message exchange but possibly room for improvement in clarity or frequency.
  - Coordination: This group was strong in coordination, effectively managing tasks, people, and resources, indicating an organized approach to their work.
  - Cooperation: Cooperation was at a medium level, suggesting that while the group works together, it was challenging in joint efforts or sharing a space for task execution.
- *G4 (Medium Communication, Medium Coordination, High Cooperation)*
  - Communication: G4's communication was also at a medium level; they likely communicate sufficiently but might lack

the open and free-flowing exchange of ideas seen in groups G1 and G2.
  - Coordination: Coordination was medium, which points to an adequate level of organizational ability but with potential for improvement in synchronizing their efforts.
  - Cooperation: They score high in cooperation, which suggests that despite some communication and coordination issues, the group worked together very effectively when executing tasks.

The evaluation of prototypes produced by the student groups was based on their levels of communication, coordination, and cooperation. The following analysis details the performance of each group (G1, G2, G3, and G4) in these areas, highlighting how these elements impacted their collaborative efforts and the quality of their developed prototypes.

- **Communication:**
  - G1: G1 was very effective at exchanging messages and ideas, suggesting a highly communicative environment. This high level of communication likely facilitated the generation of innovative ideas and solutions during the prototyping phase.
  - G2: Similar to G1, G2 excelled in communication, indicating strong dialogue and information sharing within the group. This effective communication facilitated a clear understanding of project requirements and creative brainstorming.
  - G3: G3 had a moderate level of communication, indicating a fair amount of message exchange but with room for improvement in clarity or frequency. This level of communication might have lacked the depth needed for solving the problem identified.
  - G4: G4's communication was also at a medium level; they likely communicated sufficiently but might have lacked the open and free-flowing exchange of ideas seen in groups G1 and G2. This limited the group's ability to fully leverage diverse perspectives and insights during prototyping.
- **Coordination:**
  - G1: Despite their strong communication, G1 struggled with organizing people, activities, and resources. This lack of coordination led to inefficiencies and delays in the development process, affecting the coherence of their prototypes.
  - G2: Like G1, G2 showed low coordination, leading to inefficiencies in managing and aligning the group's activities. This resulted in duplicated efforts, impacting the overall quality of their prototypes.
  - G3: This group was strong in coordination, effectively managing tasks, people, and resources. It ensured that the prototyping process was successfully conducted and that deadlines were met efficiently.
  - G4: Coordination was medium, which points to an adequate level of organizational ability but with potential for improvement in synchronizing their efforts.
- **Cooperation:**
  - G1: The group exhibited high levels of cooperation, working effectively together in a shared space to execute tasks. This cooperative spirit likely mitigated some of the issues caused by poor coordination, ensuring that the group still produced functional and creative prototypes.
  - G2: Despite the coordination challenges, G2's ability to work together toward common goals remained high. This strong cooperative effort ensured that the group supported cohesive and good prototypes, even if the process was not properly conducted.
  - G3: Cooperation was at a medium level, suggesting that while the group worked together, there might have been some challenges in joint efforts or sharing a space for task execution. These challenges slightly hindered the seamless integration of ideas and efforts in their prototypes.
  - G4: the group scored high in cooperation, suggesting that despite some communication and coordination issues, the group worked together very effectively when executing tasks. This cooperative dynamic helped them overcome organizational challenges and produce medium-quality prototypes.

## 4.3 Assessment Questionnaire

**Auxiliary Question AQ2**

What is the participants' perception of practicing collaboration through the use of DT?

To assess the use of DT to encourage collaboration, participants responded to a questionnaire detailed in Table 4.

Initially, inspired by the 4-level Likert Scale (Excellent, Good, Average, Poor) [1], participants indicated values for their teams in an Overall assessment of using DT to foster collaboration (Table 4-Q1). The responses point out that 100% of the teams consider the application of DT as an excellent strategy for promoting collaboration among groups.

