Multifaceted Approach for Teaching Mobile Software Development: Class Experiences with Lectures, Tutorials, and Pair Programming

Mohammed Saad Seyam

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Computer Science

D. Scott McCrickard, Chair
Nicholas F. Polys
Na Meng
Hicham G. Elmongui
Woongsup Kim

April 3, 2017
Blacksburg, VA

Keywords: Pair Programming, Computer Science Education, Mobile Software Development

Multifaceted Approach for Teaching Mobile Software Development:
Class Experiences with Lectures, Tutorials, and Pair Programming

Mohammed Saad Seyam

ABSTRACT

The currently mainstream mobile application development became part of several programming classes, and courses are being developed focused on mobile app development. There are fundamental differences in programming topics for mobile, including the small screen and finger-based interactions, connectivity using different communication channels, and a large number of sensors. Because of these differences, there is a need to explore different approaches to teach the concepts of mobile development. Integrated approaches and collaborative learning are key to handle the multi-platform environment of mobile development and the diversity of its devices.

One practice that has been used in educational contexts for collaborative learning is Pair Programming (PP); an approach that features two developers working on the same development task. Since it became popular in the 1990s, Pair Programming (PP) has been used by developers who worked on desktop and web applications. During the past two decades, PP has been studied in both industrial and classroom settings. Several studies have shown that PP is a pedagogical tool that can help students enhance their productivity and performance. However, PP has not been studied for mobile development courses before. With the challenging nature of mobile development topics, we saw potential benefits for pair-based learning.

To cover the challenges of mobile development, we developed an approach that integrates Lectures, hands-on Tutorials, and in-class Pair Programming (PP) sessions (the LTP approach). Although PP has been studied previously in classroom settings, LTP provides an adapted version that better fits the requirements of mobile application development. Integrating PP as a core element of mobile development classes aims at giving opportunities to students to collaborate, share experiences, and solve problems together. Moreover, providing multiple teaching approaches ensures that students would benefit from the variety of education methods. The LTP approach aims at helping Computer Science (CS) educators to develop curricula and manage classes for teaching mobile app development.

Multifaceted Approach for Teaching Mobile Software Development:
Class Experiences with Lectures, Tutorials, and Pair Programming

Mohammed Saad Seyam

GENERAL AUDIENCE ABSTRACT

Developing software applications (apps) for mobile devices is different than for traditional desktop and web applications. Mobile devices have small screens, short battery life, limited memory and processing power, and users interact directly with the touchscreens using their fingers with no mouse nor keyboard attached. Therefore, software developers need to be aware of such considerations, as well as communication and connectivity issues that are unique to mobile devices.

Since mobile app development is becoming mainstream, many Computer Science (CS) programs are developing courses that are dedicated to teaching mobile software development. Since these courses are pretty new, educators are still exploring ways to help navigate the challenges of developing apps for mobile devices. By investigating several teaching methods, we found out that collaborative learning might be suitable and beneficial in this sort of classes. One specific practice that we wanted to explore was Pair Programming.

Pair Programming (PP) is a development practice where two developers work collaboratively on the same machine to build software apps. It has been used in industry as well as in classrooms, and it has several benefits in enhancing students' productivity and performance. Therefore, our work studies how PP can help students in emerging mobile software development classes.

This research provides a teaching approach that integrates Lectures, hands-on Tutorials, and in-class PP sessions (the LTP approach), which aims at helping CS educators to develop curricula and manage classes for teaching mobile app development. Integrating PP as a core element of mobile development classes aims at giving opportunities to students to collaborate, share experiences, and solve problems together. However, the LTP approach provides an adapted version of PP that better fits the requirements of mobile application development. Moreover, providing multiple teaching approaches ensures that students would benefit from the variety of education methods.

To Rasha,

Salma,

and Malek ..

Acknowledgements

The PhD journey is often known to be lengthy, hard, and stressful. However, the experience gained through the research years and the outcomes of the PhD work make it a worthy trip. I have been fortunate to have people around me who helped making my journey less hard, less stressful, and – in many situations – a fun and enjoyable experience.

I want to thank my dear adviser, Dr. **Scott McCrickard**, for the guidance and support he provided me during my PhD journey. Besides the lengthy research discussions, staying up late working on papers, conference presentations, lectures, and other research activities, there have always been fun moments that made my journey very special. I will never forget the road trips across many states, inviting us to the lake house, canoeing, biking, karaoke, Christmas parties, and the food adventures! His jokes – which took me a while to get! – helped making my journey much less stressful, and his spirit made our lab a fun workspace for everyone.

I also want to thank my **committee members**, whose guidance have helped shaping the direction of my research work. I appreciate their cooperation and dedication to support this work since our first meeting.

I am very thankful to my **lab mates**, **colleagues** in the CS department, and **friends** at Virginia Tech and around Blacksburg for their support through different phases of my work. We shared stories, laughs, celebrations, and many wonderful moments that kept us motivated to get work done. I am also thankful that I had the opportunity to work with wonderful people at Virginia Tech, whose inspiration and encouragement had a great impact on my progress. My extracurricular activities and work with Dean **Karen DePauw** and Dr. **Patty Perillo** have helped me enjoy a unique experience as a graduate student, and I cannot thank them enough for their support.

The support of my friends has always been invaluable, especially the wonderful **Ahmed Alsum** and **Mervat Fahmy**. They have always been there for me, no matter how far we are from each other. I am more than thankful to have friends like them in my life.

Finally, my sincere appreciation to my big family in Egypt, who had to deal with my absence for the past years, and my small family in Blacksburg, who had to be away from home for a long time. Without the support of my parents, brother, sister, wife, and kids, I could never have got this work done. Thank you, my wonderful family.

#Ut_Prosim
(That I may serve)

Table of Contents

Chapter 1

Introduction

1.1 BRIEF BACKGROUND

Developing applications (apps) for mobile devices is different in nature from traditional desktop/web programming. Issues related to screen size, battery life, various sensors, portability, connectivity, and time-to-market make it more challenging for developers to build mobile apps. Since mobile app development is becoming mainstream, Computer Science (CS) educators started incorporating it as part of their programming classes, and some dedicate whole courses to mobile app development. Research suggests that different teaching styles and multiple exposures of different styles to material can aid in the learning process. One area that seemed to be worth investigating to handle the challenges of mobile software development is collaborative learning and programming, which has Pair Programming as one of its main practices.

Pair Programming (PP) is an Agile development practice that features two developers working on the same development task. One developer (usually called the driver) actually writes code, while the other (called the navigator) watches the driver, provides advice and feedback, and seeks to grasp the overall picture of the task under development. These two roles are exchanged at regular intervals, and the two developers together become the owners of the resulting product, and also become knowledgeable about the development topic.

During the past two decades, PP has been studied in both industrial and educational settings. For industrial environments, studies showed that PP enhanced understandability and maintainability of code and design, decreased defect rates, and supported knowledge transfer. For classroom settings, PP has been studied as a pedagogical tool for programming classes, providing ways to help students collaborate on assignments. There are several studies that showed that PP could enhance enjoyment, increase student's confidence level, reduce workload, improve course completion rate, increase homework submission rate, improve exam performance, and facilitate working efficiently on programming assignments.

1.2 RESEARCH DOMAIN

Emerging areas in CS present new challenges, which are tied to the tremendous diversity in available technologies, devices, operating systems, and programming languages. Mobile development is one area that is growing rapidly due to the recent advances in its hardware as well as software apps. With the huge demand on developing for mobile, increases the need for approaches to teach mobile development in schools, universities, and even online platforms. In particular, this research work looks at mobile software development at the undergraduate level in a research university setting. Mobile development is different from traditional development environments in many aspects, like screen size, portability, connectivity, battery life, and other criteria that is discussed in Chapter 2. Because of the multiple devices and technologies that are part of the mobile development environment, a collaborative learning environment seemed to be a reasonable direction for teaching mobile development concepts. Chapter 2 provides details on using PP as a pedagogical tool used in several domains. However, despite the promise of PP as a collaborative learning tool for mobile development classes, our preliminary experiences (presented in Chapter 3) raised some concerns, and provided some questions that our work tried to answer. Some of these questions are:

- What adjustments need to be made to the PP practices to preserve its advantages while addressing mobile development needs?
- Since there is value in traditional educational approaches, how can PP be integrated with such approaches (like lectures and tutorials) for better coverage of mobile development topics?
- How to manage the class during PP sessions by providing guidance while not interrupting students engagement and collaborative work?

This research answered these questions through our LTP approach, which is a PP-focused approach that integrates lectures and tutorials with PP in mobile software development classes for a comprehensive learning experience. The LTP approach, which is described in Chapter 4, is experientially-driven, and the results of our work with students over three semesters are provided in Chapter 5. This work provides an important path toward adapting PP to a new area, and there is much work remains and many directions to be explored, as described in Chapter 6.

1.3 RESEARCH STATEMENT

The currently mainstream mobile application development became part of several programming classes, and courses are being developed focused on mobile app development. Since mobile development is different in nature from traditional programming, there is a need to explore different approaches to teach the concepts of mobile development. To cover the challenges of mobile development, we developed the LTP approach that integrates Lectures, hands-on Tutorials, and Pair Programming (PP) sessions. Although PP has been studied previously as a pedagogical tool, we provide an adapted version that better fits the requirements of mobile application development. Integrating PP as a core element of mobile development classes aims at giving opportunity for students to collaborate, share experiences, and solve problems together. Moreover, providing multiple teaching approaches ensures that students with different learning styles will gain the required learning outcomes of the class. The LTP approach aims at helping Computer Science (CS) educators to develop curricula and manage classes for teaching mobile app development.

1.4 RESEARCH QUESTIONS

The four main questions this research is trying to answer are:

1. How can we enhance students understanding in mobile software development classes?
2. Can collaborative work enhance students' performance in mobile development classes?
3. How can Pair Programming be adapted to fit the requirements of mobile software development?
4. How to integrate Pair Programming with other teaching methods to gain the benefits of the two worlds?

The work presented in this dissertation aims at answering the above questions, and they will be revisited in section 6.2 to provide the answers we found based on our research.

1.5 RESEARCH OBJECTIVES

The main objective of this research is to help educators who want to introduce mobile software development topics to students. We achieve that by presenting the LTP approach that focuses on collaborative in-class work through Pair Programming, together with foundational theories presented in Lectures and coding practices presented in hands-on Tutorials. Other goals of this research are:

1. Studying how collaborative work can help students learn about mobile software development
2. Incorporating interactive practices as core parts of the syllabi, not just as an add-on.
3. Adapting PP to suit the modern challenges of mobile development field
4. Integrating multiple teaching methods into one approach that can be easily implemented or adapted based on the needs of different classes
5. Providing alternative teaching practices that do not rely solely on lecturing
6. Offering a sample syllabus that educators can follow, adapt, and update based on the rapid changes in technologies and the focus of each class

1.6 RESEARCH MILESTONES

This research explores how PP can be applied for developing mobile apps, and what would be the opportunities and challenges associated with students developing mobile apps in pairs. Figure 1.1 represents the five main stages that this research have gone though, which are:

1. Literature review on PP and mobile app development, suggesting how the field of mobile app development can benefit from Agile collaboration practices like PP.

2. Expert review sessions conducted with four experienced developers to understand how PP can be used during mobile app development work.

3. Exploratory in-class PP sessions conducted with students of Spring 2015 mobile app development class, where we collected feedback from students regarding their perception of the traditional PP process and assignments.

4. Developing the LTP approach that integrates Lectures, Tutorials and PP sessions. The original version of the LTP approach was applied to students of the Spring 2016 mobile app development class, where we collected feedback regarding their in-class work.

5. Based on the feedback collected on Spring 2016 class, some final tweaks were performed on specific areas of the LTP approach, and it was applied to students of the Fall 2016 mobile app development class.

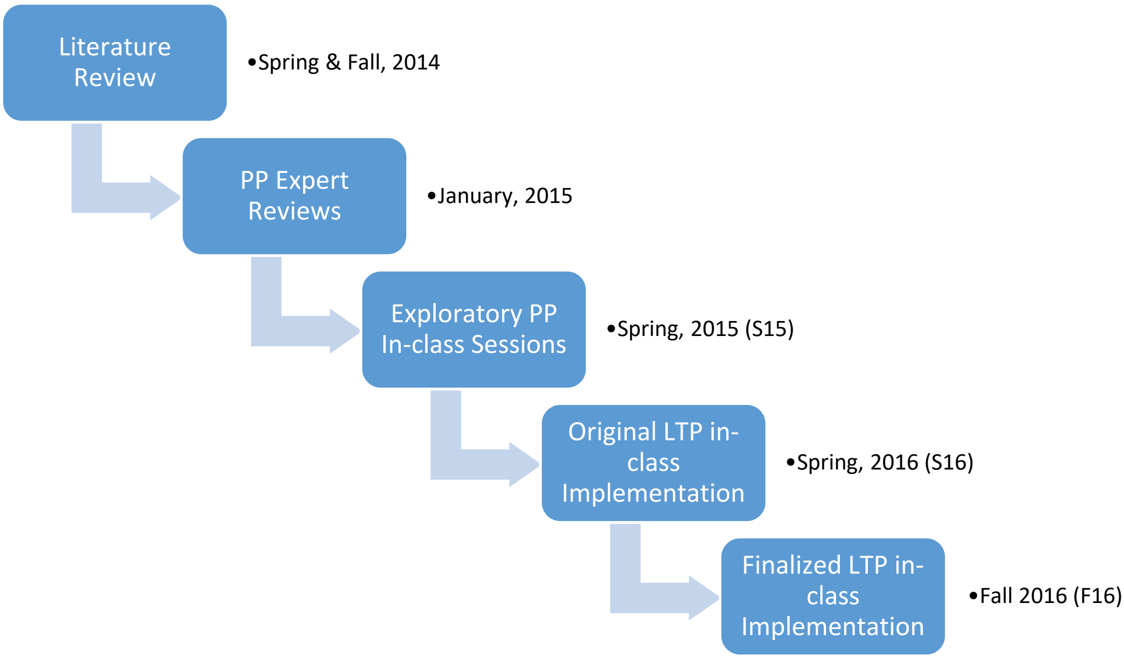

Figure 1.1 Research Milestones

1.7 DISSERTATION OUTLINE

This dissertation consists of six chapters. After this introductory chapter, five chapters are organized as follows:

- Chapter (2) provides some background for the previous work done on using Pair Programming (PP) in educational environments (i.e. programming classes). It also shows challenges associated with developing for mobile, and opportunities to handle these challenges through PP.
- Chapter (3) presents our initial steps where we tried to learn more about how programmers and students would work in pairs on mobile software assignments. This chapter provides details on

the expert reviews and the exploratory PP sessions conducted to help us understand what would and would not work when applying PP in mobile development classes.

- Chapter (4) introduces our multifaceted approach for teaching mobile software development, which integrates Lectures, Tutorial sessions, and Pair Programming (LTP) for a better learning experience. The LTP approach was developed based on our collective experience from literature review, expert review sessions, and several mobile development class experiences.
- Chapter (5) present findings and analyses of the data collected throughout the semesters we have been working with students. This chapter also contains interpretations and explanations for such results throughout the chapter.
- Chapter (6) concludes the dissertation by presenting research summary, answers to the research questions, key lessons learned, and future directions for our research to be enhanced and extended in the upcoming few years.

Chapter 2

Background and Related Work

Pair Programming: from Industry to Education

2.1 INTRODUCTION

Developing applications (apps) for mobile devices is different in nature from traditional desktop/web programming. Issues related to screen size, battery life, various sensors, portability, connectivity, and time-to-market make it more challenging for developers to build mobile apps. Since mobile app development is becoming mainstream, Computer Science (CS) educators started incorporating it as part of their programming classes, and some dedicate whole courses to mobile app development. Research suggests that different teaching styles and multiple exposures of different styles to material can aid in the learning process. In this chapter, we present some background for the work done earlier on using Pair Programming (PP) in CS classes. We also show some challenges associated with developing for mobile, which motivated us to explore new approaches to ensure that students are comprehending and applying mobile development topics properly.

2.2 AGILE SOFTWARE DEVELOPMENT METHODS

Agile methods have appeared to deal with the new problems that began to evolve with the new era of web – and then mobile – applications (Seyam and Galal-Edeen, 2011). Agile methods provides a large set of techniques, tools, and practices develop software quickly and efficiently, valuing working software and face-to-face interactions over plans, processes, and documentation (2001a, Seyam and Galal-Edeen, 2011). Agile methods have attracted a lot of attention; the main reason was that they seek to cut out inefficiency, bureaucracy, and anything that adds no value to a software product. Proponents of agile methodologies often see software specification and documentation as adding no – or minimum – value (Tichy, 2004). Agile methods are less document-oriented, usually emphasizing a smaller amount of documentation for a given task, and strongly advocate for human communication and collaboration over defined and repeatable activities as mechanisms for developing quality software (Lycett et al., 2003).

What is important about agile methodologies is not only the practices they use, but also their recognition of people as the primary drivers of project success, coupled with an intense focus on effectiveness. They stress two concepts: the unforgiving honesty of working code and the effectiveness of people willing to work together (Highsmith and Cockburn, 2001). One of the Agile practices that have been used in both industrial and educational contexts is Pair Programming.

2.3 PAIR PROGRAMMING IN INDUSTRY

Pair Programming (PP) is not a new technique, since it goes back to the mid-1950s as shown in (Williams and Kessler, 2002). However, PP has not gained the IT community attention until its revival by Kent Beck when he introduced his eXtreme Programming (XP) methodology (Beck and Andres, 2004). PP is an Agile development practice that features two developers working “at one machine, with one keyboard and one mouse” (Beck and Andres, 2004). One developer (usually called the driver) actually writes code, while the other (called the navigator) watches the driver, provides advice and feedback, and seeks to grasp the overall picture of the task under development (Williams and Kessler, 2002). These two roles are exchanged at regular intervals, and the two developers together become the owners of the resulting product, and they also become knowledgeable about the development topic.

PP, together with SCRUM daily meetings (Sutherland, 2014), have always been argued to be the most effective practices that support the agile manifesto’s first principle, which values “Individuals and interactions over processes and tools” (2001a). PP is also considered to be a typical application for the sixth principle of the twelve agile principles, which states: “The most efficient and effective method of conveying information to and within a development team is face-to-face conversation” (2001b). Therefore, PP can be considered as lying at the heart of the majority of agile software development methodologies.

2.4 PAIR PROGRAMMING IN EDUCATION

During the past two decades, PP has been studied in both industrial and classroom settings. For industrial environments, (Plonka et al., 2015) presents studies that have shown that PP enhanced understandability and maintainability of code and design (Vanhanen and Korpi, 2007), (Vanhanen and Lassenius, 2007), decreased defect rates (Phaphoom et al., 2011), (Jensen, 2003), (Pandey et al., 2003), (Cockburn and Williams, 2001a), (Phongpaibul and Boehm, 2006) and supported knowledge transfer (Vanhanen and Korpi, 2007), (Pandey et al., 2003), (Luck, 2004), (Katriou and Tolias, 2009), (Sanders, 2002), (VanDeGrift), (Vanhanen and Lassenius, 2005), (Vanhanen et al., 2007), (Williams, 1999).

For classroom settings, PP has been studied as a pedagogical tool for programming classes, providing ways to help students collaborate on assignments. One of the original studies suggested that PP is a promising approach to use as a pedagogical tool due to its capability of increasing learning capacity (Cockburn and Williams, 2001a). According to (Cockburn and Williams, 2001b), some of the foundational studies that followed are those of North Carolina State University (NCSU) (Nagappan et al., 2003a, Nagappan et al., 2003b), (WILLIAMS et al., 2002a, Williams et al., 2002b), University of California Santa Cruz (UCSC) (MCDOWELL et al., 2002), (MCDOWELL et al., 2003), (MCDOWELL et al., 2006), (WERNER et al., 2004), University of Auckland (MENDES et al., 2005), (Mendes et al., 2006), and the Pondicherry Engineering College (KUPPUSWAMI and VIVEKANANDAN, 2004). These studies have shed light on PP as a pedagogical tool that can help students enhance their productivity and the quality of their work (Simon and Hanks, 2007). Several studies have been performed later (Salleh et al., 2011) with evidences suggesting that PP could enhance enjoyment (Simon and Hanks, 2007), (Muller, 2006), (Nawrocki and Wojciechowski, 2001), (Choi, 2004), increase student’s confidence level (Muller, 2006), (Bipp et al., 2008), (Katira et al., 2004), reduce workload (Canfora et al., 2006), improve course completion rate (Muller, 2006), (Shaochun and Rajlich, 2006), increase homework submission rate (Katira et al., 2004), improve exam performance (Phongpaibul and Boehm, 2006), (Nawrocki and Wojciechowski, 2001), (Shaochun and Rajlich, 2006), and facilitate working efficiently on programming assignments (Lan and Ramesh, 2004), (Mujeeb-u-Rehman et al., 2006).

