TÂNIA MARA DORS

THE REFLECTIVE PRACTICE IN A SOFTWARE DEVELOPMENT STUDIO

Master´s degree dissertation submitted as a partial requirement for the Master´s Degree in Informatics in the Graduate Program in Informatics of the Pontifícia Universidade Católica do Paraná, Brazil.

Curitiba
2019

TÂNIA MARA DORS

THE REFLECTIVE PRACTICE IN A SOFTWARE DEVELOPMENT STUDIO

Master´s degree dissertation submitted for the Master´s Degree in Informatics in the Graduate Program in Informatics of the Pontifícia Universidade Católica do Paraná, Brazil.

Major concentration field: Computer Science

Supervisor: Prof. Dr. Sheila Reinehr
Co-supervisor: Prof. Dr Frederick Van Amstel

Curitiba
2019

Curitiba, dia do mês do ano de xxxx.

Assinatura do Autor

Assinatura do Orientador

FICHA CATALOGRÁFICA

Dors, Tânia Mara
Título da Dissertação/Tese
/ Iniciais seguidas dos sobrenome. -- Curitiba, ANO.
Número de páginas p.

Dissertação (Mestrado) – Pontifícia Universidade Católica do Paraná. Curitiba. Programa de Pós-Graduação em Informática.

1. Palavra-chave01 2. Palavra-chave02 3. Palavra-cahve03. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.
II. t ou d.

ACKNOWLEGENTS

I would like to thank my supervisors, Prof. Sheila Reinehr and Prof. Frederick Van Amstel for their assistance and dedicated involvement throughout this process. Prof. Sheila Reinehr for your guidance, support, and understanding over the years, but also for showing me what it is to be an academic. Prof. Frederick Van Amstel for encouraging the studio investigation and for availability, helpfulness, and reflective chats on studio issues.

I would also like to show gratitude to all the staff, particularly to Prof. Fabio Binder, and the students of the Pontifícia Universidade Católica do Paraná Extension Course, which allow me to do a participant observation of their project development to this research.

Finally, I would like to thank my son for the smiles, hugs, to be the patient and comprehensive when he was private from attention, in moments I had dedicated me in depth to the study. Also, to my family that gave me incentive and constant support, specially my mother that moved home from other city to take care of my son when I needed, bringing me tranquillity to dedicate and give focus. To my father that teaching me with his example that doesn´t matter the time, we always can reach our dreams, this thought makes me to pursuit and persevere in my goals.

This study was financed in part by the Coordenação de Aperfeiçoamento Pessoal de Nível Superior – Brazil (CAPES) – Finance Code 001.

“I hear and I forget.
I see and I remember.
I do and I understand.”
Confucius

ABSTRACT

Nowadays, the technology permeates nearly every aspect of our lives then it is usual to expect that the number, size, complexity, and application domains of developed software will continue to grow, as well as the problems with cost, timelines and quality of software products development. This scenario demands even more skilled software developers that blend technical and personal skills. Corporations are complaining about the lack of professional awareness and low levels of communication and teamwork skills in engineering undergraduates. For these reasons, software engineering educators have been discussing what kind of education, what methods or approaches are appropriate to address these demands and issues related to professional practice. Over the last two decades, educators have adopted new techniques, tools and approaches for practical learning, among them the studio-based learning using reflective practice, which has been used by some universities around the world, especially in architecture courses. Although very good results in these contexts have been reported there are few studies focusing on the use of such approach in the software development field. The objective of this study is to understand the reflective practice contribution to the practical learning of mobile application development in a particular software development studio program. Data were collected using ethnographic method through participant observation and from students´ written self-reflections, which were analyzed using Cycle Coding Method supported by Atlas.ti tool. As a result, it was possible to observe that the reflective practice promotes the process of emerging new ideas, helps to build a culture that is supportive of critique and contributes to practice and development of essential software engineering skills, as collaboration, verbal or written communication, commitment, interpersonal, adaptability, flexibility and teamwork. In addition, problem solving, decision-making, planning, project management, time management, scope management and outsourcing management. The studio approach emphasizes the practical learning, supports the development of skills required for software engineering practice, and helps in developing new technical skills. Finally, studio seems to be an authentic environment of relationship between academic disciplines and real-world experiences, where students can practice and learn by practicing, thereby, it better prepares students for the real world.

Keywords: Software Engineering Education, Computer Science Practical Learning, Software Practice, Software Engineering in Practice, Software Development Studio, Studio-based Learning, Studio Course, Reflective Practice.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACM	Association for Computing Machinery
ACOT	Apple Classrooms of Tomorrow
ADA	Apple Developer Academy
CBL	Challenge Based on Learning
CC2001	Computing Curricula 2001
CE2016	Computer Engineering Curricula 2016
CMU	Carnegie Mellon University
CS2008	Computer Science Curriculum 2008
CS2013	Computer Science Curricula 2013
HCI	Human-Computer Interaction
IEEE	Institute for Electrical and Electronic Engineers
MIT	Massachusetts Institute of Technology
SEI	Software Engineering Institute
SE	Software Engineering
SE2004	Software Engineering Curricula 2004
SE2014	Software Engineering Curricula 2014
SWEBOK	Software Engineering Body of Knowledge

CHAPTER 1 - INTRODUCTION

"The student cannot be taught what he needs to know, but he can be coached: he has to see on his own behalf in his own way the relations between means and methods employed and result achieved." John Dewey, 1974, p.151.

The technology is continuously changing, so software engineers must deal with new methods, tools, platforms, user expectations and software markets, therefore an advanced education is required to prepare professionals for coming decades and the new demands (Tomayco, 1997).

Nowadays, the dependence on software permeates nearly every aspect of our lives, and then it is usual to expect that the number, size, complexity, and application domains of developed software will continue to grow, as well as the problems with cost, timelines and quality of development of software products (Hibburn, 1999). These issues have created a demand for competent computing professionals who appreciate and apply software engineering knowledge and practices.

On the other hand, since when software engineering has been considered as an engineering discipline, the frustration persists for both academics and industrial software developers about the proper balance of theory and practice. Corporations and employers have often complained publicly about the lack of professional awareness and low levels of communication and teamwork skills in engineering undergraduates. In fact, when students finish the university, they have technical knowledge, but not necessarily its means professional competence.

For these reasons, software engineering educators have been discussing what kind of education, what methods or approaches are appropriate to address these demands and to deal with issues related to professional practice. As a result, we have specific recommendations in software engineering curricula Computer Science 2001

(CC2001[1], p. 59), Computer Science Curriculum (CS2013[2], p.15), Software Engineering 2014 (SE2014[3], p.17) and Computer Engineering 2016 (CE2016[4], p.9-10)

A successful software engineer must possess a wide range of skills and talents and the challenge of what to teach software engineers evolves over time as technologies, applications, and requirements change, according to Jazyeri (2004) that argued the educating software engineers has taken on new form and become more complex and urgent.

> The challenge of designing a curriculum for informatics today is to find **a way to combine formal with practical learning, technical with non-technical skills, and informatics with interdisciplinary knowledge.** To do this, we need to, as much as possible, create a real-world environment at the university. The purpose is **to enable the learning of non-technical skills in a formal way**. (Jazyeri, 2004, p. 6, highlighted by author)

In addition, Lethbridge et al. (2007, p.1) argued, "software engineering (SE) community could have a significant impact on the future of the discipline by focusing its efforts on improving the education of software engineers".

Thereby, in the last two decades, educators have been exploring new techniques, tools and approaches for practical learning, such as, laboratory instruction, the use of electronic whiteboards and tablet computers, problem-based learning, active learning methodology, flipped or inverted classroom and various studio approaches that integrate laboratory, lecture, and discussion.

At the same time, some educators consider that practical learning as a curricular discipline of graduation programs is the key to develop the technical competence and the skills needed to make tradeoffs between theoretical and practical issues. One of the pedagogical approaches to practical learning is the studio-based learning using reflective practice proposed by Donald Schön in 1983, which has been used in some universities, worldwide, since 1990.

Prior et al. (2014, p.129) argued that studio-based learning is one response to the mismatch between what employers perceive as important abilities and how

---

[1] CC2001 refers to the 2001 version of final report of the Computing Curricula 2001 project that contains a set of recommendations for undergraduate programs in Computer Science. A joint undertaking of the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) and the Association for Computing Machinery (ACM) to develop curricular guidelines for undergraduate programs in computing.

[2] CS2013 refers to the 2013 curriculum guidelines for undergraduate degree programs in Computer Science proposed by the joint task force on computing curricula ACM and IEEE.

[3] SE2014 refers to the 2004 version of the basis curricula for Software Engineering undergraduate courses proposed by the joint force established between IEEE and ACM.

[4] CE2016 refers to the 2016 version of the basis curricula for Computer Engineering undergraduate courses proposed by the joint group established between ACM and IEEE.

universities prepare graduates for employment, particularly regarding non-technical skills and the changing expectations and learning styles of students.

Although, over the last twenty-eight years, some universities programs had adopted the reflective practice with studio-based approach, it was not found any study directly dealing with the reflective practice in software development studios. Some of the studies describe the experience of implementing software development studios, but they are mainly focused on facilities implementations, concepts and definitions (Tomayco, 1991), (Kuhn, 2001), (Broadfoot and Bennet, 2003), Cennnamo et al., 2011), (Hokanson, 2012), (Prior et al., 2014). Therefore, the motivation of this study is the lack of previous field research regarding the contributions that this kind of practice brings to the software engineering education in a software development studio.

1.1 Research Objective

The objective of this study **is to understand the reflective practice contribution to the practical learning of mobile application development in a software development studio**.

Specific objectives are:

- Analyze the contributions of reflective practice to software development in a software development studio;
- Analyze the contributions of reflective practice to the development of individual competence and the artistic talent in a software studio. In other words, the contributions to the development of key competences required for professional practitioners.

1.2 Research Process

In order to organize the research, at the beginning the phases, activities and expected results of the process were defined, as illustrates the Figure 1.1. The phases of the process are:

- Phase 1 – Research Preparation: phase that corresponds to the delimitation of the study area, collection and analysis of bibliographic references, delimitation of the theme and establishment of objectives, questions and propositions. As a result, this phase contains the theoretical and methodology references to provide a background to the study.

- Phase 2 – Research Definitions and Planning: phase to select the research method and to define and plan the research steps according to it, as well as, the resources, techniques and protocols for data collection and data analysis.
- Phase 3 – Research Execution: phase to execute the defined steps of the research following the research method selected.
- Phase 4 – Research Data Analysis: phase to consolidate results of the data analysis and to relate the contributions and conclusions.

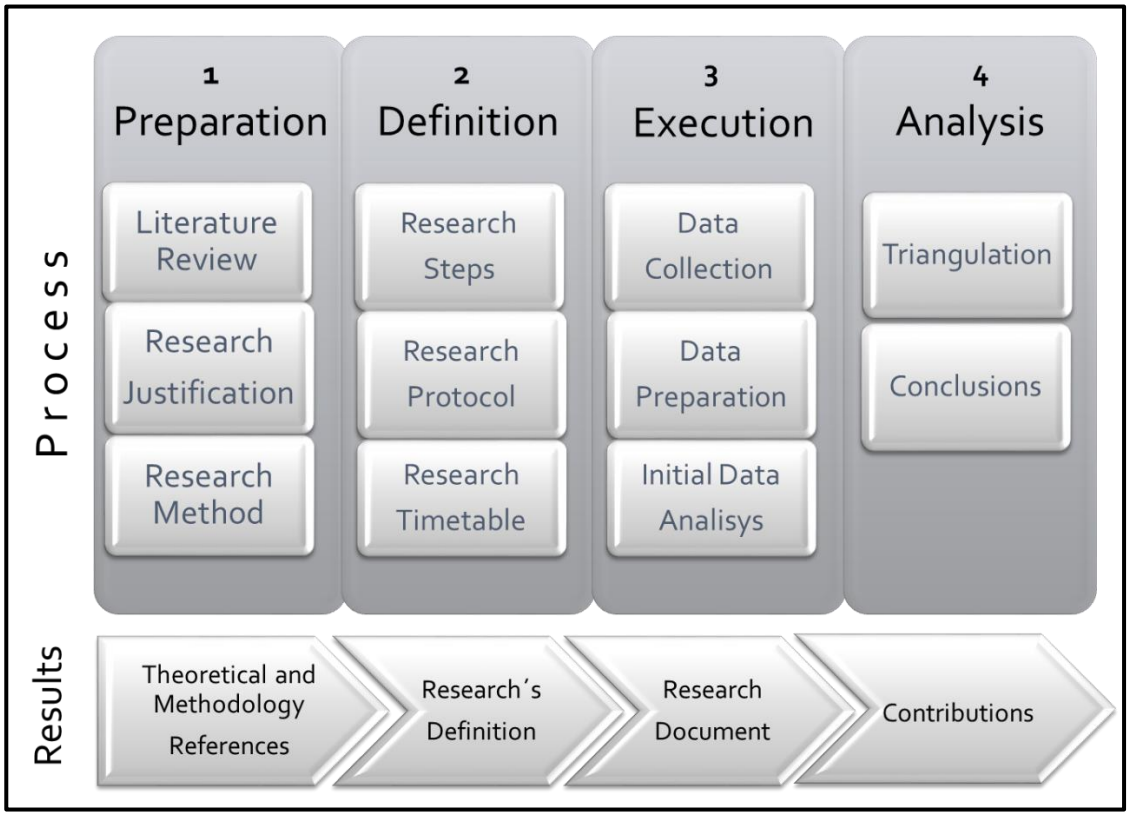


Figure 1.1 - Research Process

1.3 Document Structure

This document was structured in 6 chapters that were organized as follow:

- Chapter 1 introduces the relevance of the theme for the area of interest, its motivation, delimits the scope of work and objectives.
- Chapter 2 contains the literature review which focuses is why practical learning is important to software engineering education, what is the reflective practice, its concepts and its application in education studio, which are the

roots of the studio education and the concepts and structure of a software studio.

- Chapter 3 presents the method chosen for the research and describes the steps of the research providing in depth the details about the definitions and planning of data collection, the techniques and protocols for the data analysis, the research execution and the data analysis.
- Chapter 4 describes step-by-step the process of executing the research according to the planning of the research described and defined in the previous chapter.
- Chapter 5 relates and discusses the reached results.
- Chapter 6 concludes this study, presents final considerations about it, the contributions and limitations of the research and suggests what it is possible to do as future works.

1.4 Conclusions

In this chapter, introduced the relevance of research object in software engineering area and its motivation. It presented the research objectives, the scope of this research and how the work was organized in terms of research process and phases, as well as how the document was structured.

CHAPTER 2 - LITERATURE REVIEW

"To know what you know and what you do not know,
that is true knowledge"
(Confucus).

The objective of this chapter is to present the relevance of practical learning for software engineering education, to introduce the concept of reflective practice and the concept of studio education, as well the software studio to provide the background for this study.

The first aim is to understand the practice of learning in software engineering education. The second aim is to explain the concept of reflective practice proposed by Schön the using studio approach. The third aim is to relate the origin of studio education in architecture. The fourth aim is to explain the more recent concept of a software studio education and how to apply it for teaching software engineering and development, including software engineering, Computer Science, and information technology.

2.1 Practical Learning in Software Engineering Education

Software engineering education has been developed along with the maturation of the profession itself over the last 50 years. Since the beginning, the mismatches between curricula and industry needs have been quarreled. The solution to this problem depends heavily on the ability to design and implement curricula that not only emphasizes Computer Science, information science, and technology, but also focus on the practice of software engineering with the inclusion of the equally critical people and process issues (Hilburn, 1999).

In 1976, at the University of California, in a one-day industry/academia interface workshop, Antony Wasserman and Peter Freeman realized the wide divergence between what skills industry said it needed in university graduates and what skills are transmitted by typical Computer Science courses. This characterized what was called 'software crisis', and as a result, at late 70's, the IEEE-CS (Institute of Electrical and Electronics Engineers Computer Society) curriculum project stimulated the creation of

several Master programs on Software Engineering (MSE) in United States (Tomayco, 1998, p.7).

Some years later, Frederick Brooks (1995, p.193) stated that “The gap between the best Software Engineering practice and the average practice is very wide - perhaps wider than in any other engineering discipline”.

In 2001, in the final report of the computer curricula 2001 (CC2001), elaborated by the Joint Task Force on Computing Curricula IEEE Computer Society and the Association for Computing Machinery (ACM), there was a chapter dedicated to professional practice, calling the attention to this point:

> As we enter the 21st century, an unprecedented opportunity exists to make professional practice a seamless part of the curriculum in Computer Science and other computing disciplines. […] **The need to incorporate professional practice into the curriculum is based upon real-world issues, such as the needs of the public and private sector**, the public’s demand for higher quality products, the increasing number of software liability cases, and the need to promote life-long learning after graduation. […] **Both the private and public sectors have a vested interest in students learning professional practice** (ACM/IEEE 2001, p. 59, highlighted by author)

In an historical retrospective, Lethbridge et al. (2007) highlight some events and activities that occurred in late 90´s, which characterized the explosion of progress in both Software Engineering professionalism and education:

- The establishment of the Software Engineering Code of Ethics;
- The announcement by the State of Texas that it would license software engineers beginning in 1999, and similar moves by a few other jurisdictions;
- The establishment of undergraduate programs, which had initially appeared in Australia, but now started rapidly appearing in Europe, Canada and the U.S;
- The establishment of Ph.D. programs followed;
- The adoption of accreditation criteria in the U.S. for educational programs in software engineering ABET accepted the final set of criteria in 1999;
- The development of SWEBOK (Software Engineering Body of Knowledge) and SE2004;
- The development of the IEEE Certified Software Development Professional designation (CSDP).

All these events seem demonstrate a continuous concern to better prepare the undergraduate students for professional practice.

It is important to highlight that SE2004 document was produced following an important guiding principle, which has remained the same in SE2014, “The education of all software engineering students must include student experiences with the professional practice of software engineering.” (IEEE/ACM, 2014, p. 17). As a result, a software engineering curriculum emphasizes the student work experience, more over the practical exercises, and recommend, “A software engineering program needs faculty who possess both advanced education in computing with a focus on software, and sufficient experience in software engineering practice.” (IEEE/ACM, 2004, p. 71).

According to Jazyeri (2004), a successful software engineer must possess a wide range of skills and talents and the challenge of what to teach software engineers evolves over time as technologies, applications, and requirements change. He argued that, educating software engineers has taken on new form and become more complex and urgent.

> The challenge of designing a curriculum for informatics today is to find **a way to combine formal with practical learning, technical with non-technical skills, and informatics with interdisciplinary knowledge.** To do this, we need to, as much as possible, create a real-world environment at the university. The purpose is **to enable the learning of non-technical skills in a formal way**. (Jazyeri, 2004, p. 6, highlighted by author)

Jazyeri (2004) proposed that curriculum be integrative, holistic, interdisciplinary, and project-based. He related various reasons for the adoption of a project-based approach, as that it engages the students and motivate them more than the traditional classroom, and in certain fields, the learning by doing is most effective. In addition, this approach should enable the students to apply system-level thinking, see technologies in use, and appreciate the difficulties and benefits of working with others on a team.

Nevertheless, Ghezzi and Mandrioli (2005, p. 637) call attention to the same situation of practical learning: “The main dichotomy that we face in engineering is learning by studying (at school) vs. learning by doing (at work)” and added that the software engineers must be able to:

> Keep his/her knowledge current with respect to the new approaches and technologies; interact with other people (often not from the same culture); understand, model, formalize, analyze a new problem; recognize a recurring problem, and reuse or adapt known solutions; manage a process and to coordinate the work of different people. […] **Some of these skills are better suited to be taught and learned at school, others are fully understood only after some practice in the real world** (GHEZZI, 2005, p.637, highlighted by author).

Currently, a variety of educational materials is available from traditional libraries to electronic sources, such as IEEE and ACM digital libraries, plus some shared resources as interactive repository and notable websites for educators, which serve as sources for students and educators.

Despite this, Lethbridge et al. (2007) states that one of the challenges to improve the quality of software engineering education is communicating real-world industrial practices to students in a more effective way. In addition, they said that besides of gradual adoption of a variety of innovative approaches, it is also expected to see SE practitioners adopting approaches that reduce the amount of lecturing which includes Studio and Problem Based Learning approaches. In the end, they conclude:

> The majority of quality and budgetary issues with software have their root cause in human error or lack of skill. These in turn arise in large part from inadequate education. Therefore, improving education should go a long way, in the long run, towards improving software and software practice. (Lethbridge et al., 2007, p. 15).

The conclusion is that software engineering education must combine theory, practices, and application experience. The practical learning is essential to consolidate the use of best practices, code of ethics and to develop practical competences required by practitioners.

Prior et al. (2014, p.129) argued that studio-based learning is one response to the mismatch between what employers perceive as important abilities and how universities prepare graduates for employment, particularly regarding non-technical skills and the changing expectations and learning styles of students.

Studio-based learning using the reflective practice is one of the education approaches for professional practice that some universities worldwide are adopting in their curricula and can be seen in next section.

2.2 Reflective Practice

Reflective practice is a form of reflective learning by doing, with the help of coaching. Donald Schön introduced this concept from observations of students and their professors in architectural studios (Schön, 1983). He was an educator, a Ford Professor Emeritus on Urban Studies and Education, and Senior Lecturer in the Departments of Urban Studies and Planning, and Architecture, of the Massachusetts Institute of Technology (MIT), from 1968 until his death in 1997.

At late 70´s, he had the opportunity to participate in an architecture education case study supported by the Andrew Mellon Foundation, leaded by William Porter, director of MIT´s School of Architecture and Planning, and Dean Maurice Kilbridge, director of Harvard Design Post-graduation School. As a result, of this case study, he wrote two books, “The Reflective Practitioner: How Professionals Think in Action” (Schön, 1983), and “Educating the Reflective Practitioner: Toward a new design for teaching and learning in the professions” (Schön, 1987).

The first influence on his thought was John Dewey’s theory of inquiry, that was the basis for his doctoral thesis: "The student cannot be taught what he needs to know, but he can be coached: he has to see on his own behalf in his own way the relations between means and methods employed and result achieved.” (Dewey, 1974, p. 151).

Thirty years later, when he was writing about the reflective practitioner, he realized that: “I was attempting, in effect, to make my own version of Dewey’s theory of inquiry, taking ‘reflective practice’ as my version of Dewey’s ‘reflective thought’” (Schön, 1992, p. 123). Schön proposed a fundamental reorganization of how to think about professional practice and the relationship of theory to practice. He formulated his view of design in terms of “reflective activity” and related notions, especially “reflective practice”, “reflection-in-action”, “knowing-in-action” and “reflection-on-action” and reflective conversations with the material of a design situation.

In his second book (Schön, 1987), he elaborates his position with special attention to what he calls "the reflective practicum”, the specific experiences that he believes help students to acquire knowing-in-action under the coach of expert practitioners. Reflection-in-action is the reflective form of knowing-in-action, its means, the reflection during the problem-solving process. In reflection-in-action, “doing and thinking are complementary. Doing extends thinking in the tests, moves, and probes of experimental action, and reflection feeds on doing and its results. Each feeds the other and each sets boundary for the other” (Schön, 1983, p. 280).

For him, Knowing-in-action is the knowing built into and revealed by our performance of everyday routines of action. The knowing-in-action sometimes is labeled as “intuition”, “instinct” or “motor skills”, in such cases we continually control and modify our behavior in response to changing conditions.

> This capacity to do the right thing … exhibiting the more that we know in what we do by the way in which we do it, is what we mean by knowing-in-action. And this capacity to respond to surprise through improvisation on the spot is what we mean by reflection-in-action. (Schön, 1987).

After problem solving has occurred, a practitioner may consider what he could have done differently or would do differently next time. This is what Schön called reflection-on-action or reflecting on experience. “We may reflect on action, thinking back on what we have done in order to discover how our knowing-in-action may have contributed to an unexpected outcome” (Schön, 1987 p. 26). It does not matter, if it happens after the fact or during a pause, during action, in both cases, our reflection has no direct connection to present action.

Schön noted that the practice of any profession involves the use of special esoteric “knowledge in action”, that according to Polanyi (Schön, 1987, p. 22) it is tacit knowledge learned not in the abstract but in use. He stated that there are three ways of acquiring such knowledge. The first, very unusual, is via self-instruction. The second is via apprenticeship, learning “on line” in “real world” contexts. However, because this is both inefficient and can have serious negative real-world effects, the standard site of learning is the ‘*practicum’*. The practicum is an off-line situation that approximates the world of practice.

Since the beginning, it was Schön’s assumption that:

> Competent practitioners usually know more than they can say. They exhibit a kind of knowing in practice, most of which is tacit.… Indeed, practitioners themselves often reveal a capacity for reflection on their intuitive knowing in the midst of action and sometimes use this capacity to cope with the unique, uncertain, and conflicted situations of practice. (Schön, 1983, pp. 8-9).

To reveal the central role of reflection-in-action in professionals' practice, he explained that in their reflective conversations with design situations, designers "frame" and "reframe" problems. Such a professional knowledge is developed within the action, and the reflection-in-action improves the proficiency and professional performance.

> In such conversations, the practitioner's effort to solve the reframed problem yields new discoveries, which call for new reflection-in-action. The process spirals through stages of appreciation, action, and re-appreciation. The unique and uncertain situation comes to be understood through the attempt to change it. Furthermore, the practitioners' moves also produce unintended changes, which give the situation new meanings. The situation talks back, the practitioner listens, and as he appreciates what he hears, he reframes the situation once again. (Schön, 1983, p. 131-132).

According to him, the reflective practice helps students acquire the kind of artistic talent essential for competence in undetermined areas of practice. Professional artistry refers to kinds of competence that practitioners demonstrate in certain practice situations that are unique, uncertain, and conflicting (Schön, 1987).

Yet, he highlighted two points. The first point is that the knowing-in-action characteristic of competent practitioners in a professional field is not the same as the professional knowledge taught in the schools. The second point is that competent professional practitioners often have capacity to generate knew knowing-in-action through reflection-in-action undertaken in the indeterminate zones of practice (Schön, 1987).

Besides, Schön argued that reflection in action necessarily involves experimentation, whereas reflective conversation is a reconstruction experiment of conception and within the experiment to define the problem, local experiments of various types are located: exploratory, action test and hypothesis testing.

Architectural design was the first professional domain studied by Schön to develop his epistemology of professional practice based on the concepts of reflection-in-action and knowledge-in-action. He analyzed design education on-site, providing and studying audiotaped protocols from teaching-learning sessions in the design studio. His first objective was to understand these protocols, to grasp the central features of education in design. After that, he extended his analysis to other professions, testing his hypothesis that all professions are “design like’ in some relevant aspect. For him ´design like’ professions are those where there is a pre-conceptualization of design for subsequent execution. Such, according to Waks (2001), “Schön’s proposed ‘design’ for professional education is design itself”. In fact, Schön concluded that the reflective practice teaching is the key to professional education, and he considered that the architecture design studio could be a model for practitioners of other sciences (Schön, 2000).

Hazzan (2002) made an analysis on the application of the reflective practitioner perspective to the profession of software engineering that resulted in a framework for adopting of this perspective in general and the studio method of teaching in particular into SE education. He concluded that the “Analysis of the kind of tasks that architecture students are working on and a comparison of these tasks to the problems that SE students are facing, suggest that the studio may be an appropriate teaching method in SE as well“. He had already suggested that the adoption of reflective practice methodology as a cognitive tool might help programmers in developing software systems and the students understanding of software-development methodologies (Hazzan, 2002, p. 164).

In 2003, Hazzan and Tomayco, stated that software engineering usually accomplishes and supports the adoption of the reflective practice. Moreover, the construction of ladders of reflection and the transition between levels of abstraction is an important skill for software developers and may improve their performance:

> The importance of reflection as a habit-of-mind in the context of SE is derived mainly from two factors: The first factor is the complexity involved in developing software systems, regardless of whether one examines this complexity from an engineering, social or cognitive point of view; the second factor is the crucial role of communication among teammates for the success of developing a software system. (Hazzan and Tomayco, 2003, p.1).

Recognizing the importance of reflection in practice applied to software, IEEE Software devoted a special track in one of the 2014 issues to reflective practice. According to its guests' editors Dybå, Maiden and Glass (2014):

> Reflection often takes place in cycles of experience followed by conscious application of learning from that experience, during which a software developer might explore comparisons, ponder alternatives, take diverse perspectives, and draw inferences, especially in new and/or complex situations. […] The concept of reflective practice centres on the idea of lifelong learning. (Dybå et al., 2014, p. 32-33).

Prior et. al (2016) concerning about teaching and learning reflective practice in a software engineering studio had argued that "reflective practice is now recognised as important for software developers and has become a key part of software studios in universities". In addition, there are many papers that claim that reflection in the studio is mandatory, however he argues that learning how to reflect is non-trivial exercise for undergraduate students, such is required an investigation of how best to teach and learn reflection (Prior et. al, 2016, p.7).

The design studios observed by Schön, as well as their instructional methods based on practical learning with coaching, have inherited in the historical tradition of the École des Beaux-Arts and the Bauhaus and its atelier model, which will be explained in the next section. Additionally, the studio education applied for software engineering and its educational model will be deeper discussed.

2.3 Roots of Studio

The roots of the design studio can be traced back as far as Medieval Times where guilds would take on apprentices (Schön, 1983). Most literature that discusses the origins of the design studio often describes two main schools of studio education: École des Beaux-Arts and the Bauhaus. This section briefly summarizes them and

their main contributions to design studios, followed by a section that brings the concept of studio education oriented to software engineering.

2.3.1 École des Beaux-Arts

École des Beaux-Arts (France, 1819-1914), or 'School of Fine Arts', was a French school of art which taught subjects such as drawing, painting, sculpture, engraving and architecture. The environments of this school were known as "ateliers", French for 'workshop', where students worked under the mentorship by a *Patron* (tutor). More recent studios are said to "have inherited the historical tradition of the École des Beaux-Arts and its atelier model" (Oh et al., 2013, p.303).

The École Des Beaux-Arts, from the beginning was controlled by an established system of teaching architecture and the instructors were closely allied with practice. The character of the studio varied from time to time, representing contemporary conditions and the best French thought during succeeding periods (Wheatherhead, 1941, apud Salama 1995). The evolution of the Beaux-Arts is divided into two periods: first, from founding the Royal Academy of Architecture (1671) to mid-nineteenth century, and second, from mid-nineteenth century to 1968.

According to Egbert (1981, apud Salama, 1995), the French Revolution broke the official academic tradition from 1793 to 1795 when the National Institute of Science and Art was established. Despite of this, Egbert argued that although the revolution ended the Royal academies, it did not cause a profound break in the tradition of design education in architecture.

In 1789, Julian David Leroy opened his own "atelier" for the specific purpose of providing special training in design to students in the Royal School of Architecture. This was the direct ancestor of the system of "ateliers", privately run and connected with the section of architecture in the Ecole Des Beaux-Arts, that was continue until 1968 (Chaffee, 1977, apud Salama 1995).

The objective on the "Atelier" was to provide the home base for the duration of the student´s life at the school. The "Atelier" was where all the design exercises, the core of the École´s educational system, took place.

Considering that the purpose of École des Beaux-Arts was to allow working and learning to occur simultaneously, while working under their Patron, the students would also be studying a theory of design; each atelier had a different "acquired reputation" that would attract different students depending on their taste or intended direction

(Carlhian, 1979, p.7). One of the most important factors of the atelier was that the "design problem and learning by doing superseded the lecture as the primary method of teaching"" (Broadfoot and Bennett, 2003, p.1). Many traditions in studios today evolved from the École which "provided the basis of a pedagogical method that is still the core of design and architectural education" (Broadfoot and Bennett, 2003, p.1). For instance, the use of the ´esquisse´ (initial sketch to a problem that would be further developed), the teaching of design by practicing professionals and the final evaluation of student work by a jury.

Another aspect that occurred at the ´Ecole, was the tradition where *'anciens'* (older students) helped younger students and there were also monthly architectural competitions called *'Concours Mensuels d'Emulation'*, in which the students were expected to enter at least twice a year. The reason the École closed in 1914 was because its "buildings were reserved as a military hospital" (Salama, 1995, p.49).

2.3.2 Bauhaus

The Staatliches Bauhaus, or School of Building (Germany, 1919-1933), was founded by Walter Gropius. Bauhaus is well known for its fusion of theory and practice, but more importantly, art and craft; it taught these new approaches in a workshop environment that formed the basis of today's studios. Some of the topics taught at this school were art, architecture, graphic design, interior design, industrial design, and typography.

In both schools Beaux-Arts and Bauhaus the end product was similar despite different professional, technological, and socio-cultural contexts. The main concern of Bauhaus was to combine arts and crafts to form a universal idea with the requirements of technology. Thus, an important way that Bauhaus differs from ´Ecole des Beaux-Arts is that Bauhaus' founder, Walter Gropius, placed an emphasis on production and technology.

In the early years of the Bauhaus, emphasis was on craftsmanship rather than machine production, and the creation of an ideal community in miniature. The teaching program aimed to develop the students' personality as well as technical skills. Gropius divided the hierarchy of the school into masters, journeymen and apprentices. Apprentices were those who passed the first exam set the by local guilds and the journeymen were those who provided a link with professional practice outside the school.

What differentiated the Bauhaus was a system of workshop teaching that attempted to equate craft with art and equipped graduates with as much technical expertise as theoretical and creative. Apprentices were instructed by masters of each particular craft (Workshop Masters) as well as by practicing fine artists (Masters of Form). Another differential of Bauhaus education is the foundation course instituted by Johannes Itten to prepare apprentices that focused on the study of materials to develop an understanding of their qualities and was designed to bring to life students' hidden creative abilities. According to Groupious (1965),"the objective of the Bauhaus was not propagating any style, system or dogma, but simply to exert a revitalization influence on design". (Salama, 1995, p. 53).

What made Bauhaus particularly unique was that "apprentices were to be instructed, not only by 'masters' of each particular craft, but also by fine artists" (Salama, 1995, p.50). One final, yet important, aspect was that the teaching program aimed to develop the personality as well students technical skills (Broadfoot and Bennett, 2003, p. 2), as this was considered key to the future design experiences.

2.4 Design Studio

Traditionally studios are organized around Schön´s concept of design studio, described as learning by doing. As earlier described, he referred to studios as a reflective practicum: "A *practicum* is a setting designed for the task of learning a practice. In a context that approximates a practice world, students learn by doing, although their doing usually falls short of real-world work." (Schön, 1987, p.37*)*.

Schön argued that the fundamental concepts of designing can only be understood in the context of the doing, through the experience of designing. He believed that reflection-in-action was the basis of any design process. For the new student learning to design this poses the problem that they are seeking to learn things they cannot grasp ahead of time.

Coaching and critique support reflective practice in a studio education. Coaching is about providing instruction or advice. Schön stated that teaching staff "function as coaches whose main activities are demonstrating, advising, questioning and criticizing" (Schön, 1987, p.38).

> Coaching would consist in observing student performance, detecting errors of application, pointing their correct responses. […] But, depending on one´s view […] coaches may emphasize either the rules of inquiry or the reflection-

> in-action by which, on occasion, students must develop new rules and methods of their own. (Schön, 1987, p.39).

Feedbacks help students understand their problems, eliminate errors from their proposed solutions, also eventually construct their own solutions, and they are provided in different ways in the studio.