Next, the teams indicated the Positive aspects of DT for exploring collaboration (Table 4-Q2). The responses suggest that:

- "team activity and reasoning about solving current problems" – G1
- "Learning in teamwork, stimulates creativity" – G2
- "It greatly helped to develop teamwork and also to improve our way of presenting. It also taught us about important methodologies, such as design thinking, in addition to the knowledge we gained to create low and high fidelity prototypes" – G3
- "Working dynamically, professionally, and smartly. Set deadlines and proposals in the same way that occurs in a corporate environment" – G4

Regarding the Perceived difficulties during the application of DT (Table 4-Q3), participants from G2 pointed out that part of the agenda could be carried out remotely, aiming to facilitate the conduct of field actions, such as data collection through interviews. The other groups did not identify specific points of difficulty, suggesting that their experiences with the application of DT were adequate.

Finally, in terms of Improvements for the next application (Table 4-Q4), G2 again indicated the importance of using remote activities as a way to facilitate the execution of tasks. G4 suggested that there could be a link with code implementation activities for understanding subsequent stages. The other groups did not offer specific suggestions for improvement, indicating overall satisfaction with the current DT process.

## 5 DISCUSSION

This section analyzes the use of DT to foster collaboration among software development students. The unanimous rating of DT as an excellent strategy by all groups underscores its effectiveness in promoting high levels of communication and cooperation. This aligns with G1 and G2's high scores in these areas, suggesting DT's iterative and human-centered approach improves information exchange and collective problem-solving. However, their low coordination scores indicate a disconnect between DT's ideation and execution phases, likely due to an emphasis on creativity and ideation over project management and task delegation.

The varying levels of communication, coordination, and cooperation among the groups highlight the multifaceted nature of collaboration in a DT context. For instance, G1 and G2's high communication and cooperation scores suggest that these groups excelled in exchanging information and working together towards common goals, possibly due to a strong emphasis on open dialogue and collective brainstorming. However, their low coordination indicates challenges in organizing and managing tasks, pointing to a potential gap in applying project management techniques within the DT framework. In contrast, G3's high coordination reflects a well-organized approach to managing tasks and resources, likely due to an effective adoption of DT's structured phases, which emphasizes iterative planning and resource allocation. G4's medium communication and coordination, coupled with high cooperation, suggest that while this group effectively worked together to achieve tasks, there were areas for improvement in maintaining consistent communication and task alignment. These differences underscore the importance of integrating project management and coordination strategies within the DT process to ensure that all aspects of collaboration are effectively addressed, thereby enhancing the overall quality and coherence of the developed prototypes.

G3's balanced scores and G4's medium communication and coordination with high cooperation suggest that while DT encourages cooperation, there may be variability in how different groups internalize and implement DT's principles. G3's high coordination reflects an organized approach, likely due to the team's effective adoption of DT's iterative process for managing tasks and resources. This suggests that when teams fully embrace DT's structured phases, coordination can improve significantly.

The perceived difficulties and proposed improvements for remote activities indicate a need to adapt DT processes to hybrid environments. G2's and G4's feedback on remote activities and linking DT with code implementation activities reflect a growing trend towards distributed collaboration. This feedback may guide future iterations of DT in educational settings, ensuring that it remains relevant and effective in an increasingly digital and remote collaboration landscape.

The results highlight the role of DT in educational settings for fostering collaboration. However, they also shed lights on opportunities for improving coordination within DT frameworks and adapting DT techniques for remote collaboration. Integrating project management tools such as Jira[5] and Trello[6], and techniques within the DT process could enhance coordination, while incorporating remote collaboration tools could ensure DT's effectiveness in a hybrid or fully remote environment.

Our study advances beyond those mentioned in the literature by delving deeper into the practical implementation of Design Thinking in a real classroom setting. We analyze interactions, project artifacts, and questionnaires over an 18-week semester. We applied the 3C collaboration model (Communication, Coordination, Cooperation) to highlight how the iterative, user-centered process of DT fosters teamwork, idea sharing, and collaborative problem-solving. Furthermore, our study provides practical recommendations for educators and broader implications for industry and researchers, emphasizing the dual competencies—technical and collaborative skills—required in professional life.