2.5 PAIR PROGRAMMING FOR MOBILE DEVELOPMENT

Since it became popular in 1990s, PP has been used by developers who work on desktop applications. With the rise of web applications, PP joined the wave and became a popular practice for web development. During the past decade, mobile applications became more mainstream, and several approaches and techniques have been developed to serve the goals of mobile application developers. However, there are fundamental differences in programming topics for mobile, including a very different user interface style centered on a small screen and finger-based interactions, a greater emphasis on parallelism and asynchronous operations, and a large number of sensors. However, PP has always been the same, and the practice did not evolve to fit with the different requirements of the mobile era. Computer programming has always been widely considered a highly complex task (Bryant et al., 2008). With the limited resources and many restrictions associated with mobile devices, programming for mobile is even more complex than traditional computer programming. Rhimian and Ramsin in (Rahimian and Ramsin, 2008) show that mobile development has certain issues that differentiate it from regular desktop/web development, which are:

- Wireless communication issues (e.g. availability and disconnection, bandwidth variability, heterogeneous networks, and security risks)
- Mobility issues (e.g. address migration, and management of location-dependent information)
- Portability issues
- Various standards, protocols and network technologies
- Limited capabilities of terminal devices (e.g. low power, data integrity, small-sized user interfaces, and low storage capacities)
- Special privacy and customizability needs
- Strict time-to-market requirements

Researchers have tried to come up with modern teaching approaches that can accommodate the special nature of mobile application development. Tigrek and Obadat proposed a pedagogy for teaching smartphone programming, which incorporated PP as a practice to encourage guided teamwork (Tigrek and Obadat, 2012). However, PP was just a supporting practice and not at the core of this pedagogy, and the research did not provide details on the results of implementing PP. A recent research paper demonstrated teaching programming for smart watches as part of an undergraduate mobile development class (Esakia et al., 2015). Williams et al. (Williams et al., 2008) presented a study of using PP for teaching Human Computer Interaction (HCI) class, but they did not show the impact of applying PP on the user interface design.

2.6 CONCERNS ON PP FOR MOBILE DEVELOPMENT

Based on the unique characteristics of mobile development, we believe that students can benefit from PP when working on mobile development in-class activities, with three concerns:

1. Multi-screen mobile development environment: Mobile design relies on multiple screens to craft a realistic and meaningful design experience. PP should be adapted to address this issue.
2. Connectivity issues: A definitive feature of mobile devices is that they rely on connectivity both with external devices like smart watches and health sensors as well as internal hardware like GPS, accelerometers, light sensors, and cameras. PP should account for these features.
3. User interface (UI) issues: Mobile devices are very different from traditional desktop, and web environments, with challenges and opportunities for PP.

2.7 DISTRIBUTED COGNITION AND PAIR PROGRAMMING

Distributed Cognition is a hybrid approach to studying all aspects of cognition, from a cognitive, social and organizational perspective (Rogers, 1997). The concept was first introduced by Edwin Hutchins to explain how an individual can resolve problems through means beyond his internal cognitive processes (Hutchins, 1995). Distributed cognition observes how problems are resolved through the cognitive system around one or more minds (Lavallée et al., 2013). As a theory of learning, it has been applied in the field of distance learning, and it has been used in the analysis of Computer-Supported Cooperative Work (CSCW) as well as Human Computer Interaction (HCI) analyses (Sharp and Robinson, 2006). For example, distributed cognition has been used in CSCW to study the impact of new technologies on collaborative work such as call centers (Halverson, 2002, Jones and Chisalita, 2005) and communities of practice (Hoadley and Kilner, 2005). In HCI, distributed cognition has been adapted to support the development of interactive systems (Hollan et al., 2000, Wright et al., 2000). However, one major problem with the distributed cognition approach is that it is not a methodology that is easy and ready to be applied to a design problem. Moreover, it is no silver bullet for CSCW or HCI. The reason is that a huge amount of time needs to be spent understanding the concepts and learning to interpret and re-represent data captured in work settings. For example, a tremendous amount of time can be spent analyzing just a short 2-second clip of a video (Rogers, 1997).

One property of distributed cognition is that the knowledge possessed by members of the cognitive system is highly variable and redundant (Rogers, 1997). Moreover, (Pennington et al., 1995) claims that the ‘‘major problem for programmers is to coordinate fundamentally different problem spaces”. Therefore, there have been multiple studies the considered knowledge sharing and communicative practices among teams, and particularly among pairs. One of the most cited work that applied the distributed cognition theory in studying two programmers working together during a maintenance activity is the study of Flor and Hutchins (Flor and Hutchins, 1992). In this study, the authors identified seven properties of the cognitive system that consisted of the two programmers and their immediate environment. These properties were: reuse of system knowledge, sharing of goals and plans, efficient communication, searching through large spaces of alternatives, joint productions of ambiguous plan segments, shared memory for old plans and division of labor (Sharp and Robinson, 2006).

Other studies showed some potential cognitive benefits to PP (Bryant et al., 2008). One benefit is that the presence of a second programmer may help to minimize confirmation bias, which is a phenomenon where an individual would more likely filter information that confirms their current hypothesis, and discarding other potentially useful information that does not (Hutchins, 1995). This benefit has been discussed also in (Williams and Kessler, 2002) when the authors claimed that PP lessens the likelihood of "tunnel vision". Another benefit is the effect of "pair pressure", which is a positive peer pressure that suggests working with a critical colleague may result in positive changes in behavior, which would improve the software quality. The third benefit according to (Bryant et al., 2008) is that working in pairs encourages programmers to talk. Studies suggested that this type of verbalization alone may result in improved understanding and creating a more correct mental model of the problem (Chi et al., 1994). This verbalization has been referred to in (Ainsworth and Th Loizou, 2003) as "cognitive off-load" that frees up working memory. In addition, (Sharp and Robinson, 2006) claims that the ability for pairs to overhear and be overheard appears to support the distributed nature of problem-solving, where relevant expertise is offered from within the pair or from other pairs when it is needed.

According to (Bryant et al., 2008), the methods by which the navigator assists via cognitive off-load are "navigator as reviewer", where the navigator reviews the code pointing out any syntax and spelling errors, and "navigator as foreman", where the navigator thinks about the overall structure of the code and whether it is solving the intended business problem. However, it was found that the driver and navigator roles may not be as distinguished as has been originally suggested. Therefore, the pair works more as a "tag team" (Bryant et al., 2008). Based on that observation, (Bryant et al., 2008) suggested that the driver and navigator model provides a "cognitive tag team", where the effort of keeping the navigator up to speed is "countered with the benefits of having a partner to take over when the driver becomes overloaded".

2.8 CONCLUSION

This chapter provided some background on the challenges of teaching computer programming in general, and provided some extra issues that are related to mobile software development in particular. Using PP has been introduced to help CS educators achieving better results in their classes after it was proven successful in industry. There are several advantages for using PP in class, and there have been several studies that have been done to investigate the benefits of in-class PP. However, not enough work has been done to explore how PP could help with the challenges of mobile software development. Using PP for teaching mobile development has some benefits that have been shown in several distributed cognition studies, and that encouraged us to focus on PP as a practice to implement while working with students in mobile development classes. Knowledge sharing and exchange of ideas are important for software development, and we believe it should be of more importance considering the particular challenges that face mobile development. The next chapter discusses our initial work where we tried to learn more about how programmers and students would work in pairs on a mobile software assignment. Then we will show how we used the experience we gained from the exploratory phase in addition to the collective literature experience provided in this chapter have helped us framing our LTP approach that will be presented in Chapter 4.

Chapter 3

Exploratory Steps towards Developing the LTP Approach

3.1 INTRODUCTION

Since developing mobile apps is different from developing desktop and web apps as discussed in Chapter 2, we wanted to get better understanding of how developers would work in pairs while developing a mobile app. Our goals were to investigate developers' performance and behavior while working in pairs, and to get their feedback on the experience of PP while working on developing a mobile app. We began by collecting professional feedback through expert reviews, and then we got students' feedback after they tried PP for five sessions in the mobile development class. This chapter provides more details on those two stages of the research, which were important for us in understanding what would and would not work when applying PP in mobile development classes. These two steps also helped us building the LTP approach, which will be presented in Chapter 4. Several parts of this chapter appear in (Seyam and McCrickard, 2015) and (Seyam and McCrickard, 2016).

3.2 EXPERT REVIEWS

To better understand how developers interact with each other while working in pairs, and to ensure that we have better understanding for the context of developers in a classroom setting (i.e. students), we conducted expert reviews with two pairs of experienced Android developers. One pair of experts were graduate students, and the others were undergraduates:

1. For graduate students: they were CS PhD students who worked in the field of mobile development for a long time. They have also worked as Graduate Teaching Assistants (GTAs) for the mobile development class. Therefore, they are experienced in programming in general, mobile development, and education. It was important for our research to get this level of experienced developers to practice PP so that they share their thoughts and feedback on how the process was for them, as well as what worked and what did not work when it comes to practicing PP.

2. For undergraduate student: they were senior students in CS department, and both of them worked as Undergraduate Teaching Assistants (UTAs) for mobile development class. Their Android development experience level is lower than the graduate level, but they are experts in dealing with students in programming classes.

Based on that background, we moved forward to conduct two PP sessions to get the participants expert reviews about the process. We observed their interactions throughout the session, and they shared their insights during and after the sessions. We – together with the experts – agreed that the programming assignment that was to be worked on during the sessions should have some major criteria, which led us to come up with the following task requirements (categorized based on the corresponding criteria):

1. To be a familiar task: to implement a calculator that performs the basic operations and runs on android devices.
2. To be flexible: the task was open for whatever assumptions and decisions made by pairs. A calculator can be anything from two simple text boxes that use the devices keyboard to a full-sized screen with all buttons and operations.
3. To have usage context: that assignment definition stated that this calculator will be used by fourth and fifth graders (ages 9-11 years old) to introduce them to calculators and get them to be familiar with them.
4. To have no specific interface requirements: as we wanted to observe how pair would come up with interfaces that fit the required task based solely on their understanding of the usage scenarios.
5. To be time-limited: pairs should deliver their working prototype within one-hour period. We wanted to see how time constraints would affect interactions as well as decisions made by developers.

The experts collaboratively decided that for one-hour session, exchanging roles between driver and navigator would occur every 15 minutes, with the session facilitator working as a timekeeper. After being seated and prepared with the required information and tools, the sessions were ready to start.

3.2.1 Session 1 (Graduate experts)

Developer A was more experienced in Android development than developer B, as he worked more time with it and taught more classes related to it than B. However, both of them were experienced developers and they defined themselves as "we don't know everything about Android programming, but we are confident that we can find solutions for the problems we face even if they are completely new to us". Although they dealt with each other before, it was their first time to work together as partners on a programming assignment.

The first decision made by them was about which Integrated Development Environment (IDE) they should use. Both of them were experienced with Eclipse, while only B was the one familiar with the newer Android Studio platform. From that point, A had the suggestion to go with the "common ground", which the Eclipse, so that they can save time to focus on the application rather than getting to know about the new tool.

It was important to notice that before making this decision, B explained the main benefits of using the newer tool, which were unknown to A. B has even did a short demo showing some of the "nice" features provided by Android Studio. However, both of them were satisfied by working on the IDE that they both knew about. Therefore, their decision of working on the older tool did not prevent A from learning some features about the tool that they decided not to use. This is an important point about PP and collaborative environments, which is that developers do not only learn about what they are using, but also about the options/tools/approaches that they decided not to use.

Once developer A – the first driver – was done with setting up the new project, they stopped to talk about the layout that they should use for their application. After exchanging some verbal ideas, they hold markers and began drawing on a whiteboard right behind them. They approved using the "grid view" element, and they were confident about their ability to try anything even for the first time as long as it will achieve the required results. Therefore, they directly moved forward and began implementing their basic solution idea.

3.2.1.1 Handling Different Opinions

Opposite to their expectations, dealing with grid view was not that easy, and they had to handle some issues related to sizing, positioning, and alignment of cells. Suggestions to solve such problems came from the two developers, as they faced those problems during the first three rounds of the four-round session. Most of the times developer A's suggestions were approved and out to action even if B had some other solutions, but that did not cause any problems as A's suggestions were based on experience and usually provided better solutions. However, two main points affected this type of interaction:

1. Developer A suggested a line of code to be written by B, while B showed him that this might not be the right way to do it. However, A insisted that it was a good way to do it, so B just did as what A proposed and they moved forward. Starting from this point, B's level of interaction got lower and his suggestions became fewer than before.

2. Later when an error was found by the debugger, developer A tried to fix it by editing some parts written by B. However, they discovered that the problem aroused from the previous suggestion of A. It then became clear that B was right on the first place. Starting from that point, B's level of interaction increased as his suggestions, comments, and insights were much more than before. Moreover, A began to consider B's inputs and asked for his approval more carefully than the previous round.

These two points show the importance of self-confidence, mutual respect, and openly sharing thoughts for the success of PP sessions. For this specific session, the two experts had good levels of self-confidence and even higher level of mutual respect for each other. However, A seemed to be more confident about his abilities, which made him – to some extent – disregard B's suggestions. This style of interaction does not lead to a "good" PP session. However, it was smoothly corrected with B's trials to always share his thoughts and with both of them discussing what they are working on. Facing situations like the two shown above could lead PP sessions to success or failure based on how the pair handle it.

3.2.1.2 Discussing Usage Scenarios

The discussion about users and usage scenarios appeared early when the pair worked on the main layout. They began asking and answering questions about what the users (9-11 years old kids) would need to find in such application. Based on their assumptions, they decided to include only the main arithmetic operations (addition, subtraction, multiplication, and division). They excluded fractions, and that is why they did not include the “dot” button in their final version. They planned to work on some graphics to better fit the young users but the time did not allow for this. However, they skipped an important function, which is the ability to delete one or more digits in case of errors. That mistake appeared only on their final test before delivering the product. Although it was easy for them to correct that problem, it still showed that their discussions have missed some major usage scenarios. This shows that PP enhances the developers’ awareness about user experience, but it still needs supporting steps to ensure that developers fully understand how users will interact with their application.

It was clear for the pair on the final round that they will not be able to deliver a fully functional prototype, so they decided to go for implementing only one operation (which was the addition). They skipped some interface requirements in favor of providing a prototype that works even with some errors (exceptions). On that final round, the discussion between them was minimized to save time, and developer B was driving the keyboard, with A’s intervention only to correct mistakes or to suggest ideas that help B goes faster. At the end, they were able to provide a working Android application that can be used to input two numbers and calculate their sum.

3.2.1.3 Experts’ Comments

The first words told by the experts after they were done with the session were “it was fun” and “it was exciting”. It was obvious that they were practicing a game-style pairing where their team was playing against time. Even if they did not provide the complete required product, they were able to deliver a small working version of it. The two experts declared that they enjoyed the time spent during that session. Developer A showed that it would have taken him more time to work on such assignment if he was to do it on his own, because developer B used some coding practices that A would otherwise not use them. Developer B indicated that it would have taken him almost the same time to do the same job, but he believed that the quality of the task coming from the pair was definitely better than of the one he would have worked on by himself. They both agreed that the one who drives the keyboard is usually in a better position to decide than the navigator is. That clearly appeared on the final round where developer B decided to go for a coding practice that A did not prefer (using hard-coded listeners), but A did not stop him because time was running and they wanted to have a working demo. They also felt that sometimes the navigator wanted to get the keyboard to do something that would be faster than leaving it for the driver to do, but they did not because it was not allowed during the PP session.

Developer B showed that he was not able to try some solutions using his own way because developer A was usually deciding on the fly while he was driving the keyboard. This obviously changed after the second round, which is related to what developer A admitted: "B is really a good programmer, I liked working with him. He is really better than what I thought!". This feedback made it clear that even with some pre-assumptions that can exist among developers; PP usually helps to correct some of those false assumptions on the early rounds. The two developers said that they both learned new things from each other. Those new things were more related to coding practices, tips, and habits. They also showed that they enjoyed learning how to deal with new layout structures together, which made their learning time shorter. As for their personal feelings regarding the PP session, they concluded with almost the same sentence: "That was pretty awesome, and I would like to do that more".

3.2.2 Session 2 (Undergraduate Experts)

Both developers C and D had almost the same experience level in programming in general, and in Android programming specifically. They knew each other before this session, but they never worked together on programming assignments. In assessing their experience level, they described themselves as "being able to get the required knowledge to get the job done".

Both of them were familiar with Eclipse, but only C worked with the newer Android Studio. Unlike the pair of the first session, they decided to go for Android Studio after C explained its advantages and how it made some issues easier for her. Starting from this point, almost all decisions have been made the same way: one developer suggests something showing her rationale, the other approves. It was clear that this pair dealt in a peer-to-peer interaction level, rather than the senior-junior interaction model that appeared on the first round of the first session.

3.2.2.1 Discussing Usage Scenarios

Developer C was the first driver as she wanted to introduce the new IDE to developer D. After a very short introduction, they decided to write some basic lines of code. They then stopped after minutes when they realized that they have not agreed on how the interface would look like. After spending three minutes talking about that, they got some markers and began drawing their ideas on a whiteboard. Each of them had her own design idea, with C suggesting a very basic and simple interface, while D suggesting a more attractive interface with more features. Both of them agreed that D's idea was better, but they also agreed that C's proposal would be more feasible because of time constraints. Once agreed on the initial components of the layout, they began implementing the code to make it functional without giving any time to discuss the positioning, look, or any details regarding the UI elements.

3.2.2.2 Interactions between the Pair

The transition from being driver to navigator went smoothly every time, and the two developers easily approved all the implementation decisions. It was noticeable that they did not have to perform any online search for their work. They relied heavily on their previous knowledge and what they already know. That led them to be more conservative in their implementation choices, so that they will not have to face some sudden new situations that would prevent them from being able to deliver a functional prototype on time.

One of the important observations about this session is how the pair were so careful about "getting the assignment done" rather than building a usable application. They were dealing with the task as a class assignment that will be graded, without considering how users will deal with it. The sizes and locations of the text boxes, the alignment and positions of buttons, and all the aspects related to the interface were left to the last round. They wanted to make sure that users could input numbers, click the required operation, and get results to appear on the screen. At the end, they provided a functional product with a poor design that lacks some basic usability requirements.

3.2.2.3 Handling Usability Requirements

Although the two developers claimed that they were affected with the time limits and that they would have consider usability issues if they had more time, it was clear that their interest in user experience was not a priority regardless time constraints. They did not design a complete layout before coding, they did not talk about interface components but in implementation context, they left the layout design for the last round, and they did not consider usage scenarios.

The two experts showed that they did not consider "designing for school kids" or "serving as an introductory calculator" as requirements of the application. Their suggestion was "we can later add some colors and graphics to be more appealing to students". Developer C showed that if there were something that she would change if she was to repeat that PP session, it would have been to "spend more time on design for program structure and interface". Both of them thought they would have provided a better product if they have paid more attention to design.

3.2.2.4 Experts' Comments

When it came to interaction and communication between them, developer C liked that her partner was always talking with her, and that she was not the only one who talked all the time. Being able to talk and listen while coding was an important issue that the two experts emphasized its value for the success of a PP session. They also felt that they were learning together rather than learning from each other. Since both of them were of a "similar academic intelligence level" as stated by developer C, it was easy for them to express their ideas and to be sure that the partner will understand the implementation suggestions and coding practices.