> Studios are typically organized around manageable projects of design, individually or collectively undertaken, more or less closely patterned on projects drawn from actual practice. They have evolved their own rituals, such as master demonstrations, design reviews, desk crits, and design juries, all attached to a core process of learning by doing. And because studio instructors must try to make their approaches to design understandable to their students, the studio offers privileged access to designers' reflections on designing. It is at once a living and traditional example of a reflective practicum. (Schön, 1987, p. 43).

Central to the studio experience and the development of the ability of the student to learn to design in a thoughtful manner is the informal critique or desk crit (Schön, 1983, p. 43). Desk crit is a collaborative activity where the teacher and the student do design work together, discussing and sketching possibilities and imagining the consequences of design choices. Design teacher works to understand what the student is trying to do with his or her design work and provides feedback on these ideas and works with the student to further develop them.

Another form of crit is the design jury, which is when the student presents their drawings and describes their design to three to five local architects, instructors from other studios at the same school, other non-studio faculty members, or representatives of the client if there is one.

Broadfoot and Bennet (2003, p.3), stated that the "studio is well established as a physical place and a unique pedagogic method", which are usually problem-solving settings where "educators who are experienced in the act of design tutor students individually or in groups". In the studio, the problems set for the students are "wicked", at least in part ill defined, uncertain or incoherent. The answer is unclear and changes in the process of searching for it. The instructor is the one who raises questions and presents dilemmas to the students. Then, the students are required, according to that feedback, to debug, elaborate and refine their project.

Schön also emphasizes the importance of the studio teachers and their professional and coaching skills:

> A reflective practicum is unlikely to flourish as a second-class activity. The professional schools must give it high status and legitimacy… Coaches must be first-class faculty members, and criteria for recruiting, hiring, promotion,

> and tenure must reflect this priority. Moreover, the process of coaching and the learning experiences of the practicum must become central to the intellectual discourse of the school. (Schön, 1987, p. 171).

Hokanson (2012, p.74) stated that “Studio-based education is learning through designing, a complex and rich form of experience. This complex form of instruction includes learning through applying analysis, synthesis, judgment, and arguing ideas”. He also stated that “instead of a classroom experience that includes lecture as a common means of distributing information, the studio concentrates on direct interaction between learner and instructor, and in some views, between student and master“.

Critiques are an essential pedagogical tool in studio, as previously stated by Schön (1987), and these are based on the instructor’s expertise and professional experiences. Hokanson (2012, p. 71) provided a more recent view of critiques, its importance, types and definitions. “The critique methodology and practice is how design skills are developed around the world within a studio […] In itself, it is a challenge to the designer’s abilities”. For him, critique means a systematic and objective examination of an idea, phenomenon, or artifact (Hokanson, 2012, p. 74) and it may take place at any time in the sequence of a project, encouraged by the open nature of the studio environment. About the three styles of critique cited by Schön, design reviews, desk crits, and design juries, they were defined as:

> “Critique” is often shortened to “crit”. […] The term “crit” will be used to describe individual or small group critiques; “desk crit,” a central element in critique will be used to describe one-on-one sessions generally between learner and teacher, and student and critic. Final, formal, summative critiques are called “final reviews,” “juries,” or “final critique.” (Hokanson, 2012, p. 75).

He cited a list by rough level of formality of different forms of critique provided by Blythman et al. (2007) that include: peer crits, desk crits, online crits, formative crits, seminars. In his work he examined the final critiques, desk crits and peer crits and about peer crits argued that, although they are the least formal of the critique formats they are the basis for an extended professional understanding of the use of critique. In addition, at the same time that they can provide an external review of design decisions, also provides the critic with the opportunity to expand your own critical skills and the ability to review validity and logic of a specific design idea or set of design options. That is, any criticism develops both the critic and the designer.

Lastly, he stated that “much of the modelling of cognitive apprenticeship directly correlates to the individual critique, and to some extent, to multiperson generative

group critiques and seminars", and the peer crits may have comparable cognitive value as a significant component of some methods of cognitive apprenticeship.

In their critique studies, Oh et al. (2013, p. 305) identified eleven factors of design critique, and divided these factors into two groups: methods and conditions. They defined methods as the various ways that studio teachers use to convey their design knowledge and skills to their students and conditions as the contexts in which critique occurs. They considered the five factors of methods are as critiquing settings, teacher-student relationships, communication modalities, delivery types, and delivery. Additionally, the six factors of conditions are design phases, individual differences, knowledge/experiences, student response types, design artefacts, and learning goal.

He argued that design education researchers refer to several types or settings of critiques, which the instructor uses to interact with students. In his point of view, was Bailey (2004) that provided the most comprehensive list of them: desk crit, group crit, interim review, final review, and informal interaction (Oh et al., 2013, p. 306). Bailey´s findings were the result of his analysis of the history of architecture education considering his interpretation of Schön's observations. However, in design studio, the projects were individual, then not all these types or settings of critique necessarily occur in a software engineering studio, where usually there are group projects. Considering this list, the definitions are:

- Desk crit is an individual critiquing session involving an instructor and a single student, usually at the student's desk. It can occur along the entire period of a studio course.
- Group crit engage a small group of four to six students, frequently scheduled once a week. In this session, students expose the work on the wall or around the student´s desk and discuss the student´s presented work.
- Interim reviews involve the entire class at key milestones during a studio project. These reviews occur when the instructor realizes that all students can benefit by sharing their progress and knowledge with others in the class, or when many students have similar problems or opportunities in their projects. Usually, there are more than one interim reviews, the first often occurs after students have performed an analysis of the requirements of a project. Another common time for an interim review is as students prepare for their final review at the end of the studio course.

- Final reviews are sometimes formal like a ceremony or ritual, because it occurs at the end of the course and external critics are often invited. At the formal final review, students usually present their projects and the jury comment on each work publicly. The jury can directly teach individual students by discussing and evaluating their designs.
- Informal interaction is informal discussion among the students that often occurs into studios, therefore they are organized to embrace them. Furthermore, as studios are open spaces, students naturally monitor each other's progress, comment informally on each other's work, compare approaches, share and exchange technical knowledges, or even learn certain skills with others.

Another important point is that informal critiquing sessions tend to be more constructive and formal critiquing sessions more evaluative.

2.5 Software Studio

A software studio is an attempt to utilize similar environments to that of a design studio but for software related disciplines. Many names have been given to these studios, which include software studio, software development studio, and software design studio.

Studio-based approach emphasizes the development of reflective skills and sensibilities. "The essence of the studio concept is 'reflective practice'" (Tomayco, 1991, p. 301).

The studio-based learning model incorporates some elements of cognitive apprenticeship with components of problem-based learning and creativity-enhancing strategies: "The studio is a method of teaching problem solving that has worked in these other fields. It develops talent, rather than suppressing it […] It is a method of personal reflection and external criticism" (Tomayco, 1991, p. 301).

Kuhn (1998) referring to the MIT creation of a software design studio course in 1995 said that this course offers to the practitioners "a provocative look into a different culture of design and of design education, one inspired by architects' professional practice and system of education". She concluded that software studio helps to reflect on the strengths and weaknesses of current software design practice, beyond it encourages us to consider what we might want to borrow from the culture of architecture to promote good design of software and systems. In addition, she stated,

“The studio format supports a relationship with a real client and introduces students to user participation in the design process” (Kuhn, 1998, p 70). Yet, Kuhn (2001, p.349) argued, “Experience from a studio course in software design provokes creative reflection on engineering design education, and on how it may be improved”.

Regarding to software engineering education, Kuhn et al. (2002, p. 236) stated that the studio approach is also suitable educational environment for design studies in general.

In 2013, to facilitate and guide the creation of software studio as a learning practice environment, Bull et. al. defined a software studio framework with categories and parameters, as represented in Table 2.1, that are: Physical Environment, Facilitation of Studio, Mode of Education, Awareness, Critique, Culture, Individual’s Characteristics, Inspiration, Collaboration and Digital Technology. The framework is a result of interviews with studio educators from the disciplines where studio education originated, and subsequently it ‘was performed an analysis of transcript interviews to find out what people native to studio education think it consists of. This framework is represented at Figure 2.1.

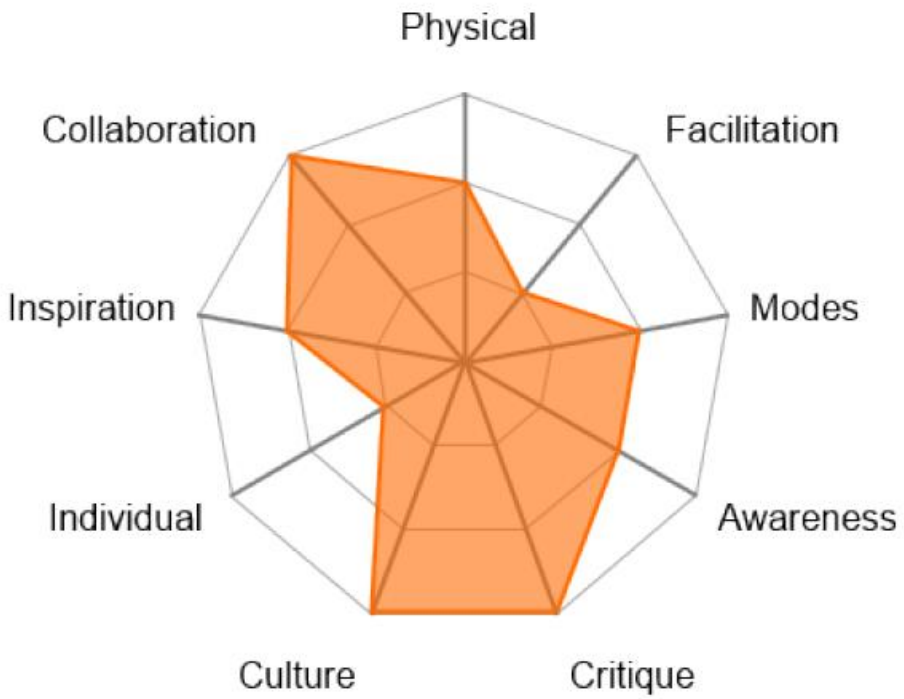


Figure 2.1 - Representation of Studio Framework (Bull, 2013, p.1071)

Interpretations of studio education for software disciplines (Computer Science, software engineering, information technology etc.) have been explored for nearly 28 years, with the earliest known software studio implemented at Carnegie Mellon University (CMU) in 1990 (Tomayco, 1991). This institution offers a Master on Software Engineering program with two years of studio experience.

Worldwide, there are other studio-based approaches in Australia, USA, Poland, and United Kingdom, implemented in different ways, as a single discipline, as a studio course or as some semesters of studio experience integrated in the curriculum of undergraduate or master's degree (Bull and Whitlle, 2014).

Table 2.1 - Categories and Parameters of Studio Framework - adapted from Bull and Whittle, 2014, p. 76

SOFTWARE STUDIO FRAMEWORK	
Categories	**Parameters**
Physical environment	The room needs to be supportive of the categories in this list by generally being open and reconfigurable, providing students with control of the room, and also providing opportunities for a variety of group, individual and social spaces.
Facilitation of studio	This relates to how the studio is managed. The students should be encouraged to use the space as they wish – encouraging a sense of ownership. Rules regarding the use of the space should not be restrictive, e.g. 24 hour access and allowing food and drink. Further, there should be small groups of students (approximately 10), and high availability of staff, encouraging richer interactions.
Modes of education	A studio should provide a variety of education methods. Teaching staff fall into a coaching/mentoring role. There is a large emphasis on the self - learning process, supported by peer-learning elements, and further supported by flexible and impromptu teaching.
Awareness	Studios should support greater awareness amongst its students. Visual work is recommended, as well as placing work on display (as work -in-progress or final products). Visibility of work helps students see other 's work, improves capability to reflect, and increases and improves social interactions.
Critique	This is an important part of reflective practice. Critique is used for providing feedback and developing ideas. It occurs in multiple formats (formal and informal, group and individual) and should come from peers (e.g. peer-coaching), as well as staff.
Culture	Widely agreed as the most important aspect of studio education. A studio culture should be social and foster a sharing culture, and yet sensitive to supporting a good work ethic – which also helps support peer-learning elements. Students' attitudes should point towards treating the studio like a second home. Serendipitous interactions are also very important.
Individual Characteristics	Despite the studio often being described as open and for groups of students, the studio should support the students as individuals too. This is achieved through offering private and quiet spaces, and also allowing and encouraging personalization of space.
Inspiration	When designing, students should be encouraged to be creative in their designs and solutions, which is helped by supporting inspiration. This is improved by students being in close proximity with each other and allowing the studio to be playful. Having the studio contain extra materials or media relevant to their work can also help.
Collaboration	Collaborative activities are common in studio education. To better support collaboration a studio should support spaces for organised and impromptu collaboration, and also contain equipment to support these interactions
Digital Technology	Studios do not require digital technology; whilst all of the other categories refer to aspects that should exist within a studio, this one is a warning about the use of certain digital technologies potentially diminishing the studio; e.g. reducing social interactions and visibility of work. However, it can improve access to work.

2.6 Related Works

Related works of software studio experience has been found in the literature, some of them presenting a report of software studio implementation, concepts and definitions, as previously presented, while others refer to the learning experience, as can be seen in this section.

A studio-based teaching and learning model in IT - Monash University (Australia)

Carbone and Sheard (2002) reported on first year students' reactions to four aspects of the new learning environment, which are physical space, the new teaching approach, IT facilities and the new assessment method. This space is a studio approach established in 2000 in the School of Information Management and Systems, of the Bachelor´s in Information Management and Systems (BIMS), at Monash University. The studio-based teaching and learning approach adopted it was based on the Bauhaus School of Design's model for teaching and learning, which required a radical change from the traditional teaching model, which is based in lecture theatre, tutorial room, and laboratory environments. To build the new design of the physical teaching and learning space they count on the contribution of educational developers at several Australian Universities.

The teaching and learning philosophy behind the studio was to provide students with an opportunity to develop strategies to cooperate and collaborate yet be individual. The findings of this study are that in general most first year students enjoyed learning in the studio environment and an unexpected finding of the study was the evidence of student developing metacognitive skills. The research highlighted four aspects of learning environment which should be considered when constructing them and they also had shown that these issues affects the students' level of satisfaction with their learning. Thus, it is intended the results act as a guide for other institutions planning to implement a studio-based approach.

Baker, Van Der Hoek - 2009 - An experience report on the design and delivery of two new software design courses

Baker and Van Der Hoek (2009) described the implementation of studio-based learning in the senior capstone of the Bachelor of Science in Software Engineering program at Monmouth University. They had emphasized how in the process of working on their projects, students were exposed to industry practices regarding soft skills such as teamwork, interdisciplinary teams, communication skills, and life-long learning skills. In addition, they had compared their implementation with the framework proposed by University of Lancaster and concluded that it brings a new element to Studio-based Learning, as it introduces a learning process almost entirely student-driven, guided only by an outline framework.

Poznan University of Technology (Poland)

Kopczyńska et al. (2012) described the process of implementation of the software development studio at Poznan University of Technology and the way that it supports practicing some typical roles defined in software development methodologies, i.e., project manager, analyst, architect, and software developers. The studio is included in both Bachelor (B.Sc.) and Master (M.Sc.) curricula of Computer Science studies at Poznan University of Technology.

Gaining hands-on experience via collaborative learning: Interactive Computer Science Courses - University of Aizu (Japan)

Danielewicz-betz and Tatsuki (2014) performed an analysis of the practical outcomes of Software Studio in an undergraduate course and in a graduate Software Engineering for Internet, which focus was to analyze the interaction between students and customers to determine how and to what degree the students transform through project based collaborative learning. In their study, during the final self-reflection the students reported that they improved their project management, communication, presentation, writing, business, and software development skills.

Although this study took place in a Software Studio, as its concerns to the collaborative learning with focus on the relation of the students and customers, it differs of our proposed study that focuses on outcomes of reflective practice that concentrates in the relation between the instructors and students.

Promoting creativity in the Computer Science design studio - Ethnographic study

Cennamo et al., (2011) drove an ethnographic study to examine studio-based courses in the traditional design disciplines of architecture, industrial design and human-computer interaction (HCI) classes, in order to identify techniques that can lead students to develop innovative design solutions and how studio-based activities guide students to creative insights in their works.

In their study, four barriers to creative thought were observed in the HCI classrooms and identify ways that the architecture and industrial design instructors helped students to overcome similar challenges. They concluded that the studio method is gradually becoming an accepted method of teaching design skills to Computer Science students. By the end, “projects and associated critiques can be used to encourage students to think broadly and engage in divergent as well as convergent thought, so that, as designers, they are able to creatively ‘put things together and bring new things into being’”. (Cennamo et al., 2011)

Software Studio: Teaching Professional Software Engineering

Nurkkala and Brandke (2011) described a curriculum model designed to train students as professional software engineers in a studio-based learning (SBL) at Taylor University, from reflecting on the missteps and successes in their implementing it over the past five semesters. As a result of a partner with real-world customers to create production-quality software, students had a clear sense of mission and purpose. In addition, some students had graduated and assumed industry positions and the feedback from graduates and their managers regarding the Software Studio experience and its efficacy in preparing students to work as software professionals has been uniformly positive.

Talking about Code: Integrating Pedagogical Code Reviews into Early Computing Courses

Hundhausen et. al. (2013) searching for an answer to: ‘How to foster the development of soft skills in computing education?” developed an active learning approach for computing education called the *Pedagogical Code Review* (PCR). Inspired by computer degree programs, where a common approach to teaching soft skills is to have students work in teams on senior capstone software development projects; and inspired by observations of colleagues in architecture and fine arts education, which applied the *Studio-Based Learning* (SBL) method educate their

students in a completely different way—one that seamlessly blends opportunities to develop both technical and soft skills. PCR is an adaptation of the “design crit” component of the studio-based learning model—can effect positive changes in these attitudes, while also providing students with opportunities to practice soft skills by engaging with experts and peers in multilevel discussions of coding practices.

They had suggested that the design of early computing courses need to rethink in the growing importance of “soft skills” in the computing profession and argued that experiences requiring soft skills ought to be an integral part of the early computing curriculum, in order to provide students the opportunities to build a positive sense of community, peer learning and self-efficacy, not only to give the skills more time to develop.

Things coming together: learning experiences in a software studio - Ethnographic Study

Prior et al. (2014) made an ethnographic study in an undergraduate software studio prototype with two student groups and their mentors and had evidenced that software studio provides learning that genuinely prepares students for professional practice.

> Learning that entails dealing with complex technical problems and tools. Learning that involves working effectively in groups. Learning that results in the building of students’ self-confidence and the conviction that they can successfully deal with the challenges of modern software system development. Learning that allows the accomplishment of the more elusive professional competencies. (Prior et al., 2014, p. 134)

The project goal was to develop a system to track feral animals for a state Wildlife and Parks department. The students used agile SCRUM approach for development and an industry mentor participated in the weekly studio sessions along with the academic mentor. This mentor had a consultative role, answering students’ queries on development issues or related to architecture, scalability and usability.

The first finding of this study was the collaborative learning appeared as one of the most significant characteristics of a studio environment, “with students working out how to do things and to develop their own skills by learning together and from each other”. This occurred into the groups or between groups, moreover, each group appeared to be genuinely interested in the work of the other groups.

The second finding refers to the holistic nature of the learning experience in the studio, called ‘things coming together’. The studio prototype could be seen as a network of people, software tools, subject policies and procedures, a development methodology, processes, techniques, documents, practices and products. It is a virtual network, not static and nor pre-configured, and it is continuously and dynamically reconfigured over time. Whence, all these elements interconnect dynamically providing a network or web in which software development knowledge and skills are co-created.

Whilst this is an ethnographic study like our research proposal, the focus of it diverge from ours, which is to analyze the reflective practice in the relations of instructors and students, peer-to-peer.

Interdisciplinary Projects in the Academic Studio

Gestwicki and Mcnely (2016), defined and described an academic studio, as a model for university courses that brings together students, faculty, and community partners to engage in product-oriented and authentically academic inquiry. This model was developed: for students to build disciplinary knowledge and skills; for students to develop multidisciplinary collaboration skills including communication, coordination, estimation, and empathy; to connect students, faculty, and the university to the wider community; and to address interdisciplinary research questions.

This model incorporates agile software development practice with situated and cognitive theories of learning to produce a unique educational experience. The academic studio is a constructionist approach through which students and faculty, in collaboration with a community partner, collaboratively create artifacts as a way of addressing an open-ended academic question. The incremental and iterative practices adopted by the academic studio are particularly useful in educational game development—a contemporary, interdisciplinary problem domain that is motivating to both students and community partners.

Among the conclusions, stand out that the academic studio encourages students to learn many of the skills championed by liberal arts education and industry, including critical thinking, communication, collaboration, and persistence. It also serves as an opportunity to reflectively practice and build competence in disciplinary skills; both students and potential employers value the balance of theory and practice.

Challenge Based Learning Applied to Mobile Software Development Teaching

Binder et al., (2017) presents the use of the active learning methodology Challenge Based Learning (CBL) for the teaching software development for mobile devices. Besides, the complex environment in which the project was inserted, and that during the project some difficulties appeared and were solved, related to monitoring the CBL method and in the construction of applications. By the end, CBL showed to be an interesting active learning methodology for teaching Mobile Software Development and promising to be applied in other areas.

Acquiring professional software engineering skills through studio-based learning

Rosca (2018) described the implementation of studio-based learning in the senior capstone of the Bachelor of Science in Software Engineering program at Monmouth University, which consisted in two-semesters experience during the senior-year. Students on software project are exposed to real-world environment and also industry practices that develop professional skills, such as working in groups, develop good communication skills, acquire strong life-long learning skills, and be able to function in interdisciplinary teams. It was made a comparison from this approach with the framework defined by the University of Lancaster researchers, and is believed that this implementation brings an element of novelty to SBL by introducing an almost fully student led learning process, with only an outline framework to guide them.

What is the Effect of a Software Studio Experience on a Student´s Employability?

Prior et. al. (2019) described a study based on open-ended interviews and ethnographic observation to understand from students' point of view how their participation in a software studio contributed to employability. They concluded that the studio provides collaboration, communication, project management, supporting each other to resolve technical issues, seeking help from industry mentors and academics, social aspects of work, reflection skills and technical skills which were important skills for employability. These findings seems to give good coverage of employability skills when compared with the Career EDGE Employability Development Profile.

2.7 Conclusions

This chapter presented the literature review to support the understanding of relevance of this study and to provide background for the research process. Such, it was described the importance of practical learning education in software engineering, had been introduced the concepts of reflective practice approach, the concept of studio education and software studio, its application and related works.

CHAPTER 3 - RESEARCH STRUCTURE

"True wisdom comes to each of us when we realize how little we understand about life, ourselves, and the world around us."
Socrates

The aim of this chapter is to provide the background of the chosen research method, in order to define the resources that will be used, as well as, the steps to guide the execution and the research´s data analysis protocol.

3.1 Research Classification

This study focuses on understanding the reflective practice use in a software development studio, how it is being applied, what are the results and the contributions of its use for software engineering, to the practical learning and to the students. So, according to the research objectives, this can be considered as a descriptive and exploratory study. Considering the data features and the data collection procedures this can be classified as a qualitative study, which consists in observing a specific phenomenon in a group, using the ethnography as the qualitative method. Ethnography was the method used for data collection and the Cycle Coding Method proposed by Saldaña (2010) for Data Analysis with support of Atlas.ti Software was used for qualitative data analysis.

3.2 Research Method

3.2.1 Ethnography

Ethnography is the study of social interactions, behaviors, and perceptions that occur within groups, teams, organizations, and communities by observations during a period. Bronsilaw Malinowski (1884–1942) introduced this term in 1922. This method emphasizes the study of ethnic rituals and practices, and studies that describe and explain a range of social phenomena within various culture-sharing groups. It provides

an in-depth description and analysis, of the ways in which culture-sharing groups interpret their experiences and create meaning from their interactions.

According to Spradley (1979) ethnography requires three information sources: what people speak, how people act and what people use.

The key features of an ethnography study are:

- A strong emphasis on exploring the nature of a particular social phenomenon, rather than setting out to test hypotheses about it;
- Primarily there is a tendency to work with “unstructured data”, that is, data that have not been coded at the point of data collection as a closed set of analytical categories;
- Investigation of a small number of cases in detail, perhaps even just one case;
- The analysis of data that involves explicit interpretation of the meanings and functions of human actions; the product of this analysis primarily takes the form of verbal descriptions and explanations;

Ethnographic studies typically gather participant observations and interviews. Through using these methods, ethnographers can immerse themselves in settings and can generate rich understanding of the social action that occurs, it means that they need direct engagement and involvement with the world they are studying. It is important to consider that the participation in the lives of the people under study requires maintenance of a professional distance to allow adequate observation and recording of data.

> Participant observations characterize most ethnographic research and is crucial of effective fieldwork. Participant observation combines participation in the lives of the people under study with maintenance of professional distance that allows adequate observation and recording of data. (Fetterman, 2010, p.45).

During their observations, ethnographers use informal or conversational interviews, which allow them to discuss, probe emerging issues, or ask questions about unusual events in a naturalistic manner.

According to Spradley (1980), in their work ethnographer may consider the following observational dimensions:

- Space - Physical layout of the place(s);
- Actor - Range of people involved;
- Activity - A set of related activities that occur;

- Object - The physical things that are present;
- Act - Single actions people undertake;
- Event - Activities that people carry out;
- Time - The sequencing of events that occur;
- Goal - Things that people are trying to accomplish;
- Feeling - Emotions felt and expressed.

The use of participant observation enables ethnographers to “immerse” themselves in a setting, thereby generating a rich understanding of social action and its particularities in different contexts. Participant observation also gives ethnographers opportunities to gather empirical insights into social practices that are normally hidden.

Because of the relationship the ethnographer shares with research participants, reflexivity occupies a central element in this type of research. The reflexivity is the relationship that a researcher shares with the people and world they are studying. Thereby, in their reports the reflexivity is presented in the form of a description of the ethnographer’s ideas and experiences, which can be used by readers to judge the possible impact of these influences on a study. Regarding this, Flick (2009) explained that in qualitative research:

> The subjectivity of the researcher *and* of those being studied becomes part of the research process. Researchers' reflections on their actions and observations in the field, their impressions, irritations, feelings, and so on, become data in their own right, forming part of the interpretation, and are documented in **research diaries** or context protocols. (Flick, 2009).

Analysis of ethnographic data tends to be undertaken in an inductive thematic manner: data are examined to identify and to categorise themes and key issues that emerge from the data. Through a careful analysis of their data, using this inductive process, ethnographers generate tentative theoretical explanations from their empirical work.

In addition, ethnographic work commonly uses methodological triangulation that is a technique to compare different types of findings. This technique helps a most comprehensive insight into the phenomenon under study. It can be very useful, because sometimes what people say about their actions can contrast with their actual behavior. Such, ethnographers commonly triangulate interview and observation methods to enhance the quality of their work.

Flick (2009, p.101) argued, “Triangulation refers to the combination of appropriate research perspectives and methods that are suitable for taking into account as many different aspects of a problem as possible.”

An ethnography “truth” is a result of triangulation, it means a combination of different data collection´s techniques or methods to construct better conclusions (O´Grady, 2006).

(SHARP et.al., 2016) argued that there were few studies from software engineering perspective using ethnographic methods to investigate software practice and a described study that presents four different roles for ethnographic studies to reach the goals in empirical software engineering, which are: to strengthen investigations into the social and human aspects of software engineering; to inform the design of software engineering tools; to improve method and process development; and to inform research programmes answering specific research questions and complementing other research methods, as code analysis and quantitative studies.

3.2.2 Cycle Coding Method

Saldaña (2016) states that coding process is one of the ways for qualitative analysis, but it is not the unique. He focuses on the coding process in its various forms, drawing from this different coding techniques, suggesting that the choice of one of these techniques should be directly associated to the type of question proposed by the researcher.

A code for qualitative research can be “a word or short phrase that symbolically assigns a summative, salient essence-capturing, and/or evocative attribute for a portion of language-based or visual data” and the data can consist “of interview transcripts, participant observation field notes, journals, documents, literature, artefacts, photographs, video, websites, e-mail correspondence, and so on”. Moreover, he proposed coding the data in two cycles:

> The portion of data to be coded during first cycle coding processes can range in magnitude from a single word to a full paragraph, an entire page of text or a stream of moving images. In second cycle, coding processes, the portions coded can be the exact same units, longer passages of text, analytic memos about the data, and even a reconfiguration of the codes themselves developed (Saldaña, 2016, p. 4).

In these two cycles, there are distinct possibilities of codifying. Before the first cycle, he proposed a pre-codifying stage and a stage of writing analytic memories or

labelled memos. Thereby, it is possible to write your code words or phrases completely rather than abbreviating them to mnemonics or assigning them reference numbers.

> The purpose of analytic memo writing is to document and reflect on your coding process and code choices; how the process of inquiry.is taking shape; and the patterns, categories and subcategories, themes, and concepts in your data – all possibly leading toward the theory. (Saldaña, 2010, p. 32)

Concerning to coding patterns, Hatch (2002) apud Saldaña (2010, p.6) refers to patterns not just as stable regularities but as varying forms that can be characterized by:

- similarity (things happen the same way)
- difference (they happen in predictably different ways)
- frequency (they happen often or seldom)
- sequence (they happen in a certain order)
- correspondence (they happen in relation to other activities or events)
- causation (one appears to cause another)

About coding filters Saldaña (2010, p.6) states that how you perceive and interpret what is happening in the data depends on what type of filter cover the researcher´s analytic lens.

In addition, coding is a cyclical act. Rarely the result is reached in the first cycle of coding data, “the second cycle (and possibly the third and fourth, and so on) of recording further manages, filters, highlights, and focuses the salient features of the qualitative data record for generating categories, themes, and concepts, grasping meaning, and/ or building theory.” (Saldaña, 2016, p. 9).

Such, he argued that qualitative codes are “essence-capturing and essential elements of the research story that, when clustered together according to similarity and regularity, (i.e., a pattern), actively facilitate the development of categories and thus analysis of their connections” (Saldaña, 2016, p. 9). According to him, coding is to arrange things in a systematic order, to make something part of a system or classification, to categorize. Thus, coding is a method that enables you to organize and group similarly coded data into families or categories because they share some characteristic. (Saldaña, 2010, p.9).

Considering the data characteristics and the research objectives, the methods selected for this research, among those listed by Saldaña for the first and second coding cycle, are briefly described below.

Structural coding is one of the Elemental coding methods, which is applied in the first coding cycle. According to Saldaña (2016), elemental coding methods are primary approaches to qualitative data analysis that focuses filters for reviewing the corpus and they build a foundation for next coding cycles. Thereby, the “Structural Coding applies a content-based or conceptual phrase representing a topic of inquiry to a segment of data to both code and categorize the data corpus”, such they are generally foundation work for further detailed coding.

Taxonomic coding is a Procedural coding method that is a second cycle method. Saldaña explains that Procedural coding are prescriptive coding methods and consist of pre-established coding systems or very specific ways of analysing qualitative data. Thus, it is appropriated to analyze the reflective practice.

3.3 Research Context

The object of this research was to observe a project of mobile application development in a software engineering studio of a University Extension Course at Pontifícia Universidade Católica do Paraná through participant observation ethnographic technique.

The project under evaluation was developed in a two months period, and it had environmental constraints and specific customer focus. The researcher planned the schedule for data collection according to the schedule of the software studio mobile application development.

3.4 Research Structure

In observance to the ethnography method, its concepts and the singularities of this study, the research was organized as showed in Figure 3.1.

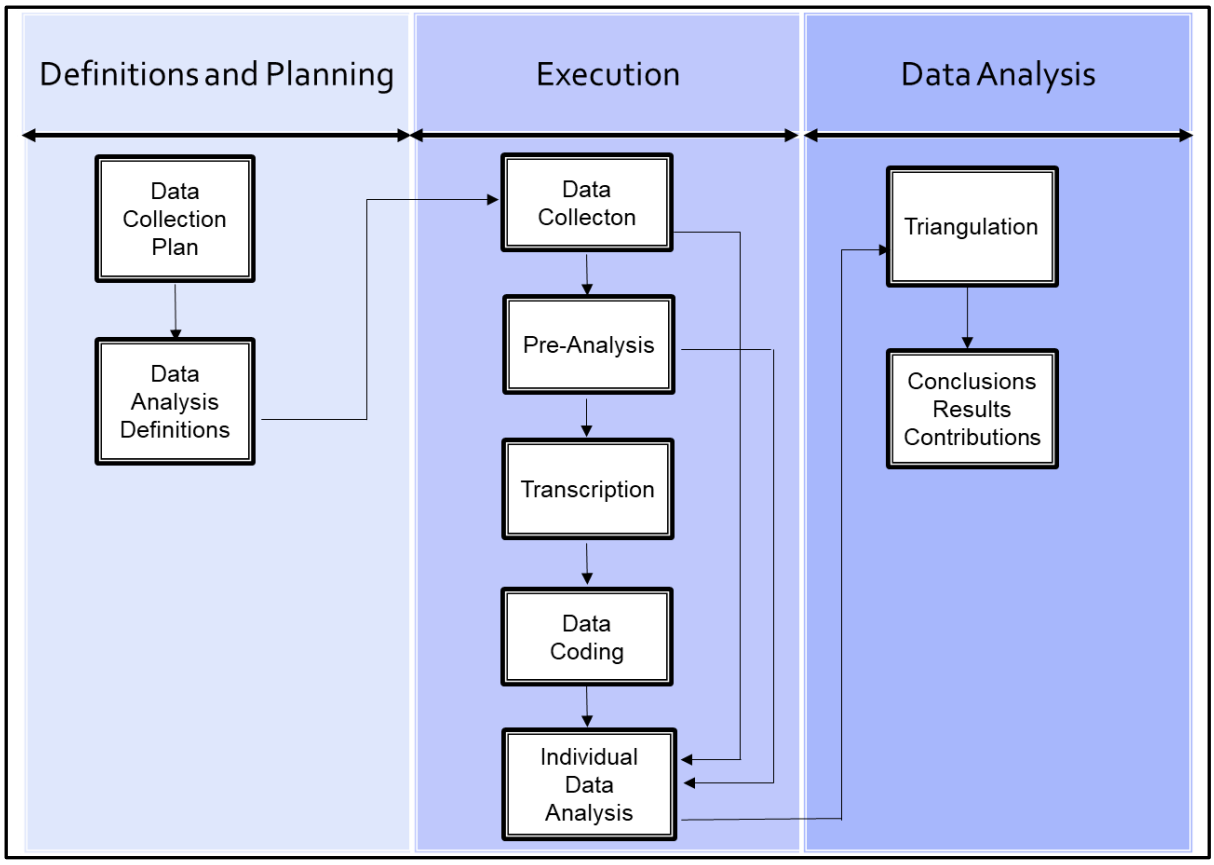


Figure 3.1 - Research Structure

3.4.1 Phase: Definitions and Planning

This phase describes the activities planned for each research step, as well as, what was needed to support them, in terms of environment, instruments, time and resources. It encompasses the steps of Data Collection Plan and Data Analysis Definitions.

Data Collection Plan

Considering the characteristics of the research object, the data collection was defined to occur from three different sources: participant observation along the project development, interviews and written final project reports.

For participant observation it was defined what resources was needed: (i) pen and paper for hand-written notes; (ii) audio recorders; (iii) personal computer for backups of collected data; (iv) availability of time from the researcher to be physically present daily throughout the project development of the two teams.