Thus, the insights from our study suggest that while DT is a powerful pedagogical tool for fostering collaboration, there is an opportunity for educators and practitioners to refine its application. By addressing the challenges highlighted in the assessments, DT can be more effectively leveraged to prepare students for the collaborative demands of the modern software development industry.

## 6 THREATS TO VALIDITY

Our study included some elements that pose potential threats to its validity. One such element is the subjective evaluation of the 3C levels (Communication, Coordination, and Cooperation), which could introduce researcher bias. Although we took steps to ensure a diverse research team was involved in the data analysis process, subjective interpretations remain a risk. To mitigate this, we invited a professional from the software development industry with over four years of experience in Design Thinking and Requirements Engineering to assist with the evaluation of the prototypes and provide an external perspective.

First, there is the risk of researcher bias, as the interpretation of qualitative data, particularly in case studies, can be subjective. To mitigate this, we ensured a diverse research team was involved in the data analysis process.

Another threat is about results generalization. Given that the study was conducted within a specific educational context at a Brazilian Institution, the results may not be directly transferable to different educational settings or cultural contexts. It is important for future research to replicate this study in varied environments to strengthen the generalization of the findings.

The limited scope of DT techniques applied in the study also presents a validity threat. While we explored a range of techniques, the full spectrum of DT's potential in software engineering education might not have been fully captured. Future studies should aim to incorporate a wider array of DT methods to comprehensively assess their impact.

[5]https://jira.com
[6]https://trello.com

Finally, the participant selection could also influence the study's outcomes. As the participants composed themselves as groups, there may be an inherent bias towards more motivated or engaged individuals, which could skew the results. Ensuring a random and representative sample in future studies would help in addressing this concern.

## 7 FINAL CONSIDERATIONS

This study presented an investigation of the use of Design Thinking to enhance collaborative skills among software development students. Through artifact analysis and student questionnaires, we have gained insights into a practical application of DT in an educational context and its perceived impact on collaboration.

### 7.1 Recommendations for Educators

Based on the insights gained from this study, the following recommendations are provided for educators who wish to use Design Thinking to foster collaboration among students:

(1) **Integrate Project Management Techniques**: Incorporate project management tools and techniques within the DT framework to address coordination challenges. This will help students better organize tasks, manage resources, and synchronize their efforts, leading to more efficient and coherent project outcomes.
(2) **Encourage Open Communication:** Foster an environment that encourages open and frequent communication among students. This can be achieved through regular check-ins, collaborative tools, and structured discussion sessions, ensuring that all team members are engaged and information flows freely.
(3) **Adapt DT for Remote and Hybrid Environments:** Modify DT processes to accommodate remote and hybrid learning environments. Utilize digital collaboration tools and platforms to facilitate remote participation, data collection, and ideation, ensuring that students can collaborate effectively regardless of their physical location.
(4) **Provide Clear Guidelines and Expectations:** Set clear guidelines and expectations for each phase of the DT process. This includes defining roles, responsibilities, and deliverables, helping students understand what is expected at each stage and how to achieve their goals efficiently.
(5) **Incorporate Reflection and Feedback Sessions:** Schedule regular reflection and feedback sessions to help students evaluate their collaboration processes and outcomes. These sessions should focus on identifying strengths and areas for improvement, encouraging continuous learning and adaptation throughout the project.

### 7.2 Implications for Industry and Researchers

The findings of this study have several implications for both industry professionals and academic researchers.

For industry, the high marks given by students to DT for fostering collaboration suggest that introducing DT practices in professional development teams could enhance communication and cooperation. The identified need for improved coordination within DT suggest need for the development of new DT facilitation tools that better integrate project management elements. Additionally, industry professionals could benefit from the insights into adapting DT for hybrid environments, which is becoming increasingly relevant.