It was important for the pair that their experience levels were close to each other. Developer C talked about her reaction when she deals with someone with more experience, showing that she easily gets intimidated in such situations, and that she gets shy and stressed, which lead her not to gain from working with the experienced developers. On the other hand, developer D had no problems in dealing with experienced developers, but her reaction would be to leave him/her do the required job, trusting that she would only be called if she was needed. For D, the experienced developer will be the leader who is responsible for the hard work, while she will be the assistant who will help only when required.

The two experts agreed that for a simple assignment like that one, PP would not enhance their performance nor quality, while it might does for larger projects with more requirements and sophisticated implementation issues. However, their opinion might have been affected by the fact that they had not dedicated enough time to the requirements that they should have focused on (i.e. design, usability, and usage scenarios). The effect of PP for them was not clear because they jumped directly into coding, which led them to miss the main and important advantages of PP that would have helped them designing a better application.

3.2.3 Collaboration Pitfalls During PP Sessions

Based on our literature review on PP, together with the expert reviews discussed above, we came to highlight some pitfalls that can affect developers' performance and interaction during PP sessions. The following eight issues were ranked (from higher to lower) based on the experts' evaluation for which pitfall would have the most negative effect on PP sessions conducted by students. The first listed issue would cause severe problems, while the last would have minimal negative effects.

1. Developers coming from different backgrounds

When introducing PP for the first time, it is hard to get developers to talk to each other if they do not have a common background to start from and move forward. This, however, can be useful with developers who are experienced and comfortable with PP, where the diverse backgrounds will add to their skills and widen the scope of their discussions about the product under development.

2. Developers with different skill/experience levels

This can work well with pairs who are familiar with PP, so that knowledge transfer can be a major benefit from practicing PP. However, for beginners in PP, it is more important to get familiar with PP through being able to talk with their partners as peers rather than as students or learners. As we showed on the first session, the lack of peer-to-peer interaction had affected the first two PP rounds until that was fixed as they progressed in coding. Having pairs with partners of different experiences levels will be required for those who are already comfortable with PP, as that enhances the learning curve, helps transferring knowledge, and puts collaborative environment into action for the benefit of the whole team.

3. Lack of planning and time management

The two pairs did not work as timekeepers during their sessions, and our reason for that was to allow them to focus more on their work rather than checking the time every now and then. However, they knew that they were only allowed on hour to finish their work. It was noticeable that during the two sessions none of them has mentioned anything about time remaining. They did not have any tentative plan on how work will go on through the 1-hour time slot. That was why the two pairs faced the same problem on the final round, when they had to wrap their work up to be able to deliver a working demo. Therefore, it is important to consider time management between pairs as something that they should consider early on the first PP round.

4. Jumping directly into coding without working on design

The second session showed how the lack of proper software design has affected the developers' ability to deliver a quality product, and it also affected their coding, debugging, and testing experience. Less-experienced developers may oversight some important aspects of software design when they get excited about trying some new approaches (such as PP). That is why it is important to direct pairs on their first PP sessions and guide them throughout the development process to make sure that they maintain the basic guidelines of software quality procedures while working with their partners.

5. Thinking about the assignment as a "task to be graded"

That problem was clear when dealing with undergraduates, who were keen to follow the problem specifications and translate the vague requirements in the safest and simplest possible ways. Since they were not exposed to development environments other than their programming classes and projects, everything for them seemed to be a "graded task" that they should get an "A" in it. Therefore, they ignored any contextual issues related to the assignment, while focusing only on the clear functional requirements stated on the problem definition. If PP is to be applied with undergrads, a more collaborative environment should be encouraged with some other supporting agile practices, which promotes the concepts of "collective ownership of code" and "whole team participation".

6. Considering on-time software delivery over product quality

Although this is the case with most of development teams, it comes into focus with PP teams. One of the major benefits of PP is to ensure software quality because of the instantaneous testing and the ideation that occurs within two minds instead of one. Therefore, if the "quality" is not achieved, PP loses one of its main advantages. The reason for developers – either individuals or in pairs – to sacrifice quality is the limited time. However, pairs in PP sessions should manage to get the best use of available time to produce the required functional product with an acceptable quality level. The problem with the two pairs in our two sessions was with their main goal, which was to "deliver a working piece of software on time”, not to “deliver quality software on time”. More practices should be put to use to ensure that quality is part of the deliverable, not a complementary feature.

7. Disregarding creative ideas in favor of traditional solutions

Trying new coding approaches, working on unfamiliar tools, and implementing uncommon solutions have always been discouraged with the excuse of “time limits”. This fully contradicts with the objectives of PP, which aim at promoting creative solutions and build an innovative environment. Introducing PP to developers should focus on the real objectives behind PP, not just to deal with it as a development technique. The two sessions witnessed some ideas that have been rejected because of the 1-hour time limit, while the objective of PP is to encourage pairs to work on their ideas and try to manage their time to be able to work on their ideas (even by asking for more time if required, as creative solutions are always easy to get approved for more time).

8. Giving less consideration to UI design

The pair on the first session discussed some aspects related to UI design that led them to assume certain usage scenarios and helped them decide on some interface design issues. However, that part was given a very small amount of time when compared to actual implementation time. On the other hand, the second session’s developers did not consider UI design until the very end of the process, and they did not discuss user preferences or any usage scenarios. Although discussion and collaboration between pairs would lead to better design decisions (as shown on the first session), UI design should be assigned more time and should be treated carefully by the pairs.

3.2.4 Preliminary Recommendations for Introducing PP to Mobile Development Students

One of the main advantages of collaboration is to promote the culture of agility, where innovation is considered a core value, rather than being a side gain. Based on our discussion above, we came to decide on certain practices that would help us achieve better results when we introduce PP to mobile development.

1. For the first PP sessions, students are better not to be paired randomly

It will be better to either let students select their partners, or pair them based on their GPA (or their grades on previous programming classes). This, however, should be changed in later sessions so that students can be paired with new partners to enhance knowledge transfer among students.

2. Assign the first PP round to high-level design and session planning

Students should be encouraged to spend reasonable amount of time only for software design, UI high-level design, and planning for their development timeline and milestones. Later, students will naturally start working on design and planning without being “required” to do so, as the advantages of spending some time on those non-coding tasks will positively affects their programming work.

3. Introduce quality as a basic requirement, not as a bonus

When presenting the problem statement, ensure that quality should be considered carefully by developers, and include examples on how users will assess the product quality. Quality attributes should address design, code, and UI.

4. Emphasize the role of talking and listening

PP session facilitator should make sure that pairs are in continuous two-way conversations. Students who are found to be silent for long times should be asked for the reasons and guided to participate in discussions with their partners.

5. Present the assignment as a challenging programming task, not as a regular class assignment

Students get motivated when they feel they are solving a real problem that requires them to be noticeably smart. Working in teams of two makes this feeling even stronger with the higher levels of competition among teams. Therefore, the facilitator should shift students' thinking from "what grade will I get for that task?" to "How good is my task compared to the required level of quality?" This can be done by presenting the culture of agile development, where teams are competing to present the highest quality within the allowed time limits.

6. Students should be asked to explicitly consider user context and usage scenarios

Aside from UI issues, students need to spend enough time discussing who will use their software, how will users deal with it, and what are the non-functional requirements (that might have not been stated on the problem definition). On one hand, it gives them more insights about the application they are working on. On the other hand, that will ensure software quality as it considers the unclear – yet important – non-functional requirements. Moreover, such brainstorming will get students to come up with innovative ideas that might have not appear if they work based on the given assignment definition on its own.

3.3 INTRODUCING PP ASSIGNMENTS TO MOBILE DEVELOPMENT CLASS

After the feedback we got from the expert review sessions, we wanted to explore how students would perceive PP as a component of their mobile software development course. We applied PP as part of in-class activities for one class, and our objectives were to:

1. Introduce students to PP as a grounded well-known agile practice that has been used widely in several environments
2. Present different models for software engineering
3. Encourage students to evaluate, compare, and critique different approaches for software development
4. Assess student understanding for mobile topics

5. Study how PP would help students in particular and developers in general to develop better mobile applications
6. Investigate how applying PP can differ between regular web/desktop application development and mobile application development

To achieve these goals, we facilitated five 75-minute PP sessions for the 53-student class, which covered five different topics related to mobile development:

1. Dealing with interface and data management through fragments
2. Using the camera and processing images
3. Connecting two mobile devices via Bluetooth
4. Using accelerometer and GPS data from mobile device
5. Recording and playing audio via mobile device

The driver and navigator exchanged roles every 15 minutes. The class was managed by a main facilitator for PP and two assistants to answer programming-related questions. Every session started with a 10-minute introduction that worked like a retrospective for previous sessions and preparation for the new one. At the end of every session (except the first introductory one), students were referred to an online questionnaire that they could fill-in right after they deliver their application. It was not mandatory for students to answer every question, which means that questions might have different number of responses within the same session. Below is a summary of what was performed in every session, and some reflections on each session.

3.3.1 Session 1

This was an introductory session featuring a 10-minute overview of PP and a 5-minute presentation about the activity, which was on creating a simple shopping cart app. PP rounds were 15 minutes each, so each student worked twice as a driver and twice as a navigator. Students were allowed to select their partner, select seating location, and create their own UI.

3.3.1.1 Observations

Most students had prior basic knowledge of PP, but with minimum previous practice. They welcomed the idea during the presentation, but seemed uncomfortable during the session. Right after the introduction they immediately started coding, giving no time to consider the application design. Indeed, they did not design a draft interface until the second or third rounds. At this stage, many students struggled to adapt their work to fit the activity's minimal requirements. Therefore, core functionality was present, but with poor attention to UI concerns.

The class instructor noted that the number of questions raised during this session was comparable to other sessions. Students did not pay attention to the time constraints, so the facilitator managed time and ask students to switch roles every 15 minutes.

3.3.1.2 Questionnaires

There was no questionnaire for this session as it was meant only to get students familiar with PP.

3.3.1.3 Findings

1. Students got annoyed when asked to work on only one screen (i.e., no second laptop)
2. Students did not like to switch while in the middle of coding certain code segments, and they ignored the time checks until the driver finishes the part in hand
3. In many cases, the navigator put his/her hands on the keyboard to write a piece of code instead of explaining what he wants to do to the driver. Some students said that it happened subconsciously while others argued that it makes them work faster.
4. Upon delivery, when students were asked about UI/UX factors they kept showing how the application is "working as required for the activity", which was the only success factor for them, rather than usable interfaces.

3.3.2 Session 2

The activity exercised camera and image processing tasks. The session started with instructor comments on student work during the previous PP session, mainly related to the roles of driver and navigator. Students were paired based on a seating location rubric. It was announced that an online questionnaire is available to be filled after finishing the assignment.

3.3.2.1 Observations

Students were surprised by not being able to self-select pairs. However, they accepted it smoothly and started working with their assigned pairs. The activity involved using the mobile camera, so the navigator has always been responsible for holding the mobile and using the camera. Since the application required students to do image processing on selfies, pairs spent time having fun testing their applications by taking selfies. The fun nature of the application was reflected on the coding and class atmosphere. During this session, the number of questions raised by students was noticeably fewer than in session 1. Similar to the first session, little effort was put towards UI.

Navigators were allowed to hold the mobile devices to test code, augmenting to the traditional PP tasks of watching, testing, and collaborating with the driver on coding. This is a special case for mobile application development as rapid parallel testing is done on another screen, rather than the development screen as in web/desktop apps.

3.3.2.2 Questionnaires

Out of the 22 responses to the open ended question: "How would you describe your experience of today's Pair Programming session?" 18 students provided positive feedback about the session, while 4 had minor complaints (classroom environment, problems with the code, not being comfortable with the experience). The positive comments were mainly from students who delivered their code on time. It is important to note that students' satisfaction with the activity correlates to their ability to deliver it on time.

3.3.2.3 Findings

1. Students got more used to switching roles and stick to the assigned turns. However, they needed the facilitator to manage time and remind them when to switch
2. Students managed to give themselves more flexibility in switching time. i.e. to wait until driver finishes what s/he is working on before switching
3. It was still hard to keep pairs working on the same screen, as the navigator sought to use another laptop screen for other supporting functions (online search, displaying class material, looking for previous related assignments)

3.3.3 Session 3

This session involved using Pebble smartwatches to learn about connectivity issues among wearable devices and mobile phones. Students had freedom to choose their partners to incentivize them to work with the same partners who will work with them on that week's homework assignment. Although students seemed, based on our observations, relieved that they would freely choose their partners, the questionnaire responses did not comply with such observations.

3.3.3.1 Observations

Since it was students' first time to program for Pebble, they were not comfortable dealing with it. Connection issues and multiple coding languages had to be addressed, leading to most of them not to be able to deliver a working piece of code, or even to make much progress either in learning or development. They complained about having short time to work on the activity, affecting the atmosphere of the PP session. There were many questions for instructors during that session, and at times partners were not able to help each other.

3.3.3.2 Questionnaires

Work distribution among pairs was not as good as the previous session. The 50-50 percentage appeared in free-partner-selection mode less than with random pairs. One major reason for this is that when students work with "strangers", they tend to adhere to rules, be more competitive, and do what is required from them to do. Out of the 34 answers to the open ended question: "How would you describe your experience of today's Pair Programming session?" 14 were negative, mainly discussing the insufficient time to complete the activity. 12 students of those who provided negative feedback delivered less than 50% of the required functionality. More positive feedback came from students who delivered 50%-75% of the required functionality. The scope of the activity was a major drawback for this session, and PP needed to adapt to deal with such situation.

3.3.3.3 Findings

1. In-class activities should balance the difficulty of the task with session time.
2. Topics related to connectivity demand more time even for simple activities. PP sessions need to be tailored to give students time to learn new topics together.
3. For new topics, students may need extra time and preparation in advance prior to working on PP.

3.3.4 Session 4

This session featured an activity combining GPS and gyroscope. (Note that students had a previous activity on GPS.) Linking the two topics was new for students. Students were assigned to random pairs. To get students to feel more involved with the session, the facilitator displayed anonymous comments from the previous session's questionnaire and discussed them with students during the introduction. That step sought to encourage students to give deeper feedback.

3.3.4.1 Observations

Since students had a previous lab on GPS, they kept referring to prior material and finding complementary resources online. Discussions among pairs seemed richer and questions to the instructors were much fewer. Differences in knowledge levels between pairs led them to have useful discussions and to get advice on how to proceed with coding. Checking previous materials and online searching was time-consuming but helped pairs complete the activity. A majority of the pairs again objected to the "one screen" rule in PP.

Although the activity introduction asked students to consider how users would use the application, only five groups included guidance for users to be able to use their applications. When students delivered their assignments at the end of the session, the facilitator discussed with them how a user will use their application and why they did not pay attention to that issue. As in previous sessions, students were more worried about providing a functioning project than building a "usable" one.

3.3.4.2 Questionnaires

25 out of the 33 recorded responses indicated they were able to deliver more than 75% of the required functionality. This rate caused the level of satisfaction in students' answers to the open ended questions to be higher than the previous session. 23 of the 29 responses provided positive feedback. Many students indicated that they were able to understand how to use GPS through discussions with their partners.

3.3.4.3 Findings

1. Navigators did not spend the whole time just watching and observing drivers; they wanted to be more active by looking things up while the driver is coding

2. Students, again, tried to use another laptop screen for tasks other than coding, not stopping until the facilitator asks them to go back to a single screen
3. PP seems to work better when students have prior knowledge about the activity, as this increases their ability to exchange knowledge and learn from each other
4. In situations where a navigator tried to do something other than watching the driver, s/he becomes fully attendant when the driver is facing a problem. This kind of self-organized pairs was common among teams
5. Feedback provided and answers for open-ended questions were lengthier than the previous sessions, probably because of the comments discussed in the beginning of the session about the previous questionnaire answers

3.3.5 Session 5

The objective of the activity was to familiarize students with audio in/out tools. This last PP session of the semester added the statement "You are required to include any components that help users understand and use your app". Based on previous session results, students were assigned to random pairs. One objective was to study how students would consider UI issues when it was explicitly stated as a requirement. Students were also asked to be responsible for timekeeping.

3.3.5.1 Observations

The number of questions directed to instructors was at its minimum during this session. Students were able to explore how to deal with the activities on their own through discussions and web search. As in previous sessions, many tried using other screens to do the online search for some resources despite being directed not to. About half of the groups missed the first switch and needed to be reminded by the facilitator, but only 3 pairs missed it on the second round. On the third and final switch, all pairs managed to switch without reminders.

The final delivered applications had many signs of considering UI requirements, which was not the case with the previous sessions.

3.3.5.2 Questionnaires

23 out of 29 responses provided positive feedback regarding the session. Some students indicated that PP helped them get the required skills needed to work with the activity with no need to ask for instructor's help. Work distribution among pairs is getting much better as the 50-50 workload appeared more than the previous sessions.

3.3.5.3 Findings

1. UI requirements need to be explicitly requested rather than broadly referenced; otherwise students do not realize the need to dedicate time for it.
2. Students are able to practice PP on their own after training

3. Online resources and course material are important during the session. Planning for PP sessions should incorporate such resources within the development process

3.3.6 Discussion

Questionnaire results showed how PP affects student performance on in-class activities. For example, the percentage of functionality delivered per session changes with the topic being introduced in such session (Figure 3.1). In Figures 3.1 and 3.2, the horizontal axis show the names of the main topics that were covered during the PP sessions. More details on such topics are presented in section 4.2.2.3. The multi-device connectivity activity (which is the Smartwatch module in this case) was the one with the least assignments completed. Therefore, this type of activities should be planned carefully to fit within the allowed time. However, student answers to open-ended questions provided more positive feedback (based on the terms they use to describe the session)—increasingly so as sessions go on (see Figure 2). While this suggests PP was helping students deal with complex problems, it also shows that students work better with PP when they get sufficient practice time.

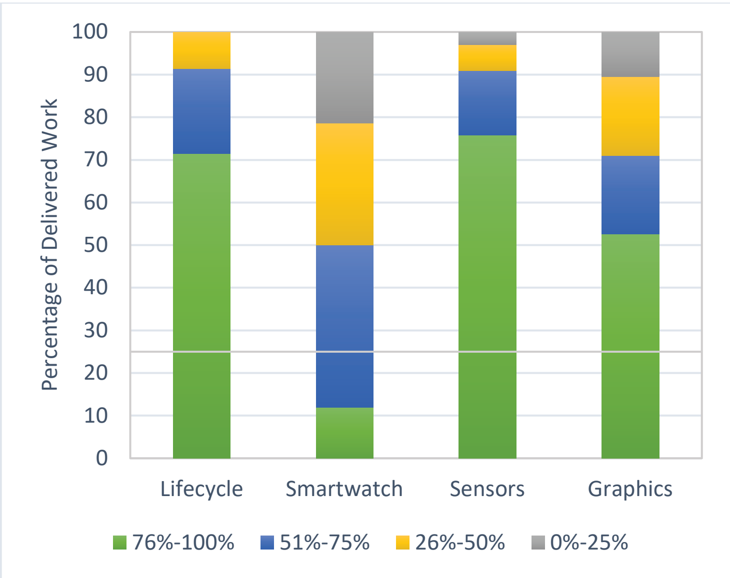

Figure 3.1 Percentage of delivered work in the four PP sessions of Spring 2015

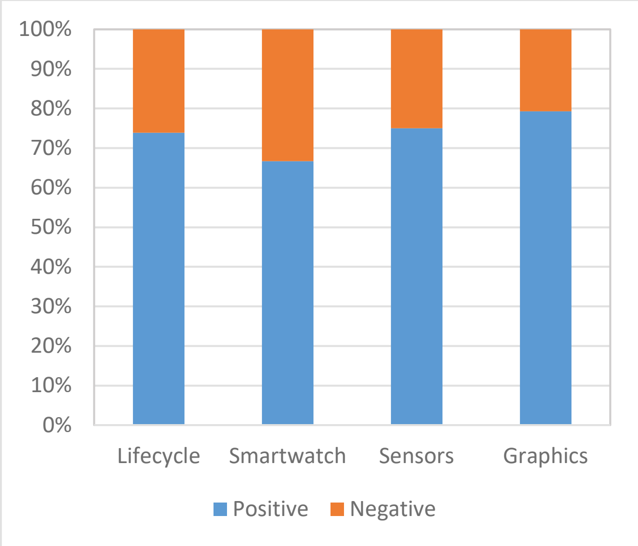

Figure 3.2 Percentage of positive and negative responses in the four PP sessions of Spring 2015

3.4 ASPECTS OF DEVELOPING FOR MOBILE

Programming for mobile devices cannot be seen as merely writing code—they also searched online resources, checked lecture materials, explored connectivity issues with their mobile devices, and learned core functionality of the new devices. PP discourages parallelism, asking students to work on the tasks as a team. However, it was hard to keep the two students (especially the navigator) working on the same aspect of the task (especially for non-programming tasks). PP does not include explicit guidelines for such situations. Further consideration is needed on this issue, perhaps resulting in a PP approach tailored for mobile design.