It was defined that the observation could be active with informal questions directed to the students and instructors along the project observation by the researcher to enrich their analysis.

It was also defined that for retrospective interviews, the researcher could show images and audio recordings of specific situations to the students or instructors to obtain their perceptions about it and, after doing the cross-analysis with their own perceptions.

By the end of the project development, it was planned that the students would write a report using the format of written self-reflections, which is part of the process of project development in this software studio. It was also planned that the researcher would receive a copy of these written self-reflections.

Data Analysis Plan

The necessary steps for the analysis were defined taking into account the research objectives and data collected by different collection methods (participant observation of the project and written self-reflections of the students):

- Step 1 – Data Preparation: After data collection conclusion, the data went through a process of pre-analysis and preparation for analysis, according to the data source. For instance, the audio recordings collected from the participant observation of the project were pre-analyzed, selected for transcription and transcribed into MS-Word digital files. The researcher´s handwritten notes collected during project development were organized and also transcribed to a MS-Word digital file. The students' written self-reflections published at studio web page were also converted into a MS-Word digital file. All files were imported into Atlas;ti software for qualitative data analysis.
- Step 2 – Reflective Practice Analysis: This step focused in understanding the reflective practice in a software studio, considering the Schon´s concepts associated with this approach. Hence, in this step it was important to identify the presence of the reflexive activities: reflection-in-action, knowing-in-action and reflection-on-action and reflective conversations with the material or situation. Thus, the contents were analyzed through the Saldaña Cycle Coding Method and Atlas;ti software for qualitative data analysis.

- Step 3 – Analysis of Contributions to Practical Learning: The focus of this step was to find the contributions of the reflective practice to the software development practical learning and to the students individually. The reflective practice is direct related to the human behavior, evidenced in the relation instructor-student, peer-to-peer or by the student with himself, in way of became a “way of thinking”, that is internalized by them. Besides, the software development is a teamwork, and teamwork is a social process, then social interactions, roles and relationships should not be ignored in the analysis of development activity performed by teams.
- Step 4 – Written Self-Reflections Analysis: The contents was analyzed using the Saldaña Cycle Coding Method and Atlas;ti software for qualitative data analysis.

3.4.2 Phase: Execution

The execution phase consisted of the observation itself as described in the Chapter 4 of this document.

3.4.3 Phase: Data Analysis

Data analysis were performed through the Saldaña Cycle Coding Method, using Atlas.ti as a support tool. Results are described in Chapter 5 of this document.

3.5 Conclusions

This chapter presented the justification for choosing the method and the main concepts related to it. Besides of describing the steps required to develop the research in detail, in terms of definition and planning of the research, execution and analysis, including its individual steps, as well as the methods and tools.

CHAPTER 4 - RESEARCH EXECUTION

"I Love the smell of coding in the morning."
(Peter B. Duffy)

This chapter details the steps of research execution, which encompasses the data collection, the process of preparation for analysis, and the data coding process, according to the previously defined methods and protocols.

The observation was carried out inside studio during the whole two months of the challenge. The researcher remained in the studio in a total of 224 hours. Table 4.1 presents the total amount of hours in each of the research activities.

Table 4.1. Hours spent in collecting, preparing, coding and analysing data

RESEARCH ACTIVITIES		
Research Step	**Description**	**Hours**
Data Collecting	Studio Participant Observation	224
Data Preparation	Data Pre-Analysis: - Audio recording selection - Student´s Self-reflections digital conversion - Typing ethnographic handwritten notes	60
Data Preparation	Data Transcription (Audio recordings of two selected teams)	380
Data Coding	Data Coding Team A (382 quotations)	152
Data Coding	Data Coding Team B (932 quotations)	260
Data Analysis	Data Analysis Studio	6
Data Analysis	Data Analysis Team A	100
Data Analysis	Data Analysis Team B	236
Data Analysis	Data Analysis Student´s Self-Reflections	24

4.1 Data Collection

This section describes the method and instruments used for data collection, the characteristics of the studio observed in terms of environment and working, and the project selected for observation.

The ethnography was performed using participant observation, as illustrated in Figure 4.1. The researcher used equipment and instruments as pen and paper for hand-written notes; audio recorders to record the meetings held during the project, as

planned in data collection plan. The researcher was physically present daily at the studio to execute the data collection in which she spent 224 hours.

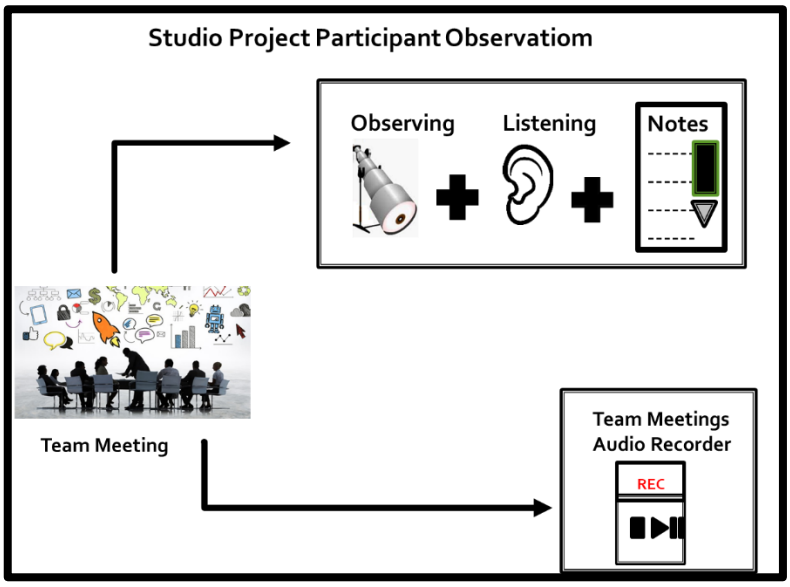


Figure 4.1 - Project Data Collection

Informal questions directed to the students and instructors were audio recorded or handwritten along the project observation by the researcher to enrich their analysis, therefore it was no needed to do retrospective interviews.

By the end of the project development, the students did a report using the format of written self-reflections, which was part of the process of project development in this software studio, as well as to publish them into studio web page. The researcher received a copy of these written self-reflections for analysis.

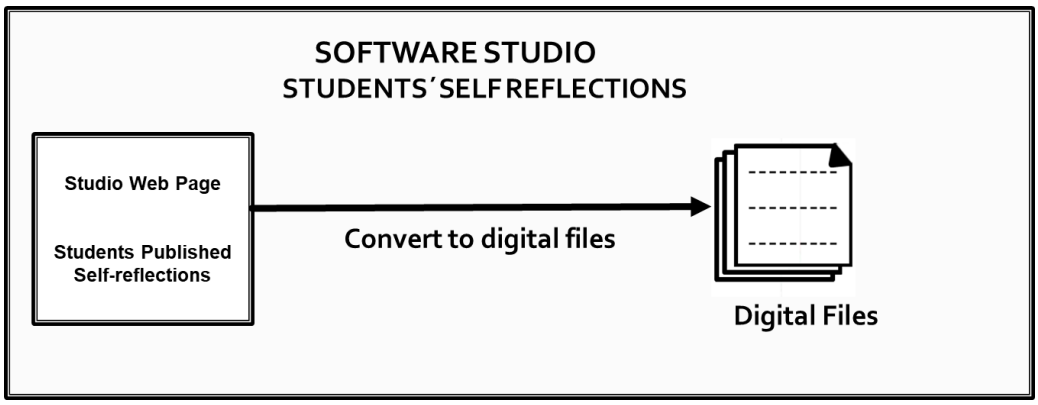


Figure 4.2 - Data Collection - Students´ Self Reflections

4.1.1 The Studio

As previously stated, the studio where this study was conducted is an educational space of a university extension course at Pontifícia Universidade Católica do Paraná, created in 2012. The main extension course purpose is teaching the development of mobile applications in practice.

The studio under observation is called Apple Developer Academy (ADA) and is the result of a partnership between university and industry where mobile applications (apps) are developed according to the partnership industry guidelines and requirements. The partnership states that the studio must apply the Challenge Based Learning (CBL) framework to promote learning while developing apps.

CBL is a framework emerged from the “Apple Classrooms of Tomorrow—Today” (ACOT) project initiated in 2008 to identify the essential design principles of a 21st-century learning environment. The framework objective is to empower Learners (students, teachers, administrators and community members) to address local and global Challenges while acquiring content knowledge in math, science, social studies, language arts, medicine, technology, engineering, Computer Science and arts. Through Challenge Based Learning, students and teachers are making a difference and proving that learning can be deep, engaging, meaningful, and purposeful (Nichols et al., 2016). It can be defined as “[…] a motivational, collaborative and multidisciplinary approach that encourages the use of common technologies for knowledge acquisition and real-world problem solving.” (Binder et al., 2017).

Studio staff consists of six instructors, including four programmers and two designers, which are available daily at the studio, 55 hours a week. For the two-years extension course were selected graduated students or undergraduate within six months, that correspond to one of the following profiles:

- Developers: Student who love developing and customizing their own technologies, with strong logical reasoning, high abstraction ability and lots of curiosity. Faced with a complex problem, he does not settle down until he finds a possible solution. His skills include logical problem solving, data structuring and efficient coding. He can see beauty where others do not see, such as in a well-written code.
- Designer: Student with ability to graphical expression through digital interfaces, able to create visual identities and original navigation structures for any type of interface. On the one hand, it values the usability of the

interface; on the other, he values the branding of experience. His extensive cultural repertoire allows him to create images with various meanings. Caprice and attention to detail are trademarks of his work.

- Devigner: Student able to work with different ways of thinking at the same time. The devigner is aware of the differences and therefore can make connections. For example, it is able to notice the application of software in other areas of knowledge, realizing new use cases. In discussions between developers and designers, devigner tends to position itself as a mediator, being able to use technical terms from both sides to aid mutual understanding. His interest is diverse and covers both programming and graphic design.

In the current cycle, fifty students were selected, including 12 Designers, 12 Devigners, and 26 Developers. They were supposed to dedicate fifteen hours a week to studio. In the first year, students learn about concepts and essential practices for Designers and Devigners. In the second year they learn complimentary practices, besides of workshops about usability, monetization, interface design, game design, and others.

Students are encouraged to develop projects not only to meet the course curriculum, but also to make their products marketable with the support of a university business department. Instructors evaluate the students through a series of presentations and discussions. On the other hand, students continually reflect and revise their projects through the process of working on them and presenting their work, often publicly, and in this case, receiving feedback from the instructor and colleagues, or sometimes external visitors.

Interaction between instructor and student occurs based on the weekly feedback that students get from instructors at different stages of the development path.

The curriculum of the course includes individual and group projects development and proposes to publish the last mobile application developed on the App Store. Each project of the course has different characteristics, customer focus and time for conclusion.

In this context, each project is called challenge. Before each challenge begins, the instructors give presentations on tools and contents related or required for the challenge. This includes technical lectures or workshops, including subjects as, usability, monetization, interface design and game design.

4.1.2 The Challenge (Mobile Application Project)

The project selected for data collection is one of the projects or challenges developed during the two-year course.

The first challenge of the course is individual. For the second and third challenges, groups are formed. The selected project was the third challenge called Mini Challenge, which goal was the development of a game for a target customer chosen by the students with a free theme and short deadline, as demonstrated in Table 4.2. The schedule is defined by the instructors. The staff divided the students in teams, each one composed of 4 to 5 members. The team's composition considered that their members should not have worked together on the previous challenge and that the students had complementary skills. For this challenge it was not defined the devigner as a team leader.

Instructors gave some guidelines for students about the general objectives of the challenge and its goals. The main goal for the challenge was to publish an iOS game, TVOs or WatchOs in the App Store, reduced time to perform a very complex task and teamwork with different people in order to form teams with students that not worked together in previous challenges. In addition, the secondary goals were to make better use of available production resources, to agile parallel development (game design, art, development, business and marketing) and do what was necessary beyond the assigned roles of each to accomplish team goals.

Table 4.2 - Mobile Application Project Schelude

MINI CHALLENGE MILESTONES	
Date	**Description**
October, 16	Beginning of the Challenge
October, 19	CBL Engage – Play concept presentation
October, 20	Week Meeting CBL Investigate – Project Planning
October, 31	Prototype presentation with gameplay - CBL Solution
November, 01	Design Crit Date
November, 10	Send Game to Test Flight and begin the play testing/usability test
December, 08	Submit to Application Store
December, 04 to 7	Presentation to studio partner
December, 05	Work process reflection
* Challenge Goal: Developing of a gameplay	

As one can observe in the schedule presented in Table 4.2, the project lasted 54 days and some challenge´s milestones were defined. In terms of the project development process, the schedule sets 3 days to the design phase, 49 days for the development and testing phase, being 12 days for developing the initial prototype of the solution.

For data collection, the researcher randomly selected two teams of the Mini Challenge for observation, each one with individual characteristics described in next chapter.

4.2 Data Preparation

As the researcher defined to use Saldaña Cycle Coding Method with support of Software Atlas.ti for qualitative data analysis all data needed to be converted into digital files to import into Atlas.ti.

Once the collected data came from distinct sources, data went through different stages of preparation to generate the digital files for analysis as illustrated at Figure 4.3 and detailed in next subsections.

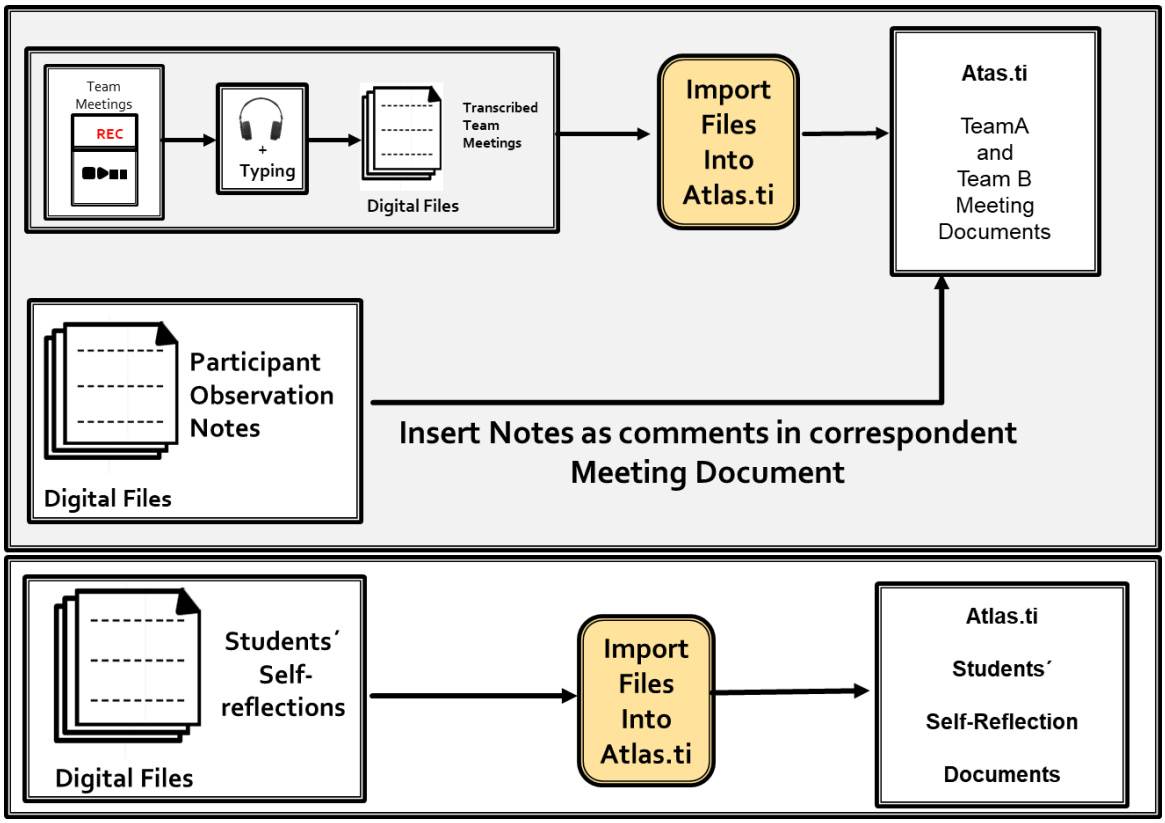


Figure 4.3 - Data Preparation

4.2.1 Data Pre Analysis

To prepare for data analysis, the audio recordings collected from the project participant observation went through a pre-analysis step to select for transcription.

It was selected audio recordings of all the meetings held during all phases of the Mini Challenge project development, from the first project definition meeting until the final presentation of the product. This means that audio recording transcripts include interactions made by students on their own team, with students from different teams, with third-party developers, or instructors during project development.

As the written self-reflections of the students were published into a software studio web page, the researcher had to convert them into a word document.

Finally, the ethnographic handwritten notes collected during project development were typed in word document by date and time.

For this step 60 hours were spent.

4.2.2 Data Transcription

In this step, the audio recordings selected in pre analysis step were transcribed and the notes of ethnographer were typed, so that each one produced an individual word document.

First, the researcher named the two observed teams, as Team A and Team B, then defined a standard for transcriptions in order to indicate and identify the actor of each speech, italics to foreign words and quotation marks for languages expressions. The transcriptions was in Portuguese, because this is the language of the audio recordings.

Writing ethnographic notes and transcribing recording audios took 380 hours and generated 252 pages.

4.2.3 Data Coding

The researcher imported each project document into Atlas.ti Software for coding. The coding process for documents from self-reflections and audio recordings follow distinct paths. In addition, the ethnographic notes where analyzed and inserted as comments or added complementary into the memo writings in the corresponding audio recording document during the first coding cycle.

In the case of the audio recordings, each quotation of a document from a different speaker was analyzed and codified, individually and required different coding cycles. As Saldaña argues “coding is a cyclical process that requires you to recode not just once but twice (and sometimes even more)”, so these research´s coding cycles are illustrated at Figure 4.4.

In the first coding cycle, besides of the addition of the ethnographic notes, it was applied the structural coding to organize the data identifying the speaker (actors of the dialogue), the project phase, studio session and type of interaction.

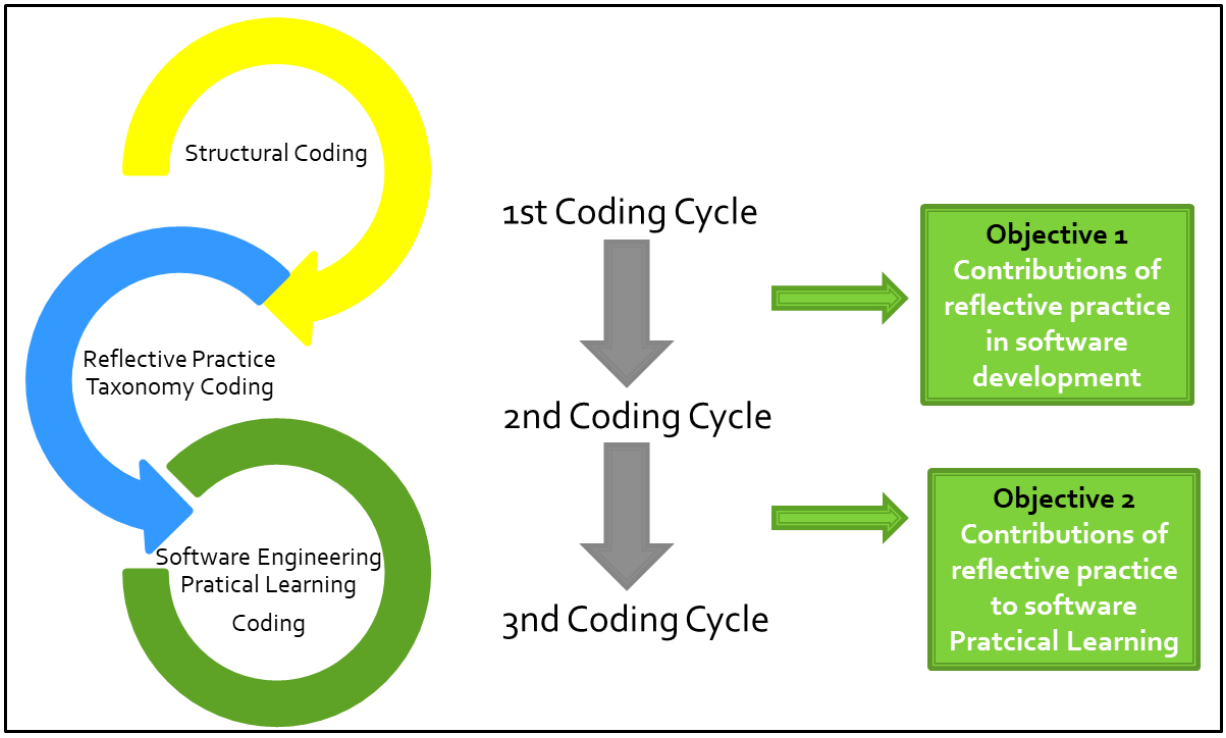


Figure 4.4 - Research´s Coding Cycle

The identified speakers were named one by one, among them, 5 instructors, 4 students of Team A, 4 students of Team B, and 9 students from other teams that made some interaction or reflection in the presentations made by Team A and B or even during of the development of their projects. Although the studio staff has 6 instructors, one of them is responsible for defining the challenges, evaluating the students along the course, the studio modes of education and administrative issues, so is not directly involved with project development.

The types of interaction identified refer to the interaction between the student and the instructor, interaction among teammates, interaction with students from

another team and interaction with third-parties. These interactions were defined and named as follows:

- Student-instructor interaction: are interactions between instructors and students. They were codified as instructor instructive interaction, instructor guidance interaction and instructor reflective interaction. It was considered that instructive is when the instructor adds information about something. Guidance, when the instructor gives guidelines related to software engineering based on the literature or technical knowledge. Finally, instructor reflective interaction is when the instructor provokes the student reflection about the subject;
- Team interaction: are interactions among students of the same team (teammates);
- Student Other Team interaction: are interactions that involves students from another studio team;
- Sound third-party interaction: are interactions performed with the sound third-party developers.

Schön emphasized the instructors reflective interactions as reflective practice, however in the instructor-student relationship other kinds of instructions usually happens, than it was also identified to observing its occurrence and results.

The project phases related to software engineering process were Design, Development, Test and Project Delivery.

The sessions related to software studio were Group Crit, Interim Review Design Idea, Interim Review Design Crit, Final Review, Final Presentation and Peer Critique, defined according to the studio concepts explained in section 2, as follows:

- Group Crits: are studio sessions where instructor and students participate, and they occurred along all phases of project development;
- Interim Review: are project´s presentation sessions that includes instructors and students of all teams, where each team presents its project to other teams, and instructors and students can contribute with questions, suggestion or ideas. The Interim Review Design Idea was a session where teams presented the Project Design Definition and it coincides with the end of Project´s Design phase. The Interim Review Design Crit is a session where teams presented the Project Prototype during the Project´s

Development phase. The instructor that attended this session was an experienced game instructor;

- Final Review: is the final project presentation that occurred when the Test phase is finished with the participation of all instructors and students;
- Final Presentation: is a session where the students presented the final project again, however it included the final customer too, which was a member of the partner´s industry;
- Peer Critique: is the meeting held between the team and a colleague from another team to criticize the idea of team design. This meeting wer

Beyond that, Team Meeting were identified, which refers to the meeting held by members of a team.

The relation of correspondence between the Mini Challenge milestones, studio sessions and project development phases is represented in Table 4.3.

Table 4.3 - Mini Challenge Milestones

MINI CHALLENGE MILESTONES			
Date	**Description**	**Studio Session**	**Project Phase**
October, 16	The beginning of the Challenge		
October, 19	CBL Engage – Play concept presentation	Interim Review	Design
October, 20	Week Meeting CBL Investigate – Project Planning	Group Crit	Design
October, 31	Prototype presentation with gameplay - CBL Solution	Group Crit	Development
November, 01	Design Crit Date	Interim Review	Development
November, 10	Send Game to Test Flight (begin the play testing /usability test)		Test
December, 08	Submit to Application Store	Final Review	Delivery
December, 04 to 7	Presentation to studio partner	Final Presentation	
December, 05	Work on process of reflection	Students´ Self-Reflection	
* Challenge Goal: Developing of a gameplay			

Thereby, in the first coding cycle, each quotation was associated to these previous codes. For the second coding cycle, it was applied Taxonomic Coding using codes from the reflective practice concept: reflection-in-action, reflection-on-action and conversation with the material. The objective of these two initial coding cycles was to analyze the reflective practice, its occurrence by type of interaction, frequency and outcomes rightly related to the individual codes of reflective practice, whose results meet the first research objective.

After that, it was required to execute one more coding cycle with focus in practical learning to analyze the second research objective.

Cycles were revised and discussed with a peer of the research group.

For practical learning´s coding, the researcher observed the data from the point of view of the students' practical learning in software developing process, either individually or as a team. After coding, they are related to competences recommended to be developed in computer science and software engineering curricula, which are required for their professional practice, either technical or non-technical, as personal attitudes. (CS2013, p.15; SE2014, p.23).

CHAPTER 5 - RESULTS

"Success is the result of perfection, hard work, learning from failure, loyalty, and persistence."
(Colin Powell)

This chapter details the data analysis process step by step. The analysis was performed according to the data source and their results were triangulated to reach the final outcomes. So, this session is organized in subsections, as Studio Characterization, Data Analysis of Team A, Data Analysis of Team B, Students Self-reflections Analysis from Team A and Team B and finally the conclusion from the triangulation of the results of each previous analysis and the discussion of results.

5.1 Studio Characterization

The first step of this research was recognizing the target studio as a studio, regarding to Bull and Whittle (2014) characteristics. The results are detailed in Table 5.1. As can be seen, all the listed characteristics were present in the software development studio under study.

Table 5.1 - Studio´s adherence to Bull and Whittle´s Framework (2014)

SOFTWARE STUDIO FRAMEWORK ACCORDING TO BULL AND WHITTLE (2014)		STUDIO ADA ANALYSIS
Categories	**Parameters**	**Description**
Physical environment	The room needs to be supportive of the categories in this list by generally being open and reconfigurable, providing students with control of the room, and also providing opportunities for a variety of group, individual and social spaces.	The room is reconfigurable, enabling students to take control of the room, and also providing opportunities for a variety of group, individual and social spaces. Even the desks are height adjustable to facilitate peer interactions.
Facilitation of studio	This relates to how the studio is managed. The students should be encouraged to use the space as they wish – encouraging a sense of ownership. Rules regarding the use of the space should not be restrictive, e.g. 24-hour access and allowing food and drink. Further, there should be small groups of students (approximately 10), and high availability of staff, encouraging richer interactions.	This studio is managed as a shared space. The students are encouraged to use the space as they wish, with a sense of ownership. The students have no restriction to the use of the space and, as the students are divided by period, usually they respect it, but it is not mandatory. Food and drink are allowed in a specific space dedicated to this. They have shared water, tea and coffee, kitchen appliances, and their own mugs. Furthermore, there are small groups of

		students, approximately 5 or 6, and the availability of staff, encouraging richer interactions.
Modes of education	A studio should provide a variety of education methods. Teaching staff fall into a coaching/mentoring role. There is a large emphasis on the self - learning process, supported by peer-learning elements, and further supported by flexible and impromptu teaching.	This studio provides and allows a variety of education methods, moreover, there is a partnership recommendation to use the Challenge Based Learning (CBL) method. Instructor staff uses and also encourages reflective practice and they are available for coaching/mentoring. There is a large emphasis on the self-learning process, supported by peer-learning elements and flexible and impromptu teaching, as previous teaching of tools and contents required to the challenge.
Awareness	Studios should support greater awareness amongst its students. Visual work is recommended, as well as placing work on display (as work -in-progress or final products). Visibility of work helps students see other's work, improves capability to reflect, and increases and improves social interactions.	Students place work on display using whiteboards, digital screens and Post-it® notes. They write project key definitions and keep the work-in-progress displayed. Visibility of work helps students to see each other's work, improves capability to reflect, and increases and improves social interactions.
Critique	This is an important part of reflective practice. Critique is used for providing feedback and developing ideas. It occurs in multiple formats (formal and informal, group and individual) and should come from peers (e.g. peer-coaching), as well as staff.	Critique is used for providing feedback and developing ideas. There is a variety of critiques, formal and informal critiques, individual or group critiques, private and publicly, by instructor coaching or peer coaching, in desk crit, group crits, interim reviews and final reviews.
Culture	Widely agreed as the most important aspect of studio education. A studio culture should be social and foster a sharing culture, and yet sensitive to supporting a good work ethic – which also helps support peer-learning elements. Students' attitudes should point towards treating the studio like a second home. Serendipitous interactions are also very important.	This studio promotes a sharing culture, it stimulates the values of ethical work, creativity, collaboration, peer learning, and peer working. Students are encouraged to treat the studio like a second home with responsibility and respect to the people and to the common structure.
Individual Characteristics	Despite the studio often being described as open and for groups of students, the studio should support the students as individuals too. This is achieved through offering private and quiet spaces, and also allowing and encouraging personalization of space.	Despite the studio offering open and shared spaces, students respect each other space keeping the environment private and quiet. They have also some private space (meeting rooms) that can be used when needed.
Inspiration	When designing, students should be encouraged to be creative in their designs and solutions, which is helped by supporting inspiration. This is improved by students being in close proximity with each other and allowing the studio to be playful. Having the	Students are encouraged to be creative in their designs and solutions, moreover instructors provide extra materials or media relevant to their work that can also help them. Additional materials available include seven design and business card-based toolkits and a large number of Lego usually used by

	studio contain extra materials or media relevant to their work can also help.	students for proof of concept, development and prototyping. Each student receives, at the beginning of the project, a notebook and a smartphone. The studio also buys any other needed equipment to help the idea development.
Collaboration	Collaborative activities are common in studio education. To better support collaboration a studio should support spaces for organized and impromptu collaboration, and also contain equipment to support these interactions	Despite of students being divided into teams for project development they usually collaborate and interact with each other. The organization space, as well the available equipment, supports these interactions.
Digital Technology	Studios do not require digital technology; whilst all of the other categories refer to aspects that should exist within a studio, this one is a warning about the use of certain digital technologies potentially diminishing the studio, e.g. reducing social interactions and visibility of work. However, it can improve access to work.	It is available a wide variety of digital technology, such as smart tv, smart watch, mobile phone, tablet, notebooks, removable set of smart tv with interactive screen board and 3D laser printing. Students share these equipments and can develop new ideas integrating them. Technology is in the centre of the whole endeavour.

Regarding to the reviews, as previous explained, according to Schön (1987), critiques are an essential pedagogical tool in a studio approach, and according to Oh et al. (2013) it was Bailey (2004) who provided the most comprehensive list of critiques: desk crit, group crit, interim review, final review, and informal interaction, as detailed in chapter 2, session 2.4.

Hokanson (2012) argued that critique is the central education point in design, and its methodology and practice is how design skills are developed around the world within the studio. He stated that “much of the modelling of cognitive apprenticeship directly correlates to the individual critique, and to some extent, to multiperson generative group critiques and seminars”. In addition, he highlighted the peer critiques may have comparable cognitive value as well, as a significant component of some methods of cognitive apprenticeship.

In the studio under study, during this challenge development, there was no desk crit session because it was organized in group of students, then the studio sessions were performed as represented in Figure 5.1. It was possible to observe that Group Crit, Interim Review, Design Critique, Final Review and Final Presentation occurred according previously detailed in section 4.2.3. In addition, the Peer Critique, occurred at Interim Review sessions with the participation of all groups and also privately, when

a student of another studio team criticized a team. The last one was not scheduled, it happened by student's initiative.

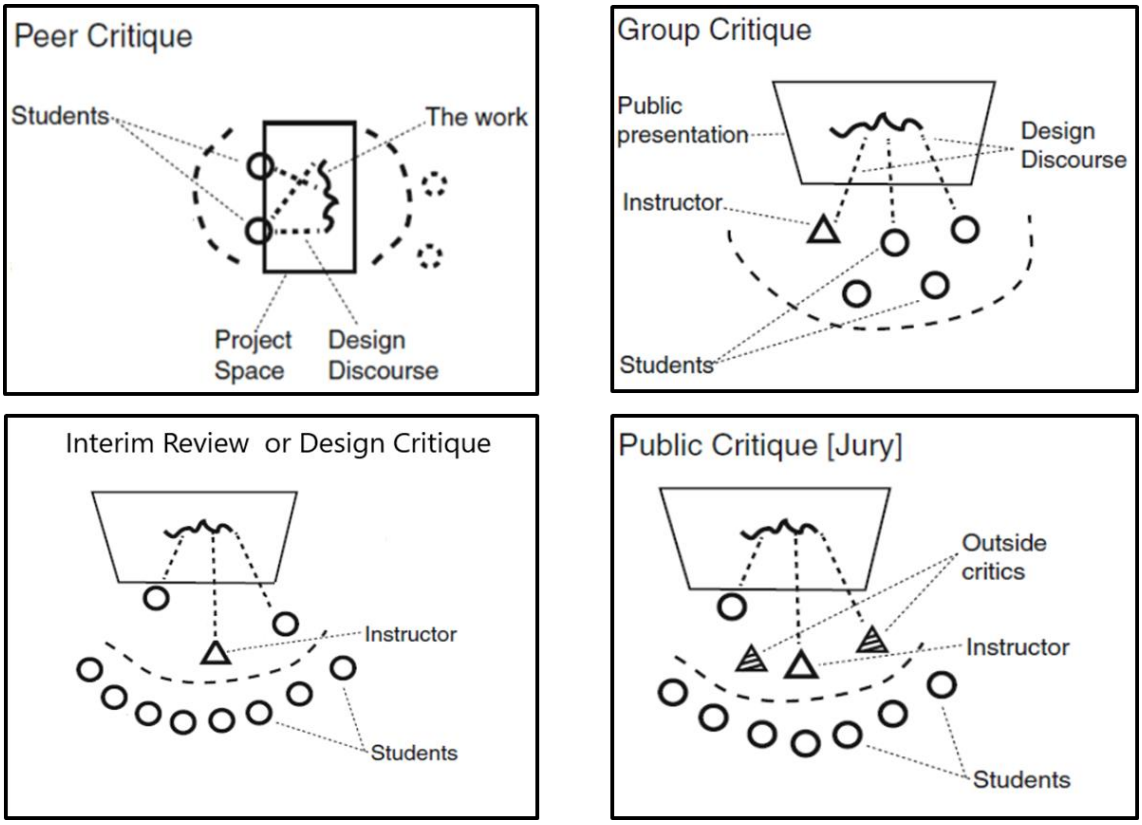


Figure 5.1 - Studio sessions adapted from Hokanson (2012)

In ADA studio, the Public Critique (or Jury) was represented by the Final Presentation. All of them occurred as represented in Figure 5.1.

5.2 Data Analysis´ Codes

Next sections will show our analysis over the data collected in the study. Table 5.2 shows the codes resultant of our analysis and will be related to the excerpts of dialogues that will be further presented, as quotations.

Table 5.2 - Data Analysis' Codes

Code	Description
[Analytical]	When student reports using analysis or logical reasoning.
[Availability]	When the student reports or the ethnographer realizes the quality of being able to be used or to be available.
[Creation]	When the idea result in design creation.
[Collaboration]	When student reports or the ethnographer realizes that two or more people are working together to complete a task or achieve a goal.