For researchers, this study provides empirical evidence of applying DT in a real-world educational setting, which can support future studies on DT. The variability in how different groups adopted DT's principles suggests that further research could explore the factors that influence this process, including cultural, pedagogical, and individual personality traits.

Our study provides the data necessary for replication. We have made available the schedule that we followed and the tools we used, such as Google Docs and Figma, Google Meet for communication, Google Whiteboard for brainstorming.

### 7.3 Future Work

Future research from this study could take several directions. Firstly, longitudinal studies could examine the long-term impacts of DT on students' collaborative skills as they enter the industry. Secondly, experimental studies could compare the outcomes of DT with other collaborative frameworks in software development education. Finally, there is a need for the development and testing of specific tools and techniques that could help bridge the gap between DT's ideation focus and the need for effective project coordination.

By addressing the nuances and challenges of DT as revealed by this study, future research and practice can refine the application of DT, ensuring it remains a robust and effective tool for fostering collaboration in both educational and professional software development environments.

The synthesis of the results and discussions from this study not only highlights the strengths of applying DT in an educational setting but also opens rooms for improvement and adaptation in response to the evolving landscape of collaboration in the software development industry.

## ARTIFACT AVAILABILITY

The artifacts that we produced in this study are available at https://doi.org/10.5281/zenodo.12585344: (i) Instructions for creating the activity log; (ii) DT techniques; (iii) Feedback collection form; (iv) Example of an activity log; (v) Course syllabus; and (vi) Example of a revision history.

## ACKNOWLEDGMENTS

Dariane Abich and Rafael Parizi thank IFFar (Farroupilha Federal Institute) for providing research support. Sabrina Marczak thanks CNPq (Grant 313181/2021-7).

## REFERENCES

[1] Ankur Barua et al. 2013. Methods for decision-making in survey questionnaires based on Likert scale. *journal of asian scientific research* 3, 1 (2013), 35–38.
[2] Henri Bomström, Markus Kelanti, Elina Annanperä, Kari Liukkunen, Terhi Kilamo, Outi Sievi-Korte, and Kari Systä. 2023. Information needs and presentation in agile software development. *Information and Software Technology* 162 (2023), 107265.
[3] Silvia Bordin and Antonella De Angeli. 2016. *Communication Breakdowns in the Integration of User-centred Design and Agile Development* (1 ed.). Springer, Heidelberg, Germany, Chapter 6, 137–161.
[4] Rafael dos Santos Braz, José Reinaldo Merlin, Daniela Freitas Guilhermino Trindade, Carlos Eduardo Ribeiro, Ederson Marcos Sgarbi, and Fabio

de Sordi Junior. 2019. Design Thinking and Scrum in Software Requirements Elicitation: A Case Study. In *Design, User Experience, and Usability. Design Philosophy and Theory*, Aaron Marcus and Wentao Wang (Eds.). Springer International Publishing, Cham, 179–194.

[5] Luis Corral and Ilenia Fronza. 2018. Design Thinking and Agile Practices for Software Engineering: An Opportunity for Innovation. In *Proceedings of the Conference on Information Technology Education* (Fort Lauderdale, Florida, USA) *(SIGITE '18)*. Association for Computing Machinery, New York, NY, USA, 26–31.

[6] Design Council. 1944. The Design Process: What is the Double Diamond? designcouncil.org.uk/

[7] Hugo Fuks, Alberto Barbosa Raposo, Marco Aurélio Gerosa, Mariano Pimentel, Denise Filippo, and CJP de Lucena. 2011. Teorias e modelos de colaboração. *Sistemas colaborativos* (2011), 16–33.

[8] Marco Aurélio Gerosa, Mariano Pimentel, Hugo Fuks, and Carlos José Pereira De Lucena. 2006. Development of groupware based on the 3C collaboration model and component technology. In *Groupware: Design, Implementation, and Use: 12th International Workshop, CRIWG 2006, Medina del Campo, Spain, September 17-21, 2006. Proceedings 12*. Springer, 302–309.