Along with information from questionnaires, our observations provided different aspects on how students worked with PP. After the five sessions, we discuss in this section what we found relevant and important to consider regarding the differences between developing for mobile and other platforms.

3.4.1 Unique Nature of Mobile Development

During the PP sessions, we allowed a mobile device in the hands of the navigator. We noted that as the driver writes code, the navigator transitioned between the driver's laptop screen and the mobile device. The navigator also looked at the mobile screen to the driver occasionally to discuss some issues. Having two screens seems inevitable when students work on mobile applications, but perhaps the small screen size leads to minimal distraction—allowing the navigator to remain connected to the driver's actions.

3.4.2 Connectivity, Sensors, and Smartwatches

Sessions 3 and 4 required students to program for multiple devices or sensors. Session 3 leveraged Bluetooth to connect mobile device with Pebble watch, while Session 4 leveraged mobile's GPS sensor to control application responses. During these sessions, we observed that interaction between students peaked, although most of them did not deliver the required application on time. Since many mobile applications require multiple-device connectivity (e.g., watches, sensors, wearables), it takes more time and effort of students to be able to learn, understand, and know how to deal with such applications.

3.4.3 Introducing UI Concepts

Students often feel they can start writing code for their assignments without dedicating time for UI issues. However, UI issues are acknowledged as critical for application functionality due to the different factors that affect mobile interfaces (e.g. screen size, orientation, touch and swipe controls). Hence, UI concepts need to be handled carefully in mobile development environments.

During the several PP sessions conducted (both during expert reviews and in-class activities), there have been two approaches to deal with UI/UX concepts:

1. Present only functional requirements of the activities

In multiple cases, students start by writing code segments related to the familiar parts, rather than considering the interface. Only after 2-3 rounds do they find themselves with functional code that lacks a usable interface. At this point, they usually delivered products that lack even the basics of UI design concepts.

This approach is more student-based and it helps when the student is a self-learner. The PP approach helps students to elaborate on how users would deal with their applications, shifting from pair "programming" to pair "design". Moreover, this approach is important for mobile development as there are many UI issues that need to be discussed and evolved by the students' own work—not always the case with traditional programming with its less dynamic components.

2. Specify UI issues as essential requirement

As seen in the fifth session, when usage scenarios were a requirement and part of the activity description, students did dedicate time on their first round to discuss interface issues. When they delivered the application, they were keen to explain its usability, and how users can deal with it in different ways.

This second approach can fit with classes that do not have time to do many PP sessions (when lab time is very limited). It directs students from the beginning to consider UI aspects so that they better plan for how to use their time. However, it will lack the ability to show students through practice the importance of these aspects, especially for mobile development.

Both approaches will educate students on UI concepts during PP sessions. However, the first approach encourages experiential learning if student activities are carefully scoped. Whether an instructor decides to go with the first or second approach, it is important not to disregard UI concepts when working on in-class activities, particularly for mobile platforms. PP can help students discuss and focus on those aspects, especially when they work on applications that require extra attention to how users will interact with such applications.

3.5 RECOMMENDATIONS

Based on our exploratory study for how to introduce mobile development concepts to students using PP, we came to find some practices that can help instructors get the best of PP while achieving their course learning objectives. Below is a list of our recommendations.

1. **Allow students to choose partners on the first session, then move to random pairing**

The first session lets students unfamiliar with PP focus on learning how PP works instead of figuring out how to deal with an unfamiliar work partner. Based on questionnaire answers and our observations, students tend to be more productive when they deal with partner they may not know as well (sessions 2, 4, and 5). Two possible reasons for this can be:

- Students are more relaxed when they work with someone they know (no pressure for competency and excellence)
- Students gain more knowledge from someone whom they did not have interaction with before (as opposed to a friend, whose knowledge about new topics is usually shared among his/her close mates/group).

2. **Allow the navigator to look at the mobile device screen while the driver works on coding**

Although this contradicts a PP core tenet, this practice seems inevitable for mobile development. Navigators usually need to check the mobile device they are working on, test and debug the application on the mobile device, and work on issues like connectivity and sensors. Asking pairs to keep focused on the coding screen was a constant disappointment for many pairs, with claims that it hindered their progress. It is also important to monitor that navigator gives full attention to coding when the driver faces issues.

3. **Decide on the strategy to be used to introduce UI requirements based on the amount of time available for each session**

Sessions 4 and 5 show students giving more attention to UI issues much more than the previous sessions. As shown in section 5.3, the instructor should dedicate the first round to only work on UI issues if students will not practice PP themselves. Otherwise, students may be left to learn the importance of including UI thinking the hard way during the multiple sessions. In either case, PP can address UI issues as students exchange views and create scenarios in pairs on how users would deal with their applications. Moreover, since mobile development requires more attention to details when it comes to UI, PP seems that it can be a good fit—but with careful attention from the instructor.

4. Plan for extra time and pre-class preparation for activities that involve using multiple devices and connections

Most mobile development applications require dealing with multiple platforms and connections, new techniques and technologies, different programming interfaces, and multiple sensors. Instructors need to carefully consider whether students can handle all the required tasks within the class time. Moreover, students may be asked to prepare for the task before they come to the class, so that they become ready to work on the assignment within the allowed session time. The value in PP lies in the ability to discuss and exchange ideas, but the reward lies in completing a task. Students' performance and feedback are affected by the ability to complete a task. Therefore, it is important to size activities to fit within the allowed time so that students can keep working with good spirit.

3.6 CONCLUSION

This chapter showed that although there has been lots of research on applying PP in CS classrooms, mobile development requires further investigation on how PP will perform—and how it might change. Our in-class observations showed that activities in a mobile development course would differ from regular CS programming course activities in several aspects. Issues related to mobile UI, sensors, and multi-device connectivity introduce unique challenges to students in mobile development classes. PP seems to provide a reasonable approach to handle such challenges, though with changes put in place. For PP to perform better in mobile development classes, instructors should consider how PP has traditionally been applied, as well as why some changes may positively influence student learning and performance. The next chapter provides our proposed approach that was built based on the experience we collected through the work presented in this chapter.

Chapter 4

The LTP Approach for Teaching Mobile Software Development

4.1 INTRODUCTION

Research suggests that different teaching styles and multiple exposures of different styles to material can aid in the learning process (Mayer, 2013, Kolb and Kolb, 2005, Johnson et al., 2000). For a challenging new field like mobile software development, multiple teaching approaches are important to help students comprehend the topics that need to be covered. This chapter introduces our multifaceted approach for teaching mobile software development, which integrates Lectures, Tutorial sessions, and Pair Programming (LTP) for a better learning experience. For simplicity, we will refer to the Spring 2014 semester as S14, Spring 2015 as S15, Spring 2016 as S16, and Fall 2016 as F16. Some sections in this chapter appear in (Seyam et al., 2016)

4.2 OVERVIEW OF THE LTP APPROACH

Based on the literature review presented in Chapter 2, and expert review sessions and the Spring 2015 class feedback presented in Chapter 3, we came to realize that there was a need for a teaching approach that aims at helping CS educators who want to teach mobile development to undergraduate students. Basically, we wanted to implement an approach that satisfies certain requirements:

- It considers integrating several teaching approaches for better understanding of mobile development topics
- It has PP as a core component for the benefits of using it in building a collaborative learning environment (as explained in section 2.7)
- It implements PP in a way that fits the mobile development challenges and serves students' needs

Having these guidelines in mind, we developed the LTP approach that integrates PP sessions with lectures, tutorials, and homework assignments. Figure 4.1 shows the main components of the proposed approach.

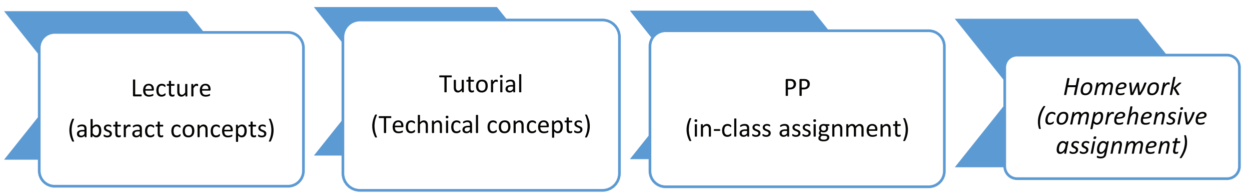

Figure 4.1 Main components of the LTP approach

4.2.1 Different Roles in the LTP Classroom

Before discussing the different components of LTP, Table 4.1 provides a summary of the different roles recommended for a successful implementation of LTP. To ensure flexibility of the approach, these roles do not have to be played by different persons. For example, based on the experience levels of instructor on PP and mobile development, the same person can do the jobs of instructor, facilitator, and LA (Lab Assistant). In some other cases, when the instructor does not have the required technical experience, or the time to work closely with students on programming tasks, then s/he can work only as an instructor. If there is a facilitator that is experienced in mobile development, or a LA who is experienced with PP, so they can be the same person.

Table 4.1 Different roles in the LTP Classroom

Role	Responsibilities
Instructor	Preparing and delivering lectures Covers the abstract concepts Incorporates Agile thinking into lectures Reviews assignments with the other members
Facilitator	Introduces Agile and PP Maintains PP rules during sessions Assigns pairs every session Keeps track of time, switching, and screen usage (pair roles) during sessions Answers questions related to PP implementation Provides feedback and comments at the beginning for each session Help with creating the PP assignments
Lab Assistant (LA)	Creates assignments (with instructor and facilitator) Prepares homework assignments (with instructor and facilitator) Makes sure that students' machines are running and have all the required software installed Answers technical questions during sessions Evaluates students' homework
Student	Installs all required software before coming to class Has the mobile device ready prior to the first session Gets the required device (e.g. smartwatch) ready before the session that needs it Be ready to pair with whoever student assigned by the facilitator Submits code and screenshots by the end of the class Submits homework by the due date

The idea of Table 4.1 is to provide different roles that can be played by different persons based on the resources available (experience levels, number of available personnel, number of students). Table 4.2 provides the different roles and numbers we had in S14, S15, S16, and F16.

Table 4.2 Overview of the roles and numbers of the four semesters studied

Semesters	No. of Students	No. of Instructors	No. of Facilitators	No. of LAs	Notes
S14	63	1	-	2	No PP
S15	54	1	1	2	"traditional" PP
S16	69	1	1	3	Original LTP applied
F16	37	1	1	2	Modified LTP applied Instructor and Facilitator are the same person

There is usually one instructor and one facilitator needed to run the lectures, tutorials, and PP sessions. Based on our experience, it is easy for the instructor to lead the PP sessions as well if they are familiar with basic agile development concepts (as in F16). If this is not the case, there might not be the same person (as in S15 and S16). However, what is more critical is the number of LAs, since they interact directly with students during the tutorial and PP sessions. For tutorial sessions, the number of questions asked and the need for LAs around the classroom is much less than what is needed for PP sessions. Based on our observations of several sessions over the past three semesters, and considering the number of questions asked to LAs in each session (presented in section 5.5.1), our recommendations are:

- For tutorial sessions: two LAs will be needed for classes up to 60 students.
- For PP sessions: one LA is needed for every 20 students.

The numbers provided in those recommendations are rough estimates based on our experience. It is usually better to have more LAs so that students can always find a "free" LA to consult. However, we have found that students tend to keep exploring the problem in hand if they are waiting for LA to help. Therefore, a small waiting time usually leads to students exploring solutions on their own. However, longer waiting times would lead to student disappointment if they cannot figure out how to resolve the problem. The goal of our recommendations is to provide an estimate of the minimum number of LAs needed to run a successful PP session.

4.2.2 Course Contents

This section provides details on the content that was provided to students during the mobile software development classes. Since our work was mainly with Android as the platform for software development, we have to explain how this affected our choices, and what the modules included in our course are.

4.2.2.1 Android development platform

Mobile software development can be taught in two environments: Android and iOS. In the classes we worked with, they were all Android classes. The main reason we chose to work on Android over iOS are:

1. Development tools are available for free
   Students are able to download the official Android Studio (the IDE) for free, and they can start working on it directly after installing it on their machines
2. Android is built on Java
   Since mobile development requires some prior programming knowledge, student can use their background in Java for delving into Android programming without having to revise the basics of Java programming
3. Variety of mobile devices are available for low prices
   In contrary to iOS, which only runs on the $700+ iPhones and iPads, Android devices with minimum requirements needed for the class can be found starting from $40. On one hand, this price is very affordable for students to buy a device if they do not already have one. On the other hand, it makes it easy for instructors to request support from the department to buy number of devices for students who may not be able to afford buying a device. Moreover, Android smartwatches are also much more affordable than Apple watches.
4. Publishing apps is easier and subscription fees are more affordable
   Compared to publishing apps to the Apple App Store, Google Play store offers very affordable fees for the developer account, which enables developers to upload their apps to the store. This is a great incentive for students to build better apps knowing that they can easily have real users downloading their apps.

Although Android is the vehicle we worked on for this research, we believe that our LTP approach can also work with iOS mobile development classes. The major difference will be the contents of lectures, and what topics to be covered in each of the LTP components. However, having the three main components of the LTP approach, and applying our recommendations regarding lectures, tutorials, and PP sessions can be helpful for any mobile development environment since they can be regarded as generic practices for mobile development classes.

4.2.2.2 Computer Science Curricula 2013

To decide on the modules that should be covered in the mobile software development course, we started by reviewing the Computer Science Curricula (Joint Task Force on Computing Curricula and Society, 2013). In this curricula, there was a new area added to the Knowledge Areas (KA) under the name of "Platform-Based Development" (PBD). Mobile development courses (as well as web development) were listed under this KA since it was about environments that are constrained by hardware, APIs, and special services, which are different from the "general purpose" programming. For the mobile platforms section of this KA, the topics that need to be covered are:

- Mobile programming languages
- Challenges with mobility and wireless communication
- Location-aware applications
- Performance / power tradeoffs
- Mobile platform constraints
- Emerging technologies

Together with the listed topics, the learning outcomes of the mobile development courses are:

- Design and implement a mobile application for a given mobile platform.
- Discuss the constraints that mobile platforms put on developers.
- Discuss the performance vs. power tradeoff.
- Compare and contrast mobile programming with general purpose programming.

We followed the recommendations of this ACM/IEEE Computer Science Curricula, and then we started developing our own suggested syllabus, which is presented in the next section.

4.2.2.3 The Six Main Modules

While there is a huge variety of topics that can be covered when teaching mobile software development, the limited time of any course makes it mandatory to carefully select certain topics to be covered during the course period. For our course, we decided to go with the topics that are "unique" for mobile development; meaning that they are radically different from the regular Java programming that can be used for desktop or web applications. After reviewing Android material, provided mainly by Google (Google, 2017), we came to focus on six modules:

1. Graphical User Interface (GUI)
   One of the main criteria that differentiate mobile devices from other development environments is the screen. The size of screen is very small compared to laptop and computer screens. Moreover, touchscreens makes the interaction with fingers radically different from the interaction through a keyboard. Therefore, we decided to start our course with talking about this major difference, and how to handle GUI requirements and elements on mobile devices. Example topics covered in this module are: MVC (Model-View-Controller), GUI Architectures, wireframes, Android UI elements, and Design Best Practices.
2. Activity Lifecycle
   Since activities are the main component of any mobile app, this module focuses on the several stages of activity life cycle, with the goal of getting students to understand how to approach solutions to problems with a mobile developer's perspective. Building apps that require

multiple activities (or fragments) is the norm for Android developers, and this module makes sure that students can do this whenever needed. Example topics covered in this module are: Activity Lifecycle, Intents, Fragments, and handling device orientation and data exchange among screens.

3. Services, Broadcast Receivers, and Content Providers

   After introducing Activities as the main component for Android apps, this module covers the three other components that can help developers control their apps and provide required functionality. Example topics covered in this module are: modes of Services, Service Lifecycle, Broadcast Receivers, and Content Provider Capabilities.

4. Sensors

   Mobile devices currently have many sensors that collects vast amount of data. Managing sensors is a topic that is very different from what traditional programmers had to deal with for desktop apps. Therefore, this module provides examples of sensors, how they can be used, and how the data collected can be integrated and presented in useful apps. Example topics covered in this module are: accelerometer, GPS, Gyroscope, Android Permissions, Coordinate System, and Sensor Best Practices.

5. Smartwatch

   Using Bluetooth connections is a topic that developers did not have to deal with when developing desktop or web apps. Therefore, it was important to provide students with examples on how their apps can run on multiple devices simultaneously. Example topics covered in this module are: Multi-device coordination, Android Wear UI, Connectivity (Bluetooth and WiFi), and Android notifications.

6. Graphics and other topics

   Graphics in this module is about basic 2-D animations and painting. However, this topic also covers other useful mobile resources like the microphone, camera, speakers, and gestures. Example topics covered by this module are: 2D Graphics, Canvas Class, Camera, Animations, Speech Recognition, Recording & Playing Audio, and Touch & Multi-touch Gestures.

The order of modules presented above is the order we believe to work better with students, based on how the modules build on one another, and how advanced students need to be to comprehend a module's topics. We have tried different orders over the three semesters, and we believe that the order presented above worked better with students (in F16) because it took into consideration the comments of students in the previous semesters on how difficult some topics are, and how they felt unprepared for some topics when presented early in the semester. As an example: the Smartwatch module was the third module in S16, and it has been moved to be the fifth module in F16. This change in order was one of the factors that provided better results when comparing students' performance over the two semesters. Details on the results for Smartwatches module is provided in section 5.7.2.

It is important to note that there are other modules that can be added to this syllabus, like Databases and Data Management, Mobile Security, Connectivity and client/server architectures, among others. However, we believe that there are lots of similarities between these modules for mobile and traditional development environments. Therefore, we focused on the topics that are unique to mobile devices, which was a recommendation in the Computer Science Curricula PBD section (Joint Task Force on Computing Curricula and Society, 2013). However, if there is more time allowed, some of these topics can be added to the syllabus and would be of great value.

4.3 COMPONENTS OF THE LTP APPROACH

To cover the contents of each of the modules presented in section 4.2.2.3, the LTP approach provides three main components plus an assessment component to help students understand and comprehend such modules. The following sections provide an overview of the main components of the LTP approach.

4.3.1 Lectures

The lectures are taught by the main instructor, who focuses on the abstract concepts that do not require technical details. Lectures were used to introduce topics and explore underlying theories of development. The lectures included time for questions from and for the students, but otherwise did not have an active learning component. The goal of lectures is to provide the required background of abstract concepts to students before delving into technical details and the actual implementation.

One recommendation that we adapted in F16 was to include introductions to code segments within the lectures. This would help students make the linkage between the lecture concepts and the tutorial code segments. It is important, however, to make sure that the lecture do not go deep into technical details that would get students focus away from the wider scope of the topics covered in the lecture.

Example: while explaining implicit intents in Android, one way to introduce it is to just mention what are they and what is their benefits, which we found to be very high level and not easy to connect with the tutorial code. The other way would be to work on Android Studio on an example that demonstrates what implicit intents are, which would talk long time to develop, and will interrupt the flow of the whole lecture. Our LTP recommendation is to present what implicit intents are, what are their benefits, as well as a code segment that would show how they can be used without actually having to write a whole example to explain it. Figure 4.2 shows one slide as a demonstration of the implicit intents example.