[Commitment]	When the ethnographer realizes or student reports dedication to completing a particular task in preference to doing other things the willingness to work hard and give your energy and time to a job or an activity.
[Communication]	When student reports or the ethnographer realizes students are developing the ability to make effective presentations to a range of audiences about technical problems and their solutions. This may involve face-to-face, written, or electronic communication.
[Communication Problem]	When student reports a peer communication problem or when the ethnographer observed it.
[Conflict Management]	When the ethnographer realizes, or student reports approaches to resolve conflicts.
[Correction Request]	When students realize they need to correct something in the product.
[Conversation with Material]	Reflective conversations with the material of a design situation (in this specific case, the software product or its other artefacts).
[Design Learning]	When the ethnographer realizes students´ Design Learning or even when student reports it.
[Decision-Making]	When the ethnographer realizes student are identifying and choosing alternatives based on the values, previous experiences and/or technical experiences. Every decision-making process produces a final choice, which may or may not prompt action.
[Game Assets]	When the subject is related to Game Assets.
[Game Stages]	When the subject refers to the stages of the game.
[Game Development (1st Experience)]	When student reports that it is his first game development experience.
[Game Narrative]	The Game Narrative.
[Game Narrative initial idea]	The Game Narrative first idea.
[Game Mechanics]	When students deal of mechanics of the game.
[Game Final Idea]	The final idea of the game design.
[Game Asset Out of Synchrony]	The game asset out of synchrony with the player touch
[Interpersonal]	When student reports abilities that include being able to negotiate, persuade, and instruct people as well as coordinate our actions with them and read their body language, also known as non-verbal cues. The ability to sympathize and empathize with others and know when something will offend someone are also interpersonal skills.
[Interpersonal Problem]	When student reports or the ethnographer realizes students' interpersonal problems.
[Instructor Instructive Interaction]	When the instructor adds information about something.
[Instructor Guidance Interaction]	When the instructor gives guidelines related to software engineering based on the literature or technical knowledge.
[Instructor Reflective Interaction]	When the instructor tries to provoke a student reflection about the subject.
[Learning by practice]	When student reports learning by practice.
[Learning Experience]	When student reports a learning experience.
[New Idea]	When a new idea emerges.
[Object Size]	Refers to the size of the object on the game´s screen.
[Out of Requirement]	When the product delivered not match with the requirements.
[Planning]	When students were involved in defining what the project will accomplish, when it will be completed, how it will be implemented and monitored and who will do it.
[Personal Learning]	When student reports a personal learning.

[Problem-Solving]	When the ethnographer observe students developing the ability of solving the product problem, identifying it, searching for solution and to take the appropriate action to do it.
[Programming]	When student reports to learn programming.
[Project Management]	When the student reports it or when the ethnographer observes the students developing the ability to use manage the planned activities of the project or managing activities resulted from project required changes along of the development process.
[Project Scope Management]	When student are required to take appropriate action to manage the scope of the project.
[Reflection-in-action]	Reflection-in-action is the reflective form of knowing-in-action, its means, a reflection during the problem-solving process.
[Reflection-on--action]	Activity of reflecting on experience, reflect on past action, thinking back on what we have done in order to discover how our knowing-in-action may have contributed to an unexpected outcome.
[Self-Confidence]	When student report the belief that he can do things well, as well as, the feeling of trust in one's abilities, qualities, and judgement.
[Soundtrack Dissatisfaction]	When students were not satisfied with the soundtrack produced / delivered.
[Sound interface Error]	When students find an error in game sound interface.
[Student Other Team Interaction]	When a colleague of another studio challenge team did an interaction with the students of the team.
[Teamwork]	When Students are working in Team.
[Test Feedback]	Game test feedback.
[Third-Party Instructive Interaction]	Third-party sound developer instructive interaction.
[Third-Party Reflective Interaction]	Third-party sound developer reflective interaction.
[Unforeseen Situation]	When students find a situation that was not foreseen.

5.3 Data Analysis of Team A project

5.3.1 The history of Team A

Initially, the group of students named Team A consisted of four members, including two programmers, a designer and a devigner, called as StudentA1, StudentA2, StudentA3 and StudentA4. As for this challenge it was not defined a leadership role for devigner, then the devigner acted as team member in the designer role only.

Table 5.3 presents the meetings that were held during the project of Team A. The table lists the purpose of the meeting, the main activity, the corresponding studio session, the project development phase, and the meeting date.

Table 5.3 - Team A project´s meetings

Meeting	Objective	Studio Session	Project Phase	Date
Meeting 1	Define Design Idea - Watch Games in search of ideas	Team Meeting	Design	10/16/2017
Meeting 2	Define Design Idea - Informal instructor interaction	Group Crit	Design	10/17/2017
Meeting 3	Define Design Idea - Organize Ideas	Team Meeting	Design	10/17/2017
Meeting 4	Define Design Idea - Instructor Interaction	Group Crit	Design	10/17/2017
Meeting 5	Define Design Idea - Prepare Project Presentation	Team Meeting	Design	10/18/2017
Meeting 6	Interim Review Design Idea	Interim Review	Design	10/19/2017
Meeting 7	Development - Instructor Interaction - Game Mechanic	Group Crit	Development	10/26/2017
Meeting 8	Development - Instructor Interaction - Prototype Test	Group Crit	Development	10/31/2017
Meeting 9	Interim Review Design Crit - Prototype Critique	Interim Review	Development	11/1/2017
Meeting 10	Development - Instructor Interaction - Game player interaction	Goup Crit	Development	11/7/2017
Meeting11	Development - Soundtrack interface problems	Team Meeting	Development	11/8/2017
Meeting12	Development - Instructor Interaction - Scope Reduction	Goup Crit	Development	11/20/2017

According to the mobile application project schedule, the students had only three days to define the game to be developed, so in these days they had all-day meetings that were very rich in terms of interactions, definitions and decisions.

On the first day of the project, students gathered to observe some mobile games in search of an idea using the available resource of TVOs. They decided to develop a mobile application for the iPhone, however, they did not define the type and the theme of the game.

On the second day, each student listed the type of game they would like to do, and the team registered it on the whiteboard. They used the Post-it® notes to organize and sort games by types as thrillers, adventure, musical, puzzles and so on. At this point, the instructor asked informally what kind of game they intended to do and provided some related game references to consider.

After that, students discussed how the game could make people reflect on their own behavior, considering the theme of the game and how many actors the game would have.

Then there was the first session of Group Crit, where the students explained that they plan to develop an adventure game, which it will explore seven deadly sins

and each phase of the game will have a specific demon. The instructor commented that it was an interesting idea and maybe it would be good if the game brought some kind of reflection. The instructor called attention to avoid being tied to just one religion and being careful to respect and try to reach all beliefs.

Moreover, the instructor provoked reflections on the students, such as:

- Does the game give the actor the option to have addictions or vices?
- Can the player determine the fate of the actors of the game who commit sins, for example, can the player send him to hell or to heaven?
- Can the player do something to go to heaven?

Besides reflecting on instructors' comments and questioning, students discussed the work of Dante's Divine Comedy and, by the end, decided to change the game´s theme from the sins to the human vices.

On the third day of the project, each student listed some guiding questions that they had prepared as homework in order to organize the ideas and prepare the presentation for a next day Interim Review Design Idea session.

At Interim Review Design Idea, the team presented their project design idea to peers of other teams and the instructor, that contributed with questions, critical view and giving suggestions about the design idea. This session marks the end of design project phase and the beginning of the project development phase.

During the development phase, students looked for the instructor to talk about the game mechanics, doing a Group Crit. In this specific situation, the interaction was more instructive and guidance than reflective, because they were unknown and had no previews experience on it. They developed a game prototype and did a Group Crit, one day before the Design Critic Session date. In this Group Crit, the instructor tested the prototype and questioned reflexively the students about the game mechanics, design and storytelling. Specifically, the instructor called attention about the number of design assets and feasibility of building them taking into account the project deadline, but students argued that they could manage them without problem, and it would be able to comply with the schedule.

For Interim Review Design Crit, students presented the project idea and its prototype to the instructor and students of other teams. In the beginning of the session, the instructor remembered that there were 6 teams, so the time was about 8 minutes for presentation and 10 to 12 for discussion. The focus of the presentation should be the design of the game and showing the game´s prototype. Teams should explain how

they applied the theoretical concepts of the design in practice. When the presentation finished the instructor prepared the TVs on *Camp Fire* for students to use the interactive online software for questioning, comments and suggestions about the design´s project.

The instructor criticized the project idea and students of other teams contributed with some related game references and suggestions. The instructor that attended this session is the more experienced game design instructor of the staff.

By the end, they reflected on the critiques and questions from this crit session and realigned to continue the project developing. During the development, students needed instructor support on game player´s interaction then in this Group Crit session, the instructor´s interaction was more guidance and instructive than reflective.

At this point, the team received the sound interface developed by a third-party based on the requirements previously sent and unfortunately, realized that the features of the sounds did not match to the defined requirements. So, students reflected on action to understand if they had correctly communicated the requests in the sound requirements document, in order to prepare a change request document and send it to the third-party for correction.

Elapsed ten days of beginning of the tests and therefore, twenty days after presenting the project prototype in Design Crit session, the team realized that they would not comply the deadline unless they changed the scope of the project. Then students looked for instructor to communicate they will reduce the scope of the project. Initially, they thought of developing three stages of the game, but now they estimate they could only develop one stage and change the storyline to make sense. Instructor agreed with them and reinforced that it is better to have one well done stage than three poorly done.

During the test phase, students asked for peers of other teams to test their game and some of them had difficulty and spent more time to advance in the game, then the team done a meeting to reflect on the level of difficulty of the game and on the possibility of inserting tips for the player.

Next sections show the analyses performed upon the data collected from the observation of Team A.

5.3.2 Reflection-in-action

As previously stated, Schön (1987) argued that the fundamental concepts of designing can only be understood in the context of the doing, through the experience of designing and that reflection-in-action was the basis of any design process.

Reflection-in-action stands for the reflective form of knowing-in-action, its means, the reflection during the problem-solving process, or the capacity to respond to surprise through improvisation on the spot (Schön,1987). In reflection-in-action, “doing and thinking are complementary” (Schön, 1983).

In the Design phase, students performed extended meetings to define the project design. In these meetings, except when students were doing brainstorming, they did reflection-in-action in a natural way. At this stage, the more evidenced outcomes of reflection-in-action was the creation, it means that, when students reflected during the action of thinking in the design, they could imagine the product feature previously and decided to change or to improve the initial idea, as a result of this reflection. Thus, a large number of new ideas emerged as a consequence of continuous doing reflection-in-action at this stage. Also, it was possible to observe that reflection-in-action occurred as a result of instructors´ interactions from Group Crit and Interim Review Design Idea session, peer of other teams’ interactions from Interim Review Design Idea and of teammates from team meetings. Some examples of the dialogues that shows these situations will be next detailed.

Observing the interactions of the instructors during the Group Crit, it was possible to notice that they privileged a reflective interaction over instructive or guidance ones. This is in accordance to the software development studio principals. Questions that provokes reflections are preferred over simple instructive moments. The next dialogue (Table 5.4), held in the Group Crit session shows a reflective interaction of the instructor. When students introduced the instructor to the theme of the game they intended to develop, he found the idea very interesting and commented that it would be good for the game to bring some kind of reflection, which resulted in the student's reflection-in-action.

Table 5.4 - Team A Dialogue 1 – Reflection-in-action – Group Crit session

Code	Quotation
[Instructor Reflective Interaction]	“[...] (Instructor1**):** I consider interesting the idea of a game that stimulates sins, however I think it will be nice if the game brings some type of reflection. [...] If you can do something that is ambiguous enough, it will generate an interesting

	debate […] The game will not suggest to commit a sin, but it can give the choice to do it or not [...]. It is a moralistic game, such you can make the choice. […]
[Reflection-in-action]	(StudentA3): I do not know. If you think about the Catholic Church and commit a sin, not always doing these sins, you are screwing yourself. [...]
[Instructor Reflective Interaction]	(Instructor1): No, but I think if you go to this idea of absolute sin, you can 'hum'! Perhaps one way could be relative or changeable sin. Then the game challenge is conquering sins. You will have to overcome and commit sins, this is an entire phase and you have conquer that sin. Then next time this sin could be changed, and you will suddenly choose sin, that you consider a sin.[…]" (Team A Meeting 4)

As students reflected on each other's ideas, a new idea emerged for the theme of the game. They decided to consider Dante's Divine Comedy work as their reference, which deal with vices rather than sins, as can be seen in Table 5.5.

Table 5.5 - Team A Dialogue 2 – Reflection-in-action – Group Crit session

Code	Quotation
[Reflection-in-action] [Creation]	(StudentA1): I was thinking that is a very broad topic. We could make a game where there was a demon of a dead person. And, the demon that tries to leave is the demon of gluttony. It is a fat one. (Laughter) He is already dead. (StudentA2): He is already in the hell. [...]
[Instructor Reflective Interaction]	(Instructor1): Hence, the boss of the stage is a demon. Hence, each stage is a hell [...]
[Reflection-in-action]	(StudentA1): The last one could be proud. [...]
[Instructor Reflective Interaction]	(Instructor1): Exactly! Because you can be proud of not having the others 6 [...]
[Reflection-in-action] [New Idea]	(StudentA3): Divine comedy has 9 [...]" (Team A Meeting 4)

After the Group Crit, at the team meeting, the students did reflection-in-action on this new design idea. The dialogue in Table 5.6 shows students doing reflection-in-action to ensure that the most appropriate terminology was vices instead of sins, taking into account Dante´s Divine Comedy.

Table 5.6 - Team A Dialogue 3 – Reflection-in-action – Team Meeting

Code	Quotation
[New Idea]	"[...] (StudentA1): 7 sins were changed to vices. (StudentA2): [...] Not to get too attached to an idea of religion, because the seven sins are closely linked to an idea of religion. [...] Everyone understood that we changed [...]
[Reflection-in-action]	(StudentA3): Would the right name be vice? (StudentA2): This is how he describes virtue and vice. Virtue is a positive habit and vice is negative. (StudentA3): This I wanted to know. Nice! Perfect then. Because as long as people can do what you said, Dante and all. Because the seven deadly sins ended up delimiting to a Catholic church and its definition, right. (StudentA4): Exactly!

	(StudentA3): So, that is interesting, it opens our minds to see. Because sometimes we find some of the 9 sins interesting, more interesting than one of the 7. (StudentA2): Also, have heresy and lies. [...] (StudentA2): It also has relationship with the Catholic Church." (StudentA4): Our culture is around of Catholic Church. (StudentA3): But the interesting thing about vices is that we can get into the discussion we started yesterday about current vices like this. What would you consider a vice? [...]" (Team A Meeting 5)

In addition, as follows, to define the narrative of the game they reflected on: How could the game stimulate the player's reflection on their own vices? What did psychology have to say about vices? Could vices change depending on culture and geographical region or religion? What would be the most popular vices? Could language´s vices be considered vices? What could be consider a vice in the West or East? What would be the current vices? Could anorexia be considered a current vice? Could a vice lead to death? This can be seen in Table 5.7.

Table 5.7 - Team A Dialogue 4 – Reflection in action – Team Meeting

Code	Quotation
[Reflection-in-action]	[...] (StudentA2): What does psychology have to say about vices? [...] Vices could change depending on culture and region. (StudentA3): Ah, could it be religion, too? [...] (StudentA2): What are the most popular vices? (StudentA1): This thing that StudentA2 talked about, as I am from outside man I can answer: Regionalism [...] (StudentA1): You want a very clear vice. Linguistic vice, man. [...] People from Brazil´s province of Minas Gerais says "Uai" all the time. It is a linguistic vice. But, there is no wrong, it is regionalism. You cannot say it is wrong. (StudentA2): Yes. And there is also a very big change, for example, between what is considered wrong here in the West and the East. There are things completely different. [...] (StudentA3): What are the current sins? [...] An example for better understanding. People talk about gluttony because people eat too much. Today, anorexia is fashionable, right? Like, people who eat nothing. So this would be a sin, like contemporary sin. [...] (StudentA2): What was the profile of a person with vices? [...] (StudentA2): How important is eliminating a vice? [...] (StudentA2): Can people be punished because of their vices? [...] (StudentA2): Can vice lead to death? [...]" (Team A Meeting 5)

Observing the interactions of the instructors during the Design Idea Interim Review session, it was possible to notice that they once more privileged a reflective interaction over instructive or guidance ones. We can see that this is a regular practice in this studio.

In the next dialogue (Table 5.8) the instructor said they should be careful dealing with religion´s cultural symbols to not offend any player and not to be connected to any specific religion. Then, students argued that they wanted to keep distance from any religion to not have controversy. The instructor explained that controversy is not a problem, but considering only one side is, so ambiguity is the best option.

Table 5.8 - Team A Dialogue 5 – Reflection-in-action – Design Idea Interim Review session

Code	Quotation
[Instructor Reflective Interaction]	"[…] (Instructor1): But I think the most important, just to repeat the same thing as the first team. You should be careful when dealing with cultural symbols from religion, because some people may be offended.
	(StudentA2): From our bias, we would take more the occult side, but not very attached to (pause).
[Instructor Instructive Interaction] [Instructor Reflective Interaction]	(Instructor1): Combining different elements from different traditions, mostly traditional ones, like play bags, or things, has another name too. It is pagan traditions. Hence, it does not get any official religion like that. [...]
[Reflection-in-action]	(StudentA3): Actually, we really want to distance ourselves from religions, so as not to have so much controversy. [...]
[Instructor Instructive Interaction]	(Instructor1): The controversy is not bad. The problem is when you are just one side. If you only have one side, then people blow you away. If you have multiple sides, and play in ambiguity, then you can work it out. [...]" (Team A Meeting 6)

Moreover, in this session students of other teams had contributed with reflective questions and giving some game references related to the idea of the presented project for further analysis.

In the following example (Table 5.9), a student from another team (StudentO4) tried to contribute by recalling the question of the indulgence of the Catholic Church as a possibility to add or consider in the game. StudentA2 explained that the goal was to stay as far away from the Catholic Church as possible. Then the instructor remembered that it was not necessary to make a direct reference.

Table 5.9 - Team A Dialogue 6 – Reflection-in-action – Design Idea Interim Review session

Code	Quotation
[Student Other Team Interaction]	"[…] (StudentO4): One thing I remembered now, that is just an idea. It is like those indulgences that the Catholic Church had in the middle ages. Maybe an idea could be you buy something and then your sins are forgiven.
	(StudentA2): We wanted to get away from the Catholic Church, as much as possible.
[Instructor Reflective Interaction]	(Instructor1): But you do not have to make a direct reference. […]" (Team A Meeting 6)

Another student from another team (StudentO9) shared the reference of a game that depicted a hell from which the player should try to escape. The game had interesting scenarios that created a horror environment suitable for the purpose of the game. Therefore, this student tried to collaborate by showing how important the game's scenario, mechanics, and narrative being well connected to achieve the game's goal. This made clear the importance of the interaction with other teams, provided by this studio. This dialogue is shown in Table 5.10.

Table 5.10 - Team A Dialogue 7 – Reflection-in-action – Design Idea Interim Review session

Code	Quotation
[Student Other Team Interaction] [Game Reference]	(StudentO9): Suggestion A game called Agony. He tries to be a representation of hell and you are a being who was there trying to escape. It is an 'outlet' style game, which you have to escape. For example, certain things, as a part of the game that is a fireplace, which is a mouth full of teeth, and the scenario is a whole set of bodies, which is a terribly disgusting thing like that. I think they managed to do well in their goal. […]" (Team A Meeting 6)

Due the project characteristics, mainly the short time to produce a result, the development phase encompassed both development and test. We are calling such a phase simply as Development.

At Group Crit sessions in the Development phase, instructors worked with reflective instructions, instructive instructions and guidance instructions. Despite this, almost all of them led to reflection-in-action.

In one of these sessions, when the instructor finished testing the prototype, he asked the students if they had already considered the feasibility of meeting the deadline, considering the volume and complexity of the assets, as presented in Table 5.11.

Table 5.11 - Team A Dialogue 8 – Reflection-in-action – Group Crit session

Code	Quotation
[Instructor Reflective Interaction] [Game Assets] [Project Scope]	(Instructor1): It's a lot of production, you won't get it at the time of the challenge. It's a lot of assets production. It is a different game style and different mechanics. (StudentA4): No! So, what we have shown so far, which is this room, will be the first phase. There will not be much! (Instructor1): How are you going to solve this? This problem. (StudentA1): The whole first phase it is okay! (Instructor1): No. I'm saying that you have an extremely detailed game, which requires very realistic objects to bring the cases you want. There are many characters that are not trivial, they are characters that mix diabolical characteristics with normal ones. How will you produce all these assets on time? (StudentA2): I guarantee myself.

	(StudentA1): It's kind of when Rodrigo and I finished our models, I even told her [...] I can help. (StudentA2): I do not think asset production is a problem, honestly. (Instructor1): Is it true? (StudentA2): Yes. [...]" (Team A Meeting 8)

Students argued that asset production should not be a problem, but programming should be a more complex task as many mini games must be developed. The instructor agreed that programming could be more difficult, but still believed that building assets would be a problem.

Table 5.12 - Team A Dialogue 9 – Reflection-in-action – Group Crit session

Code	Quotation
[Instructor Reflective Interaction] [Game Assets] [Project Scope]	(StudentA4): The programming, we will make the games. (Instructor1): Programming to me is not looking the hardest, anyway. (StudentA2): Yeah, but there's a lot of programming that will blur. (StudentA1): We have a lot of scene with mini games. Almost our entire script. (Instructor1): I think asset production is going to be difficult. But, if you guarantee yourself. […]" (Team A Meeting 8)

At Design Crit Interim Review session most of the instructor interaction were reflective, and few of them are of the type guidance. Students of other teams contributed with reflective questions. Both, instructors and students of other teams contributed giving some game references related to the presented project for team reflection, as shown in Table 5.13.

Firstly, to contribute to this team, a student from another team (StundentO5) brought in a reference from a puzzle game that he considered interesting for them to watch as they were developing a puzzle game. This game works with images that must be unveiled, for example, the game only widens one eye of a particular statue and, based on that image, the player must find out which statue is.

Table 5.13 - Team A Dialogue 10 – Reflection-in-action – Design Crit Interim Review session

Code	Quotation
[Student Other Team Interaction] [Game Reference]	"[…] (StudentO5): Have you ever played Doubt? (StudentA4): No. (StudentO5): Doubt is a browser game that you have to unravel. It is an image, sometimes it is a crazy image. You put the answer by URL, after the bar, and then you are in and can play only per URL. (StudentA4): It is like NotPron. (StudentO5): Oh, I do not know this one. There is the DNA, which is an organized "roll" that pays the prize, National Academic Challenge. Research it! They make things available from previous years. It is just a few outages, as

	an absurd zoom in on a statue that shows only the eye of the statue and you have to find out what statue it is. […]" (Team A Meeting 9)

This reminded another student (StundentO2) to share with them the reference (Table 5.14) to a puzzle-based game that begins and ends with the release of a book sequence. The game is based on a sequence of seven books, each one is associated to a stone, so each stage of the game has a stone name. The goal is to solve a puzzle and find a right word to post on the game´s site. The first person to post the word won a prize. The puzzles were associated with things of real world, but hey were masked. Then he suggested that the team look at this game and reflect on it.

Table 5.14 - Team A Dialogue 11 – Reflection-in-action - Design Crit Interim Review session

Code	Quotation
[Student Other Team Interaction] [Game Reference]	"(StudentO2): What you said reminded me an interesting event, which in truth, was a game that begins and end in the release of a sequence of books, which was the Eudora line. It is called Eudora Quest, if I am not mistaken. Therefore, it was an event with seven stages because it was seven books, each book associated to a stone. So, there was the amethyst stage, the emerald stage, and so on. However, this quest involved solving a puzzle, finding a right word and posting it on the site. The first person to post the word won a prize, that was 2000 reais at the time or something like that and plus the collection. It was not Eldora, it was Deltora, I remembered now. This sequence of puzzles was like this, everything related to real world stuff, only masked out of the real world. I heard you talking there, zooming in on the eye of a statue. I remember at the time, the first puzzle like that, it was very interesting that the solution of the first puzzle, it was guiding you until you arrived in Mexico. Hence a pun of words that they were giving a clue a day, but the challenge lasted for weeks, until someone thought the answer was La Moneda, because it was the name of a hostel in a city of 3000 inhabitants in the interior of Mexico. (Instructor4): I have read about this, the story is very long. (Laugh) (StudentO2): However, there were seven challenges and each challenge lasted weeks. The latter were getting a little faster, because then people already understood the logic, how the puzzles were built. It was very interesting. And, all puzzles somewhere on the internet you must find them mapped. […]" (Team A Meeting 9)

At the end of this session, another studio instructor who attended this session shared the reference (Table 5.15) of a puzzle-based game whose story was about a guy trapped on an island who was supposed to solve the puzzles to get out of there.

Table 5.15 - Team A Dialogue 12 – Reflection-in-action - Design Crit Interim Review session

Code	Quotation
[Instructor Reflective Interaction] [Game Reference]	(Instructor4): The time is up. Any more suggestion? (Instructor5): A puzzle and exploration game that recently sold well. It's from the same creator of Upgrade. It is The Witness. Of course it is a 3D game, super beautiful and such. (Instructor4): Its puzzles are really cool, right! Well crafted. I like puzzle game that (interruption).

	(Instructor5): It was the same idea. The guy stuck on an island trying to get out. [....]" (Team A Meeting 9)

So, the instructor who is managing the session shared he once read that a good puzzle was like a serial killer who acted with the desire to be discovered. In addition, he instructed the students to be careful not to create something that no one quickly discovered, because a puzzle created for the purpose of not being discovered is a very frustrating puzzle (Table 5.16).

Table 5.16 - Team A Dialogue 13 – Reflection-in-action – Design Crit Interim Review session

Code	Quotation
[Instructor Instructive Interaction] [Instructor Reflective Interaction]	(Instructor4): I read once that a good puzzle is like a serial killer, it acts with the desire to be discovered. A puzzle that is created with the aim of not being discovered, it's a very frustrating puzzle. So be careful not to try to do something that, oh! No one will unravel fast. It just frustrates people, that's fine! Or at least make the progression. (StudentA4): We want to do something simple, but one that needs at least some thought. (Instructor4): Oh, that's right, perfect. And enjoy this abstract level you're playing with. [....]" (Team A Meeting 9)

As noted at team meetings (Table 5.6 and Table 5.7), each member acted individually as a reflection-in-action stimulating agent for the other teammates.

Throughout the development process, new ideas emerged as a result of students' reflection-in-action. Therefore, reflective interactions with instructors, between team members, or with colleagues of other teams lead to reflection-in-action, and the process of reflection-in-action seems to stimulate the emergence of new ideas and thus helps to promote the process of creation.

5.3.3 Reflection-on-action

As previously stated, reflection-on-action refers to reflecting on past experience, it means, thinking back on what we have done in order to discover how our knowing-in-action may have contributed to an unexpected outcome.

Table 5.17 and Table 5.18 detail two examples in which the reflection-on-action took place from the conversation with the material. In the case of software development, we consider material as being the software itself or any other related artefact.

The first happened when the third-party developer delivered the sound interface for testing. As a result of Conversation with the Material (the delivered soundtrack

interface) the team realized that the interface has not meet the expectations of the team neither the submitted requirements. The team did a reflection-on-action to understand if they had correctly communicated the requests (sound requirements) to the third-party developer, before preparing the change request document. They concluded that they have correctly communicated the requirements, then did a reflection-in-action on how to do the third-party understand their requirements. They decided to submit other video examples and more references with comments on the requirements. The quotations are presented in Table 5.17.

Creating sound assets based on descriptive requirements and sound references could not be accurate the first time, because it is difficult to communicate the imagined soundtrack, as well as there are different perceptions between who defines and who develops.

Table 5.17 - Team A Dialogue 13 – Reflection-on-action – Conversation with the material (soundtrack)

Code	Quotation
[Conversation with Material] [Reflection-on-action] [Soundtrack dissatisfaction] [Out of requirement]	"(StudentA2): [...] I want to solve this because I'm not very satisfied. (StudentA4): I'm not very satisfied either. Do the following! Ask him, like it was for 10 seconds, right! (StudentA2): I'll say that and I'll talk too (interruption) (StudentA4): About the music´s "corpocity".
[Correction Request]	(StudentA2): It's to be a similar pace as the reference, slower and darker; (StudentA4): It is darker, with a deeper background and sharper notes. [...] (StudentA1): Send a youtube video of a guy passing like a fork, like on a plate. (StudentA4): Also send a video of the reference. (StudentA2): I will send all.
[Conversation with Material] [Reflection-on-action] [Out of Requirement]	(StudentA1): So man, but it really has nothing to do with reference. [...]" (Team A Meeting 11)

The second situation happened when, in the middle of the mobile application development, the team realized that a scope reduction was needed to meet the Mini Challenge schedule and approached the instructor to explain this, as illustrated in the next dialogue in Table 5.18. It is important to remember that the instructor had already pointed out this situation, as previously detailed, when testing the prototype at the Group Crit session. Instructor in a guidance and reflective interaction, called attention to the number and complexity of the design assets to be built on a short-term project, but at that time, the team believed they could meet the schedule.

Table 5.18 - Team A Dialogue 14 – Reflection-on-action – Development Phase

Code	Quotations
[Conversation with Material] [Reflection-on-action] [Game Stages] [Scope change]	"[…] (StudentA1): So the point is that we are worried about, because we are making the phase very, very good. However, our initial goal were three stages, if we spend (interruption). If we cut to make these two other stages, all three will be poorly made. We cannot handle everything well done. (StudentA2): We prefer to deliver a very round phase, good kind of play. (StudentA1): We are changing the phase 1 puzzles to not give the feeling that something is missing.
[Instructor Reflective Interaction]	(Instructor3): How long do you think you will have to play in phase 1, more or less? How long to pick up the device and play?
	(StudentA2): With the puzzles, are we working on now? (StudentA1): I think like 40 min or not even that. […]
[Instructor Reflective Interaction] [Instructor Guidance Interaction]	(Instructor3): Oh, I think that is right. Then make a round phase to give the taste of the game. I think it is better to have a short round than to have several "half mouth".
[Planning] [Scope management]	(StudentA1): We are thinking of launching as being really teaser and evolving to a phase 2 and 3 later. (StudentA2): And, to leave a type of tip finger when it is over too. When the first phase ends, to be like, there is something else. […]" (Team A Meeting 12) (StudentA1): One thing we thought about implementation was having a character customization, for example when the player would kill the boss, in the previous solution he would get a key, now he gets the key and something to increase his weapon, which is a baseball bat, we were going to put some wall nails, etc.

The solution for this situation first required the team to rethink the scope and the design of the game, in other words, to review the three initial planned stages for the game, to deal with the impacts in the storytelling, to think in a new distribution of the game stages, and to consider the time necessary for development. As a conclusion they had to rethink the scope and change the initial design solution. They presented the possible solution to the instructor according to the dialogue 14 (Table 5.18).

As a consequence of this reflection-on-action they practiced the problem solving, decision-making, planning, project management, time management, scope management, the scope reduction and more than exercising and enhancing those skills, they had a practical learning experience.

5.3.4 Conversation with the Material

As previously stated, conversation with the material refers to reflective conversations with the material of a design situation, in this case, the soundtrack or the game. Conversation with the material can result in reflection-in-action, reflection-on-action, or sometimes both of them, as detailed, in the examples of the last section.

For instance (Table 5.17), when one encountered a product error, the team first did a reflection-on-action to understand what went wrong, and after a reflection-in-action to problem solving, as well as taking the actions needed to do so. On the other hand, the conversation with the material that refers to a conversation with the developed game resulted in a product dissatisfaction (Table 5.18), this led to a reflection-in-action and required a product change, as well as other activities arising from this. In this case, it demanded a product change and of course, a decision-making on what was the better solution taking into account the project´s deadline.

Thereby, throughout the development´s process new ideas have emerged as a result of reflection-in-action of the students. I, as a researcher observed that, at design phase, the reflective interactions led to reflection-in-action and the process of reflection-in-action stimulated the emergence of new ideas and thus it helped to promote the process of creation. At the development phase, the reflection-in-action contributed to problem-solving and/or to improve the solution or final product.

5.3.5 Practical Learning Contributions

One of the research objectives was to analyze the contributions of reflective practice to the development of individual competences and the artistic talent in a software studio, once Schön argued that studio helps to develop them. Some of them were competences or skills required throughout this team mobile application development. In some cases, the students acquired these competences or skills, as they were unknown; other times they practiced and developed them.

Technical research is a technical skill practiced in design and development phases, when they were in searching of references of the games related to the game they will develop as previous shown in Table 5.10, Table 5.13, Table 5.14 and Table 5.15.

Pair programming is another technical skill learned by a student of this team during project development. StudentA4 had never done pair programming before, so this student acquired a new skill in this challenge and during this course of study as well.

Project Management, time management, planning, problem solving, decision-making and scope management are skills required for practicing of software engineering that the students practiced during the project development, as illustrated in Table 5.18. In those examples, the team realized that they would not meet the project

deadline unless they had reduced the scope of the initial project. So, first the team had to make the decision of changing the scope, and then they reviewed the three stages initially planned for the game, thinking about a new distribution of the game's stages and adjusting the story to be consistent with a single stage. Moreover, it was necessary to plan and manage the activities necessary for this change in order to meet the schedule.

For instance, collaboration, oral and written communication, leadership, teamwork, interpersonal savvy and conflict management are skills required for practicing of software engineering too, and they were practiced during the project.

The researcher realized that along the project development the students worked in team and collaboratively. In the design phase, students worked in group immersively, with a complete composition of the team, researching and studying some game references related to the types of games that students would like to develop during the project. They brainstormed ideas collaboratively, and as they discussed the ideas, they wrote them on Post-it® notes and put them on the whiteboard in order to define the design of the project game. In the development phase, they divided the tasks, being that the developers worked on the code at the same time, the designer built the design assets, and the devigner was responsible for documenting, organizing and making available to the team all documents with project definitions, as well as the presentations.

They practiced oral and written communication, either to prepare and/or present the project in the studio sessions, or to communicate verbally or digitally within the team.

Furthermore, throughout the development of the mobile application students demonstrated engagement, commitment, flexibility and adaptability, from team organization (once the students had no worked together in previous studio projects) to technical challenges and new learnings.

The researcher observed that some interpersonal relationship problems occurred with the team during project development. Depending on the point of view, it could be understood as a conflict of leadership or role, or unbalanced distribution of project activities or lack of empathy among team members. It was possible to observe that in some situations, StudentA2 segregated StudentA3, not taking her ideas and opinions into account and leading the other team members to act the same way, isolating her. As a result of the growing conflict, StudentA3 was dismissed at the end

of the challenge. Although this situation is quite common in practical situations at any kind of organization, it was not possible to observe that the studio brought any new tools so the students could learn to deal with it during the challenge and in the future. Despite of this lack of direct instructions, StudentA1 realized during his written self-reflection that he could have taken a different posture and influenced the course of actions, as seen in Table 5.19.