[9] Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology.* Sage publications.

[10] Michael Lewrick, Patrick Link, and Larry Leifer. 2020. *The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods* (1 ed.). Vol. 1. John Wiley & Sons, New Jersey, USA.

[11] Anna B. Marques, Bruna Ferreira, Adriana Lopes, and Williamson Silva. 2020. Stimulating the development of soft skills in Software Engineering Education through Design Thinking. In *Proceedings of the Brazilian Symposium on Software Engineering* (Natal, Brazil) *(SBES '20)*. Association for Computing Machinery, New York, NY, USA, 690–699.

[12] Alexandra Matz and Panagiotis Germanakos. 2016. Increasing the Quality of Use Case Definition Through a Design Thinking Collaborative Method and an Alternative Hybrid Documentation Style. In *Proceedings of the International Conference on Learning and Collaboration Technologies*. Springer, Toronto, Canada, 48–59.

[13] Sofia Ouhbi and Nuno Pombo. 2020. Software engineering education: Challenges and perspectives. In *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 202–209.

[14] Maria Palacin-Silva, Jayden Khakurel, Ari Happonen, Timo Hynninen, and Jari Porras. 2017. Infusing Design Thinking into a Software Engineering Capstone Course. In *Proceedings of the Conference on Software Engineering Education and Training*. 212–221.

[15] Rafael Parizi, Marina Moreira, Igor Couto, Sabrina Marczak, and Tayana Conte. 2022. A Tool Proposal for Recommending Design Thinking Techniques in Software Development. *Journal of Software Engineering Research and Development* 10 (Mar. 2022), 3:1 – 3:15.

[16] Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte. 2022. How has Design Thinking being Used and Integrated into Software Development Activities? A Systematic Mapping. *Journal of Systems and Software* 187 (5 2022), 1–27.

[17] Lauriane Pereira, Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte. 2021. Towards an Understanding of Benefits and Challenges in the Use of Design Thinking in Requirements Engineering. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing* (Virtual Event, Republic of Korea) *(SAC '21)*. Association for Computing Machinery, New York, NY, USA, 1338–1345.

[18] Lauriane Pereira, Rafael Parizi, Matheus Prestes, Sabrina Marczak, and Tayana Conte. 2021. Towards an Understanding of Benefits and Challenges in the Use of Design Thinking in Requirements Engineering. In *Proceedings of the Annual ACM Symposium on Applied Computing*. ACM, Virtual Event, Republic of Korea, 1338–1345.

[19] Matheus Prestes, Rafael Parizi, Sabrina Marczak, and Tayana Conte. 2020. On the Use of Design Thinking: A Survey of the Brazilian Agile Software Development Community. In *Proceedings of the International Conference on Agile Software Development*. Springer, Copenhagen, Denmark, 73–86.

[20] Anderson Souza, Bruna Ferreira, Natasha Valentim, Lauriane Correa, Sabrina Marczak, and Tayana Conte. 2020. Supporting the Teaching of Design Thinking Techniques for Requirements Elicitation Through a Recommendation Tool. *IET Software* 14 (Dec 2020), 693–701. Issue 6.

[21] Anderson Felipe Souza, Bruna Ferreira, Natasha Valentim, and Tayana Conte. 2018. An Experience Report on Teaching Multiple Design Thinking Techniques to Software Engineering Students. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering* (Sao Carlos, Brazil) *(SBES '18)*. Association for Computing Machinery, New York, NY, USA, 220–229.

[22] United Nations. [n. d.]. Sustainable Development Goals. https://sdgs.un.org/goals. Accessed in 06/02/2024.

[23] Yunwen Ye. 2006. Supporting software development as knowledge-intensive and collaborative activity. In *Proceedings of the International Workshop on Workshop on Interdisciplinary Software Engineering Research* (Shanghai, China). ACM, New York, NY, USA, 15–22.

[24] R.K. Yin. 2017. *Case Study Research and Applications: Design and Methods.* SAGE Publications.