Do not name a specific component, but instead declare a general action to perform

If you want to open a webpage, you can use an implicit intent to request that another capable app (browser) show the desired webpage.

```
Uri webpage = Uri.parse("http://www.Google.com/");
Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);
```

Figure 4.2 Example on integrating code segments in lecture slides

Figure 4.2 shows three parts: the first sentence is what the intent service is. The second part is a usage scenario to get students to understand why and when it is needed. The third part is two lines of code that focus only on the scenario provided in the example. This is an example of what we believe to be a balanced lecture content.

4.3.2 Hands-on Tutorials

Following the lectures, come the hands-on tutorial sessions, which can be led by the instructor or an LA, depending on their experience level of in mobile development. We, however, assign these sessions to the LAs since they get to deal with student questions during the PP sessions, homework grading, and office hours. These sessions aim to cover the basic technical details that allow students to implement the concepts that they learned about in lectures. The LA prepares a tutorial that explains how to turn the lecture concepts into functioning app, as well as including some code segments for the basic topics of each module. During the session, the LA may refer to the tutorial, while demonstrating certain sections of it via live coding session with students. Students typically follow the LA and are encouraged to apply what they see on their machines as the LA is working on the app.

Because we started injecting code segments in the lecture component in F16, LAs did not have to do the introductory linkage with the concepts provided in the lecture during the tutorial session. Therefore, we were able to save more time for students to follow the LA's instructions and ask more questions during the session.

4.3.3 Pair Programming Sessions

After covering the abstract concepts of a module in the lecture, and then demonstrating implementation and technical details in the hands-on tutorial session, comes the time for PP session. The goal of PP session is to ensure that students have comprehended the information provided during the class and the tutorial session, and that they are able to use it on their own. During the PP session, students to work collaboratively (in pairs) on an assignment that requires them to develop a mobile app in the period of a class session. The importance of using PP and our recommendations for applying it are provided in detail in section 4.4.

4.3.4 Homework assignments

To evaluate students' understanding of the topics covered during a module, homework assignments are supposed to include topics from lectures, tutorials, and PP sessions. After the different ways of collecting information in the classroom, students would work individually on the homework assignments.

To provide an example to what can be covered on the three in-class components, Table 4.3 provides our recommendations of the topics that can be covered in Lectures, Tutorials, and PP sessions.

Table 4.3 Topics that can be covered in Lectures, Tutorials, and PP sessions

Module	Lecture	Hands-on Tutorial	PP Session
GUI	MVC GUI Architectures Wireframes Android UI elements Design Best Practices	Android Architecture Android Studio (IDE) Implementing various UI elements Code behind MVC	UI elements (dropdown menus, checkboxes, and toasts)
Lifecycle	Android Lifecycle Intents Four primary application components Activities and Fragments	Code for activity class files Callback methods Implementing intents and fragments Lifecycle events AsyncTask	Intents and Fragments
Connectivity, Services, and Broadcast receivers	Forms of Services Service Lifecycle Broadcast Receivers Content Provider Capabilities Query and Cursor	Implementing Services Bind Activity to the Service Implementing Broadcast Receivers	Services and Broadcast Receivers
Sensors	Different sensors GPS Android Permissions Accelerometer Coordinate System Sensor Best Practices	Location Manager Google Location Services Accelerometer Gyroscope Magnetic Field Sensor	Accelerometer and Gyroscope
Smartwatches	Multi-device coordination Connectivity (Bluetooth, WiFi, USB, NFC)	Android Wear UI Wear Lifecycle Accessing Data Layer Syncing Data Items	Exchange data between devices Syncing Data Items
Graphics & other topics	2D Graphics Canvas Class Camera & Light Sensors Animations Speech Recognition Recording & Playing Audio Touch & Multi-touch Gestures	Real-time Drawing (Canvas) Handling Touch events Detecting Gestures SoundPool Code for Camera	Camera and Gestures

4.4 RECOMMENDATIONS FOR IN-CLASS PP AS PART OF THE LTP APPROACH

As shown in section 2.4, several studies show that PP could enhance enjoyment, increase student's confidence level, reduce workload, improve course completion rate, increase homework submission rate, improve exam performance, and facilitate working efficiently on programming assignments (Salleh et al., 2011). We also showed that PP seems to be a useful approach to teach mobile development, considering the challenges that come with the relatively new industry of mobile applications. However, when we applied PP for in-class assignments in the S15 mobile development class students (as discussed in section 3.3), we came to realize some concerns:

- Students did not know the value of using PP until the final sessions
- Students had to split work between laptop and mobile
- Students wanted to review course material and online resources while keeping the development window open
- Students struggled when the PP assignment was not closely related to information presented in lectures
- Students had concerns on whom to pair with

To address these issues, we proposed some practices to implement PP in mobile development classes, that takes into consideration the special nature of developing for mobile.

1. Dedicate One Session for Introductions

Before asking students to work in pairs for in-class assignments, they need to know why they would do it, how they would, and some background on where is that coming from. Therefore, we recommend dedicating one full session for introductions on agile development and PP. The topics to be covered in this session are:

- Introduction to Agile development methods and practices
- History of PP and its current implementations in industry
- Why we believe PP would help students learning about mobile development

After finishing the introduction, student are asked to pair with the one sitting right next to them, for two 10-minute PP rounds to develop a very simple mobile app (in our S16 and F16 experiences, it was a simple calculator with basic operations). The objective is not to actually develop the app, but rather to make sure that students are familiar with the environment that they will be working on for the rest of the in-class PP sessions. The goals of the two short PP rounds are:

- Physically move to work with a partner
- Decide on who will be the driver and the navigator
- Experience the two different roles at least once
- Switch roles after certain amount of time

- Submit their code via the online submission system

By the end of this session, students should be familiar with the logistics of the class, as well as with the nature of developing in PP environment. Since this session is done very early in the semester, students are typically having no previous experience with mobile development, so they would spend the whole time exploring the development environment (IDE, which is Android Studio in this case). Therefore, they are not expected, nor required, to submit a working app. However, working in pairs for such short time on this session would help all the participants in the following PP sessions to save the time of talking about logistics, and would allow everyone to directly work on the assignment.

2. Allow Students to Use More Than One Screen

The roles of driver and navigator in PP traditionally have been defined to ensure that there will always be one developer who writes code (the driver), while the other developer (navigator), who monitors what the driver writes, corrects the code if there are mistakes, suggests ideas, and help in solving coding problems. From one side, this would enhance code quality. From the other side, it will help in knowledge transfer among developers with different skill levels and backgrounds. However, in the case of mobile app development, and based on our experience with the S15 class, students were not strictly following the PP roles, mainly for two reasons:

1. Because of the inherent need for having a mobile device in hand while testing the app, navigators assumed the role of tester while drivers were monitoring the notifications on the laptop screen while execution. Therefore, having the two students looking at the same screen for the whole time would interrupt their naturally-developed work organization.
2. Mobile development has lots of online resources that help developers find efficient ways to solve coding problems. Moreover, class material provide help for students to guide them through their work on the assignments. Therefore, students work during the sessions is not only about coding, but also searching online resources, looking up lecture material, and reviewing tutorials. Working on those tasks on the same machine is a burden for students, and we experienced in S15 that they used to complain about the waste of time and effort due to their need to switch windows on the same machine.

After studying the above feedback from students, we decided to allow them to use multiple screens. Such screens can be other laptop screen, mobile phones, tablets, and smartwatches. The two restrictions we had were:

1. Only one screen can have the coding IDE
2. The other screen(s) should have nothing but material that can help towards working on the assignment in hand

It is the responsibility of the facilitator (with the help of LAs) to make sure that these two rules are not broken during the session.

The main concern about not strictly following the traditional PP rules is that students may not be working together. However, based on our observations during S16 and F16, we found that:

- The other laptop screen always has either Google search results, Android Studio help pages, or lecture and tutorial materials from the previous sessions
- Navigators used the other screen only when drivers were working on what some of them called "straightforward" programming tasks.
- When the driver has problems with coding, or the navigator has problems with the material, their attention gets directed towards where the problem is. Whether it is the coding screen, the support screen, the mobile device, or even – sometimes – paper sheets, both the driver and navigator provided attention to the device that required their attention.

Therefore, our recommendation to allow students to work on multiple screens has been proven successful, especially when comparing the feedback of S15 (where we did not allow the multi-screen work) to S16 and F16 (where we allowed it). A detailed comparison on feedback is provided in Chapter 5.

3. **Consider The Tutorial Session and The Homework while Preparing for The PP Assignment**

One main objective of the PP session is to work as a bridge between the lecture and tutorial session from one side, and the homework assignment from the other side. The PP session should then cover two main components:

- A topic (or more) from those which have been covered in detail on the tutorial and demo session
- A topic (or more) of those which they will need to explore on their own, and then can help them work on their homework assignment

The reason for this is that, with the many topics that need to be covered in a mobile development course, there will be no way to cover everything during any of the in-class sessions. Therefore, after the lecturer covers the abstract concepts of a certain module, comes the time of the facilitator to work together with LAs to decide what topics will be covered on the tutorial, demo session, PP session, and then the homework. The general guidelines for this are:

- The topics that are common (or similar) between mobile development and traditional development have less priority to be covered on the tutorial session
- For the topics that are new and specific to mobile development, they should be introduced during the tutorial session
- Since the in-class session time is limited, the PP assignment should cover the topics that come on higher priority in the learning outcomes for that module. The PP session is the students'

opportunity to apply what they learned during the previous tutorial session, find solutions to new problems, and gain knowledge for the upcoming homework assignment.

- The homework assignment should include the topics covered during lectures, tutorials, and PP sessions, while being more challenging since its time is not as limited.

One important strength point of PP, which makes it a better fit for the in-class sessions component of the LTP approach, is that it allows students who work in pairs to:

1. Collaboratively review the material provided in the tutorial session, which ensures that they will help each other recall and understand the topics covered in the previous session.
2. Explore the new problems together, and find ways to solve it by collaboratively analyze, think, and search for solutions (as discussed in section 2.7). This ensures that the collective experience they have would help them find better solutions and prepare for the upcoming homework.

4. PP Assignment Presentation Should Include a Rough Draft of the Required App

When presenting the PP assignment to the class, we recommend that a rough draft of the required app to be provided as part of the assignment. (Lavallée et al., 2013) and (DeFranco-Tommarello, 2003) showed that if an initial model of a task is imposed upon the team, it would improve the problem comprehension. Based on our experience with three different classes, providing a rough draft of the app helped students understanding what is required, and got them to start solving the problem without having to lose time due to misrepresentation of the assignment problem.

5. Partners Should Be Assigned by The Facilitator

This recommendation emerged after working with the S15 class. Getting students to pair with different partners every session will ensure that:

- Different partners will provide different experience levels and different knowledge every session
- In some sessions, a student will be paired with a less-experienced partner, while in other sessions the same student would be paired with a more-experienced partner. On one hand, this will enhance the knowledge transfer among class students. On the other hand, it will make students appreciate the value of PP, since they know that for the times they may be disappointed for slowing down because of a less-experienced partner, there will also be other times they will learn from other more-experienced partners.
- Students will not be talking about irrelevant issues that do not relate to the class, which can happen when pairing with friends. This helps students to focus only on working towards the assignment in hand.
- Students will know who they are pairing with once the "pairing rule" is set by the facilitator. This works better than allowing students to choose partners as time is wasted trying to figure

out who to pair with. Moreover, this will prevent some situations from happening, like a student that cannot find a partner, a student that has more than one partner who want to pair with, or students with no one willing to pair with them, which is the case described by (Melnik and Maurer, 2002) as "all good ones are taken" attitude.

The "pairing rule" mentioned above is the way set by facilitator every session for students to pair with. Some examples we tried with the S16 and F16 students are:

- Every student should pair with the one sitting on the right
- Every student should pair with the one sitting in front of them
- Every student should pair with the one sitting on the same position of a different row
- Class is divided into two groups based on their locations, and everyone in the first group shouts out a number sequentially, and then the second group shouts out the same numbers, and then everyone from the first group gets to pair with the student with same number of the second group.

The main idea is that no student would expect who would be his/her partner until they start moving based on the facilitator's "rule".

The feedback we received in S16 and F16 showed that several students have expressed their appreciation to their partners because they learned new things from them. Some students stated that a good outcome from PP sessions was that they got to know new people. There have been complaints about less-experienced partners, which were expected, and have been a regular complain in similar studies about PP (Dybå and Dingsøyr, 2008, Melnik and Maurer, 2002). However, as explained above, these cases are not repeated because the partners change every time, and the number of complains got much lower in S16 and F16 compared to S15. Section 5.4.3 provides more details on the feedback we received from students regarding their work with partners.

4.5 CONCLUSION

This chapter presented the LTP approach for teaching mobile development. The LTP approach was developed based on our collective experience from literature review, expert review sessions, and several mobile development class experiences (S15, S16, and F16). Although there are different components in the presented approach, Pair Programming (PP) is considered the core element of it due to the benefits we see in using it for handling mobile development challenges. Therefore, the chapter described in detail how to apply PP for in-class assignments, and how to make sure that all the other components are well-integrated with the PP sessions. Chapter 5 provides findings and detailed analyses of the students' performance and feedback we collected during the different semesters we worked with mobile development classes.

Chapter 5

LTP in Action:
Findings and Discussion

5.1 INTRODUCTION

Chapter 3 provided our background work to explore how PP should be introduced to deal with the challenges of teaching mobile software development courses. This work led to the LTP approach presented in Chapter 4, which we applied its original version in Spring 2016 and its updated version in Fall 2016. In this chapter, we present findings that we collected through the semesters we have been working with students. We also provide our interpretations and explanations for such results throughout the chapter.

5.2 DATA COLLECTION

To be able to evaluate students' performance and perception regarding using LTP in their mobile software development class, we wanted to collect information in different ways through different stages of the teaching process, balancing personal reflections and observations with performance-based metrics, both in the moment and afterwards. Therefore, we used questionnaires, grades, and observations from students, Lab Assistants (LAs), and facilitators to have a wider coverage of the LTP experience. Figure 5.1 shows the different data collection methods we adopted.

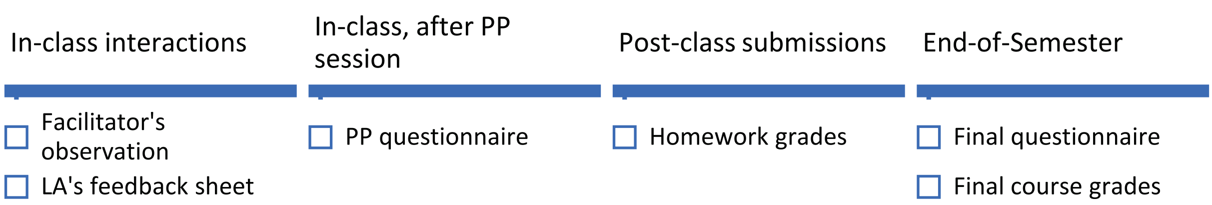

Figure 5.1 Data collection methods

5.2.1 Data Collection Methods

1. **In-class interactions**

a. Facilitator's observations: At the beginning of each PP session, the facilitator was responsible for providing guidelines to successful implementation of the session. During the session, the facilitator should keep track of any incidents that might happen. Examples are: non-talking pairs, non-collaborating partner, and students not working on assignment-related tasks. In addition to the facilitator's intervention in these situations, they took notes on such cases to be used for analysis purposes.

b. LA's feedback sheet: LAs were handed sheets that asked them to record short notes on questions that they get from students. Information like: the nature of question, time spent answering it, and how many questions are answered during PP session. This information was collected and analyzed to evaluate whether the PP assignment fits the time allowed, and whether the level is difficulty is suitable for students. The template of the sheet used during the PP sessions is shown in Appendix D.

2. **In-class post-PP session:** Questionnaires

Before students leave classroom after PP session, they were asked to fill-in an online questionnaire about their PP experience. The objective of this questionnaire was to collect feedback on students' perception of PP, and to note the difference in answers across the different sessions. This was used to help craft the facilitator plan for future sessions and for planning other classes. Appendix B shows questions that have been used throughout the PP sessions we conducted. The shown questionnaire is presented as it appeared on Canvas, which is the course management system we used for collecting questionnaire results.

3. **Post-class submissions:** Homework

After students submit their homework, the LAs graded the assignments and provide feedback to students. This worked as a means to ensure that students gained the required knowledge to cover the learning outcomes of specific module.

4. **End-of-Semester**

a. Final questionnaire: This questionnaire aimed at getting students' feedback regarding the different LTP components, and how they saw the role of PP in helping them understanding specific topics throughout the semester.

b. Final grades: The final course grades were used as a reference for comparing different implementations of LTP with difference classes (S16 and F16). It has also been used to identify patterns across different groups of students.

5.2.2 Available Data

We have data from four classes with different levels of PP and LTP integration, which enabled us to investigate the effect of our LTP approach on students' work in mobile software development classes.

1. **Spring 2014 (S14)**
    - A traditional class where students attended lectures, studied tutorials, and worked on homework assignments.
    - Number of students: 63
    - Type of intervention: none
    - Data available: homework grades and final grades

2. **Spring 2015 (S15)**
   - Students were asked to work in pairs on five in-class assignments.
   - Number of students: 54
   - Type of intervention: five typical “raw” PP sessions
   - Data available: homework grades, final grades, feedback on 4 out of 5 PP sessions
3. **Spring 2016 (S16)**
   - An initial version of LTP was developed and integrated with the syllabus of this course based on prior feedback.
   - Number of students: 69
   - Type of intervention: integration between lectures, tutorials, and PP assignments, and 7 “modified” PP sessions (1 introductory + 6 full sessions for the 6 modules)
   - Data available: homework grades, final grades, feedback on the 7 PP sessions, end-of-semester questionnaires, and LA sheets.
4. **Fall 2016 (F16)**
   - A full updated version of LTP was applied to students of this class.
   - Number of students: 37
   - Type of intervention: full framework implementation, with 7 “modified” PP sessions (1 introductory + 6 sessions for the 6 modules)
   - Data available: homework grades, final grades, feedback on the 7 PP sessions, end-of-semester questionnaires, and LA sheets.

To understand how our approach affected students’ work, we categorized the data we have into three main categories:

1. Students understanding and performance
   - Analyzed through homework and final grades, and the performance in the PP sessions
2. Students perception and feedback regarding the PP experience
   - Analyzed through questionnaires (post-submission and end-of-semester questionnaires)
3. Observations
   - LA sheets provide insights of how students worked during the PP sessions

Throughout this chapter, facilitator observations will be used to provide context for the data and explanation of the results.

5.3 STUDENTS UNDERSTANDING AND PERFORMANCE

In this section, we study students’ final grades for the data that was collected. The goal is to explore how students’ overall understanding of the mobile development concepts has improved after applying the LTP approach. We then show detailed grades for the homework assignments that students worked on over the four semester. This will help us get in-depth information on how students performed in different modules, which will provide more insights on how the LTP approach affected students' understanding of certain topics. Finally, we present the amount of delivered work by students during the PP sessions, so that we can compare the performance over different semesters where different levels of intervention have been applied.

5.3.1 Final Grades

Figure 5.2 shows that the mean final grades are collectively getting higher on the semesters that followed S14. To better understand the diagram, we performed statistical analyses on the final grade data from each semester, including ANOVA analysis for group comparisons. For all of the tests we performed, we used the threshold of p=0.05.