Table 5.19 - Team A – Students´Self-reflections - Interpersonal and Communication Problem

Code	Quotation
[Communication Problem] [Interpersonal]	(StudentA2): Throughout the execution process, we had some difficulties, as well as any team, mainly related to communication.
[Communication Problem] [Interpersonal]	(StudentA3): I know I have a lot of trouble making friends and being sociable, and I've been trying to change that every day. It hurt me so much to know that I hurt a lot of colleagues and that I came up with the idea of being sloppy, lazy and doing nothing. I honestly thought I was doing a good job and found that not too late. It is with great sadness that I leave the Academy but with many great memories, besides the numerous contacts I made here. I can only leave my thanks to you all and apologize again for not being the best member of your team and maybe hindered your work
[Communication Problem] [Interpersonal]	(Student1): However, despite the idea that the game is good and the team's commitment, not everything was like an arrangement of roses in this challenge. We had some internal relationship problems, which caused a certain amount of nerve damage. At this point, I am sorry that I am no longer involved to refresh my spirit and avoid some unnecessary arguments. I could also have been a little tougher on the team's accusations.

5.3.6 Team A summary

Table 5.20 summarizes the results of Team A Mini Challenge obtained from the data coding cycles and ethnographic observation of the studio project development experience for later cross analysis with Team A Self-Reflection results.

We observed that team's reflections-in-action resulted from interactions with the instructors, team meetings, conversations with the material and reflection-on-action. The last one, is justified, for instance, when ones encountered a product error, so first the team did a reflection-on-action to understand what went wrong, and after this a reflection-in-action was taken to problem solving, as well to define the actions that should be taken. On the other hand, the conversation with the material - that refers to a conversation with the developed game - resulted in a product dissatisfaction, this led to a reflection-in-action and required a product change, as well as other related earlier described activities.

Table 5.20 - Team A – Summary of the results

Team A - Coding cycles & Participant Observation Results	
Reflection-in-action	Creation
Reflection-on-action	Decision-Making
	Planning
	Problem Solving
	Project Management
	Scope Management
	Time Management
Conversation with Material	Decision-Making
	Planning
	Problem Solving
	Project Management
	Scope Management
	Time Management
Studio Development Experience	Collaboration
	Commitment
	Communication
	Conflict Management
	Interpersonal
	Leadership
	Pair Programming
	Teamwork
	Technical Research

5.4 Data Analysis of Team B project

5.4.1 History of Team B

Team B was composed by two developers, a designer and a devigner, called as StudentB1, StudenB2, StudentB3 and StudentB4. Devigner acted as designer because in this challenge it was not defined a leadership role for devigner.

The main activities related to the meetings held by Team B during the development process are represented in Table 5.21.

Table 5.21 - Team B Project's Meetings

Meeting	Objective	Studio Session	Project Phase	Date
Meeting 1	Define Design Idea - Watch Games in search of idea	Team Meeting	Design	10/16/2017
Meeting 2	Define Design Idea - Instructor Interaction	Group Crit	Design	10/17/2017

Meeting 3	Define Design Idea - Define Design	Team Meeting	Design	10/17/2017
Meeting 4	Define Design Idea - Third-party Interaction	Third-party Meeting	Design	10/17/2017
Meeting 5	Define Design Idea - Instructor Interaction	Group Crit	Design	10/17/2017
Meeting 6	Define Design Idea - Define Design	Team Meeting	Design	10/17/2017
Meeting 7	Define Design Idea - Peer Other Team Interaction and Prepare of Presentation	Peer Critique Team Meeting	Design	10/18/2017
Meeting 8	Interim Review Design Idea	Interim Review	Development	10/19/2017
Meeting 9	Development - Team Checkpoint Meeting	Team Meeting	Development	10/23/2017
Meeting 10	Development - Team Checkpoint Meeting	Team Meeting	Development	10/23/2017
Meeting 11	Development - Team Meeting	Team Meeting	Development	10/25/2017
Meeting 12	Development - Instructor Interaction - Prototype Test	Group Crit	Development	10/31/2017
Meeting 13	Interim Review Design Crit - Prototype Critique	Interim Review	Development	11/1/2017
Meeting 14	Development - Third-party Meeting - Soundtrack interface	Third-party Meeting	Test	11/23/2017
Meeting 15	Development - Test Feedbacks	Team Meeting	Test	12/4/2017
Meeting 16	Development - Third-party Meeting - Soundtrack interface	Third-party Meeting	Test	12/7/2017

On the first day of the project, students watched videos of mobile games, observed them and sometimes played to find an idea to their project. Students used a removable set of smart TV with interactive screen board, then they took notes of the relevant points to consider in their project. By the end, they had decided to develop a game for iPad that combined sounds and movements, where sounds changes according to the movements. At this point, the instructor appeared to check the progress of the team, then students explained the main idea and the instructor concluded that they were focusing on games that create soundtrack dynamically.

On the second day, each student brought a list of games that were similar to the one they wanted to develop and analyzed them together. The team used the Post-it® notes to organize the main ideas on the whiteboard. Then at the first Group Crit, the instructor shared with them the references of two games that focuses on sound experience, one of them developed for blind people. In addition, he suggested that they first define the musical style, in order to facilitate next decisions and directions.

After that, students reflected on instructor interaction and whereas the sound was the main part of the game, they decided that it was crucial to talk with the third-

party developer to understand what was possible to develop and how the interface would work before defining the project.

Students that were responsible by programming had to research and learn how to manage high and low sound filter. At the beginning, the team had two different game´s idea, a musical city and the adventure of a lost spaceship in the outer space. In an attempt to choose between these ideas, they analyzed the possibilities of sound´s interface for each one, however they did not come to any agreement. So, as a homework, each one should think of a narrative related to these two ideas, and select some references to discuss again, in the next day.

On the third day of the project, the team watched videos of some games to observe the synchronicity between soundtrack and game stages in order to define the narrative of the game. The game initial idea was that there would be a planet where all kind of communication would be based on musical notes. The inhabitants of this planet would like to explore other planets. When they set off on a trip, the spaceship exploded, and they got lost in the space trying to communicate with their planet to come back. Therefore, they planned to jump from planet to planet producing musical sequence with a rescue message to send to their original planet.

Team B made a reflective interaction with a student from another team to get his opinion on their game idea. This student arose questions on every aspect of the game idea helping them to reflect on some aspects that were not defined yet, mainly related to game´s mechanics.

Finally, to organize the project they defined that it would use software Trello for documents and Slack for tasks, and discussed the items of the presentation, defining the Big Idea, Challenge, and Essential Questions.

At Interim Review Design Idea, the team presented their project design idea to peers of other teams and the instructor. They contributed with questions and critical view, giving some references of games as suggestions to evaluate for their design idea. This session marked the end of the project design phase and the beginning of the project development phase.

During the development phase, at the team checkpoint meetings, students responsible for programming, reported some difficulty to create sound combinations and to work with sound effects linked with the speed of the game design´s assets, and so forth. The greatest challenge was the integration with the sound´s interface. On the other hand, the designers had no difficulty to find images related to planets, outer

space, cosmic dust or Milky Way to use as basis for drawing the game images. However, to produce the image of the game´s character, which was an astronaut, as the designer did not find an image of the same angle required for the game, the designer took some pictures of herself with a helmet at the appropriate angle. This attitude demonstrated commitment and engagement to the project and the pursuit of accuracy in design and results.

At the Group Crit for game´s prototype evaluation, the instructor commented that the game scenario was good, but there was no sound synchronicity. He asked the students to reflect on the consistence between the game and its storytelling. Students justified that they were investigating some sound effects to improve it. They explained that it was more complex that it seemed, because there were three different sounds to compose the soundtrack, plus the effects that resulted from the interaction with the assets of game.

For Interim Review Design Crit, students presented the project idea and its prototype to the instructor and students of other teams. The instructor, who was the most experienced game design instructor of the staff, pointed out the positive aspects of the game´s project and gave some guidance instructions related to designing. In addition, students from other teams asked how the game would handle rewards and gave references of games with similar rewards principle.

After that, the team continued focusing on the project developing and developers performed some interactions with the third-party developer to deal with the sound´s interface.

During the test phase, students asked for peers of other teams and friends to informally test the game and give their feedback. Then the team considered these feedbacks and took appropriate demanded action.

Next sections show the analyses performed upon the data collected from the observation of Team B.

5.4.2 Reflection-in-action

In Design phase, besides team meetings and group crit meetings, this team did a third-party developer meeting and a meeting with a student of another team too.

In these team meetings, students listed some games related to the design idea, watched some of them, commented on their features and discussed their ideas for the

design. In the next example (Table 5.22) of one of these team meetings, they interactively discussed the different possibilities of how the sound´s interface could be built. Thus, many different ideas came up and, reflecting on them, the design was being defined. It highlights the situations of the reflection-in-action and creation at same time and, also includes a learning experience from an earlier challenge that will be discussed later.

For example, they commented that it is possible: to work with multi gravity, to distort the sound and to add speed; if using the headset, he could tinker with the stereo of the sound; as the character progresses in the game, he could gain speed.

Regarding this last idea, one student commented that it depends on how objects and obstacles will be placed on the game map. However, the goal was to know if this will generate some soundtrack or just a terrible noise.

Table 5.22 - Team B Dialogue 1 – Reflection-in-action – Team Meeting

Code	Quotation
[Reflection-in-action] [Creation]	"[...] (StudentB4): Type, in a runner like this is 'too mass' to work with 'multi gravity'. (StudentB1): Oh man, I think 'too ace' of working with sound is that it also has a lot of possibilities. Like you can add distortion, you can add speed, peach, and a lot of stuff. (StudentB4): It's running like (mimics sound): Vrum! Ohhh! Gee! (StudentB1): And I think there are a lot of inputs that might give us a link to the sound modification itself, including gyro and air, whatever. I think you can include like, each input can move some detail of the sound to leave a deal (interruption). (StudentB4): With headset you can make sound, play with sound. (StudentB1): Yes, you can! Wow! (StudentB4): As the guy runs on one side, is only in the right channel, then jump up is only in the left channel, when in the middle, makes a distortion of the two channels. [...] (Student B4): Is it? And if the character is accelerating, as it progresses in the game like that. He goes, as if he gains speed. [...] (StudentB4): But then, it depends on how we place the objects or obstacles, like, he deviate on the map. Oh, like, I said about the city, but he may be running on the street, suddenly he comes a car and he has to jump, then comes a truck and he has to jump higher. When he goes to the sky, he is running on the ceiling, he running on the ceiling, suddenly a plane comes, he goes down. Ah! A bird comes next and he has to make a leap, so yes, we try to create the object for him, we take the user to move to create these sound variations. The point is just to know if this is going to make some really crazy sound or if it's going to be a meaningless wretched noise, do you understand? [...]" (Team B Meeting 3)

At this point (Table 5.23) one student said that they should be careful because, in a previous challenge, he built many soundtracks that, in theory, should match, but when put together, became horrible.

Table 5.23 - Team B Dialogue 2 – Reflection-in-action – Team Meeting

Code	Quotation
[Learning Experience]	"[...] (StudentB1): Ah! This has to be careful because in the other challenge I had such a problem. I had generated several mini tracks so that theoretically they matched, like it looked pretty similar. I did in the same program, so when I went to get it together at Discoveryship it was horrible, horrible, because, but one of them was because the sound didn't match. [...] (Team B Meeting 3)

Next (Table 5.24) they discussed how the gravity of the planets could influence in the game soundtrack. The students had opposing opinions about when the sound would turn low or high but agreed that gravity influences the soundtrack.

Table 5.24 - Team B Dialogue 3 – Reflection-in-action – Team Meeting

Code	Quotation
[Learning Experience]	"[…] (StudentB1): Ah! This has to be careful because in the other challenge I had such a problem. I had generated several mini tracks so that theoretically they matched, like it looked pretty similar. I did in the same program, so when I went to get it together at Discoveryship it was horrible, horrible, because, but one of them was because the sound didn't match. [...]
[Reflection-in-action] [Creation]	(StudentB4): Another thing of gravity you may think I can imagine. Imagine a kind of 'deconstructed' runner that you actually have to jump from planet to planet, and every planet and has a gravity. And the gravity of this planet influences the sound to be really low or high. Planets that have a heavier gravity may fall a little more. (StudentB3): The sound is more brass. (StudentB4): Is that I imagine a pressure like (mimics sound): hummmmm! (StudentB3): It's just like I imagine, like, playing on a planet where the atmosphere is really heavier, like dimming, like (mimics sound): tumm. More bass. (StudentB4): I already imagined the opposite, because lighter gravity makes you tread slower, and slower sound in general is more (mimics sound): vohmm. Whatever, but the situation is, gravity influences loudness, and bass and treble, so you can just jump from planet to planet and try to make some kind of sound. Then you might have a micro universe there that you are traveling and trying to do something. (StudentB1): I don't know why, it reminded me a bit of a black hole. For example that sucks everything. I imagined something like sucking the sound and distorting it. (StudentB4): Then we can even play with physics, the question like ah, the planet he jumps in makes a sound, and when he jumps near a black hole, he just goes on tangent like that (emits sound): vuumm and accelerated. Play like, a lot of stuff. (StudentB3): And the little planets could have like, I don't know, vegetation or some rocks like that. When you kind of touch, and you get in touch, like it makes a different sound too, right? Or like, you jumped over, he goes over there and like (mimics sound): vum, go over and kind of pulls over (mimics sound): come on, some stops like that, right? (StudentB1): We have to check it. It would be nice if someone share an idea with ThirdParty1 to see just what a person can do with a sound, right. [...]" (Team B Meeting 3)

During the team-meetings, one realized that each member individually acted as an agent that stimulates reflection-in-action in others. In this process each one had a new idea or even something to complement in another´s idea, resulting in creation in this design phase.

As soon as students decided to develop a game sound based, they decided to meet third-party responsible by developing the sound´s interface. This third-party interaction (Table 5.25) was reflective and instructive also, like instructors interactions, due the outsourced developer expertise in the development of the sound features. They discussed issues as how the game could manage the sound effects; how the evolution of the soundtrack can affect the other features of the game, for instance, the speed that the objects appear on the screen or the background colours of the screen or even, size, colour or format of the objects could change depending on the rhythm of the music, and so forth. These discussions resulted in continuous process of reflection-in-action.

The third-party developer acted instructively when he said the generative soundtrack with gradual sound transition in his opinion is better than the predefined soundtrack.

Table 5.25 - Team B Dialogue 4 – Reflection-in-action – Third-party Meeting

Code	Quotation
[Third-party Instructive Interaction]	"[…] (ThirdParty1): Man, you can have the full experience of sound, both of this layer business, that I always hit this key, which I think makes dynamic music much cooler than closed music, you know. […] So make a sound-based game, man freaks out and try fucking that kind of thing. I think the idea is very cool. I think with an art, some really cool assets and a sound well finished, if it is possible to make gradual transitions, with programming would be very cool, man. (StudentB3): Yeah, in the arts you can roll, like you can freak out a lot, you can be very kind of different and [...]" (Team B Meeting 4)

Moreover, reflectively the third-party (Table 5.26) said they could manage visual effects such as light and contrast, suggested to create the soundtrack in layers and control them depending on the characters movements, then the students did reflection-in-action on this idea.

Table 5.26 - Team B Dialogue 5 – Reflection-in-action – Third-party Meeting

Code	Quotation
[Third-party Reflective Interaction]	"[…] (ThirdParty1): Ah! I don't know how you imagine it, but a very minimalist thing, playing well, quite with light, contrast. [...] (ThirdParty1): Another thing that I thought about, I talked about making music in layers and you control one of them with the character, right. But you can have other points, you can control the whole song, you know. I would have to test to see visually and in terms of sound what a sensation it causes. Because sometimes it gets so subtle if you move it in one place, sometimes it takes the whole song and moves it all. If you are going to heaven, you are in the highest part, it will become more acute. Not that it gets more acute. Not that it gets lighter, not that you improve the tone of the song, you'll cut the bass. You would understand better.

[Reflection-in-action] [Third-party Instructive Interaction]	(StudentB4): This is possible, it's like stirring peach, right? Of the audio. Hence, like. (ThirdParty1): No, no it is not *peach*! [....] (StudentB4): Is it really low? How to make a bass filter. (ThirdParty1): This, you can make a filter. This you can make a filter. Oh! In this format like this, as it is flat here, I will do so, as here, I am cutting bass and treble. This filter depends on where the guy had, for example, he starts here, on the floor, then he goes up, the filter goes all up together. As you cut the bass, you increase the treble. (StudentB4): I understood. (ThirdParty1): And come and go, you can do that, you know? Stir in both at the same time. (StudentB1l): Blimey! (ThirdParty1): I can show anything an application of it there in the program. I pick it up, compose a song or pick up a song that's done, and just apply it, and show you how I would do it, to see if it's going to be cool. [...]" (Team B Meeting 4)

Students shared their concerns to work with sound in real time (Table 5.27), dynamically, because had no previous experience, so they discussed how to do it with sound third-party developer, who remembered that they should also think about performance developments and finally he suggested to export three different soundtracks. The students agreed with this suggestion and asked him to send them these soundtracks examples to test this idea.

Table 5.27 - Team B Dialogue 5 – Reflection-in-action – Third-party Meeting

Code	Quotation
[Third-party Reflective Interaction]	"[…] (StudentB1): But my question is this, if it is possible to implement this. [...] (StudentB4): Anything is possible. (Laughter) You have to see the difficulty of doing this alone. (StudentB1): The biggest problem is the real time, so I doubt it, because some, at least when I used the sound program, you know. It usually took a while to process alone, but I don't know if it was because of the quality or because the process required it. I'm taking a look here, but. (ThirdParty1): Yeah, I would have to see this performance issue too. Because sometimes it was, hum! Let's suppose that it requires a lot of performance, then is going to get ugly, right! So there's only one layer, so it doesn't have to be. (StudentB4): That's what I imagine maybe, maybe have to do then. It's like having the assets exported in several. The treble and bass assets exported and make a kind a grey transition during the game you know. Load up all the sounds first, and as the guy runs, he transits. (ThirdParty1): If it's just this filter, the business of changing the volume, I don't think it requires much. I export three music tracks: one like this, one like this and one like this. Like one with medium, one with bass and one with treble. Then you put all of them to play at the same time and start with the volume 100% low, 50% mid and 100% high. And then, when the character is going up the middle, increase the medium, the bass decreases and the treble decreases. Go here to get the full bass, just medium and high. Because then, you wouldn't have to do this one, just play with the volume. I think I might be easier.

	(StudentB3): It's a, it's a stop that kind is a lot easier to do. (ThirdParty1): I imagine it is easier. (StudentB1): Can you export these three tracks? Because then we can test today. I think it's good for us to know if this works or not, before we start (interruption) You can export it later, or any time. [...]" (Team B Meeting 4)

At the Group Crit sessions of design phase (Table 5.28), the instructor´s interactions were most of them reflective and some instructive, almost no guidance. Also, the instructor gave some game references related to their idea for reflecting on them, as follows the dialogue of first Group Crit session:

Table 5.28 - Team B Dialogue 6 – Reflection-in-action – Group Crit session

Code	Quotation
[Instructor Reflective Interaction] [Game Reference]	"[…] (Instructor1): The other game I also like in this line is *Luxuria Superbia*. That is also the same thing, you must use earphones too, because it generates a soundtrack. Basically the game is a metaphor for a trip […].Each stage is a flower and you will penetrate this flower, only you have to go caressing the flower in different movements and seeing the reaction of the flower. And then the flower will bloom if you keep going say the intentions and the desires or it will close and lock and let you out if you do not tune in with it. (StudentB4): What platform is it? (Instructor1): It is IOS. It is for IPad. […] (Instructor 1): *Luxuria Superbia*. And, the cool thing is that the gameplay is multi-touch. There are various objects on the screen, you can hit the right objects and avoid objects that take off. [...]
[Instructor Instructive Interaction]	(Instructor1): I think there are few IOS games that exploit sound. They almost does not explore the sound.[...]
[Instructor Reflective Interaction]	(Instructor1): Is it something you want to do? (Student B4): We are in the beginning, trying to refer the CBL, but the idea is decide by vote, and develop the most voted of them
[Instructor Guidance Interaction]	(Instructor1): So my suggestion is that you set the musical style to begin with. If you are going to make a music style track, what would it be? Define it first that will help. [...]" (Team B Meeting 2)

When students had delimited and defined their idea for game development they made another Group Crit session (Table 5.29) to expose it to the instructor. After carefully listening to the idea, the instructor instructively suggested that they should choose between generating rhythm or harmony, explaining that they were different situations and that it would be almost impossible to randomly generate both at the same time.

Table 5.29 - Team B Dialogue 7 – Reflection-in-action – Group Crit session

Code	Quotation
[Instructor Instructive Interaction]	"[...] (Instructor1): Look you have two ways here, rhythm generation and harmony generation, and melody generation better saying, are two different things. You will have to decide which of the two ways you will go, because

	both at the same time, I think it's kind of impossible to generates rhythm and melody randomly, dynamically. It will break down. [...]
[Reflection-in-action]	(StudentB4): What I imagine is the principle type, would be the base of a corridor, right. But each type, you can set up this scenario somehow before, and each type of property would be a type of sound, an instrument. And as you walk and jump to various static objects, you create sound intensities, and the type as you move too, it goes to the beat of the music.
[Instructor Reflective Interaction]	(Instructor 1): Ah, this is already gives you an interesting idea that the city is a musical instrument. (Student B4): Yes. (Instructor 1): The city of music [...]. So it's very interesting to bring, to make a relationship between city and music.
[Reflection-in-action]	(StudentB4): It's kind of like he walks, and the windows can light up like a sample scheme, as you add the buttons. It is jumping, it will light up. It is that at the end, end, say, at the end of the phase, you can exclude real estate.
[Instructor Reflective Interaction]	(Instructor1): Don't take away the characters, make a generative deal. You make a map of a city that represents a song. So you can customize this city like a center control or maybe a three-dimensional structure.
[Reflection-in-action]	(StudentB3): Like a dimensional *pad*.
[Instructor Reflective Interaction]	(Instructor1): I imagine a kind of a sun city musical. I keep imagining this and as you build your city shapes, it would generate a kind of music, and then you would share on social networks: Look at the city I made, the city of samba. Man, that has a lot of potential.
[Reflection-in-action]	(StudentB4): You can kind of go around the city, and at a certain speed develop their rhythm. [...]" (Team B Meeting 5)

After the Group Crit session ended, students continued doing reflection-in-action on their ideas (Table 5.30), taking into account instructor comments and some game references and finally defined a new feature for the game, the game should be frantic.

Table 5.30 - Team B Dialogue 8 – Reflection-in-action – Team Meeting

Code	**Quotation'**
[Reflection-in-action]	"[…] (StudentB1): This one I was also wondering, that the guy could orient himself through, or form a sequence of actions, just like you put there, the buildings, or anything else right, and assemble something in the sequence. And one could guide on the building of the music to know if they are able to build the right level [...]. (StudentB3): So it´s like, as if you were adjusting the harmony according to the characters and the buildings, is that it? (StudentB1): Yes. (StudentB3): Wow, It´s interesting. [...]
[Game Reference]	(StudentB1): Ah! I was going to comment on that time Instructor1 talked about the song of *Hotline Miami*. (StudentB3): Ah! *Hotline Miami* is very good! (StudentB1): It's like when you start a level that makes you very tense, right. I don't know if you're tense, right? But as the song begins (mimics the sound): Tum tu ru rum tum tu rum rum, as if it were quite intense and the time that ends when you just kill the last guy is almost like absolute silence anyway. It's a very soft song like that, out of nowhere. I do not know, it is an interesting feeling of pyre is difficult to explain.

	StudentB3): Yeah, it makes you frantic even at the beginning of *Hotline Miami*, that he goes there, like you have to go crazy killing the guys. (StudentB1): Hmm! Hmm! I find it funny that when you kill the last guy (mimics sound): Pa, pa. That's all gone. It seems that the world has stopped like this. Give a relaxed.
[Game Reference]	(StudentB3): You saw *Undertale's* soundtrack that it works kind of the opposite. you start with a very calm trail and everything. And then according to the difficulty of the villain you will be getting. And also kind of according to the trail you're following, just like the *Truego Trego* soundtrack, you're in the villain like a kind of spider like that. Yeah, the little song is kind of fast-paced and then it's changing like the things you have to do for you to get away from it. There is a time that seems like you're kind of climbing a ladder and you have to go dodging the spiders, what kind of dodging their attacks and speeding up, speeding up. And when you get to the last boss, dude the song is like "full frenzy", that you kind of beat him and here comes the flow, and he says: Do you think you can do it? So, it's like that, you play all the elements like that in your face, and it's like playing music. And after you end up with everything just calm music and then you see the characters coming out of the world. Dude, it's very crazy and I think very, very well done *Undertale* soundtrack. [...]" (Team B Meeting 3)

In addition (Table 5.31), they commented that the instructor's interaction brought to them some relevant information to consider in the design project, as well as reflecting on the idea of sharing a video with player´s performance on YouTube.

Table 5.31 - Team B Dialogue 9 – Reflection-in-action – Team Meeting

Code	**Quotation**
[Reflection-in-action]	"[…] (StudentB4): It's just to inform, the thing I thought is a lot simpler, actually. Make something like he jumps out and when he finishes, the speed is just like sound waves like that, showing the construction of the sound waves he made. Like one of the things I think cool to be here, like reiterating that stop, that social freak. I don't know what, like we can make a video of the guy playing and he can share the video of his gambling routine later on youtube and such. And we can even create a channel of ours, the guy brand there and we can even monetize the channel later. Thinking about later monetization, this is just an idea. These things that Instructor1 was also talking about are very valid information.
[Game Reference]	(StudentB1): Humming! Humming! I think everything. There was a game that was really cool to play with people too. It was a bike, which was orange. [...]" (Team B Meeting 3)

When the team finished to define the game design, one day before to present it at the Design Idea Interim Review, the team did a reflective interaction with one student of another team to get his opinion on their game idea. This meeting is called a Peer Critique, as illustrated in next the dialogue (Table 5.32). This student made a checkpoint of every aspect of the game design. He asked about the overall back story, the theme of the game, the levels, rules and patterns in the game. Also, about the content design, the characters, items, puzzles, and quests. Regarding the user

interface, how the player would receive information and feedback, and how the player would interact with the game.

Table 5.32 - Team B Dialogue 11 – Reflection-in-action – Peer Critique

Code	Quotation
[Student Other Team Interaction] [Reflective Interaction] [Game Mechanics] [Soundtrack]	"[…] (StudentO1): Then the goal is to make contact with home planet, right? And, the way you are going to do that is by gathering an x score? (StudentB4): That is not right yet, but the idea is a correct sound sequence, or a certain score. (StudentO1): And once I get this sound sequence, what happens? Do I step up or is the game over? (Student B4): Step up to the next phase of the game. (StudentO1): Phase shift. Then will you make different sequences and will the complexity of the sequence increases with each phase? (Student B4): Exactly! (StudentO1): Okay. And, how will these sequences be? Or, is not set yet? (StudentB4): Hum, we already talked to the third-party developer and he will help us put together the right sequences, to do kind of harmonic things and make some sound that makes perfect sense. (StudentO1): And what will the input look like? (StudentB4): The entrance? (StudentO1): Will I touch the screen? (StudentB4): You will touch the screen. You will touch the screen at the right time to jump from planet to planet. [...] (StudentO1): But honestly, I cannot imagine the game play. I am thinking how it will be. I will have to hit the right note button. That is the mechanics. Maybe in the right position. Right note in the right position, is it? (StudentB4): Hmm! Hmm! The idea is this is how you get sound feedback. The game is not really for you to squeeze tighten, but to keep the pace so that the music is harmonious to your ears. Because when music has this kind of failure, it is boring, right. [...]" (Team B Meeting 7)

Some of this student´s questions helped them to reflect on some aspects that were not defined yet, mainly related to game´s mechanics (Table 5.33).

Table 5.33 - Team B Dialogue 12 – Reflection-in-action- Peer Critique

Code	Quotation
[Student Other Team Interaction] [Reflective Interaction]	"[…] (StudentO1): And where am I going to press? On top of the planet when it is falling or anywhere?
[Reflection-in-action] [Unforeseen Situation]	(StudentB4): I had not thought about that yet. [...]" (Team B Meeting 7)

Furthermore, in this Peer Critique (Table 5.34) the teammate could share his learning experience from his first challenge, where his team was able to make the game, but was unable to develop the narrative. For this reason, he was afraid of games that depend on history. From this experience, he learned by practice that when

developing a game that depends on the story, the narrative must be well defined to achieve success in the game development project, an evidence of practical learning.

Table 5.34 - Team B Dialogue 13 – Reflection-in-action – Peer Critique

Code	Quotation
[Student Other Team Interaction] [Reflective Interaction]]	"[...] (Student B4): Yes. Can I say a fear I have about games that depend on story? A situation happened to Him (refers to StudentO1). (StudentO1): Ha! There is! There is! (Laughs)
[Reflection-in-action] [Game Mechanics] [Game Narrative] [Learning Experience]	(Student B4): Our first challenge, the game *Fiona in the Nebola* is to have a beautiful story, but we could not develop it. The game was limited because we could not develop the story. We wanted to do things and could not implement, because we had no story. No one could think of a decent story to make the game and it is over! (StudentB3): O *storytelling* is something like (hum) (StudentB4): Guy, storytelling is something that spend a time to become good. I do not know. I think that in a *challenge*.there is not enough time to develop. (StudentO1): It is difficult . you have to know who can do it too. […]" (Team B Meeting 7)

At Design Idea Interim Review session (Table 5.35) the instructor interaction was most reflective, while few instructive and few guidance, as follows. Students from other teams contributed reflective questions and provided some game references related to the project presented. In the following example, a colleague from another team commented on a game where music determines the scene that is totally generative, drawing attention that works with multiple tracks seems to be difficult.

The instructor took this example to give an instructive explanation, clarifying that what happens in this game is what Instructor4 taught as a flow state, as the player achieves the complexity of the task and the highest level of difficulty in the game at the same time.

Table 5.35 - Team B Dialogue 14 – Reflection-in-action – Design Idea Interim Review session

Code	Quotation
[Instructor Reflective Interaction]	"[…] (Instructor1): So will be a platform game, a runner, a platform? (Student B4): More or less. (...) We are thinking of a runner, but with a slightly different mechanics of being a horizontal sliding platform, there is still something to be developed, but the idea is to be a runner. […]
[Student Other Team Interaction] [Game Reference]	(StudentO2): *Line Rider* do you know? *Line rider* is a game that proposes to synchronize a visual thing with an existing song. You can write your map in Line Rider, it is a text, it is a text file, and of course it is extremely boring and laborious to do that. Then I will send you a video on how is the end result of playing with Line Rider. But basically it's a guy with a bike riding in a line, and as he bounces and hits the line, he is not building the song, it is the opposite, right. The music determines the scene where he's bouncing and then if you zoom out the scene, you can't read music there, because it has totally generative and apparently random format, right. However, it is because sometimes it hits and come back, then it goes down, left and right. […]

	(StudentO2): Doing this with a song with multiple tracks is quite difficult, as it alternates which track the line is representing. (StudentO3): All lines that exist there are used. (StudentO2): Yes, that is the coolest part. It is extremely good there, it is not breaking, but it breaks, moves away from the bike. (StudentO6): The guy is a good player. (StudentO2): Yes, he is not upside down once, so far. Hit and exfoliate your head. Almost right!
[Instructor Instructive Interaction]	(Instructor1): What happens in a game like that is that state of Flow, which Instructor4 told you reminds. People get there and reach the complexity of the task, exactly like the difficulty of the game, right. (StudentO2): Hence, there is a vertical limit. Look at the inertia that is this guy. There are the dishes only. This spin is essential, if it does not spin, will hit the wheels on the wall, and it breaks. [...] (StudentO2): And did you noticed that he and the bike, so in rhythm […] (StudentB3): And did you see that a score appears at the end, right. All right. (Instructor1): It is a performance.

In Development phase, at Group Crit sessions, the instructor interactions were few guidance and instructive, and the majority reflective, which resulted in team reflections-in-action.

Next dialogue (Table 5.36Table 5.36) happened in development phase when students were working on a game´s prototype. It shows an example of a Group Crit session in which the instructor tested the first game´s prototype developed by the students. After testing the prototype, the instructor questioned the mechanics of the game, which consists of a character jumping from planet to planet at the touch's command.

Table 5.36 - Team B Dialogue 15 – Reflection-in-action – Group Crit session

Code	**Quotation**
[Game Narrative initial idea]	"(Instructor1): What's the story of this character? Why is he jumping from planet to planet? (StudentB4): Their idea is that they lived in a quiet and kind place. [...] They lived in a city. For now, we call this the city of sound**.** All the people did there, as the only destiny of the people who lived there was to make sounds, and the idea was to make perfect harmony. Then a "disturbing" person was born and she wanted to look for other meanings for the life. [...] So she went after the universe, and the idea is that when everyone arrived at night, they would try to draw lines in the scores and turn on the stars to make perfect sounds, but no one ever hit. [...] This born person, this born entity, wanted to visit the stars. She wanted to take the stars by hand and reorder them as she pleases. She travels through the universe and is therefore lost and therefore to her loneliness. [...] (Instructor1): So this character, is he trying to find disharmony? (StudentB4): Exact. The harmony of disharmony. […]" (Team B Meeting 9)

The instructor strived to understand the relationship between the mechanics and narrative of the game, but perceived inconsistencies between them and tried to make the students to reflect upon it, as shows next dialogue (Table 5.37).

Table 5.37 - Team B Dialogue 16 – Reflection-in-action – Group Crit session

Code	Quotation
[Instructor Reflective Interaction]	"(Instructor1): But in the game, disharmony is a bad thing? […] (StudentB4): The idea is that he makes different sounds, not disharmony. He makes different sounds than he heard his whole life, and he learns that even when he is losing, he can reach and reach goals. The idea is that there are easter eggs. (Instructor1): But is he running after a planet or what is he doing? (StudentB4): He is wandering from planet to planet. He does not know if he can make it back to his home planet because his spaceship has been destroyed. He is wandering from planet to planet and making sounds. It may be one day he can get back to his planet, or he may find that disharmony is (pause) works for him. (Instructor1): And what is good for him? What is a good thing or result in the game? If he is jumping from the planet, what is a good result, to skip at the time of music? […]" (Team B Meeting 11)

As the students tried to answer the instructor's questions, "in promptu" students have come up with new ideas, which continued to present inconsistencies. Realizing this, the instructor asked new questions for them to reflect on this situation, since the instructor's goal was not to "destroy" their game proposal, but to help defining a proposal in which there was a coherence between the narrative and the game mechanics. In the next dialogue (Table 5.38), the instructor seeks to prevent students from solving the problem by adapting narrative to mechanics without sufficient reflection on mechanics.

Table 5.38 - Team B Dialogue 17 – Reflection-in-action – Group Crit session

Code	Quotation
[Reflection-in-action] [Creation] [New idea]	(StudentB3): Well, whereas the way he communicates on his planet is music, this may be from the story of the game that can cure the loneliness within him or even be sending a signal while he is packing his things. (Instructor1): No, you are not talking in terms of story. (StudentB3): I am talking about the story, but I am also talking about possible stages of the game. We can make a first phase where if he goes at the right times he can get a better tool to fix his spaceship.
[Instructor Reflective Interaction]	(Instructor1): But this is contradictory to the idea that one was going off his planet to look for different things. (StudentB3): He left his planet and the spaceship explode. Then he got arrested. He got lost.
[Instructor Reflective Interaction]	(Instructor1): [...] what I am trying to say is that you have a story where the guy is searching other rhythms and harmonies because he was bored with that classical way from his planet. Then he went out and suddenly, the game

	engine is based on trying to be as perfect as possible […]" (Team B Meeting 11)

For students better understand, as illustrated in next dialogue (Table 5.39), the instructor quoted as reference, a game where the rhythm of the music (pace) is integrated to the game mechanics (engine), and does not matter or has nothing to do with the game narrative, because the goal is to explore the scenarios of the game, then as the player moves through the objects, some visual effects (appear) are build and different sounds that creates a pleasant player experience.