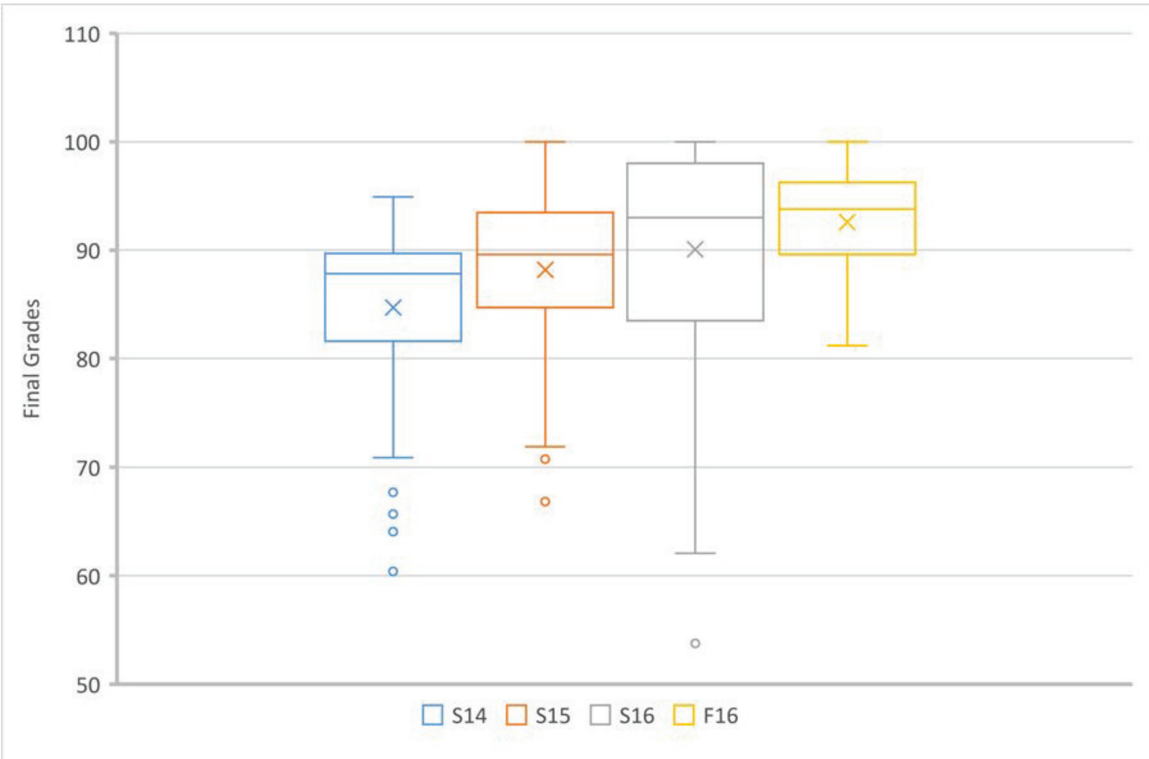

Figure 5.2 Analysis of the final grades for the four semesters

5.3.1.1 Findings

Table 5.1 shows the results of the statistical analysis we performed on the final grades of students over the four semesters: S14, S15, S16, and F16. To understand the differences among the four semesters, we performed ANOVA test to check whether there is significant difference between S14 grades where our approach was not applied (on one side) and S16 and F16 where we applied the original and updated versions of our approach (on the other side). For all classes, ANOVA notes a significant difference in the grades [$F(3,219)=8.88$, $p<0.0001$].

Table 5.1 Results of the statistical analysis of the final grades for the four semesters

Semester	N	Mean	Variance	Standard Deviation
S14	63	84.71	60.4	7.77
S15	54	88.18	56.9	7.54
S16	69	90.06	95.05	9.74
F16	37	92.59	19.58	4.42

To further understand the difference between pairs of classes, we ran t-test for each pair. The results of the t- test show that:

- There is a statistically significant difference between S14 ($M=84.71$, $SD=7.77$) and S15 ($M=88.18$, $SD=7.54$) [$t(115)=2.43$, $p=0.0164$]
- There is a statistically significant difference between S14 ($M=84.71$, $SD=7.77$) and S16 ($M=90.06$, $SD=9.74$) [$t(130)=3.46$, $p=0.0007$]
- There is a statistically significant difference between S14 ($M=84.71$, $SD=7.77$) and F16 ($M=92.59$, $SD=4.42$) [$t(98)=5.46$, $p<0.0001$]

5.3.1.2 Discussion

The statistically significant difference we found when analyzing student grades over the four semesters shows that using PP and LTP approach correlates with a better understanding of the mobile software development topics. Although using grades as a sole measure might provide misleading results, it is not the case with our results. The reason for this is that the graph in figure 5.2 shows consistent improvement over the different semesters. Therefore, we are confident that our approach was a contributing cause for such improvement in students understanding of the course topics.

However, although Figure 5.2 shows that the grades were consistently improving over the four semesters, the difference between S16 ($M=90.06$, $SD=9.74$) and F16 ($M=92.59$, $SD=4.42$) is not statistically significant [$t(104)=1.49$, $p=0.1374$]. This is normal according to how we worked with the students over these two semesters. In S16, we provided our initial version of the approach, which integrates lectures, tutorials, and PP sessions. Based on the feedback we got from the S16 students, we tweaked some components (as discussed in Chapter 4) to achieve better results with the F16 students. Since there were no radical differences between what we did in S16 and F16, we did not expect to find significant difference between the overall student grades, but we achieved better in F16 when it comes to certain homework assignments, students perceptions, and other factors presented in the following sections.

5.3.2 Homework Grades

We have access to the homework grades of S14, S15, S16, and F16. However, not all homework assignments cover every module, and some homework assignments that cover similar modules do not cover the same topics within such modules. Table 5.2 shows the topics covered by homework assignments over the four semesters.

Table 5.2 Topics covered by homework assignments for each semester

Semester	Homework assignments
S14	GUI, Lifecycle (2), Graphics, Smartwatches, and Databases
S15	GUI, Graphic, Smartwatches, and Sensors
S16	GUI, Lifecycle, Smartwatches, Sensors, and Graphics
F16	GUI, Lifecycle, Sensors & Services, Smartwatches, and Graphics

5.3.2.1 Findings

We do not have enough details regarding specifics of S14 homework assignments. Therefore, our comparison will focus on S15, S16, and F16. The common modules that had their own homework assignments for those semesters are: GUI, Graphics, Smartwatches, and Sensors.

For the LTP to be successful, we identified two criteria:

1. To see consistent improvement in the overall students' performance in their homework assignments (by comparing grades of S15, S16, and F16)
2. To have over 50% of students getting over 80% of the homework grade.

Figures 5.3, 5.4, 5.5, and 5.6 demonstrate how students performed in the four homework assignments that were common among the three different semesters. We applied ANOVA test on each module across semesters, and the results were:

- For the GUI module: there is a statistically significant difference [$F(2,155)=7.56$, $p=0.0007$]
- For the Graphics module: there is a statistically significant difference [$F(2,159)=8.78$, $p=0.0002$]
- For the Smartwatch module: there is no significant difference [$F(2,155)=0.38$, $p=0.6844$]
- For the Sensors module: there is no significant difference [$F(2,153)=2.6$, $p=0.0775$]

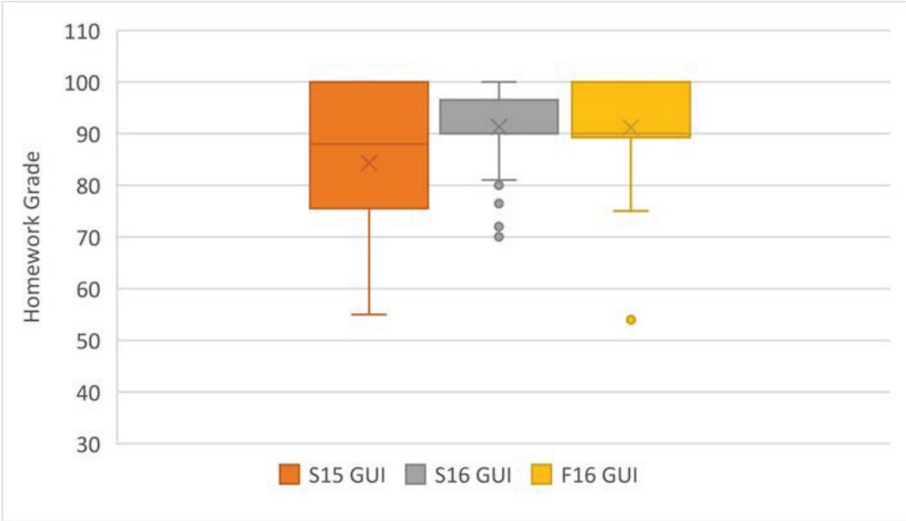

Figure 5.3 Homework grades of the GUI module

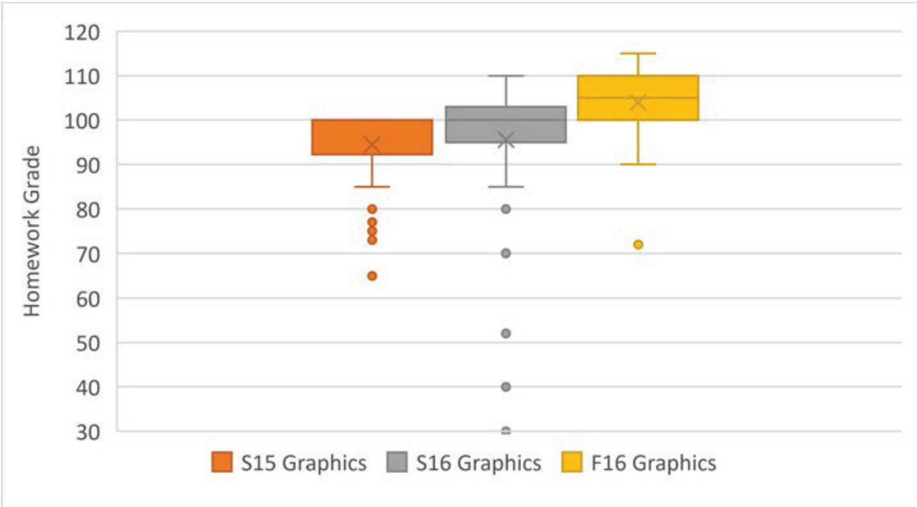

Figure 5.4 Homework grades of the Graphics module

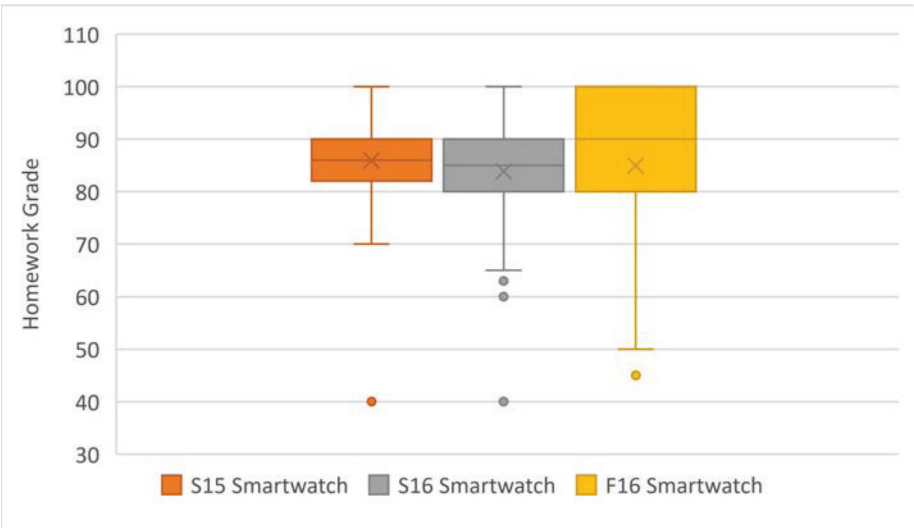

Figure 5.5 Homework grades of the Smartwatch module

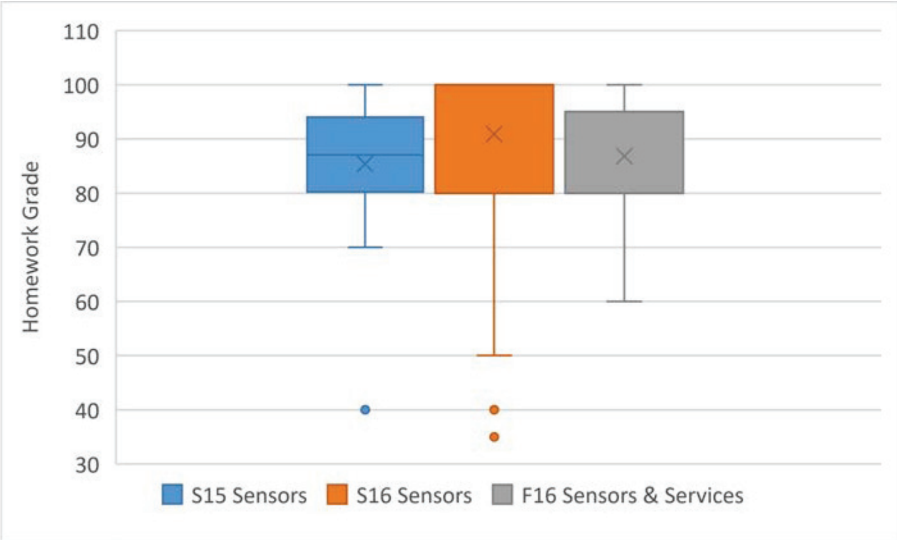

Figure 5.6 Homework grades of the Sensors module

5.3.1.1 Discussion

There are two points that should be considered while examining the above results:

1. The comparison starts from S15 where we first introduced PP for in-class assignments, continues through S16 where we implemented our original version of LTP, and ends with F16 where we applied the modified version. Because of our intervention that has been gradually integrated with the syllabus, our goal was to see a consistent improvement in students' performance in homework assignments.
2. In F16, and due to time limits, we combined sensors and services homework assignments into one homework that covers both topics. Our expectations were that the performance in F16 homework might be negatively affected due to:
    a. A time gap between students working on the services module in class and the time they worked on the homework.
    b. Covering the two topics in one homework assignment led students to face several programming issues because of dealing with various coding requirements.

    Although there is no statistically significant different between S16 ($M=90.86$, $SD=15.11$) and F16 ($M=86.76$, $SD=12.52$) [$t(102)=1.415$, $p=0.1601$], one of our recommendations is not to accumulate and combine homework of different assignments into on homework because it will not achieve the best results required by such modules.

Finally, figure 5.7 shows the grades of the Lifecycle module homework, which was assigned to students of S16 and F16. The results of t-test analysis show that there is no significant difference between S16 ($M=85.58$, $SD=21.04$) and F16 ($M=89.71$, $SD=14.56$) [$t(98)=1.063$, $p=0.2904$]. Since these two semesters are the ones where we applied the LTP approach, it was expected that the difference might not be significant for some modules, which is the case with this one.

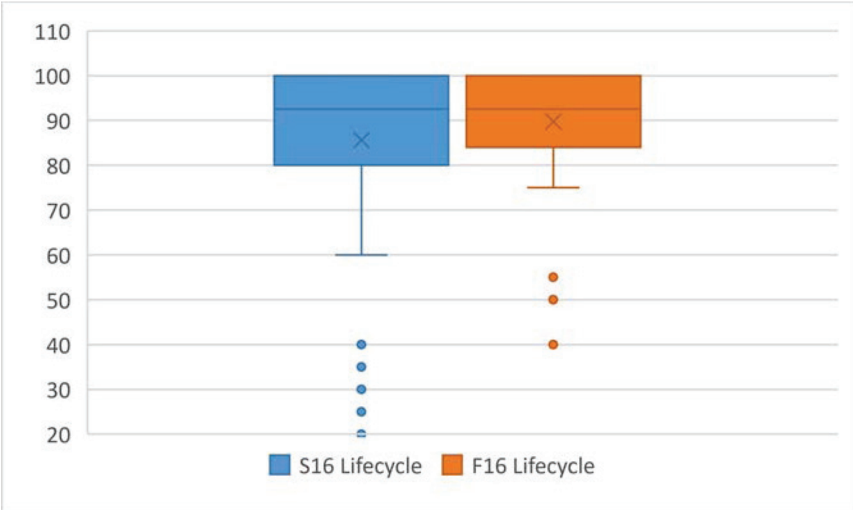

Figure 5.7 Homework grades of the Lifecycle module

The above discussion shows that:

1. Students' homework grades improved on some modules from S15 until F16.
2. More than 60% of student received grades that are over 80%

Since these were our goals to evaluate the effectiveness of our approach, the figures and the discussion followed them shoed that the LTP approach helped students work better in their homework assignments, which are milestones across the semester to evaluate students' understanding to the different module topics.

5.3.2.2 The highest and lowest grades for topics per semester

To conclude our analysis of the homework grades, figures 5.8, 5.9, and 510 show the overall homework grades for different modules per semester. We performed ANOVA test on each semester across modules to get a better understanding of how modules differ, and the results were:

- For S15: there is a statistically significant difference [$F(3,207)=9.11$, $p=0.000011$]
- For S16: there is a statistically significant difference [$F(4,328)=7.09$, $p=0.000017$]
- For F16: there is a statistically significant difference [$F(4,185)=12.84$, $p<0.00001$]

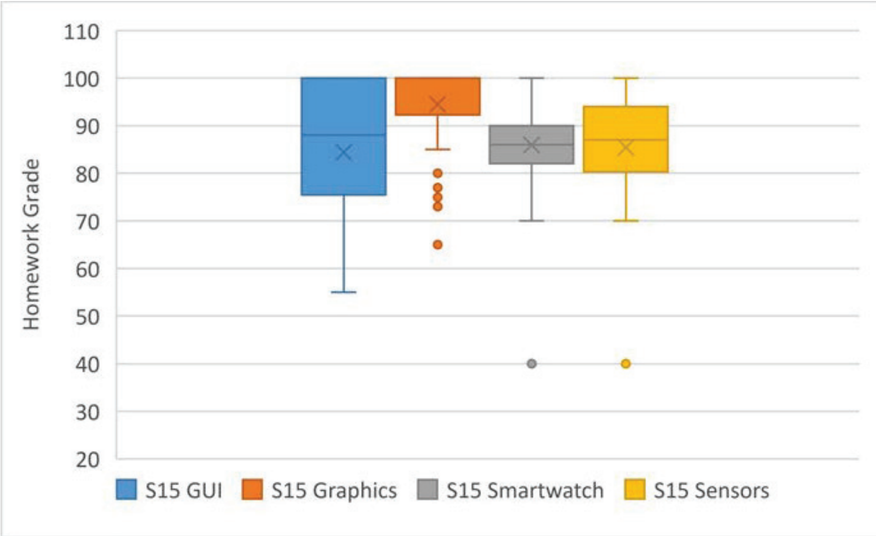

Figure 5.8 Homework grades of S15

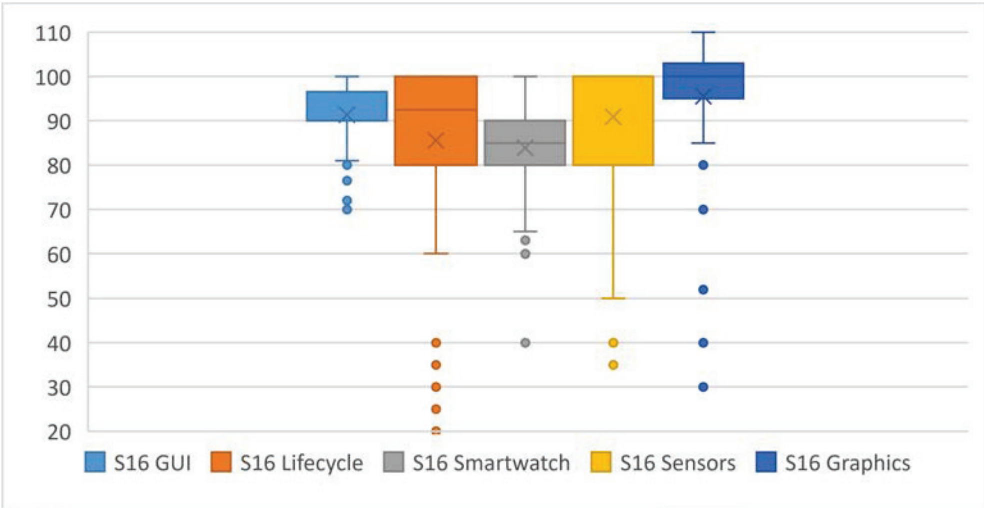

Figure 5.9 Homework grades of S16

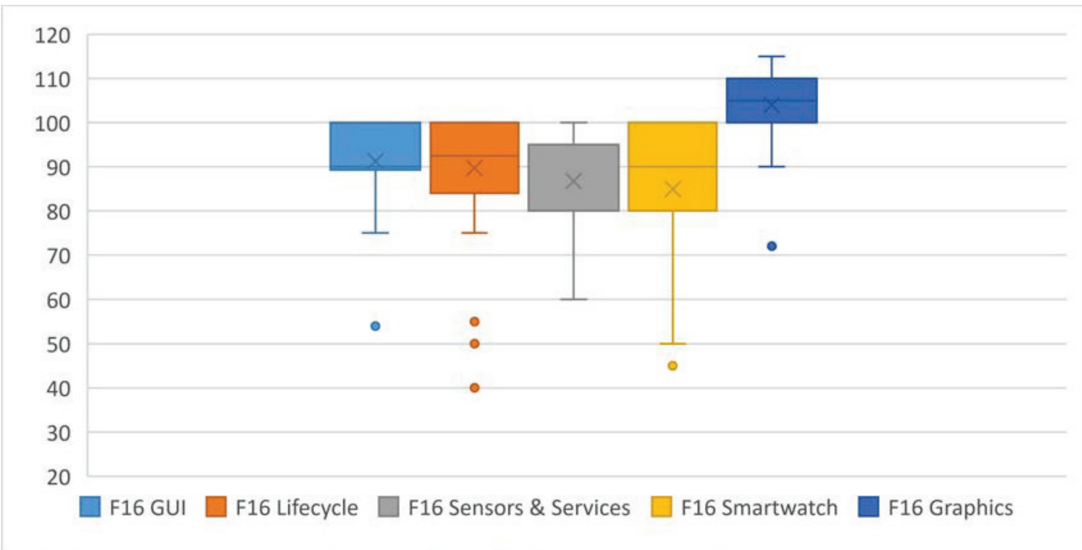

Figure 5.10 Homework grades of F16

The above figures show that two modules had students consistently scoring better in their corresponding homework assignments: GUI and Graphics. On the other hand, Smartwatch module usually had the lowest mean values. A detailed discussion on the Smartwatch module is presented in section 5.7.2.