Table 5.39 - Team B Dialogue 18 – Reflection-in-action – Group Crit session

Code	Quotation
[Instructor Reflective Interaction] [Game Reference]	"(Instructor1): Do you see the gane *Hohokun?* [...] has no goal, no right or wrong. The idea is to explore the game scenarios. As the player pass through the objects they make sounds. […] The rhythm (pace) is integrated with gameplay. Has non-explicit goals. If you move in a certain order to certain elements of the game and gather other elements like puzzles, you do not get game points, just can see the animation happen and be happy that way. […]" (Team B Meeting 11)

These discussions and reflective interaction resulted in students´ reflection-in-action and the adjustment of the game´s narrative to stay straightly connected to the game mechanics, as could be seen in the next day, when students described the game narrative at Design Crit Interim Review session, as shown in Table 5.40.

Table 5.40 - Team B Dialogue 19 – Reflection-in-action - Design Crit Interim Review session

Code	Quotation
[Game Final Idea]	"(StudentB4): [...] There is a character who lives in a world where all communication is based on sounds. The point is that the place where she lives is like a paradox, a parallel with the field like, who is born there only messes with sound and will only make sound for the rest of life, and everyone tries to find a perfect melody, a perfect harmony**.** This child, who was born there, she does not agree with this view and she wants to explore the universe, because the most she sees at night is her parents aiming, trying to create sheet music against the sky, against the stars and she wants to reach the stars. So she "moves worlds and backgrounds" there, studies hard and can build a spaceship, and goes into space trying to catch the stars. However, arriving in space, an accident happens, and her spaceship explodes, and she begins to wander through space. One fine day she wakes up lying on the surface of an unknown planet, and then she decides to jump on that planet, and sees that she can make a sound. As she is alone, going through a moment of loneliness, she ends up trying to make new compositions, to one day try to head home, who knows**.**" (Team B Meeting 13)

In this case, the narrative of the game was the character of the game is trying to build new compositions instead of sounds´ disharmony, as was in the first narrative

they had presented in Group Crit session. Moreover, the character lives in a world with sound-based communication instead of in the city of music, this way the narrative seems more connected with the game mechanics, since as the character jumps over the planet, each planet makes a different sound and, depending on the player´s interaction, the soundtrack is generated.

At Design Crit Interim Review session (Table 5.41), the majority of instructor interaction were reflective and guidance, no instructive. The instructor asked about the rules of the game, user interface design, the game mechanics, the placement of the objects in game´s layout, the influence of the object size on gameplay.

Table 5.41 - Team B Dialogue 20 – Reflection-in-action – Design Crit Interim Review session

Code	Quotation
[Instructor Interaction]	"[...] (Instructor4): So the questions: Does the time of clicking or the direction influence? For instance, is it the swipe or just the tap in the right timing that changes the relationship? (StudentB1): For now, it is the click on the planets, which one wants to choose. (Instructor4): Ah, on the planet itself. (StudentB1): It is you click on the planet and it jumps over there. (Instructor 4): Each planet, for example, a smaller planet he pulls to the note "E" and a large planet pulls to "C" or not. Does it not matter, or did you not think about it and why not? (StudentB1): We thought about creating these variations. For instance, the size is not related to the note. We are even doing many combinations to see, both the visual and the sound to see what fits best with each thing. Because when the player starts to make mistakes, one of our goals, for instance, is not that the game just ends, as the guy gives a game over. Is that it spoils the sound, spoiled in quotation marks, but distorts the sound. However, it can be even helpful if he wants to cause a little different harmony. So, we want to put these sound and visual parts together, however we have not defined exactly. We are doing some more matchings to see how it fits. […]"(Team B Meeting 13)

The instructor´s critique was if the game was interesting, cool, looks fun, different from other music games as *Patapon* and *Guitar Hero*. Finally, he recommended that the team should better define the feedback on elements such as colour, shapes and size, as presented in Table 5.42.

Table 5.42 - Team B Dialogue 21 – Reflection-in-action - Design Crit Interim Review session

Code	Quotation
[Instructor Guidance Interaction]	"[…] (Instructor4): Yeah, I put it there as a suggestion. Hmm, any questions? It is a music game! It is interesting! It is cool! So, the reading to me was very clear and it seems fun, it was going to be something I wanted to interact with, right. It is nothing like other music games I have seen, like *Patapon* right, which is very cool, but has a whole set of symbols and such, on top. Nor a linear game like *Guitar Hero* itself. I do not have the feeling of either of these, it is quite different, so it is hard for us to evaluate. Maybe that is why I put the question of feedback,

	but I just put there, have to think more about feedback, colours, shapes and size, so things are clearer. […]"(Team B Meeting 13)

At Design Crit Interim Review session (Table 5.43) students of other teams contributed with questions and suggestions, for example about game quests, scoring system, scoring sharing, when game is over does the player level up, as illustrated in next dialogue.

Table 5.43 - Team B Dialogue 22 – Reflection-in-action - Design Crit Interim Review session

Code	Quotation
[Student Other Team Interaction] [Reflective Interaction]	"[…] (StudentA1): Yeah, did you guys think of some kind of score and score sharing? (StudentB4): Yes, the score is like this. It has, as the game is rhythmic, there is an optimal time line for you to change planet. Hence, we are thinking to indicate this optimal time of exchange. [...] He only needs to make a maximum score to get it, if he stays within range, he step ahead. So he can make choices like, oh, I want to ruin my harmony right now. (StudentO5): It is like shifting gears in Need for Speed. [...] there is that optimal point that you change and the car goes away, like this. *Need for Speed* is cool! [...] (StudentO8): Does not *Guitar Hero* have a score scheme like that? (StudentB4): There is, too, except *Guitar Hero* is kind of a single point. If you hit the right point, you get point. If you hit the wrong spot, it spoils the part of the trail you are doing, so you get no point [...] (StudentO7): My question is more boring. Will the game over at the end or will you pass each specific level? Will be music blocks and (interruption) (StudentB4): It is a procedural feat, it is not going to be assembled itself, but let is just say how do you know those subway lines? You know the lines that are like grid lines, from a graphic grid like that. They leave the main point and a certain time they come back at some point. One will have these deviations to choose from, and then they can join paths and merge into certain paths. As she will have more than one way to complete the level. (StudentO7): So there will be begin, middle and end, for example. (StudentB4): That is it. (StudentO7): Okay. (StudentO5): I can suggest something else, related to that other suggestion I had given from *Need for Speed* and such. […]" (Team B Meeting 13)

While most of the team reflection-in-actions resulted from instructor's interactions, some resulted directly from team reflection, some happened as consequence of conversation with the material and reflection-on-action which are next detailed.

5.4.3 Reflection-on-action

As previously stated, reflection-on-action refers to the reflection made upon an action. During the test phase, students found an error of sound synchronicity then they

made a third-party developer interaction to show it, analyzing it together and providing the correction. Such, there was a conversation with the material in which an error was identified, next developers made a reflection-on-action, they identified the problem and did a problem solving, as illustrated in Table 5.44.

Table 5.44 - Team B Dialogue 23 – Reflection-on-action

Code	Quotation
[Third-party Interaction] [Sound Interface Error]	"[...] (ThirdParty1): It is down there. (StudentB1): It is that scheme, we had commented on that. Then the guy clicks the planet to start. (ThirdParty1): Oops! Here is a note missing
[Reflection-on-action]	(StudentB4): Yes, there is a hole in it. [...] (StudentB1): Okay. He has the comet on top. (ThirdParty1): I have an obligation to get everything right, do I not have? [...] (ThirdParty1): I do not know perhaps increasing the range a bit. [...]" (Team B Meeting 14)

5.4.4 Conversation with the Material

As previously stated, conversation with material in this context refers to the conversation with the developed software or related artefacts.

Students asked for peers of other group and friends to informally test the game and give their feedback. Then the team reviewed these feedbacks and took appropriate actions.

The first example (Table 5.45) referred to test´s feedback from friends who reported that sometimes the finger of the gamer was in front of the asset and the user lost the control, consequently the game´s character went out of the screen. In this situation, the user could not touch in the right position then they reflected on the possibility to reduce the asset speed to do not prejudice the player.

Table 5.45 - Team B Dialogue 24 – Conversation with the material

Code	Quotation
[Team Meeting] [Test Feedback] [Conversation with Material] [Object Size] [Reflection-in-action]	"(StudentB4): So [...] at weekend I asked some people to play at the beach, and even some feedbacks that comes against what I was seeing. And I could not shoot because I did not have another phone. The main point that positive feedback was for people who could not see the character, and such. And there were not many elements on the screen, that the elements were very large. [...] there is no problem that the planet keep coming smaller or the character is small. [...] but having more reaction space. The things are so big on the screen, so there is a lot of things and the players get lost. Like sometimes when there are two, three planets in a row like this, people can't play, but not because they don't have the speed and responsiveness, but because it's so close there that they actually twist their fingers on the screen.

	(StudentB1): Are the planets overlapping on the iphone or haven't you reached that point? (StudentB4): Overlap. Some cases, when it comes to three, four together like that, they take one edge over another. (StudentB1): Hum! Hum! (StudentB4): So, I think, since the screen is designed for the ipad, right, and you're testing on the ipad, on the iphone the ratio decreases a lot. (StudentB1): That is what people are going to solve, because in fact, I take like, I need to make the optimal point of the planets and not be able to define a calculable number. And then as I was testing mostly on the ipad, I set for the ipad and thought of putting a device prepared to present to the guys. But this is something that has to be solved. (StudentB4): The ideal line is at the right height of the screen, you know. Even on the iphone is not bad and I tested on the ipad too, and everything is fine. (StudentB1): But then I prepared like, the planets and the size everything relative to the ipad like that. (StudentB4): I got it. (StudentB1): So, I ended up not thinking about the iphone and now I have to adapt. (StudentB4): Okay, just think about it because it's so big. [...]" (Team Meeting 16)

Initially the application was developed for iPad, however, as friends who tested the game did it on the iPhone, the size of the objects did not fit the screen. The students reflected in the action and decided to change the scope of the project, created an extended version for the iPhone, making the necessary adaptations. An iPhone version required to resize some assets, so in terms of project, this represented first a decision-making; next, a scope management, planning, time management and project management, in order to manage the work with the third-party developer the sound interface necessary adjustments.

Next situation (Table 5.46), refers to the player´s misunderstanding of a game visual effect. They did not understand a blinking border, so they did a reflection-in-action on the original idea searching for a solution.

Table 5.46 - Team B Dialogue 25 – Conversation with the material.

Code	Quotation
[Team Meeting] [Test Feedback] [Conversation with Material] [Game Asset out of Synchrony] [Reflection-in-action]	"(StudentB4): And the people didn't understand exactly what that flashing pink border does. (StudentB1): Hmm! Hmm! Right. That's it was to be synchronized with the beat, which was for, like (mimics sound): pa, pa, pa. But I saw that at the beginning of the song, she's not synchronizing with the beat. From now on, she starts to sync, but by then the guy is like, already thought it's not that anymore. (StudentB3): It is that feeling. (StudentB4): And something like that, maybe if we could have a little less opacity on the screen so it really looked like a neon glow there, one thing would make more sense than it was hard. Because it's so hard, people don't understand what's going on and kind of ignore it. She sees nothing, it's like there, just exists. [...]" (Team B Meeting 16)

Thereby, throughout the development´s process new ideas have emerged as a result of reflection-in-action of the students. Such the researcher observed that, at design phase the reflective interactions led to reflection-in-action and the process of reflection-in-action contributed to stimulate the emergence of new ideas and consequently the process of creation. At development phase, the reflection-in-action contributed to problem-solving and/or to improve the solution or final product.

5.4.5 Practical Learning Contributions

The competences or skills required throughout this team mobile application development are listed below, being that, sometimes students acquired these competences or skills because they were unknown or had not previously developed and other times they practiced and developed them.

Technical research is a technical skill practiced in design and development phase, from distinct ways. At design phase, students practiced this skill when they were in searching of references of the games related to the game they will develop, as shown in Table 5.28, Table 5.30, Table 5.31, Table 5.34, Table 5.35, Table 5.39 and Table 5.47. On the other hand, at the development phase students practiced it, when they need to find a library of codes and the most adequate application to work with sound.

Table 5.47 - Team B – Technical Research – Team Meeting

Code	Quotation
[Game Reference]	"[...] (StudentB4): I don't know maybe it's Rapt Guy (...) and there is also one on Steam that I play that is Inner Movies. His scenery is like a cut paper, the edges are like caves. It is as if the cave had been cut out like this, you know [...] when you arrive at a dead end, you press some arrows on the keyboard and it rotates the scene and gravity changes to the orientation you rotate, it is crazy! And, it has a lot of possibilities, and the game art is super simple. (StudentB1): Gravity in general is something you can explore a lot. The Angry Birds itself, since I think that the crowd did not like it because it was saturated in Angry Birds idea. […] (StudentB4): There is one on Steam, it is also half 3D and half 8 bits, man. [..] It has a very good sound based on Steam. [...] Ah! Son of Life […] Sound of the track […] Bit Trip Runner […] (StudentB1): Ah! Do you remember Patapon [...] each key is a note [...] (StudentB3): There was a Japanese game in which he was very impulsive, but it was really cool! (...). There is Taipo who is very unhappy too, but there is this one, it was a name with S, I do not remember anymore (...) (StudentB3): And there is another sound game that I remembered now that is very cool, which is Project Diva (...) I will put it here for you [...]" (Team B Meeting 1)
[Game Reference]	"[…] (StudentB1): Ah! I was going to comment on that time Instructor1 talked about the song of *Hotline Miami*. (StudentB3): Ah! *Hotline Miami* is very good!

	(StudentB1): It's like when you start a level that makes you very tense, right. I don't know if you're tense, right? But as the song begins (mimics the sound): Tum tu ru rum tum tu rum rum, as if it were quite intense and the time that ends when you just kill the last guy is almost like absolute silence anyway. It's a very soft song like that, out of nowhere. I do not know, it is an interesting feeling of pyre is difficult to explain. […] (StudentB3): You saw *Undertale's* soundtrack that it works kind of the opposite. you start with a very calm trail and everything. And then according to the difficulty of the villain you will be getting. And also kind of according to the trail you're following, just like the *Truego Trego* soundtrack, you're in the villain like a kind of spider like that. Yeah, the little song is kind of fast-paced and then it's changing like the things you have to do for you to get away from it. There is a time that seems like you're kind of climbing a ladder and you have to go dodging the spiders, what kind of dodging their attacks and speeding up, speeding up. And when you get to the last boss, dude the song is like "full frenzy", that you kind of beat him and here comes the flow, and he says: Do you think you can do it? So, it's like that, you play all the elements like that in your face, and it's like playing music. And after you end up with everything just calm music and then you see the characters coming out of the world. Dude, it's very crazy and I think very, very well done *Undertale* soundtrack. [...]" (Team B Meeting 3)

Pair programming is another technical skill that was practiced by students of this team during project development.

Learning Experience is a skill that also appeared, as reported by a student from this team and for a student of another team, both from earlier challenges. These experiences helped the students of this team during the project´s development to take project decisions.

The first situation reported by the student (StudentB1) refers to an experience of a previous challenge related to matching soundtracks and guided them to not repeat the same mistake, as illustrated in Table 5.48.

Table 5.48 - Team B Dialogue 26 – Team Meeting – Learning Experience

Code	Quotation
[Team Meeting] [Learning Experience] [Reflection-in-action]	"[...] (Student B1): Ah! You need to be careful about this because in the other challenge I had this problem. I had generated several mini tracks that theoretically matched each other, because they were very similar, so I put in the same program. However, when I joined at Discoveryship, it was horrible, horrible. The sound did not match, and matching the sounds was a challenge. [...]" (Meeting 2)

The second situation reported by another student (StudentB4) of this team refers to a previous experience of him and a colleague from another team (StudentO1) that occurred in the first studio challenge in which they worked together. The experience concerned the close connection between narrative and game mechanics and underlined the importance of defining them well, as follows in (Table 5.49).

Table 5.49 - Team B Dialogue 27 – Peer Critique – Learning Experience

Code	Quotation
[Student Other Team Reflective Interaction] [Learning Experience]	"[...] (Student B4): Yes. Can I say a fear I have about games that depend on story? A situation happened to Him (refers to StudentO1).
[Reflection-in-action]	(StudentO1): Ha! There is! There is! (Laughs) (Student B4): Our first challenge, the game *Fiona in the Nebola* is to have a beautiful story, but we could not develop it. The game was limited because we could not develop the story. We wanted to do things and could not implement, because we had no story. No one could think of a decent story to make the game and it is over! […]" (Team B Meeting 7)

Teamwork, collaboration, communication, interpersonal, time management, planning, problem solving, decision-making, scope management and project management are skills required for practicing of software engineering that the students practiced during this project development.

Throughout project development the researcher noticed that students worked as a team and collaborated with each other. In the design phase, students worked in group in an immersive manner to define the design. They researched and observed some game references related to the types of games they would like to develop the project. Next they discussed the ideas collaboratively, and wrote them in Post-it® to put the project memory on the whiteboard. In the development phase, they divided the tasks, being that each developer prepared a game feature separately and joined in the end to work together on the code. The designer built the design assets and the devigner documented, organized and shared all the documents with project definitions and also presentation to the team.

They developed oral and written communication, when preparing and presenting their project in the studio sessions, or when making verbal or digital communication within the team. They already had to integrate with each other because they had not worked together on previous challenges.

For instance, when the students found a product error, they had to solve the problem, searching for the solution, as illustrated in Table 5.44. In this case, the problem was in sound interface, then they communicated to the third-party developer.

When analyzing the results of the game tests, students had to take action for each reported case. Table 5.45 showed the situation where the resolution required students to change the initial scope of the project to create an extended version for the iPhone. This required, first, the decision to change the scope and manage all the

necessary activities related to it, such as planning the necessary adaptations, resizing some game assets, managing changes in the sound interface with the third-party developer and the impacts on the project's schedule project.

Table 5.46 illustrated a situation where the player did not understand a game visual effect, then students reflected on this and found a solution.

Same as, the researcher observed the student's flexibility and adaptability, from team integration to the pursuit of new techniques and software applications for project developing. In several situations, students demonstrated commitment and engagement to the project development, as taking pictures of himself with a helmet to get the best angle to create the asset of game´s character or asking for friends to test their game to collect their feedback.

5.4.6 Team B Summary

The summary of the Team B Mini Challenge results from coding cycles and ethnographic observation of studio experience project developing for further cross analysis with Team B Self-reflection results, is shown in Table 5.50.

Table 5.50 - Team B - Summary of Results

Team B - Coding cycles & Participant Observation Results	
Reflection-in-action	Creation
Reflection-on-action	Problem Solving
Conversation with Material	Decision-Making
	Planning
	Problem Solving
	Project Management
	Scope Management
	Time Management
Studio Project Experience	Collaboration
	Commitment
	Communication
	Interpersonal
	Learning Experience
	Pair Programming
	Teamwork
	Technical Research
	Technical Learning

5.5 Students Self-Reflections Analysis

The students made their individual self-reflections at the end of the Mini Challenge which coincided with the end of the first year of the studio course. The studio staff gave no rules nor guidelines to write this document, so the format was free, and the goal was to collect the perceptions of the students so far.

On the particular analysis of the students' self-reflections of the observed teams, it was noticed that students analyzed their own experience in the studio as an education place and on the point of view of the proposed challenge.

I, as a researcher, observed a controversial point about the studio's written self-reflections regarding to the way the collection process was conducted. The process was carried out in free format, with the intention of capturing student feedback without 'inducing', so that were no guidelines nor addressed problems in relation to the studio or development process. As a result, self-reflections had different formats, making it difficult to analyze common points and some students even ended up talking about aspects of a personal nature instead of addressing the methodology of education, and the learning process in the studio.

5.5.1 Self-Reflections of Team A

Students of Team A reported a particular situation occurred during the development that required skills of communication (Table 5.51), interpersonal capabilities and conflict management, consequently brought them personal learning. A student reported team´s commitment to the development of the project, and another student reported collaboration and teamwork.

Table 5.51 - Team A - Student´s Self-Reflections.

Code	Quotation
[Communication Problem] [Interpersonal]	(StudentA2): Throughout the execution process, we had some difficulties, as well as any team, mainly related to communication.
[Communication Problem] [Interpersonal]	(StudentA3): I know I have a lot of trouble making friends and being sociable, and I've been trying to change that every day. It hurt me so much to know that I hurt a lot of colleagues and that I came up with the idea of being sloppy, lazy and doing nothing. I honestly thought I was doing a good job and found that not too late. It is with great sadness that I leave the Academy but with many great memories, besides the numerous contacts I made here. I can only leave my thanks to you all and apologize again for not being the best member of your team and maybe hindered your work

[Communication Problem] [Interpersonal] [Commitment]	(StudentA1): However, despite the idea that the game is good and the team's commitment, not everything was like an arrangement of roses in this challenge. We had some internal relationship problems, which caused a certain amount of nerve damage. At this point, I am sorry that I am no longer involved to refresh my spirit and avoid some unnecessary arguments. I could also have been a little tougher on the team's accusations.
[Collaboration] [Teamwork]	"(StudentB2): Our proposal started in the discussion about games and horror movies, and “memes”, mainly Brazilians. We considered the possibility of match the terror theme to the “memes” theme, but considering that the idea would not stop being comical, we preferred to follow a line where the suspense would be a guide [...] Throughout this process we realized that the majority of the team felt comfortable and liked the theme of horror games, so we decided to continue on this side. On our first day of research, we essentially looked at other games and horror stories and found a curiosity that ended up leading us to our Big Idea. [...] After some discussions, we realized that it would be better, mainly for religious reasons, to change the term “Sins” to “Vices”, thus having a name for what we would work for. These discussions at the beginning of the project were essential for what would be defined in a group, regarding to the game[...]"

Moreover, one student reported technical learning, two game development first experience; one of them reported design learning, one programming learning, two reported learning by practice, learning experience and personal learning (Table 5.52).

Table 5.52 - Team A Students´Self-Reflections

Code	**Quotation**
[Learning Experience] [Design Learning]	(StudentA1): In first challenge [...] we didn’t have much time for developing, however it was extremely important; the learning was huge, and because it was an individual challenge, I had to work on the design, which made me realize that I'm not very good with it, but I was proud of the final result
[Game Development (1st Experience)] [Learning by Practice] [Learning Experience] [Technical Learning]	(StudentA1): Last challenge of the year, we had to create a game from scratch, which for a video game lover was an amazing experience. It's extremely rewarding to create a character on paper, to imagine a game, and then to slowly see your ideas come to life. My team fell in love with the proposed theme for the project, which was a darker game, and it engaged us a lot. Unfortunately, for lack of time, we will only finish the first phase of the three planned, however, this is a project that I have been very attached to, and if my team is willing to finish, I would be extremely happy and willing to do so.
[Business Learning] [Learning By Practice] [Learning Experience] [Programming] [Design Learning] [Personal Learning]	(StudentA3): I was able to learn a lot from all the teachers, colleagues and obviously from all the professionals who taught courses that were exclusive to us. My growth was very big, I could learn a lot about technology, games, programming, design and business. I grew a lot and matured a lot too.
[Game Development (1st Experience] [Programming]	(StudentA4): [...] this course presented what I can do with programming to change the world.

From an educational environment perspective, students of Team A said it was challenging, required dedication and availability, moreover they felt motivated to work on it. Table 5.53 presents the results summary of such self-reflections for Team A.

Table 5.53 - Team A Results of Self-Reflections

Team A Results of Self-Reflections
Analytical
Availability
Collaboration
Commitment
Communication
Communication Problem
Conflict Management
Design Learning
Game Development
Interpersonal
Interpersonal Problem
Learning by Practice
Learning Experience
Personal Learning
Programming
Self-Confidence
Teamwork

5.5.2 Self-Reflections of Team B

Regarding to Team B, the self-reflections mentioned only the studio, as illustrated in Table 5.54, they reported they were commitment, had to work with team collaboration, to improve communication, interpersonal abilities and teamwork. On technical perspective, they reported technical learning, design and programming learning. As well as planning, project management, time management, learning by practice and personal learning, as follows:

Table 5.54 - Team B Student´s Self-Reflections

Code	Finding
[Design Learning] [Planning] [Programming] [Communication] [Project Management] [Time Management]	(Student B1): At ADA, I was able to develop many things that basically fit into design, programming, communication, management, planning, and a hint of business. […] However, in here we don't have all the time in the world. The deadline is short, and it is necessary to automate some tasks. That's how I learned.
[Communication] [Planning] [Teamwork]	(StudentB1): Since I had never done group work that required organization at a crucial level. I was a little lost in that regard. I did not know what tools to use, how to plan, and not even how to communicate with the group about what we should or should not do.

[Learning by Practice] [Learning Experience]	(StudentB1): So much so that in the first group challenge we suffered a lot from it, but we learned. We tried to use tools to organize tasks but did not prioritize what was important or not. We developed features with which we had affinity and the important thing was falling behind. In fact, these learnings would give a long history.
[Personal Learning] [Learning by Practice]	(StudentB3): In each challenge, I feel I've been learning a lot of things that were never even mentioned in college and probably will not be. From everyone I worked with, I learned something new, from details to really relevant things that I still consider myself behind other colleagues. The last challenge in which we all create games, I can say that I finally found what I really enjoy doing
[Collaboration] [Communication] [Interpersonal]	(StudentB3): I often asked my teammates and even asked colleagues from other teams what they thought of my work. Receiving criticism from people with different mind sets was very enriching.
[Learning by Practice]	(StudentB4): Make mistakes and make mistakes again, as long as, you are aware. Mistakes teach all the ways we should not revisit. People learn by mistake, each in their own time, but the world does not wait. How to solve it? Make more mistakes in less time. In other words, practice. Scribble. Make a version. Two. Three or twenty, like a phase selection screen for my last project. [...] technical knowledge is not everything.

The Table 5.55 summarizes the outcomes of the students self-reflections of Team B, concerning to Mini Challenge and Studio.

Table 5.55 - Team B Results of Self-Reflections

Team B Results of Self-Reflections
Collaboration
Commitment
Communication
Design Learning
Interpersonal
Learning by Practice
Personal Learning
Planning
Programming
Project Business Learning
Teamwork
Time Management
Technical Research
Technical Learning

5.6 Conclusions from data analysis

In this section, we will discuss the results of cross analysis of each team´s observation with its respective self-reflections, as well as, the outcomes related to the contributions from reflective practice to software development and to the development of the individual competences. By the end, it will be presented the outcomes

concerning to research objectives from triangulation of the results from participant observation, results from self-reflections and the literature of reflective practice findings in studio.

5.6.1 Analysis of Self-Reflections and Practical Learning Contributions

Firstly, the objective is not to compare the result of analysis of each team but consider the results of both to summarize and consolidate the research outcomes. Although teams took part of the same studio challenge, their project development had different characteristics, experiences and student´s perceptions.

To facilitate the view shows Table 5.56 the cross analysis of the Team A Self-reflection results and studio Mini Challenge practical learning contributions obtained from coding and participant observation.

Table 5.56 - Team A Cross Analysis of Mini Challenge and Self-Reflections Results

Codes	Coding cycles and Participant Observation	Self-reflections
Adaptability	X	
Analytical		X
Availability		X
Creation	X	
Collaboration	X	X
Commitment	X	X
Communication	X	X
Communication Problem		X
Conflict Management	X	X
Decision-Making	X	
Design Learning		X
Interpersonal	X	X
Interpersonal Problem		X
Game Development (1st Experience)		X
Leadership	X	
Learning by Practice		X
Learning Experience		X
Pair Programming	X	
Personal Learning		X
Planning	X	
Problem Solving	X	
Programming		X
Project Management	X	
Self-Confidence		X

Scope Management	X	
Teamwork	X	X
Technical Research	X	
Time Management	X	

In the cross analysis of the self-reflection results and the Mini Challenge practical learning contributions it can be seen that in Team A, regarding to the way students relate and interact with each other, the outcomes were Collaboration, Commitment, Communication, Conflict Management, Interpersonal and Teamwork, and regarding to technical issues, programming. On the other hand, team reported design learning and game developing first experience, and both refers to full process of software development, encompassing technical issues. As well as learning experience, personal learning and learning by practice.

The resulting of cross analysis of the Team B Self-reflections and studio Mini Challenge practical learning contributions from coding and participant observation results is illustrated in Table 5.57.

Table 5.57 - Team B Cross Analysis of Mini Challenge and Self-Reflections Results

Codes	Coding cycles and Participant Observation	Self-Reflections
Adaptability	X	
Creation	X	
Collaboration	X	X
Commitment	X	X
Communication	X	X
Decision-Making	X	
Design Learning		X
Interpersonal	X	X
Learning Experience	X	
Learning by Practice		X
Pair Programming	X	
Personal Learning		X
Planning	X	X
Problem Solving	X	
Project Management	X	
Programming		X
Project Business Learning		X
Scope Management	X	
Teamwork	X	X
Technical Research	X	X

Technical Learning	X	X
Time Management	X	X

Team B concerning to the way one related to and interact with other people the outcomes were Collaboration, Commitment, Communication, Interpersonal and Teamwork. Regarding to technical issues, programming and technical research. Team still reported design learning, planning, project management and time management, which concerns to full process of software development, including technical learning. Apart from learning experience, personal learning and learning by practice.

Therefore, behind these cross analysis it was possible to confirm some outcomes of research analysis in the perception of the students, since they matched directly. In case of the outcomes from reflective practice coding, as decision-making, problem solving, scope management and project management, although not specifically mentioned in self-reflections, they are closely connected with design learning, project business learning and learning by practice that was reported by teams.

In addition, throughout the project development, the researcher noted in the participant observation that students of both teams had to practice and develop adaptability and flexibility skills for team integration, and to search and learn new techniques and software applications needed for project developing. Also, ability to negotiate, to document and leadership.

Therefore, the practical learning contributions found in the analysis of both teams from reflective practice or from matching of self-reflections and project observations results are the research´s outcomes related to contributions to the development of the individual competences.

5.6.2 Reflective Practice Outcomes

From the cross analysis of reflective practice results of both teams throughout the Mini Challenge developing it was possible to observe the following situations. As the studio mode of education from its root is based on the principle of reflective practice when analyzing the results of reflective practice, studio results are quoted together.

Reflective practice is present throughout the process of the project development as part integrant of this and could be seen in the interactions of instructor-student,

student-student, in formal or informal situations, public or in private, as team-meeting, group crit, interim review or final review.

Reflective interactions provoke reflection-in-action and the process of reflection-in-action promotes creation. In addition, during team meetings, each student acts as an agent who stimulate others to reflection-in-action, and certainly the studio's mode of education and the tools available collaborate and encourage this behavior.

Coaching and critique are tools that support reflective practice in the studio and they usually happen in the studio sessions.

In Group Crit sessions the instructor interactions are the most of time reflective, and few of them are guidance and instructive, and usually the outcome is student's reflection that leads to new ideas, product enhancement or developmental evolution.

In the Interim Review Design sessions, the presentation and exchange of ideas among students from different teams promotes and stimulates creation. Besides, in these sessions peers of other teams contributes most of the time questioning and sharing some references of related games for team reflection. The interactions with instructor are the most of time reflective and sometimes guidance and instructive.

As one could see, the studio sessions helps and incentive students reflect on the critiques, questions and suggestions from instructors and peers of the studio course, giving support to reflective practice.

According to the milestones defined in the schedule of the Mini Challenge, students had three days to define the project idea and presenting it at Interim Review studio session. This first project phase in terms of software development process correspond to the design definition. In other words, students had only three days to define the product of the game in terms of objective, customer focus and theme, technical and environmental features, using CBL framework to support and guide the project activities. Besides the short time for designing, they are agile and reach good results.

Such, at Design project phase, the team had to decide among different possibilities for the design, in which situations students had to manage in terms of roles and relationships, planning and acting, information gathering and sharing, problem analysis and understanding, concept generation and adoption, as well conflict avoiding and resolving.

Through the participant observation it is noticed that, in terms of team interaction, this phase was very rich because team members had to negotiate

strategies and, depending on the level of commitment and alignment with other team members, students adopted appropriate persuasion strategies. They carefully moderated their commitment to their ideas to remain amenable to negotiation.

Throughout the development of the challenge students were required to use and develop competences needed for professional practice, as verbal and written communication, teamwork, leadership, flexibility, ability to negotiate, to plan, to document, to manage and to solve project timetable problems, reducing the project scope or creating distinct deliverable versions of the product.

Technical research is a technical skill practiced in design and development phases. In design, when students were in searching of references of the games related to the game they will develop. In development, for instance, when the developers of team B had to research in a library of codes, the most appropriated to work with sound.

Moreover, the students had the opportunity to use the technique of pair programming. In the case of Team A the developers had never worked in pair before this project, thus it was required to acquire and develop this new ability during the development phase. Thus, they acquire a new technical skill.

Students of Team A had to practice the scope management, scope change and scope reduction to solve failures of project´s planning, and consequently they had to practice the project management and time management, also.

Students of Team B had to learn programming the sounds effects to develop a musical game.

Students had to manage an outsourced development for sound features. A third-party developer was responsible to develop the sound assets managed by game´s application according to the guidelines and requirements of the project. Depending on the game characteristics the third-party developer acts in a different way, for instance, in Team A he acts only as an external developer. However, for Team B, as the soundtrack is the mainly part of the game, the third-party worked some stages of the project development like a member of the team, working side by side with the programmers at the same time he was developing the soundtrack feature. Frequently they made reflection-in-action together the team members, that causes some product definitions along all phase of the development, since the design to test.

Moreover, when students justify decisions or directions based on experience from a previous challenge, become evidenced that students acquired some learning experience, technical or personal throughout the studio course.

Likewise, on the cross analysis of the students' self-reflections and the practical learning contributions by team, the student's perception matched with the main findings of the Mini Challenge results, as detailed in previous subsection.

Furthermore, this studio mode of education with challenge-based development and team turnover in each challenge brings the possibility of developing interpersonal relationships, personal communication, ability to deal with conflicts and leadership issues, learning different techniques.

In addition, these studio guidelines promote team collaboration instead of competitiveness, and encourage collaborative sharing of ideas and feedback among teams during development, not just in presentations at studio sessions.

The physical environment, facilitation and digital technology are available daily at the studio and students made use of them.

Studio is an educational place where students can practice and learn by practicing, consequently, it prepares students better for real-world practice.

As a result, studio brings the students closer to the real-world practices and it proves to be effective in training students in the developing of the competences for professional practice.

5.6.3 Research Outcomes

This research´s aim was to analyze the contributions of reflective practice to software development and to the development of individual competences in a software studio, once the reflective practice is the educational foundation of the studio concept. It was not our goal to compare this approach with any other approach.

Thus, the contributions of reflective practice to software development from reflective practice codes are emerging of new ideas (creation) and development of skills such as problem solving, decision-making, planning, project management, scope management and time management.

On the other hand, by participant observation of Mini-Challenge development it was noticed the skills (practicing and enhancing) development, as collaboration, verbal or written communication, commitment, interpersonal savvy, communication, adaptability, flexibility, teamwork, negotiation, and outsourced development management. Also, the use of learning experience and exchange of experience between peers from different teams. On technical aspect, it was observed: researching

of technical references, learning of how to work and integrate sound features with the application and self-learning of new technique, as pair programming.

From the analysis of self-reflections of the teams and Mini Challenge practical learning contributions of these teams, were identified: learning by practice and skills development of collaboration, commitment, communication, conflict management, interpersonal, personal learning, and teamwork. On technical aspects, it was observed: design learning, technical research (refers to references of games for designing and programming), technical learning (refers to applicative to lead with sound or audio kits), and programming.

Considering that these data comes from two different collecting methods, to reach the outcomes these data are triangulate, as represented in Figure 5.2.

Therefore, the contributions of reflective practice to the development of individual competence and the artistic talent in a software studio are skills developing of collaboration, verbal or written communication, commitment, interpersonal, adaptability, flexibility, teamwork, and outsourcing management; learning by practice of mobile software development and pair programming.

Thus, one can observe what Schön argued on reflective practice, that it helps students acquiring a kind of artistic talent essential for competence in professional practice, which refers to kinds of competence that practitioners demonstrate in certain practical situations that are unique, uncertain, and conflicting.

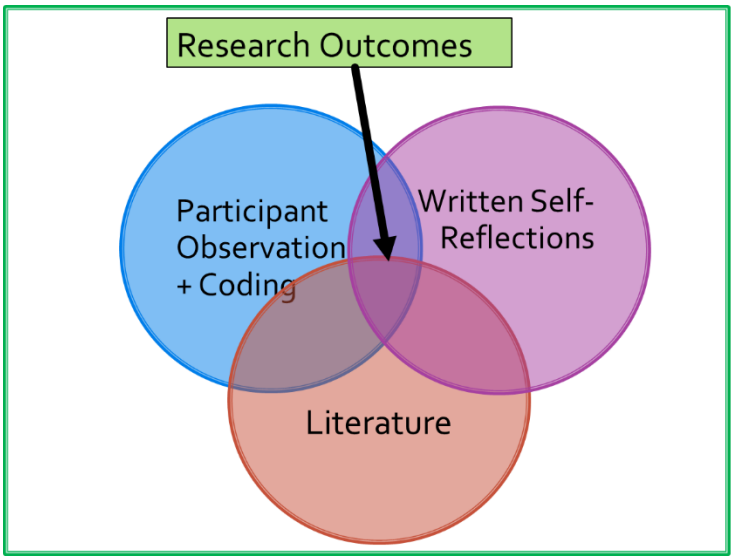


Figure 5.2 - Research Outcomes

5.7 Discussion

Although some promising results have been reported in the literature regarding studio context related to skills development, none of them focused on the outcomes of using reflective practice approach in the software studio, as we can see, in related works that concerns to this.

Nurkkala and Brandke (2011) found that the software studio experience is effective in preparing students to work as software professionals, considering feedback received from students and their managers. They do not discuss the reflective practice and its consequences.

Rosca (2018) argued that the exposition to real-world environment and industry practices in the studio provides the development of professional skills. She also related that working in groups fosters the development of good communication skills, acquiring strong life-long learning skills and the ability to work in interdisciplinary teams. She do not discuss the role of reflective practices.

Prior et.al. (2014) highlighted that the collaborative learning in the studio helps the students to develop their own skills by practice and groups appeared to be genuinely interested in the work of the other groups. Moreover, the dynamic interconnection of the set of elements in a studio like people, software tools, subject policies and procedures, development methodology, processes, techniques, documents, practices and products provide a network or web in which software development knowledge and skills are co-created. They do no study the reflective practice.

Prior et. al. (2019) argued that studio supports the development of a set of skills that cover a range of required employability skills desirable in career profile, according to Career EDGE Employability Development Profile. They observed and researched with the students which skills they developed at the studio that were listed as desirable in career profile.

This research observed the studio contributions to software development and the development of individual skills required for SE practitioners from the point of view of using reflective practice principles in the studio, listed in section 5.6.3. Therefore, this research provided an in-depth analysis of the contributions of reflective practice to software development and the development of individual competences or skills needed for professional practice and demonstrates the efficacy to achieve it in this studio

course, where the staff undoubtedly apply the reflective practice in students´ relations, beyond that other studio practices.

CHAPTER 6 - FINAL CONCLUSIONS

“The only source of knowledge is experience”
Albert Eisten

This Chapter summarizes and presents this ongoing research, its main results, contributions, and threats to validity.

6.1 Summary

As discussed in Chapter 1, it is possible to observe that educators have been looking for adequate approaches for practical learning, and studio education with reflective practice is one of these approaches.

Worldwide, there is some software studio interpretations, however it was not possible to find one study like this one, which analyzes the reflective practice contribution to the software process development in a software engineering studio in a broad and deep manner. This research aimed to contribute to fill this gap with a qualitative data analysis, investigating the effective practical learning contributions using reflective practice in a software studio.

First, the studio under evaluation is compliant to software studio definitions, fulfilling the characteristics defined in the software studio framework by Bull (2013), which are Physical environment, Facilitation of studio, Modes of education, Awareness, Critique, Culture, Individual’s characteristics, Inspiration, Collaboration and Digital technology.

For this study, the data were collected by participant observation of two mobile application development projects and were analyzed with support of coding cycles and Atlas.ti software tool. For achieving the research´s outcomes, the researcher made a triangulation of participant observation results and the results of self-reflections made by students at the end of the project development.

It was possible to observe and analyze the use of reflective practice approach in the relationship of the instructor-student, student-student, in formal or informal situations, public or in private, in team-meeting and studio meetings, as group crit, interim review and final review.

6.2 Contributions

First, it was possible to observe that the reflective practice, by reflection-in-action promotes the process of emerging new ideas.

Second, it was noticed that reflective practice in studio helps students in development of skills needed for software engineering practice, as adaptability, collaboration, verbal or written communication, flexibility, interpersonal, conflict management, teamwork, and outsourcing management. Also, project planning, time management, problem solving, decision-making and scope management. In addition, the learning and developing of new technical skills, as research for technical references, likewise programming skills, from the learning on how to work and integrate sound features with the application to pair programming. By the end, it allows students personal growth, learning from experience and design/project learning by practice.

As one could observe reflections help in process of learning by doing, either during the process of reflection or after by reflecting on experience to produce a better result.

Third, it was possible to conclude that studio helps to build a culture supportive of critique, emphasizes the practical learning and supports the development of some skills required for software engineering practice, which Schön called of development of “artistic talent” required for practitioners.

Finally, studio seems to be an authentic environment of relationship between academic disciplines and real-world experience, where students can practice and learn by practicing, thereby, it prepares students better for the real word.

Therefore, as Schön argued the reflective practice helps students acquire the kind of artistic talent essential for professional practice.

6.3 Threats to Validation

This study was performed in one specific software development studio and it is only covering mobile application development. Besides, the analysis was performed by the participant researcher and may have some bias.

6.4 Future Works

Repeat this study, with two teams in parallel, one of them using reflective practice in a software studio and another not using it, for instance, in a classic educational environment of a computer science or software engineering course.

Other possibility is to provide the same design requirements for a studio team and one team in a company that do not use reflective practice to compare results.

BIBLIOGRAPHY

(ACM/IEEE, 2001) Computer Curriculum 2001. **Computer Science. Composition of the Curriculum 2001** Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery. Final Report. Sloan, Robert; Carver, Doris; Eckhouse et. al. 2001, December 2001.

(ACM/IEEE, 2013) Computer Science Curricula 2013. **Curiculum Guidelines for Undergraduate Degree Programs in Computer Science 2013** Joint Task Force on Computing Curricula Computer Association for Computing Machinery(ACM) IEEE Computer Society. December 2013.

(ACM/IEEE, 2016) Computer Engineering Curricula 2016. **Curiculum Guidelines for Undergraduate Degree Programs in Computer Engineering 2016**. A Report in the Computing Curricula Series Joint Task Force on Computer Engineering Curricula Association for Computing Machinery (ACM) IEEE Computer Society. December 2015.

(ACOT, 2008). Apple Classrooms of Tomorrow - Today Learning in the 21st Century.

(BAKER and VAN DER HOEK, 2009) Baker, Alex and Van Der Hoek, André. **An experience report on the design and delivery of two new software design courses**. In: Proceedings of the 41th ACM Technical Symposium on Computer Science Education (SIGCSE '09). Chattanooga, Tennessee, USA: ACM, pp. 519-523, March 2009.

(BINDER et al., 2017) Binder, Fabio Vinicius; Nichols, Mark; Reinehr, Sheila; Malucelli, Andreia. **Challenge Based Learning Applied to Mobile Software Development Teaching.** In: IEEE 30th Conference on Software Engineering Education and Training (CSEE&T 2017), IEEE, 2017, Savannah, Georgia, USA, November, pp. 57-64.

(BROADFOOT and Bennett, 2003) Broadfoot, Ouita and Rick Bennett (2003). **Design Studios: Online? Comparing traditional face-to-face design studio education with modern Internet-based design studios**. In: Paper presented at the Apple University Consortium, pp. 1–13.

(BROOKS, 1995) Brooks, Frederick P. Jr. **The Mythical Man-Month: Essays on Software Engineering,** Anniversary Edition, 1st Edition, Boston, MA: Addison Wesley Longman, Inc., 1995, 17th printing, August 2002, 322 p.

(BULL et. al, 2013) C.N. Bull, J. Whittle, and L. Cruickshank, **Studios in Software Engineering Education: Towards an Evaluable Model**, International Conference on Software Engineering (ICSE 13), 2013, pp. 1063–1072.

(BULL and WHITTLE, 2014) Christopher N. Bull; Jon Whittle. **Observations of a Software Engineering Studio: Reflecting with the Studio Framework**. In: IEEE 27th Conference on Software Engineering Education and Training (CSEE&T 2014), 2014, IEEE, pp. 74-83.

(BULL and WHITTLE, 2014) Christopher N. Bull; Jon Whittle. **Supporting Reflective Practice in Software Engineering Education through a Studio-Based Approach**. IEEE Software, 2014, IEEE, v. Jul/Aug, Issue 1, p. 44-50.

(CARBONE and SHEARD, 2002) Carbone, Angela; Sheard, Judy. **A studio-based teaching and learning model in IT.** In: Proceedings of the 7th annual conference on Innovation and technology in Computer Science education (ITiCSE'02), 2002, v. 34, Issue 4, pp. 213-217.

(CENNAMO et al., 2011) Cennamo, Katherine; Douglas, Sarah A.; Vernon, Mitzi; Brandt, Carol; Scott, Brigitte; Reimer, Yolanda; McGrath, Margarita. **Promoting creativity in the Computer Science design studio.** In: Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE'11), 2011, ACM, pp. 649--654.

(DANIELEWICZ-BETZ and TATSUKI, 2014) Danielewicz-betz, Anna; Kawaguchi, Tatsuki. **Gaining hands-on experience via collaborative learning: Interactive Computer Science Courses**. In: 2014 International Conference on Interactive Collaborative Learning (ICL), 2014, IEEE, December, pp. 403-409.

(DEWEY, 1974) John Dewey. How we Think. D.C.HEATH & CO., PUBLISHERS. Collection universal library. Contributor Universal Digital Library. Language English. Barcode 104680. Call number 838. Copyright owner D.C.HEATH & CO., PUBLISHERS. Publication date 1910.

(DYBÅ et al., 2014) Tore Dybå; Neil Maiden; Robert Glass. **The Reflective Software Engineer: Reflective Practice**. IEEE Software, 2014, IEEE.

(FINLAY, 2008) Finlay, Linda. **Reflecting on 'Reflective practice'.** *w*ww.open.ac.uk/cetlworkspace/cetlcontent/.../4bf2b48887459.pdf, accessed 23rd March, 2018.

(FLICK, 2004) Flick, Uwe. Pesquisa qualitativa. Tradução: Sandra Netz. 2nd Edition, Porto Alegre: Artmed Editora S. A., 2004, 312 p.

(FLICK, 2009) Flick Uwe. Qualidade na pesquisa qualitativa. Tradução: Roberto Cataldo Costa. Porto Alegre: Artmed Editora S. A., 2009, 196 p.

(FLICK, 2009) Flick Uwe. An introduction to qualitative research. Four Edition SAGE Publication Ltd., Printed in Great Britain by Ashford Colour Press Ltd., 2009, 504 p.

(FETTERMAN, 2010) Fetterman, David M. Ethnography Step by Step, SAGE Publication Ltd., 2010, 200 p.

(GHEZZI, 2005) Ghezzi, Carlo; Mandrioli, Dino, 2005. **The Challenges of Software Engineering Education.** In: Proceedings of the 27th international conference on Software engineering, 2005, ACM, pp. 637-638, 2005.

(GESTWICKI and McNELY, 2016) Gestwicki, Paul; McNely, Brian. **Education Interdisciplinary Projects in the Academic Studio**. ACM Transactions on Computing, v.16, Issue 2 , p 1- 24.

(HAZZAN, 2002) Hazzan, Orit. **The reflective practitioner perspective in software engineering education**. The Journal of Systems and Software, v. 63, Issue 3, p. 161–171.

(HAZZAN & TOMAYCO, 2003) Hazzan, Orit. and Tomayko, Jim E. (2003). **The reflective practitioner perspective in eXtreme Programming**, *Proceedings of the XP Agile Universe 2003*, New Orleans, Louisiana, USA, p. 51-61.

(HAZZAN and TOMAYKO, 2004) Hazzan, Orit; TOMAYKO, Jim E. **The Reflective Practitioner Perspective in Software Engineering.** Position Paper: CHI 2004 One Day Workshop - Designing for Reflective Practitioners.

(HILBURN, 1999) Hilburn, Thomas B.; Hislop, Greg; Bagert, Donald J.; Lutz, Michael; Mengel, Susan; McCracken, Michael. **Guidance for the development of software engineering education programs.** Journal of Systems and Software, v. 49, Issue 2, p. 163-169, 1999.

(HOKANSON, 2012) Hokanson, Brad. **The Design Critique as a Model for Distributed Learning.** In: The Next Generation of Distance Education. Ed. by Leslie Moller and Jason B. Huett. Springer US. Chap. The Design, v. 9781461417, pp. 71–83, 2012.

(HUNDHAUSENH et al., 2008) Hundhausen, Christopher D., N. Hari Narayanan and Martha E. Crosby (2008). **Exploring studio-based instructional models for computing education**. In: Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE '08). New York, NY, USA: ACM, pp. 392–396, March 2008.

(HUNDHAUSEN et. al., 2013) Hundhausen, Christopher D.; Agrawal, Anakrati; Agarwal, Pawan. **Talking about Code: Integrating Pedagogical Code Reviews into Early Computing Courses**. ACM Transactions on Computing Education, v.13, Issue 3, article 14, 28 pages, August 2013.

(IEEE, 2004) IEEE Computer Society. **Guide to the Software Engineering Body of Knowledge (SWEBOK).** Los Amitos: CS Press, 2004, 204p.

(IEEE/ACM, 2004) Sobe, Ann; Lethbridge, Timothy C; Diaz-Herrera, Jorge L; Barrie Thompson, J. IEEE/ACM Joint Task Force on Computing Curricula. **Software Engineering 2004**, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, IEEE Computer Society Press and ACM Press, August 2004.

(IEEE/ACM, 2014) **Software Engineering 2014.** Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE/ACM Joint Task Force on Computing Curricula IEEE Computer Society and Association for Computing Machinery. A Volume of the Computing Curricula Series. February 2015.

(JAZAYERI, 2004) M. Jazayeri. **The education of a software engineer.** In: Proceedings of the 19th International Conference on Automated Software Engineering, Linz, Austria, 2004, pp. xviii-xxvii.

(KUHN, 1998) Kuhn, Sarah. **The software design studio: An exploration.** IEEE Software**,** IEEE, v. 15, Issue 2, p. 65-71, 1998.

(KUHN et al., 2002) Kuhn, S.; Hazzan, O.; Tomayko, J. E.; Corson, B. **The software studio in software engineering education.** In: 15th Conference on Software Engineering Education and Training (CSEE&T 2002), Proceedings, Kentucky, USA, p. 236-238, 2002.

(KOPCZYŃSKA et al., 2012) Kopczyńska, Sylwia; Nawrocki, J; Ochodek, M. **Software development studio - Bringing industrial environment to a classroom.** In: 2012 1st International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex 2012), 2012, pp. 13-16.

(LETHBRIDGE et. al., 2007) Lethbridge, T C; Diaz-Herrera, J; LeBlanc, R J;Thompson, J B. **Improving software practice through education: Challenges and future trends**. Future of Software Engineering FOSE 07, v. 2, p. 12-28, 2007.

(MUNBY, 1989) Munby, Hugh. **Reflection-in-Action and Reflection-on-Action,** *Education and Culture*, v. 09, Issue 1, Article 4.

(NICHOLS et al., 2016) Nichols, Mark; Cator, Karen;Torres, Marco. & Henderson, D. (2016). Challenge Based Learning User Guide. Redwood City, CA. Digital Promise.

NIPPERT-ENG, Christena. Watchig Closely. A Guide to Ethnographic Observation. Oxford University Press, 2015.

(PRIOR et al., 2019) Prior, Julia; Suman, Laudari; Leaney, John. **What is the Effect of a Software Studio Experience on a Student´s Employability?** In: Proceedings of the Twenty-First Australasian Computing Education Conference (ACE'19), ACM. Sydney, NSW, Australia, pp. 28-36, January 2019.

(O'GRADY, 2006) O'GRADY, Jenn and Kenn V. **A Designer's research manual. Succeed in Design by knowing your clients and what they really need***.* Berverly: ed. RockPort, 2006.

(OH et al., 2013) Oh, Yeonjoo Oh; Ishizaki, Suguru; Gross, Mark D.; Yi-Luen Do, Ellen. **A theoretical framework of design critiquing in architecture studios.** In: *Design Studies* 34.3, pp. 302–325, 2013.

(PRIOR et al., 2014) Prior, Julia; Connor, Andrea; Leaney, John. **Things coming together: learning experiences in a software studio**. In: Proceedings of the 2014 conference on Innovation & technology in Computer Science education, 2014, IEEE, pp. 129-134.

(PRIOR et al., 2016) Prior, Julia; Ferguson, Samuel; Leaney, John. **Reflection is Hard: Teaching and Learning Reflective Practice in a Software Studio**. In: Proceedings of the Australasian Computer Science Week Multiconference (ACSW '16). Canberra, Australia, pp. 7-15, February 2016.

(PRIOR et al., 2019) Prior, Julia; Suman, Laudari; Leaney, John. **What is the Effect of a Software Studio Experience on a Student´s Employability?** In: Proceedings

of the Twenty-First Australasian Computing Education Conference (ACE'19), ACM. Sydney, NSW, Australia, pp. 28-36, January 2019.

(ROSCA, 2018) Rosca, Daniela. **Acquiring professional software engineering skills through studio-based learning.** In: 17th International Conference on Information Technology Based Higher Education and Training, (ITHET'18), IEEE. Olhao, Portugal, pp. 1-6, Abril 2018.

(SALDAÑA, 2009) Saldaña, Johnny. The Coding Manual for Qualitative Researchers. London: SAGE Publication Ltd, 2009, 244 p.

(SALDAÑA, 2016) Saldaña, Johnny. The Coding Manual for Qualitative Researchers. London: SAGE Publication Ltd, 2016, 339 p.

(SALAMA, 1995) Salama, Ashraf. Design Education and Studio Work in the Conventional Approach**.** In**: New Trends in Architectural Education: Desiging the Design Studio**. Raleigh, N.C.: Tailor
ed Text, 1995, 191 p.

(SCHÖN, 1983) Schön, Donald A. Reflective Pratictioner: How Professionals Think in action. New York: Basic Books Inc., 1983, 374 p.

(SCHÖN, 1987) Schön, Donald A. Teaching artistry through reflection-in-action. In: **Educating the Reflective Practitioner: Toward a new design for teaching and learning in the professions**. San Francisco: Jossey-Bass. Re-printed with permission of the publisher via Copyright Clearence Center, Inc. Reproduced with permission at Teachers College, Columbia University, p. 22-40.

(SCHÖN, 1987) Schön, Donald A. Educating the Reflective Practitioner: Toward a new design for teaching and learning in the professions. First Edition, San Francisco, CA, US: Jossey-Bass, 1987.

(SCHÖN, 1992) Schön, Donald A. The Theory of Inquiry: **Dewey's Legacy to Education**. The Massachusetts Institute of Technology Cambridge, Massachusetts Published online: 15 Dec 2014. Curriculum Inquiry, v. 22, issue 2, p. 119-139.

(SCHÖN, 2000) Schön, Donald A. Educando o profissional reflexivo: um novo design para o ensino e aprendizagem, tradução Roberto Cataldo Costa, Porto Alegre: Artes Medicas Sul, 2000. 256 p.

(SHARP et al., 2016) Helen Sharp; Y. Dittrich: Cleidson De Souza. **The role of ethnographic studies in empirical software engineering.** In: IEEE Transactions on Software Engineering, IEEE, v. 422, issue 8, p. 786-804, 2016.

(SPRADLEY, 1979) Spradley James P. The ethnographic interview. New York: Holt, Rinehart and Winston, 1979.

(SPRADLEY, 1980) Spradley James P. Participant Observation. New York: Holt, Rinehart and Winston, 1980.

(TOMAYCO, 1991) Tomayco, James E. **Teaching software development in a studio environment.** In: Proceedings of the twenty-second SIGCSE technical symposium on Computer science education (SIGCSE '91), v. 23, issue 1, 1991, ACM, pp. 300-302, March 1991.

(TOMAYCO et. al. 1997) Tomayco, James E.: Garlan, David; Glush , David p.. **Agents of Change: Educating Software Engineering Leaders**, IEEE Software, v. 30, Issue 11, p. 59-65, 1997.

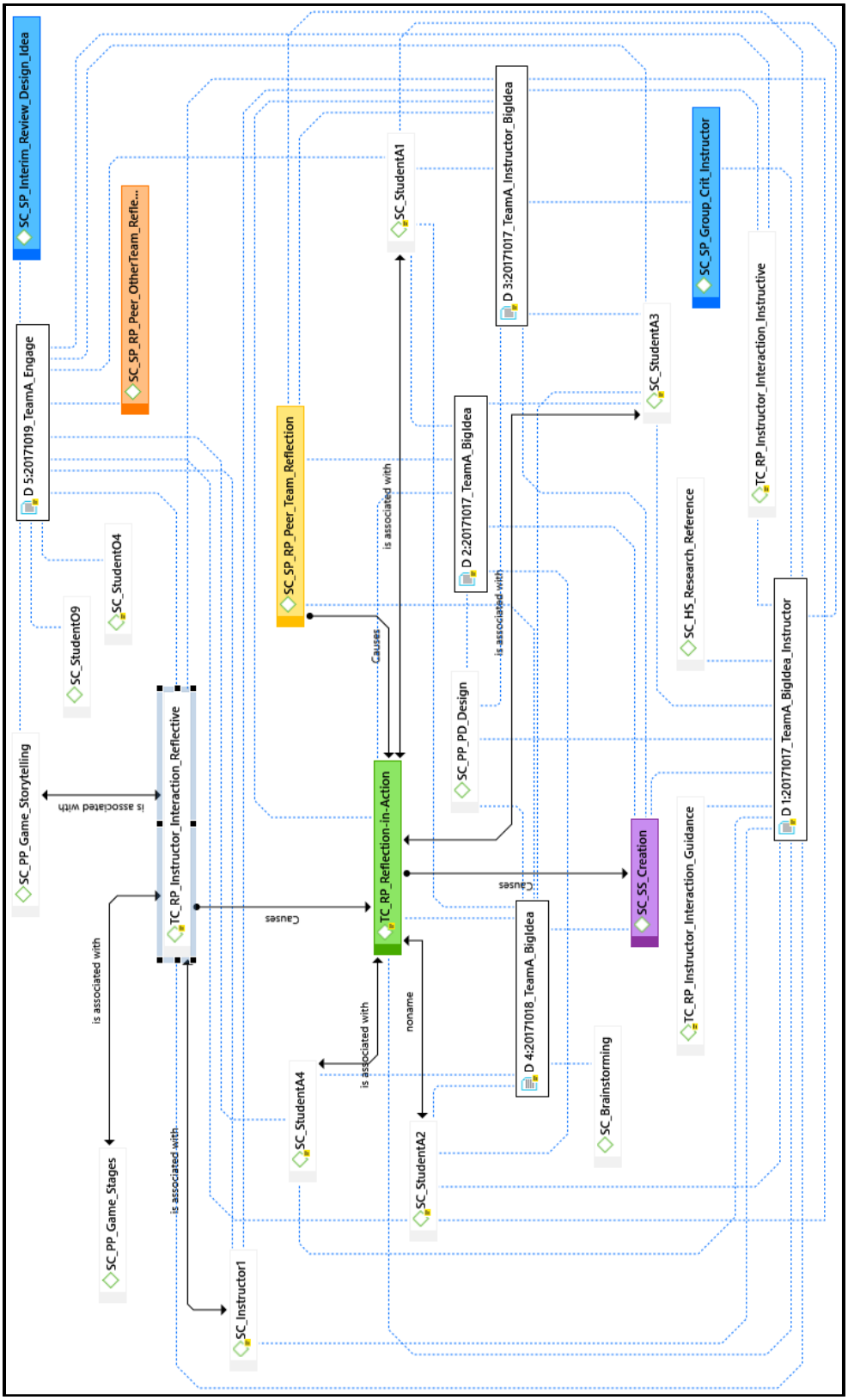
(TOMAYKO, 1998) TOMAYKO, J. E. **Forging a discipline: An outline history of software engineering education.** In: Annals of Software Engineering, v.6, 1998, Springer, pp. 3-18, March 1998.

(WAKS, 2001) Waks, L. J. **Donald Schon's Philosophy of Design and Design Education**. International Journal of Technology and Design Education, v.1, p. 37-51, 2001.

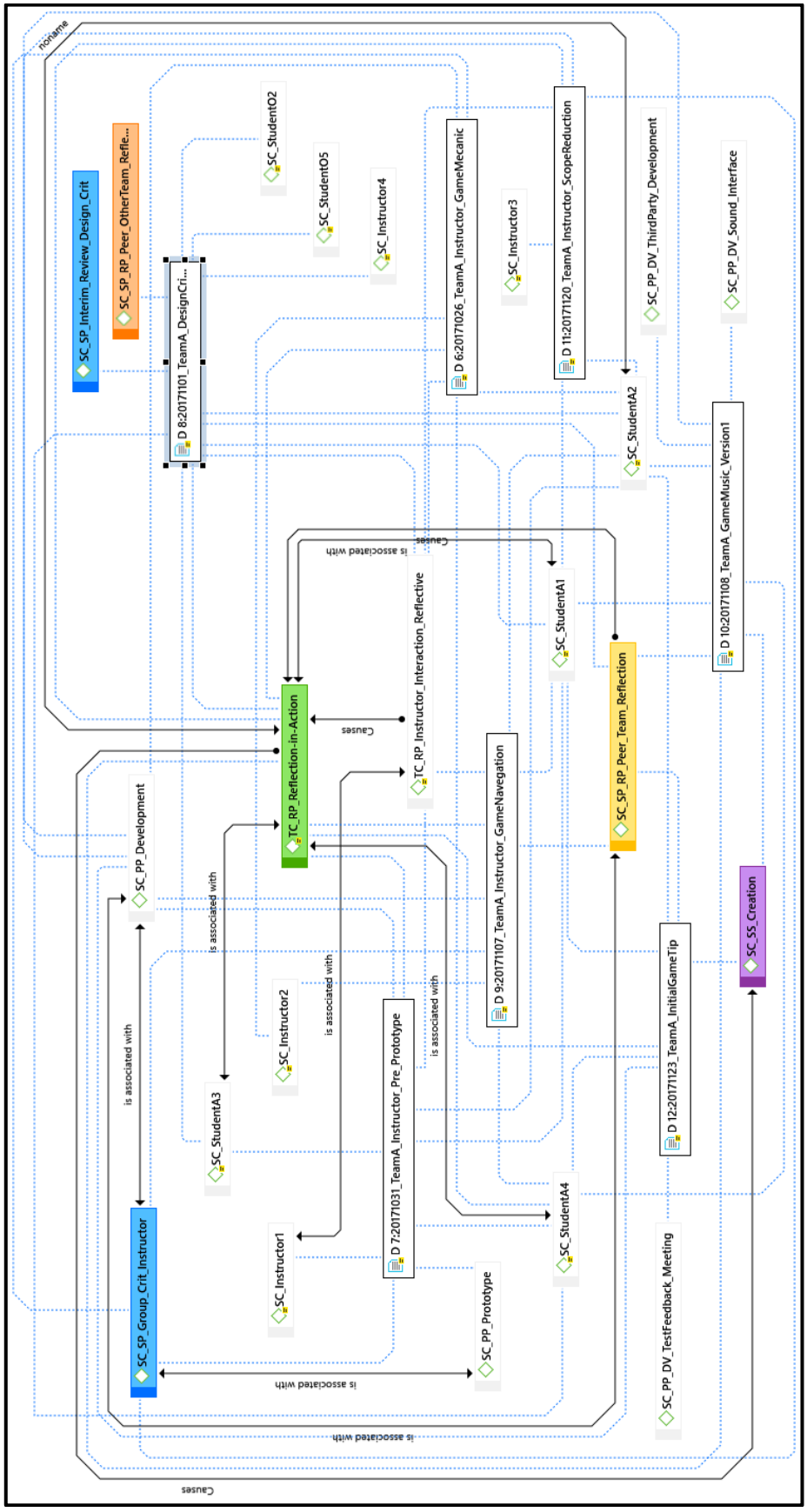
APPENDIX

APPENDIX A - TEAM A NETWORKs from Atlas.ti

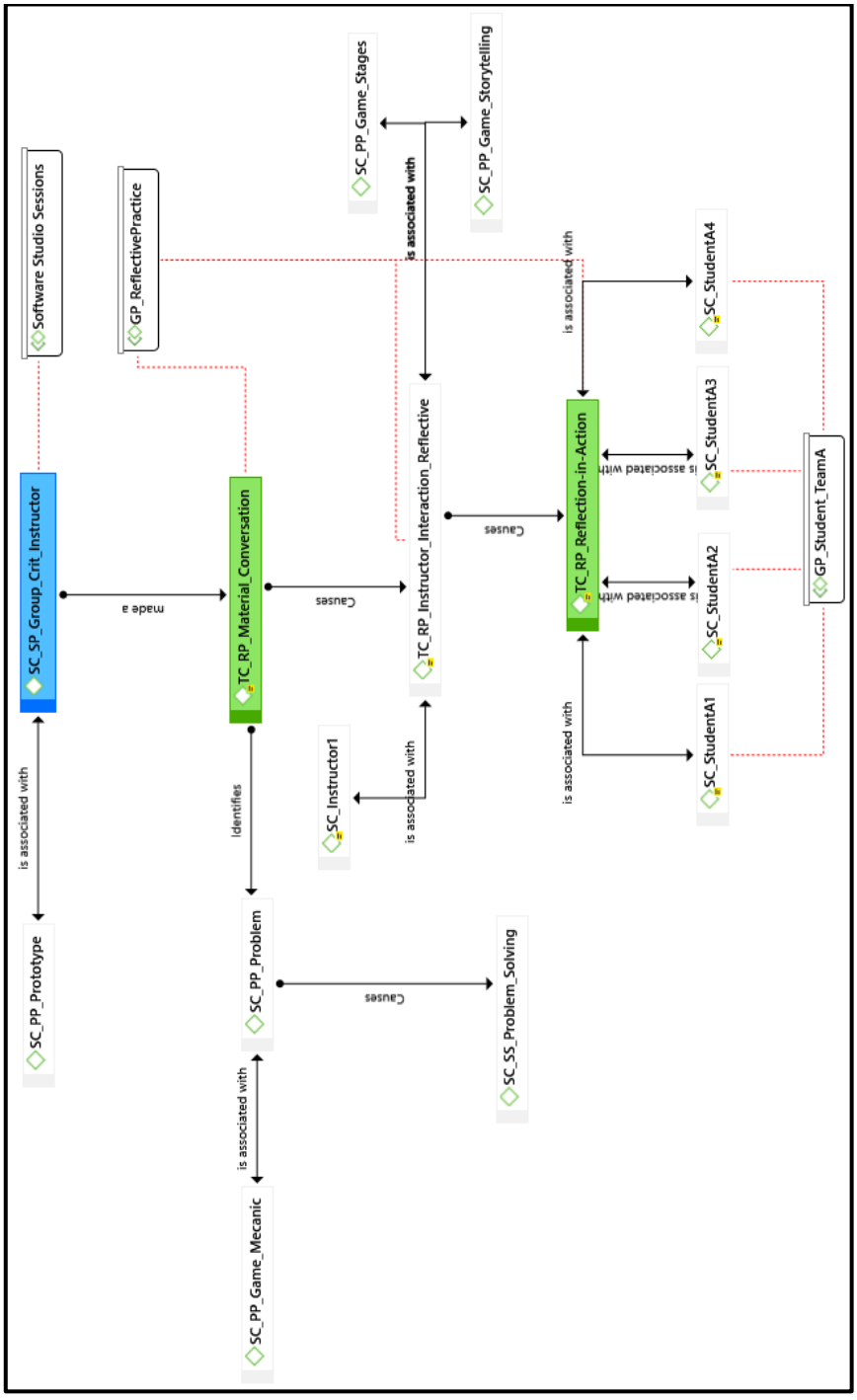
Team A Reflection-in-action at Design Network