5.3.3 Delivered In-class Work

On each PP session, students were asked to work on a short assignment that demonstrates some core topics of the module under considerations. Because the time is usually limited (75-minute class), it was understood – and accepted – that students might not be able to submit a fully functional app. They were allowed to work on their apps after the class, but they were required to submit whatever they were able to develop during the 75-minute session before leaving the class. The goal was for us to understand how students performed and whether our assignment helped students toward better understanding of the module's topics. Students we also asked to submit what they think to be the percentage of delivered work, which means the percentage of functionality covered by their submitted app versus what they would deliver if they had enough time. We then checked their responses and compare them to the actual submitted work. It was very rare to find cases where student claim they submitted more than what they actually did (2-3 cases per semester).

Our initial goals while crafting the in-class PP assignments where:

1. To have 50% of students (or more) submitting 50% (or more) of the required functionality
2. Making sure that no more than 25% of students would submit less than 25% of functionality, and trying to lower this number as possible.

5.3.3.1 Findings

Figure 5.11, 5.12, and 5.13 show the percentages of work delivered by students on the three semesters. To explain what we mean by this, we provide an example: in figure 5.11, 71% of students (the bottom green section of the first bar) submitted code that satisfied more than 75% of the requirements of the Lifecycle assignment during S15. 20% of students (the middle blue section of the first bar) submitted code that satisfies 51%-75% of the required functionality. 9% of students (the upper yellow section of the first bar) submitted code that satisfies 26%-50% of the functionality. Since there is no grey section on the first bar, it means that no students submitted code that only satisfies less than 25% of the required functionality. The second bar (S15 Smartwatch), however, shows that 21% of students (the grey section in the top of the bar) submitted code that satisfied less than 25% of the functionality.

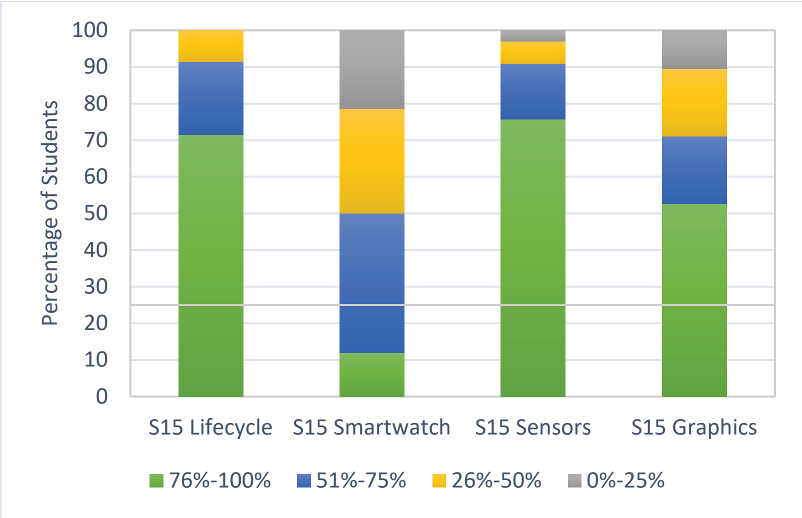

Figure 5.11 Percentage of delivered in-class work in S15

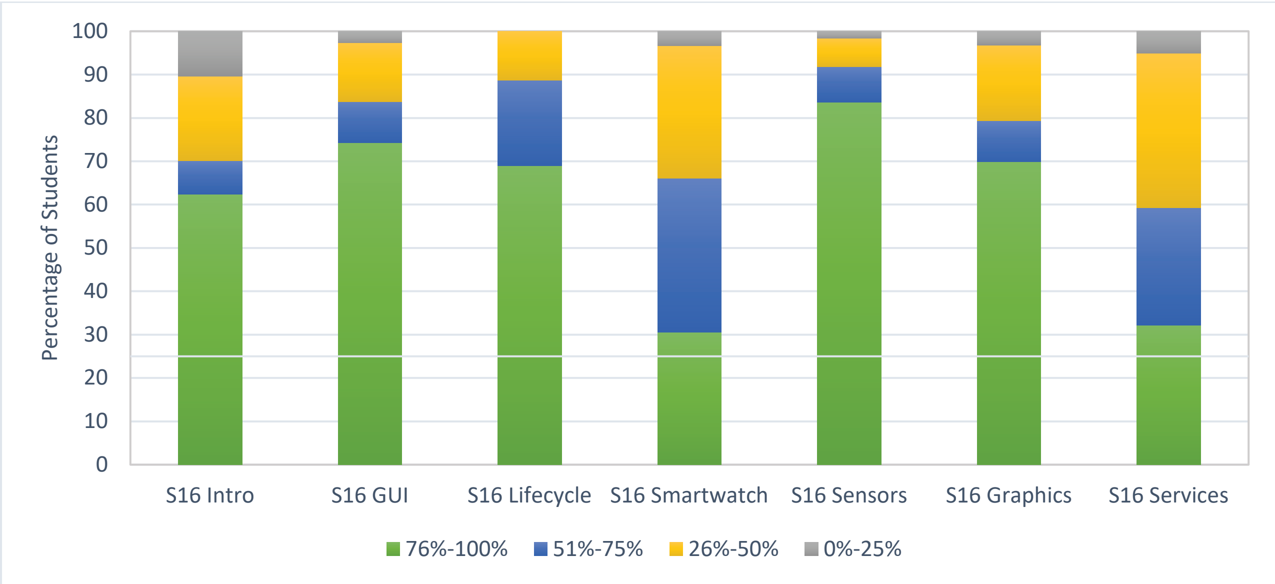

Figure 5.12 Percentage of delivered in-class work in S16

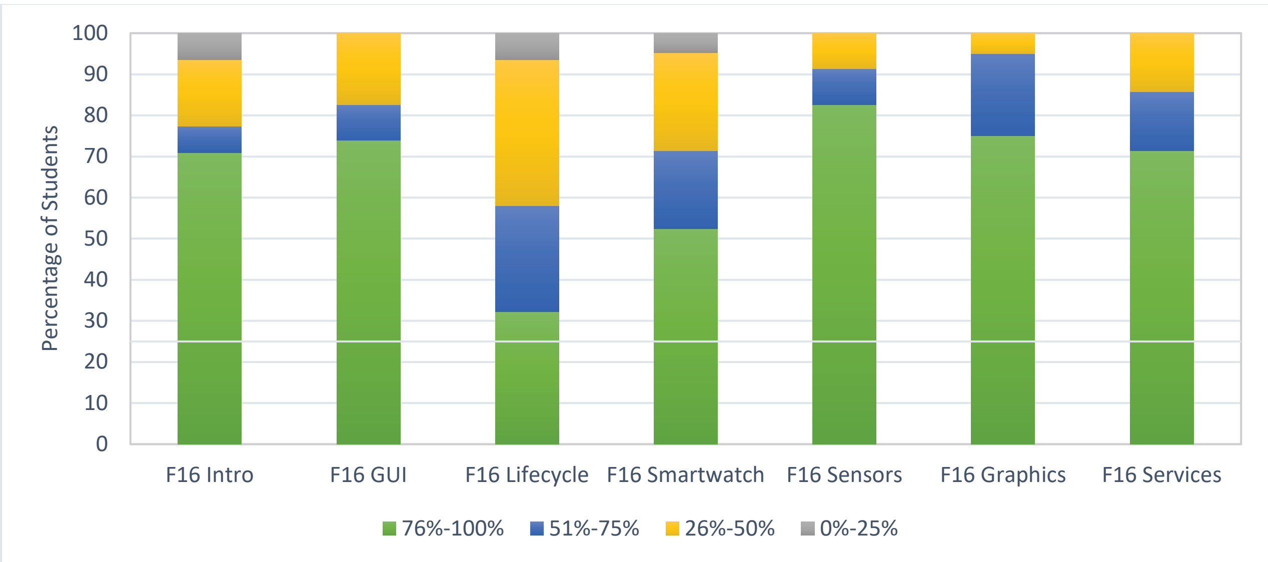

Figure 5.13 Percentage of delivered in-class work in F16

These goals stated in section 5.3.3 were set after our exploratory PP assignments in S15. However, if we want to apply such measures on S15, Figure 5.11 shows that Lifecycle, Sensors, and Graphics assignments have achieved such goals.

5.3.3.2 Discussion

Figure 5.11 shows that although the Smartwatch assignment was very close to achieving the stated goals, it raises some concerns; since only 50% of students submitted around 50% of the required work, and around 22% of students submitted less than 25% of functionality. As we mentioned in section 5.2.2, the in-class PP assignments of S15 were not well-integrated with the syllabus, and students were using the "traditional" PP; not the LTP's version. However, this semester was important for us as it worked as guidance for our work in S16 and F16.

Figure 5.12 shows that in S16, all the assignments have achieved the first goal (more than 50% of students were able to submit more than 50% of the required work). The figure, however, shows that six of these assignments (including Intro) had some students submitting less than 25% of the functionality. Although the percentage of students submitting less than 25% is between 2% and 10%, we still wanted to minimize such numbers. Figure 5.12 also shows that the assignments of Intro, Smartwatch, and Services are the ones with the lowest percentages of students submitting more than 50% of work.

Applying LTP in F16 led to the results shown on Figure 5.13. The two goals stated in 5.3.3 were achieved for each assignment, where more than 50% of students were able to submit more than 50% of functionality, and no more than 25% of students have submitted less than 25% of functionality.

Figures 5.14, 5.15, and 5.16 show how the numbers for the three modules with the lowest submission rates (Intro, Smartwatch, and Services) have changed over time. Sections 5.4.2.2 and 5.7.2 provide an explanation for why the modules of Intro and Smartwatch had the lowest submission rates.

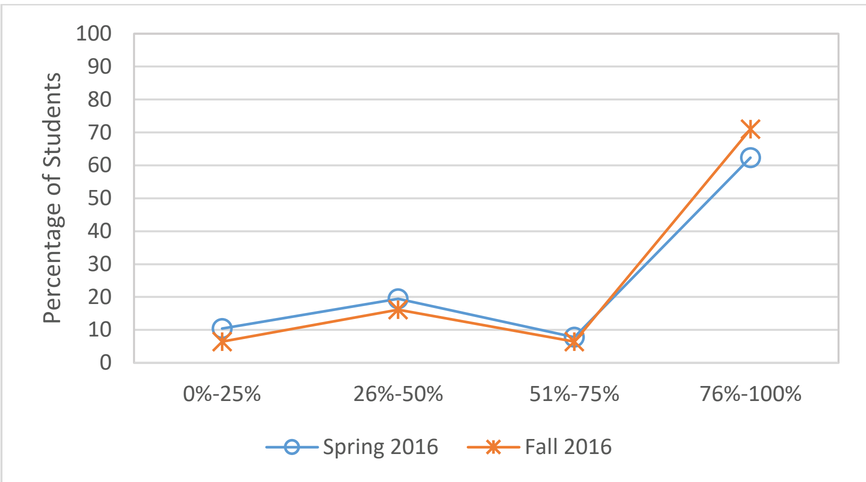

Figure 5.14 Percentage of delivered in-class work for the Intro assignment in S16 and F16

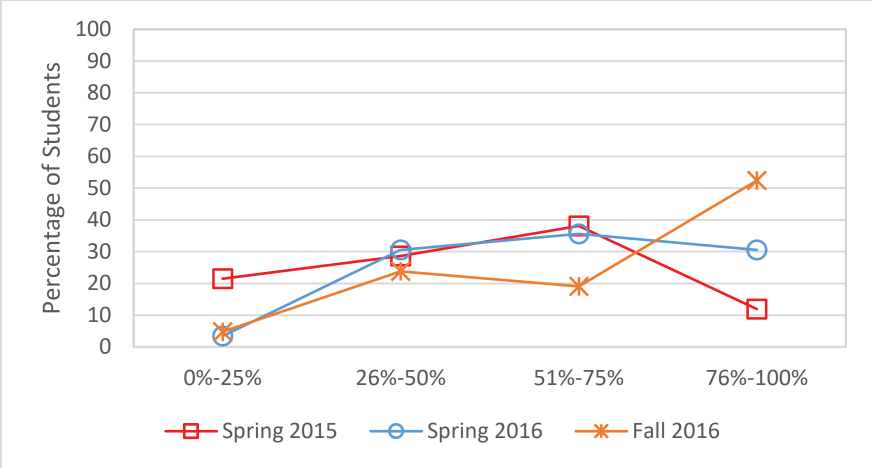

Figure 5.15 Percentage of delivered in-class work for the Smartwatch assignment in S15, S16, and F16

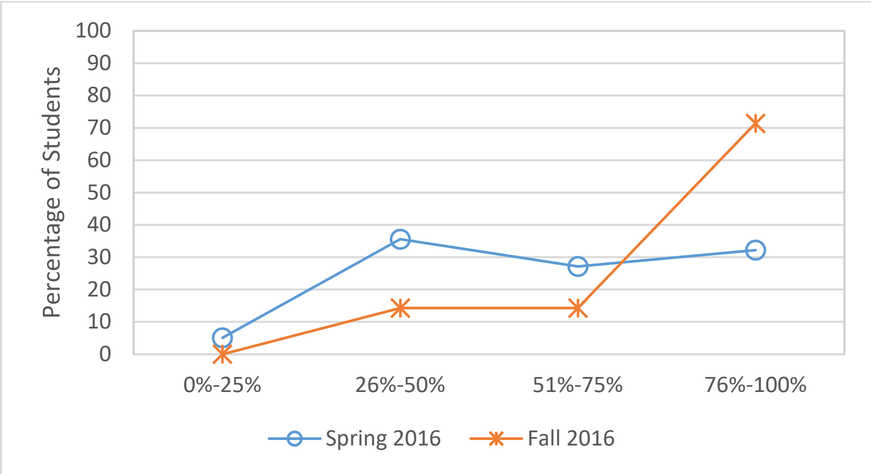

Figure 5.16 Percentage of delivered in-class work for the Services assignment in S16 and F16

We applied Chi-Square statistical test to investigate how LTP affected the percentages of delivered in-class work over the three semesters. The threshold used for p is always 0.05. We started by the modules that scored the lowest submission rates.

- For the Intro assignment (with data available only for S16 and F16): There is no significant difference: χ^2 (1, N=111) = 1.98, p=0.574
- For the Smartwatch assignment: There is a significant difference: χ^2 (1, N=122) = 19.18, p=0.0038
- For the Services assignment (with data available only for S16 and F16): There is a significant difference: χ^2 (1, N=81) = 8. 8, p=0.032

Due to the special nature of the Intro assignment (discussed in section 4.4), we will focus on discussing the Smartwatch and Services assignments. From the Chi-Square analysis above, it shows that there was significant improvements in students' submission rate after using LTP. Therefore, and despite having the lowest submission rates per semester, those rates still got improved after applying LTP.

Figures 5.17, 5.18, and 5.19 show the numbers for GUI, Sensors, and Graphics modules. We also applied Chi-Square test on these values as well, and the results were:

- For the GUI assignment: There is no significant difference: χ^2 (1, N=98) = 0.3, p=0.958
- For the Sensors assignment: There is no significant difference: χ^2 (1, N=85) = 0.58, p=0.899
- For the Graphics assignment: There is no significant difference: χ^2 (1, N=122) = 6.19, p=0.402

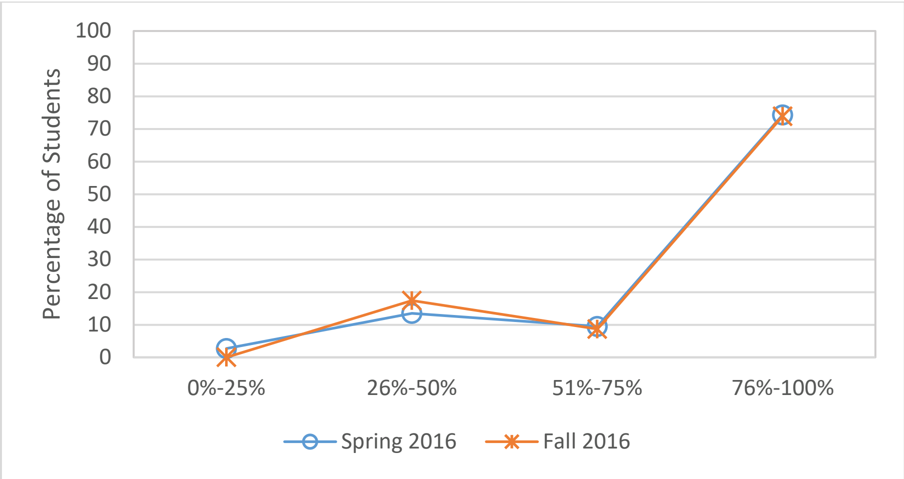

Figure 5.17 Percentage of delivered in-class work for the GUI assignment in S16 and F16

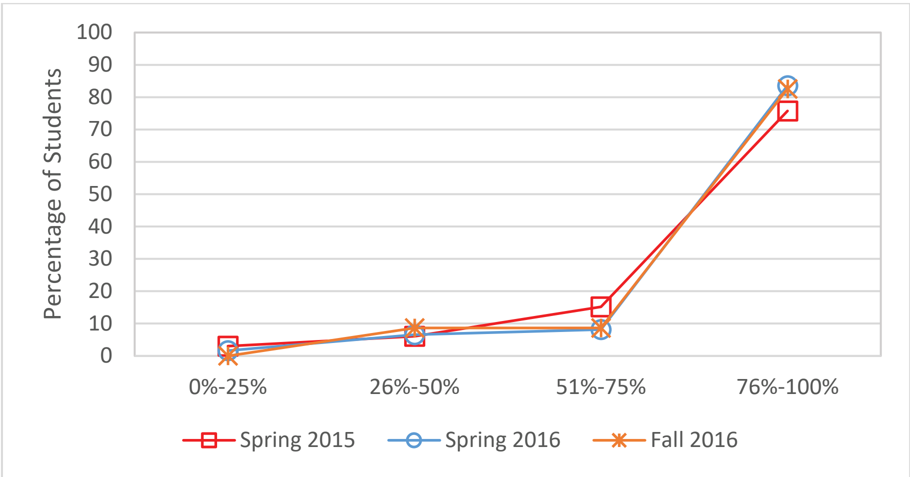

Figure 5.18 Percentage of delivered in-class work for the Sensors assignment in S15, S16, and F16

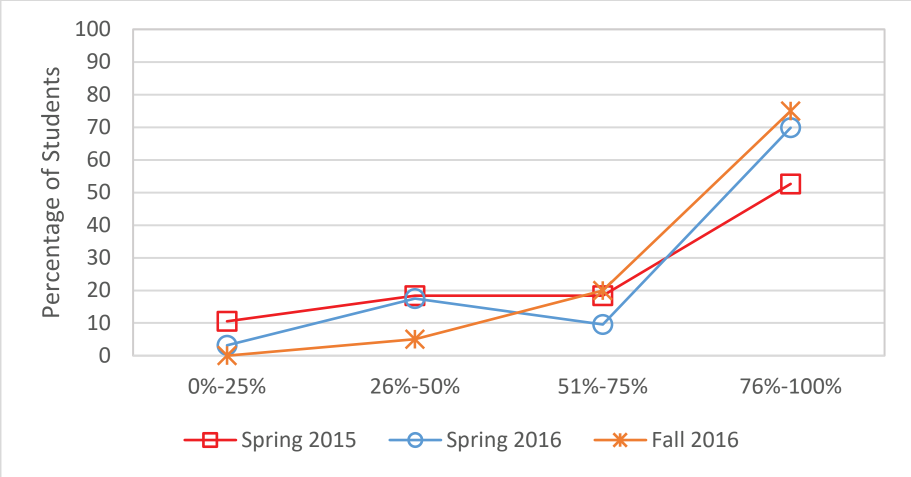

Figure 5.19 Percentage of delivered in-class work for the Graphics assignment in S15, S16, and F16

The one case that we found to be an "anomaly" is the Lifecycle assignment of F16. A Chi-Square analysis of the delivery rate of this assignment over the three semesters shows that there is a significant difference: χ^2 (1, N=139) = 17.5, p=0.007. Although it has achieved our goals, it was the only assignment that showed lower percentages of delivery than the previous semesters. Figure 5.20 shows a comparison of the Lifecycle assignment percentages of work delivered over the three semesters.