Team A Reflection-in-action at Development Network

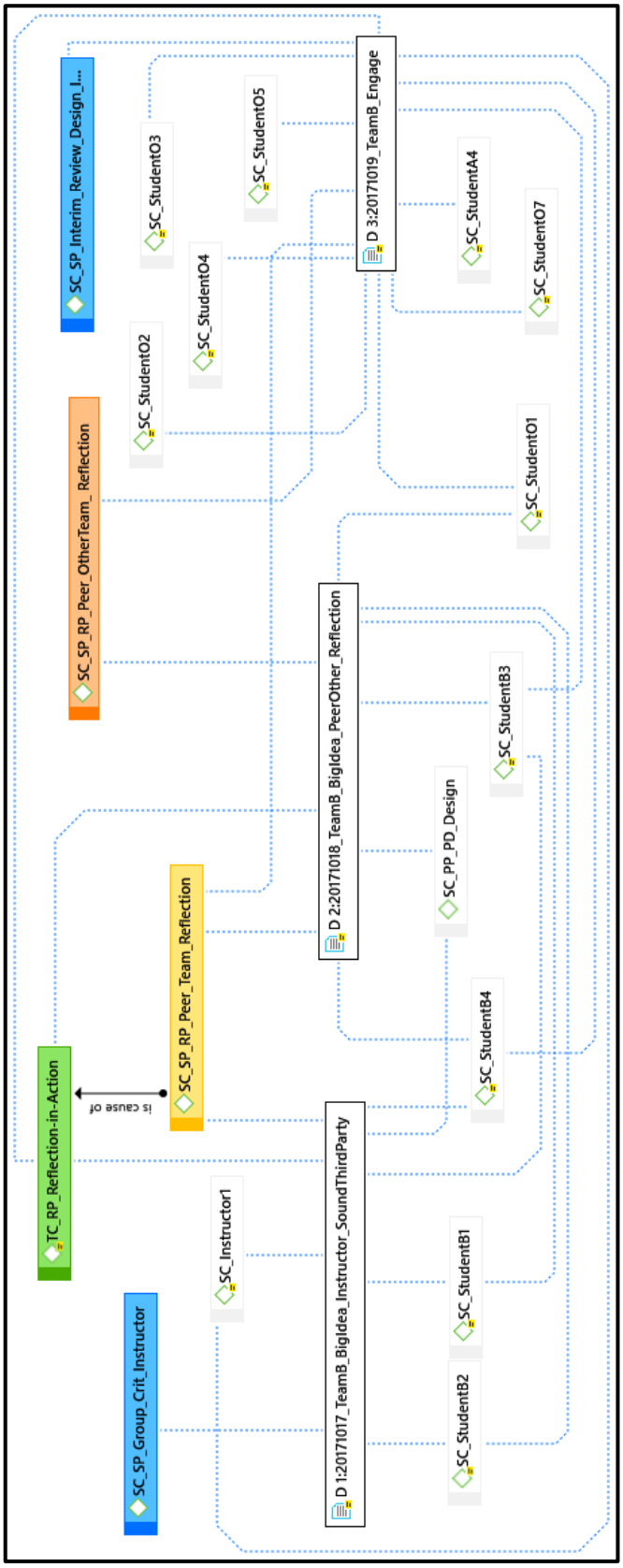


Team A Conversation with the Material at Development Network

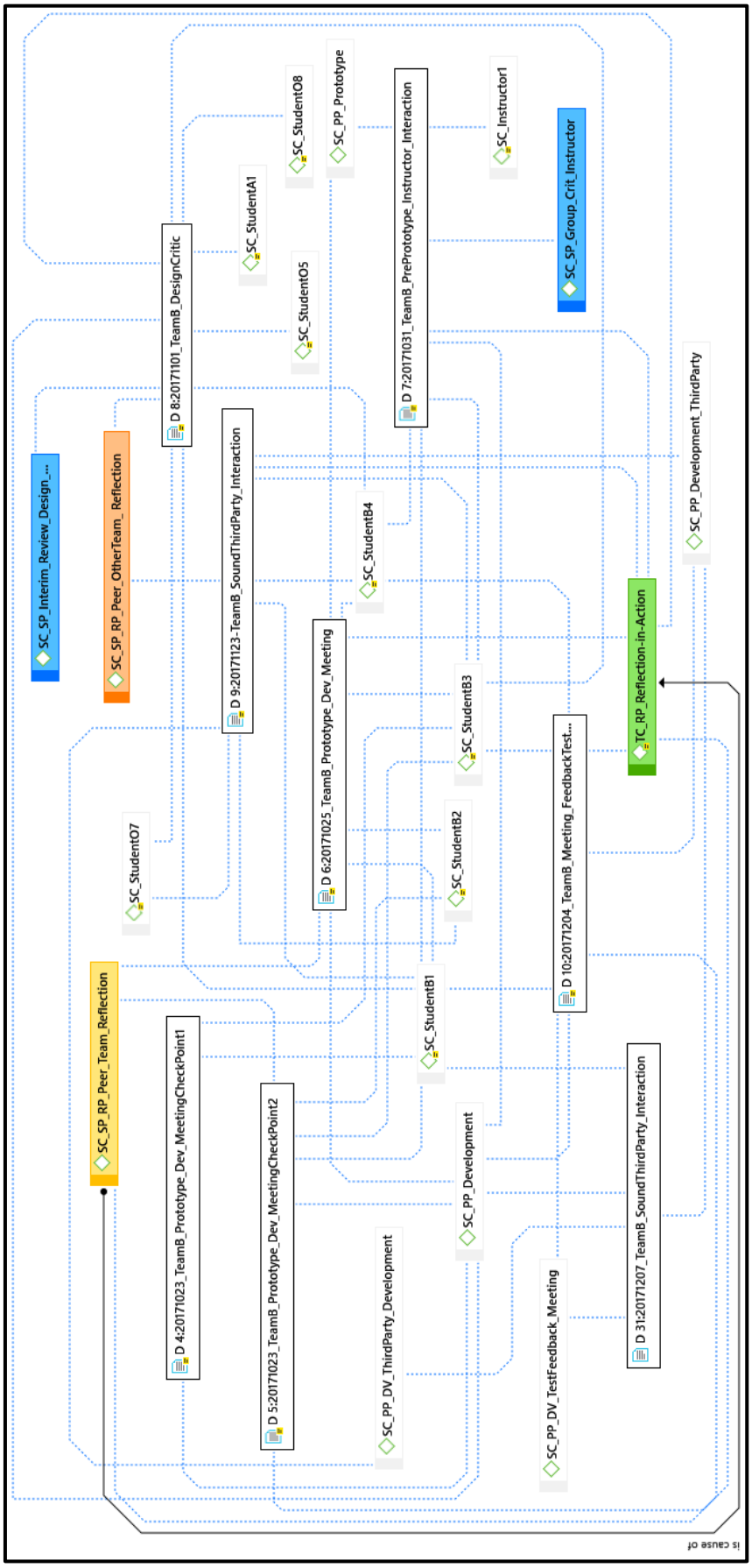


APPENDIX B – TEAM B NETWORKs from Atlas.ti

Team B - Reflection-in-action at Design Network



Team B - Reflection-in-action at Development Network


SC_SP_Interim_Review_Design_...
SC_SP_RP_Peer_OtherTeam_Reflection
D 8:20171101_TeamB_DesignCritic
SC_StudentA1
SC_StudentO8
SC_PP_Prototype
SC_Instructor1
SC_SP_Group_Crit_Instructor
D 7:20171031_TeamB_PrePrototype_Instructor_Interaction
SC_StudentO5
D 9:20171123-TeamB_SoundThirdParty_Interaction
SC_StudentB4
SC_PP_Development_ThirdParty
TC_RP_Reflection-in-Action
D 6:20171025_TeamB_Prototype_Dev_Meeting
SC_StudentB3
SC_StudentO7
SC_StudentB2
SC_SP_RP_Peer_Team_Reflection
D 10:20171204_TeamB_Meeting_FeedbackTest...
SC_StudentB1
D 4:20171023_TeamB_Prototype_Dev_MeetingCheckPoint1
D 5:20171023_TeamB_Prototype_Dev_MeetingCheckPoint2
SC_PP_Development
D 31:20171207_TeamB_SoundThirdParty_Interaction
SC_PP_DV_ThirdParty_Development
SC_PP_DV_TestFeedback_Meeting
is cause of

APPENDIX C – TEAM A – Tables of Co-occurrence from ATLAS.ti

Atlas.ti Table of codes – Descriptions and coding groups

Code	Comment	Code Group 1
SC_Instructor1	Structure Code Instructor1	Instructor
SC_Instructor2	Structure Code Instructor2	Instructor
SC_Instructor3	Structure Code Instructor3	Instructor
SC_Instructor4	Structure Code Instructor4	Instructor
SC_PP_Development	Structure Code Development Phase	
SC_PP_Development_ThirdParty	Structure Code Third Party Development	
SC_PP_Documentation	Structure Code Documentation	
SC_PP_Scope_Change	Structure Code Project Scope Change	
SC_PP_Scope_Reduction	Structure Code Project Scope Reduction	
SC_SP_Final_Review	Structure Code Studio Final Review Studio Session	
SC_SP_Group_Crit_Instructor	Structure Code Studio Group Crit Session	
SC_SP_Interim_Review	Structure Code Studio Session Interim Review	
SC_SP_Interim_Review_Design_Crit	Structure Code Sudio Session Interim Review Design Crit	Interim Review
SC_SP_Interim_Review_Design_Idea	Structure Code Studio Session Interim Review Design Idea	Interim Review
SC_SP_RP_Group_Peer_Reflection	Structure Code Group Peer Reflection - Interim Review	
SC_SP_RP_Peer_OtherTeam_ Reflection	Structure Code Peers of Other Team Reflection	
SC_SP_RP_Peer_Team_Reflection	Structure Code Teammate Reflection	
SC_SP_RP_ThirdParty_Interaction	Structure Code Interaction with Third Party	
SC_SS_Creation	Creation	
SC_StudentA1	Structure Code StudentA1 (Team A)	StudentA
SC_StudentA2	Structure Code StudentA2 (Team A)	StudentA
SC_StudentA3	Structure Code StudentA3 (Team A)	StudentA
SC_StudentA4	Structure Code StudentA4 (Team A)	StudentA
SC_StudentB1	Structure Code StudentB1(Team B)	StudentB
SC_StudentB2	Structure Code StudentB2 (Team B)	StudentB
SC_StudentB3	Structure Code StudentB3 (Team B)	StudentB
SC_StudentB4	Structure Code StudentB4 (Team B)	StudentB
SC_StudentO1	Structure Code StudentO1 (Other Team)	StudentO
SC_StudentO2	Structure Code StudentO2 (Other Team)	StudentO
SC_StudentO3	Structure Code StudentO3 (Other Team)	StudentO
SC_StudentO4	Structure Code StudentO4 (Other Team)	StudentO
SC_StudentO5	Structure Code StudentO5 (Other Team)	StudentO
SC_StudentO6	Structure Code StudentO6 (Other Team)	StudentO
SC_StudentO7	Structure Code StudentO7 (Other Team)	StudentO
SC_StudentO8	Structure Code StudentO8 (Other Team)	StudentO
SC_ThirdParty1	Structure Code - ThirdParty Developer	
TC_RP_Instructor_Interaction_Guidance	Taxonomy Code - Reflective Practice - Instructor Interaction - Guidance	Practice - Instructor
TC_RP_Instructor_Interaction_Instructive	Taxonomy Code - Reflective Practice - Instructor Interaction - Instructive	Practice - Instructor
TC_RP_Instructor_Interaction_Reflective	Taxonomy Code - Reflective Practice - Instructor Interaction - Reflective	Practice - Instructor
TC_RP_Material_Conversation	Taxonomy Code - Reflective Practice - Conversation with Material	Reflective Practice
TC_RP_Problematization	Taxonomy Code - Reflective Practice - Problematization	Code - Reflective
TC_RP_Reflection-in-Action	Taxonomy Code - Reflective Practice - Reflection-in-Action	Reflective Practice - Reflection
TC_RP_Reflection-on-Action	Taxonomy Code - Reflective Practice - Reflection-on-Action	Reflective Practice - Reflection
TC_RP_ThirdParty_Interaction_Instructive	Taxonomy Code - Reflective Practice - ThirdParty - Interaction Instructive	ThirdParty
TC_RP_ThirdParty_Interaction_Reflective	Taxonomy Code - Reflective Practice - ThirdParty Interaction - Reflective	ThirdParty

Code	Comment	Code Group 2
SC_Critical_Thinking	Critical Thinking	Learning Practice
SC_HS_Conflict_Management	Conflict Management	Learning Practice
SC_HS_PairProgramming	Pair Programming	Learning Practice
SC_HS_Planning	Project Planning	Learning Practice
SC_HS_Project_Management	Project Management	Learning Practice
SC_HS_Research_Reference	Research Reference	Learning Practice
SC_HS_Scope_Management	Project Scope Management	Learning Practice
SC_HS_TeamWork	Project Teamwork	Learning Practice
SC_SS_Adaptaability	Adaptability	Learning Practice
SC_SS_Analytical	Analytcal	Learning Practice
SC_SS_Commitment	Commitent	Learning Practice
SC_SS_Communication	Communcation	Learning Practice
SC_SS_Conflict_Management	Conflict Management	Learning Practice
SC_SS_Flexibility	Flexiblity	Learning Practice
SC_SS_Interpersonal	Interpersonal	Learning Practice
SC_SS_Learn_Change_Peer_Experience	Learning from change of experience with Peer	Learning Practice
SC_SS_Learning_ByPractice	Learning from practice	Learning Practice
SC_SS_Learning_Experience	Learning Experience	Learning Practice
SC_SS_Peer_Experience	Peer Experience	Learning Practice
SC_SS_Problem_Solving	Problem Solving	Learning Practice
SC_SS_Tech_UnKwonSkill	Technincal UnKnonw Skill	Learning Practice
SC_SS_Time_Management	Time Management	Learning Practice
SC_Technical_Reference	Technincal Reference (unkwonw applicative or technique)	Learning Practice

Team A - Reflective Practice - Co-occurrence Table at Design

atlas.ti QUALITATIVE DATA ANALYSIS	● TC_RP_Material_Conversation Gr=34	● TC_RP_Reflection-in-Action Gr=180	● TC_RP_Reflection-on-Action Gr=12
○ SC_PP_PD_Design Gr=123	0	80	0
● SC_SP_Group_Crit_Instructor Gr=194	28	87	3
● SC_SP_Interim_Review_Design_Crit Gr=18	0	10	0
● SC_SP_Interim_Review_Design_Idea Gr=23	0	0	0
○ TC_RP_Instructor_Interaction_Guidance Gr=39	0	19	0
○ TC_RP_Instructor_Interaction_Instructive Gr=50	0	19	0
○ TC_RP_Instructor_Interaction_Reflective Gr=170	23	92	0
○ TC_RP_ThirdParty_Interaction_Reflective Gr=3	0	0	2
● SC_SP_RP_Group_Peer_Reflection Gr=10	0	4	0
● SC_SP_RP_Peer_OtherTeam_Reflection Gr=7	0	1	0
● SC_SP_RP_Peer_Team_Reflection Gr=151	6	110	9

Team A - Reflective Practice - Co-occurrence Table at Development

atlas.ti QUALITATIVE DATA ANALYSIS	● TC_RP_Material_Conversation Gr=34	● TC_RP_Reflection-in-Action Gr=180	● TC_RP_Reflection-on-Action Gr=12
○ SC_PP_Development Gr=194	34	90	12
○ SC_PP_Prototype Gr=87	25	36	0
● SC_SP_Group_Crit_Instructor Gr=194	28	87	3
○ TC_RP_Instructor_Interaction_Guidance Gr=39	0	19	0
○ TC_RP_Instructor_Interaction_Instructive Gr=50	0	19	0
○ TC_RP_Instructor_Interaction_Reflective Gr=170	23	92	0
○ TC_RP_ThirdParty_Interaction_Reflective Gr=3	0	0	2
● SC_SP_RP_Group_Peer_Reflection Gr=10	0	4	0
● SC_SP_RP_Peer_OtherTeam_Reflection Gr=7	0	1	0
● SC_SP_RP_Peer_Team_Reflection Gr=151	6	110	9

Team A – Practical Learnings- Co-occurence Table at Design

atlas.ti QUALITATIVE DATA ANALYSIS	○ SC_SS_Creation Gr=17	○ SC_HS_Research_Reference Gr=7	○ SC_HS_Scope_Management Gr=11	○ SC_PP_Scope_Change Gr=1	○ SC_PP_Scope_Reduction Gr=2	○ SC_SS_Decision_Making Gr=1	○ SC_SS_Problem_Solving Gr=2	○ SC_PP_DV_ThirdParty_Development Gr=12	○ SC_PP_DV_ThirdParty_Requirements Gr=7
○ SC_PP_PD_Design Gr=123	12	2	0	0	0	0	0	0	0
● SC_SP_Group_Crit_Instructor Gr=194	3	4	11	1	2	1	2	0	0
● SC_SP_Interim_Review_Design_Crit Gr=18	0	3	0	0	0	0	0	0	0
● SC_SP_Interim_Review_Design_Idea Gr=23	0	0	0	0	0	0	0	0	0
○ TC_RP_Instructor_Interaction_Guidance Gr=39	0	0	9	0	0	0	0	0	0
○ TC_RP_Instructor_Interaction_Instructive Gr=50	0	4	0	0	0	0	0	0	0
○ TC_RP_Instructor_Interaction_Reflective Gr=170	3	0	11	1	0	0	1	0	0
TC_RP_ThirdParty_Interaction_Reflective Gr=3	0	0	0	0	0	0	0	1	0
● SC_SP_RP_Peer_OtherTeam_Reflection Gr=7	0	3	0	0	0	0	0	0	0
● SC_SP_RP_Peer_Team_Reflection Gr=151	17	0	0	0	0	0	0	12	7
● TC_RP_Material_Conversation Gr=34	0	0	0	0	2	1	2	6	2
● TC_RP_Reflection-in-Action Gr=180	17	2	8	1	0	0	1	4	4
● TC_RP_Reflection-on-Action Gr=12	0	0	0	0	2	1	1	8	3

Team A – Practical Learnings Co-occurrence Table at Development

	SC_SS_Creation Gr=17	SC_HS_Research_Reference Gr=7	SC_PP_Problem Gr=1	SC_SS_Correction_Request Gr=1	SC_HS_Scope_Management Gr=11	SC_PP_Scope_Change Gr=1	SC_PP_Scope_Reduction Gr=2	SC_SS_Decision_Making Gr=1	SC_SS_Problem_Solving Gr=2	SC_PP_DV_ThirdParty_Development Gr=12	SC_PP_DV_ThirdParty_Requirements Gr=7
SC_PP_Development Gr=194	5	2	1	1	11	1	2	1	2	12	7
SC_SP_Group_Crit_Instructor Gr=194	3	4	1	0	11	1	2	1	2	0	0
TC_RP_Instructor_Interaction_Guidance Gr=39	0	0	0	0	9	0	0	0	0	0	0
TC_RP_Instructor_Interaction_Instructive Gr=50	0	4	0	0	0	0	0	0	0	0	0
TC_RP_Instructor_Interaction_Reflective Gr=170	3	0	1	0	11	1	0	0	1	0	0
TC_RP_ThirdParty_Interaction_Reflective Gr=3	0	0	0	0	0	0	0	0	0	1	0
SC_SP_RP_Peer_OtherTeam_Reflection Gr=7	0	3	0	0	0	0	0	0	0	0	0
SC_SP_RP_Peer_Team_Reflection Gr=151	17	0	0	1	0	0	0	0	0	12	7
TC_RP_Material_Conversation Gr=34	0	0	1	1	0	0	2	1	2	6	2
TC_RP_Reflection-in-Action Gr=180	17	2	1	0	8	1	0	0	1	4	4
TC_RP_Reflection-on-Action Gr=12	0	0	0	1	0	0	2	1	1	8	3

APPENDIX D – TEAM B Tables of Co-occurrence from ATLAS.ti

Team B – Reflective Practice - Co-occurrence Table at Design

atlas.ti QUALITATIVE DATA ANALYSIS	● TC_RP_Material_Conversation Gr=52	● TC_RP_Reflection-in-Action Gr=489	○ TC_RP_Reflection-on-Action Gr=6	TC_RP_Problematization Gr=6
○ SC_PP_PD_Design Gr=381	0	260	0	4
● SC_SP_Group_Crit_Instructor Gr=109	0	25	0	0
● SC_SP_Interim_Review_Design_Crit Gr=23	0	10	0	0
● SC_SP_Interim_Review_Design_Idea Gr=45	0	7	0	0
○ TC_RP_Instructor_Interaction_Guidance Gr=8	0	0	0	0
○ TC_RP_Instructor_Interaction_Instructive Gr=6	0	0	0	0
○ TC_RP_Instructor_Interaction_Reflective Gr=52	0	4	0	0
○ SC_SP_RP_ThirdParty_Interaction Gr=180	19	128	2	0
○ TC_RP_ThirdParty_Interaction_Instructive Gr=28	1	17	0	0
○ TC_RP_ThirdParty_Interaction_Reflective Gr=150	13	109	0	1
● SC_SP_RP_Group_Peer_Reflection Gr=17	0	6	0	0
● SC_SP_RP_Peer_OtherTeam_ Reflection Gr=97	0	60	0	1
● SC_SP_RP_Peer_Team_Reflection Gr=372	33	276	2	4

Team B – Reflective Practice - Co-occurrence Table at Development

atlas.ti QUALITATIVE DATA ANALYSIS	● TC_RP_Material_Conversation Gr=52	● TC_RP_Reflection-in-Action Gr=489	○ TC_RP_Reflection-on-Action Gr=6	○ TC_RP_Problematization Gr=6
○ SC_PP_Development Gr=258	35	130	6	1
○ SC_PP_Prototype Gr=76	0	17	0	0
● SC_SP_Group_Crit_Instructor Gr=109	0	25	0	0
● SC_SP_Interim_Review_Design_Crit Gr=23	0	10	0	0
○ TC_RP_Instructor_Interaction_Guidance Gr=8	0	0	0	0
○ TC_RP_Instructor_Interaction_Instructive Gr=6	0	0	0	0
○ TC_RP_Instructor_Interaction_Reflective Gr=52	0	4	0	0
○ SC_SP_RP_ThirdParty_Interaction Gr=180	19	128	2	0
○ TC_RP_ThirdParty_Interaction_Instructive Gr=28	1	17	0	0
○ TC_RP_ThirdParty_Interaction_Reflective Gr=150	13	109	0	1
● SC_SP_RP_Group_Peer_Reflection Gr=17	0	6	0	0
● SC_SP_RP_Peer_OtherTeam_ Reflection Gr=97	0	60	0	1
● SC_SP_RP_Peer_Team_Reflection Gr=372	33	276	2	4

Team B – Practical Learnings – Co-occurrence Table at Design

View 1

atlas.ti QUALITATIVE DATA ANALYSIS	SC_PP_PD_Design Gr=381	SC_SP_Group_Crit_Instructor Gr=109	SC_SP_Interim_Review_Design_Crit Gr=23	SC_SP_Interim_Review_Design_Idea Gr=45	SC_SP_RP_Peer_OtherTeam_Reflection Gr=97	SC_SP_RP_Peer_Team_Reflection Gr=372	SC_SP_RP_ThirdParty_Interaction Gr=180	TC_RP_Problematization Gr=6	TC_RP_Reflection-in-Action Gr=489	TC_RP_Reflection-on-Action Gr=6	TC_RP_ThirdParty_Interaction_Instructive Gr=28	TC_RP_ThirdParty_Interaction_Reflective Gr=150
SC_SS_Adaptaability Gr=7	0	0	0	0	0	7	0	0	4	0	0	0
SC_SS_Analytical Gr=5	5	0	0	0	0	5	0	1	5	0	0	0
SC_Critical_Thinking Gr=16	12	0	0	1	3	13	0	2	15	0	0	0
SC_SS_Commitment Gr=1	0	0	0	0	0	1	0	0	0	0	0	0
SC_SS_Communication Gr=1	1	0	0	0	0	1	0	0	0	0	0	0
SC_SS_Creation Gr=72	67	4	0	1	1	66	2	0	70	0	0	1
SC_SS_Interpersonal Gr=1	1	0	0	0	0	1	0	0	0	0	0	0
SC_HS_Conflict_Management Gr=0	0	0	0	0	0	0	0	0	0	0	0	0
SC_SS_Learn_Change_Peer_Experience Gr=1	0	0	0	0	0	1	0	0	1	0	0	0
SC_SS_Learning_ByPractice Gr=4	0	3	0	0	0	1	0	0	0	0	0	0
SC_SS_Learning_Experience Gr=12	6	0	0	2	2	7	1	0	6	0	0	1
SC_HS_Project_Management Gr=76	15	1	0	0	0	65	10	0	40	0	3	7
SC_PP_Documentation Gr=13	0	0	0	0	0	13	0	0	0	0	0	0
SC_HS_Scope_Management Gr=16	10	0	0	0	0	15	0	0	12	0	0	1
SC_PP_Scope_Change Gr=2	0	0	0	0	0	2	0	0	1	0	0	0
SC_PP_Scope_Reduction Gr=0	0	0	0	0	0	0	0	0	0	0	0	0
SC_SS_Problem_Solving Gr=14	7	0	0	0	0	10	1	0	9	1	0	3
SC_SS_Time_Management Gr=7	0	0	0	0	0	4	2	0	6	0	1	2
SC_HS_Research_Reference Gr=64	51	13	0	4	15	30	4	0	16	0	0	3
SC_SS_Tech_UnKwonSkill Gr=9	0	0	0	0	0	9	0	0	9	0	0	0
SC_Technical_Reference Gr=9	4	0	0	1	1	4	2	0	0	0	0	0

Team B – Practical Learnings – Co-occurrence Table at Design View 2

	SC_SS_ Creation Gr=72	SC_HS_Research _Reference Gr=64	SC_Technical _Reference Gr=9	SC_SS_Tech_ UnKwonSkill Gr=9	SC_HS_ Planning Gr=23	SC_SS_Time _Managemen t Gr=7	SC_HS_Projec t_Management Gr=76	SC_HS_Scope _Management Gr=16	SC_PP_Scope _Change Gr=2	SC_PP_DV _Decision_ Making Gr=3	SC_SS_ Problem _Solving Gr=14	SC_SS_Ad aptaability Gr=7	SC_SS_ Analytical Gr=5	SC_Critical _Thinking Gr=16	SC_SS_ Communication Gr=1	SC_SS_ Interpersonal Gr=1	SC_SS_Learn_ Change_Peer_ Experience Gr=1	SC_SS_Learning _ByPractice Gr=4	SC_SS_Learning _Experience Gr=12	SC_SS_Peer _Experience Gr=10
SC_PP_PD_Design Gr=381	67	51	4	0	12	0	15	10	0	0	7	0	5	12	1	1	0	0	6	5
SC_SP_Group_Crit_Instructor Gr=109	4	13	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0	0
SC_SP_Interim_Review_Design_Idea Gr=45	1	4	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	2
SC_SP_Interim_Review_Design_Crit Gr=23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TC_RP_Instructor_Interaction_Guidance Gr=8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TC_RP_Instructor_Interaction_Instructive Gr=6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TC_RP_Instructor_Interaction_Reflective Gr=52	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SC_SP_RP_ThirdParty_Interaction Gr=180	2	4	2	0	0	2	10	0	0	0	1	0	0	0	0	0	0	0	1	0
TC_RP_ThirdParty_Interaction_Instructive Gr=28	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0
TC_RP_ThirdParty_Interaction_Reflective Gr=150	1	3	0	0	0	2	7	1	0	0	3	0	0	0	0	0	0	0	1	0
SC_SP_RP_Peer_OtherTeam_ Reflection Gr=97	1	15	1	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	2	2
SC_SP_RP_Peer_Team_Reflection Gr=372	66	30	4	9	23	4	65	15	2	3	10	7	5	13	1	1	1	1	7	6

Team B – Practical Learnings – Co-occurrence Table at Development

View1

atlas.ti QUALITATIVE DATA ANALYSIS	SC_PP_ Development Gr=258	SC_PP_ Prototype Gr=76	SC_SP_Group_ Crit_Instructor Gr=109	SC_SP_Interim_ Review_Design_Crit Gr=23	SC_SP_RP_Peer_Other Team_ Reflection Gr=97	SC_SP_RP_Peer_ Team_Reflection Gr=372	SC_SP_RP_ThirdP arty_Interaction Gr=180	TC_RP_Proble matization Gr=6	TC_RP_Reflecti on-in-Action Gr=489	TC_RP_Reflection- on-Action Gr=6	TC_RP_ThirdParty_ Interaction_Instructive Gr=28	TC_RP_ThirdParty_ Interaction_Reflective Gr=150
SC_SS_Adaptaability Gr=7	7	0	0	0	0	7	0	0	4	0	0	0
SC_SS_Analytical Gr=5	0	0	0	0	0	5	0	1	5	0	0	0
SC_Critical_Thinking Gr=16	2	0	0	0	3	13	0	2	15	0	0	0
SC_SS_Commitment Gr=1	1	0	0	0	0	1	0	0	0	0	0	0
SC_SS_Communication Gr=1	0	0	0	0	0	1	0	0	0	0	0	0
SC_SS_Creation Gr=72	2	0	4	0	1	66	2	0	70	0	0	1
SC_SS_Interpersonal Gr=1	0	0	0	0	0	1	0	0	0	0	0	0
SC_SS_Learn_Change_Peer_Experience Gr=1	1	0	0	0	0	1	0	0	1	0	0	0
SC_SS_Learning_ByPractice Gr=4	4	3	3	0	0	1	0	0	0	0	0	0
SC_SS_Learning_Experience Gr=12	3	0	0	0	2	7	1	0	6	0	0	1
SC_HS_Project_Management Gr=76	51	3	1	0	0	65	10	0	40	0	3	7
SC_PP_Documentation Gr=13	5	0	0	0	0	15	0	0	12	0	0	1
SC_HS_Scope_Management Gr=16	13	0	0	0	0	13	0	0	0	0	0	0
SC_PP_Scope_Change Gr=2	2	0	0	0	0	2	0	0	1	0	0	0
SC_PP_Scope_Reduction Gr=0	0	0	0	0	0	0	0	0	0	0	0	0
SC_SS_Problem_Solving Gr=14	4	0	0	0	0	10	1	0	9	1	0	3
SC_SS_Time_Management Gr=7	4	0	0	0	0	4	2	0	6	0	1	2
SC_HS_Research_Reference Gr=64	9	7	13	0	15	30	4	0	16	0	0	3
SC_SS_Tech_UnKwonSkill Gr=9	9	0	0	0	0	9	0	0	9	0	0	0
SC_Technical_Reference Gr=9	4	0	0	0	1	4	2	0	0	0	0	0

Team B – Practical Learnings – Co-occurrence Table at Development View 2

atlas.ti QUALITATIVE DATA ANALYSIS	SC_SS_Creation Gr=72	SC_HS_Research_Reference Gr=64	SC_Technical_Reference Gr=9	SC_SS_Tech_UnKwonSkill Gr=9	SC_HS_Planning Gr=23	SC_HS_Project_Management Gr=76	SC_SS_Time_Management Gr=7	SC_HS_Scope_Management Gr=16	SC_PP_DV_Decision_Making Gr=3	SC_SS_Problem_Solving Gr=14	SC_PP_Scope_Change Gr=2	SC_SS_Adaptaability Gr=7	SC_SS_Analytical Gr=5	SC_Critical_Thinking Gr=16	SC_SS_Communication Gr=1	SC_SS_Interpersonal Gr=1	SC_SS_Learn_Change_Peer_Experience Gr=1	SC_SS_Learning_ByPractice Gr=4	SC_SS_Learning_Experience Gr=12	SC_SS_Peer_Experience Gr=10
SC_PP_Development Gr=258	2	9	4	9	11	51	4	5	3	4	2	7	0	2	0	0	1	4	3	3
SC_PP_Prototype Gr=76	0	7	0	0	2	3	0	0	1	0	0	0	0	0	0	0	0	3	0	0
SC_SP_Group_Crit_Instructor Gr=109	4	13	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0
TC_RP_Instructor_Interaction_Guidance Gr=8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TC_RP_Instructor_Interaction_Instructive Gr=6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TC_RP_Instructor_Interaction_Reflective Gr=52	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SC_SP_RP_ThirdParty_Interaction Gr=180	2	4	2	0	0	10	2	0	0	1	0	0	0	0	0	0	0	0	1	0
TC_RP_ThirdParty_Interaction_Instructive Gr=28	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
TC_RP_ThirdParty_Interaction_Reflective Gr=150	1	3	0	0	0	7	2	1	0	3	0	0	0	0	0	0	0	0	1	0
SC_SP_RP_Group_Peer_Reflection Gr=17	1	3	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
SC_SP_RP_Peer_OtherTeam_ Reflection Gr=97	1	15	1	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	2	2
SC_SP_RP_Peer_Team_Reflection Gr=372	66	30	4	9	23	65	4	15	3	10	2	7	5	13	1	1	1	1	7	6

APPENDIX E – STUDENTS´ SELF-REFLECTION TABLE

Self-reflection co-concurrency table of Team A and Team B outcomes

atlas.ti QUALITATIVE DATA ANALYSIS	SC_StudentA1	SC_StudentA2	SC_StudentA3	SC_StudentA4	SC_StudentB1	SC_StudentB3	SC_StudentB4	SC_Studio	SC_MiniChalleng
SC_SS_Analytical	0	1	0	0	0	0	0	0	1
SC_SS_Avaliability	0	0	1	0	0	0	0	1	0
SC_SS_Colaboration	0	3	0	0	0	1	0	1	3
SC_SS_Commitment	1	0	1	0	0	0	1	2	1
SC_SS_Communication	0	2	1	0	2	1	0	3	3
SC_SS_Communication_Problem	0	1	1	0	0	0	0	0	2
SC_SS_Conflict_Management	1	1	0	0	0	0	0	0	2
SC_SS_Design_Learning	1	0	0	0	1	0	0	2	0
SC_SS_Indivitual_Work	1	0	0	0	0	0	0	1	0
SC_SS_Interpersonal	1	1	1	0	0	2	0	2	3
SC_SS_Interpersonal_Problem	1	1	1	0	0	0	0	0	3
SC_SS_Learning_ByPractice	2	0	1	1	2	1	2	8	1
SC_SS_Learning_Experience	3	0	1	0	0	0	0	3	1
SC_SS_Personal_Learning	2	0	1	0	1	1	0	4	1
SC_SS_Planning	0	0	0	0	2	0	0	2	0
SC_SS_Project_Management	0	0	0	0	1	0	0	1	0
SC_SS_Self_Confidence	0	0	1	0	0	0	0	1	0
SC_SS_Teamwork	1	4	0	0	1	0	0	1	5
SC_SS_Time_Management	0	0	0	0	1	0	0	1	0
SS_SS_Usabillity_LearningOn	0	0	0	0	1	0	0	1	0
SC_TS_Game_Develop_1stExper	1	0	0	1	0	0	0	1	1
SC_TS_Programming	0	0	0	1	2	0	0	3	0
SC_TS_Technical_Learning	4	0	1	0	1	1	0	6	1
AC_Challenge	1	0	0	0	0	0	0	1	0
AC_Study_Motivation	1	0	0	1	0	0	0	2	0
SC_Dedication	0	0	1	0	0	0	0	1	0

	SC_StudentA1	SC_StudentA2	SC_StudentA3	SC_StudentA4	SC_StudentB1	SC_StudentB3	SC_StudentB4	SC_Studio	SC_MiniChallenge
SC_SS_Analytical		X							X
SC_SS_Avaliability			X					X	
SC_SS_Commitment	X		X				X	X	X
SC_SS_Communication		X	X		X	X		X	X
SC_SS_Colaboration		X				X		X	X
SC_SS_Communication_Problem		X	X						X
SC_SS_Conflict_Management	X	X							X
SC_SS_Design_Learning	X				X			X	
SC_SS_Interpersonal	X	X	X			X		X	X
SC_SS_Interpersonal_Problem	X	X	X						X
SC_SS_Learning_ByPractice	X		X	X	X	X	X	X	X
SC_SS_Learning_Experience	X		X					X	X
SC_SS_Personal_Learning	X		X		X	X		X	X
SC_SS_Planning					X			X	
SC_SS_Project_Management					X			X	
SC_SS_Self_Confidence			X					X	
SC_SS_Teamwork	X	X			X			X	X
SC_SS_Time_Management					X			X	
SS_SS_Usabillity_LearningOn					X			X	
SC_TS_Game_Develop_1stExper	X			X				X	X
SC_TS_Programming				X	X			X	
SC_TS_Technical_Learning	X		X		X	X		X	X
AC_Challenge	X							X	
AC_Study_Motivation	X			X				X	
SC_Dedication			X					X	