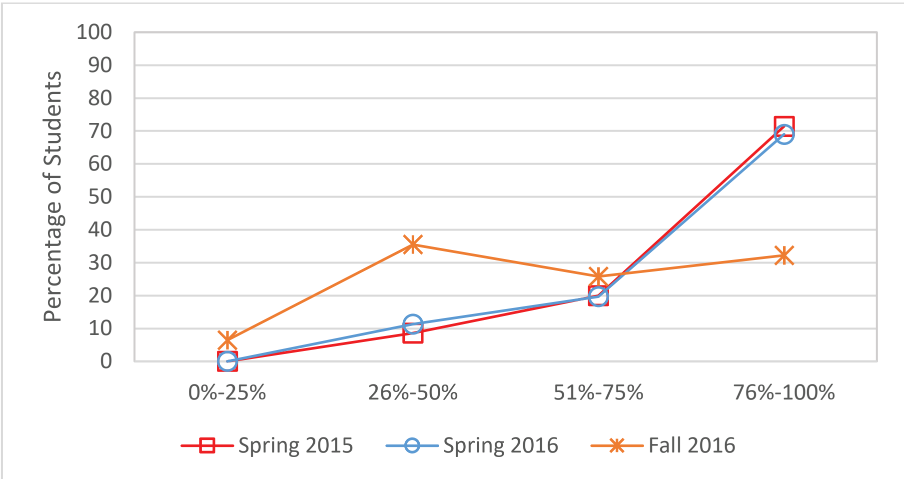

Figure 5.20 Percentage of delivered in-class work for the Lifecycle assignment in S15, S16, and F16

After reviewing the questionnaire responses related to that assignment, we found out that four students mentioned that their computers crashed while in the middle of working on the assignment, and two other students started working late because they arrived very late to the class that day. Besides that, this session started with a conversation on some logistical problems that students faced while forming project groups and submitting project assignments to Canvas (the online course management system). Such discussion took some extra 10 minutes in addition to the 10 minutes needed for introducing the

PP assignment. We believe that all these reasons have collectively affected students' performance that day, and that was what led to lower-than-expected percentages of work submitted.

5.4 STUDENTS PERCEPTION AND FEEDBACK

After every PP session, there was a questionnaire for students that they could fill-in and provide some feedback on the session. The questionnaire was not mandatory and students did not to have to answer every question. The questions were provided and answers were submitted via Canvas (the online course management system). Details on the questions are provided on Appendix B. The open-ended questions were important for us during the semesters to provide us with feedback that can help making future sessions better. Particularly, the open-ended questions were the students' space where they can provide feedback, suggestions, and criticism to the experience they had.

5.4.1 Feedback on the PP Experience

Besides helping us tweaking the assignments to mitigate certain problems, we chose to quantify some of the students' answers to a certain question, which would reflect how students perceived using PP for the in-class assignments. The question said: How would you describe your experience of today's Pair Programming session?

For every assignment, we counted the number of positive answers to this question (that have words like "good", "insightful", "useful"…) and the number of negative answers (that contain words like "bad", "disappointing", "not fun" …). Figures 5.21, 5.22, and 5.23 show the student answers over the three semesters where the data was available.

5.4.1.1 Findings

Figure 5.21 shows that in S15, where we applied the tradition PP version, the satisfaction rate among student on the four sessions we asked that question was always less than 80%, with only Smartwatch being lower than 70%.

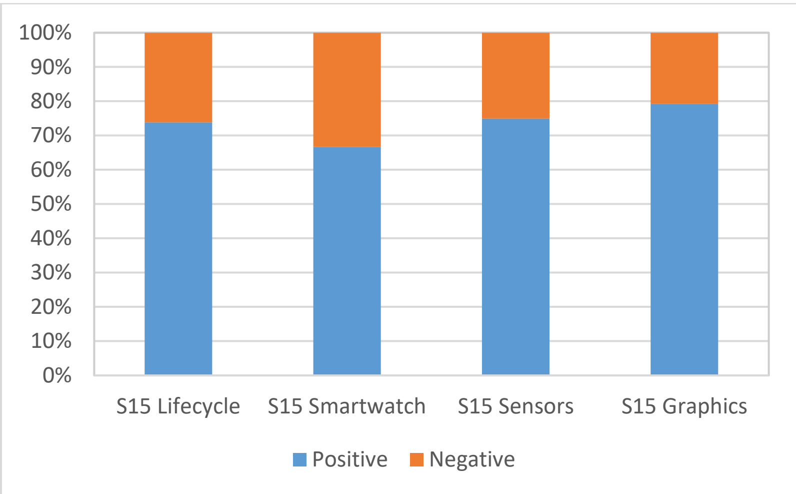

Figure 5.21 Percentage of positive vs. negative feedback in S15

In S16, as Figure 5.22 shows, the positive feedback from students was more than 80% for all the assignments but two: Smartwatch and Services. The satisfaction rate for these two modules was around 50%.

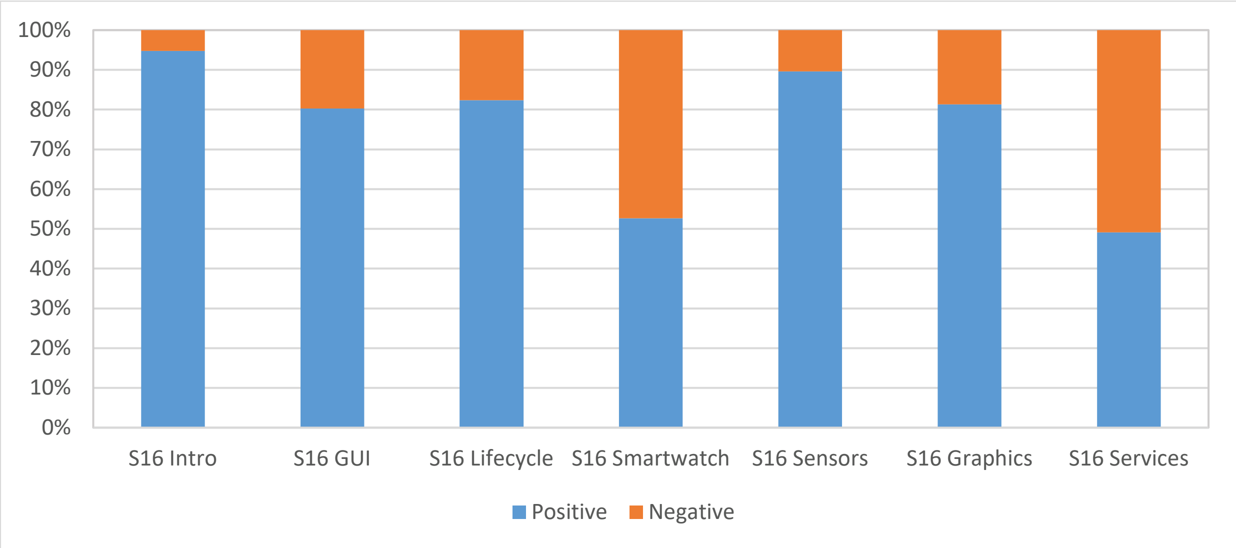

Figure 5.22 Percentage of positive vs. negative feedback in S16

In F16 the satisfaction rate was higher than 80% for all the assignments but two: Lifecycle and Smartwatch, as shown in figure 5.23.

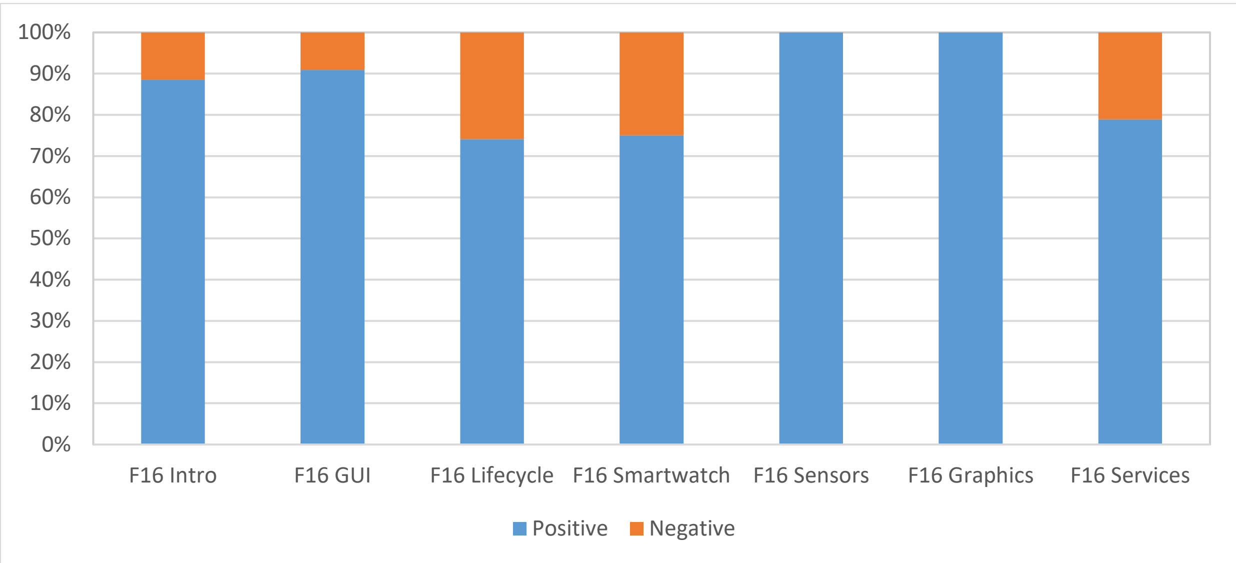

Figure 5.23 Percentage of positive vs. negative feedback in F16

5.4.1.2 Discussion

We explained in section 5.3.3.2 that Lifecycle assignment of F16 had some factors that led to poor performance from students. A detailed discussion on the Smartwatch module is presented in section

5.7.2. We applied the Chi-Square statistical test to know which modules had significant change in the satisfaction levels over the semesters and which had no significant changes.

- For the GUI assignment: There is no significant difference: χ^2 (1, N=93) = 1.33, p=0.248
- For the Lifecycle assignment: There is no significant difference: χ^2 (2, N=118) = 1.2, p=0.547
- For the Services assignment: There is a significant difference: χ^2 (1, N=74) = 5.12, p=0.023
- For the Sensors assignment: There is no significant difference: χ^2 (2, N=109) = 5.5, p=0.063
- For the Smartwatch assignment: There is no significant difference: χ^2 (2, N=106) = 3.4, p=0.182
- For the Graphics assignment: There is no significant difference: χ^2 (2, N=105) = 1.86, p=0.392

The analysis shows that only the Services module has a significant difference in the satisfaction levels, and we provide an explanation for this in section 5.7.1. Figures 5.24, 5.25, 5.26, 5.27, 5.28, and 5.29 show the assignments side by side for better presentation of the satisfaction rates with the traditional PP in S15, then the original LTP in S16, and finally after applying the modified LTP in F16.

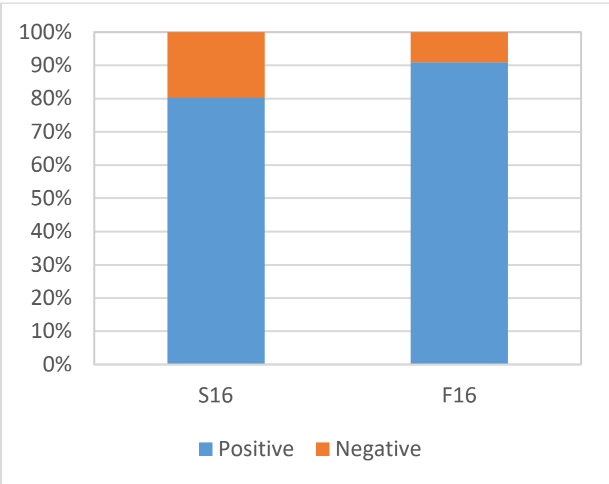

Figure 5.25 Percentage of positive vs. negative feedback for the GUI assignments

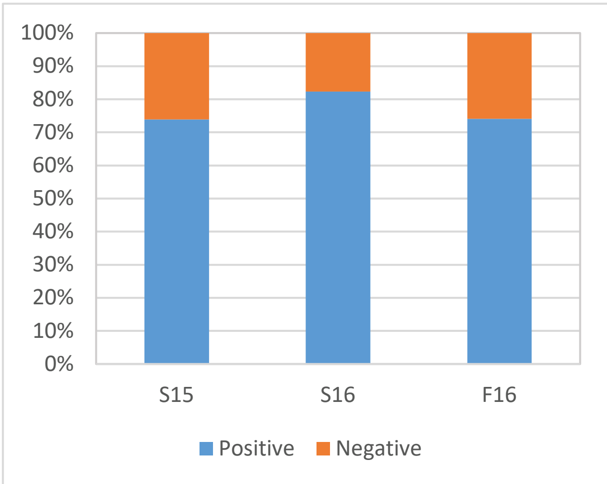

Figure 5.24 Percentage of positive vs. negative feedback for the Lifecycle assignments

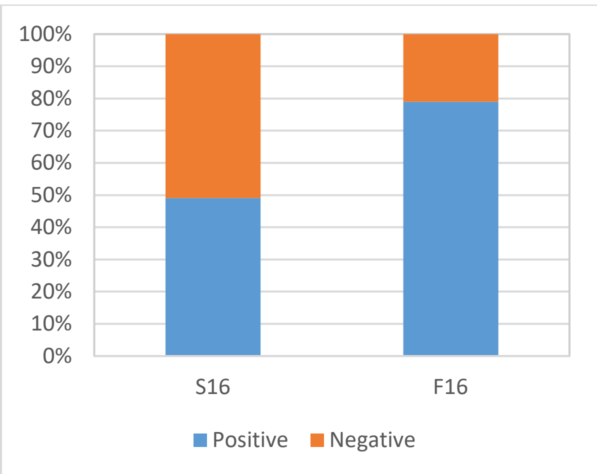

Figure 5.27 Percentage of positive vs. negative feedback for the Services assignments

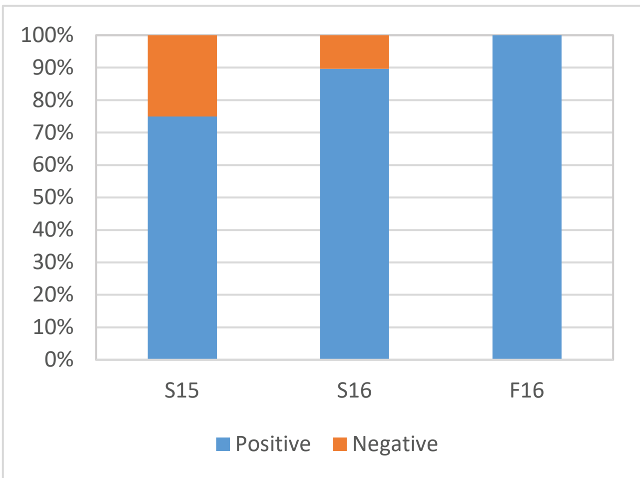

Figure 5.26 Percentage of positive vs. negative feedback for the Sensors assignments

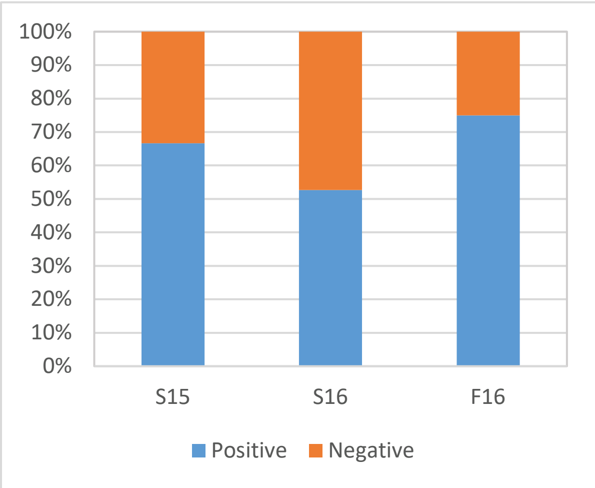

Figure 5.29 Percentage of positive vs. negative feedback for the Smartwatch assignments

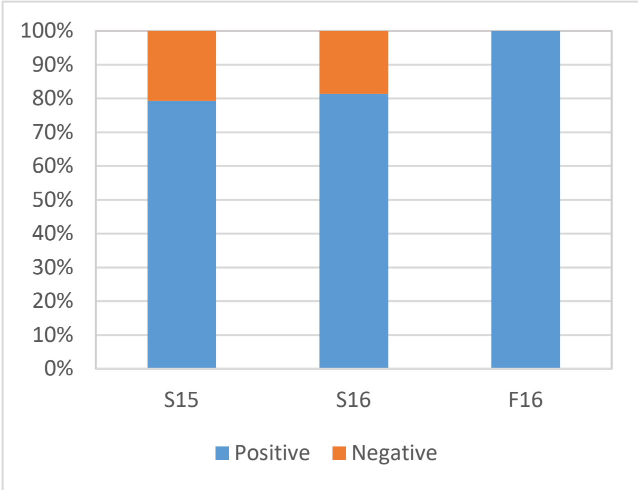

Figure 5.28 Percentage of positive vs. negative feedback for the Graphics assignments

5.4.2 Division of Workload

Students were asked to answer a question about their perception of how the work was divided between the two partners. We aimed at having more than 90% of students working in equal (or almost equal) workload.

5.4.2.1 Findings

Our definition for equal/almost-equal workload is for every student to work on 40%-60% of the required workload. Therefore, answers like 50-50, 45-55, and 40-60 were all grouped under the category of equal division of labor. If any of the students perceived that they worked more than 60% or less than 40%, their answers are added to the unequal division of labor category. Figures 5.30 and 5.31 show summaries of responses for S16 and F16.

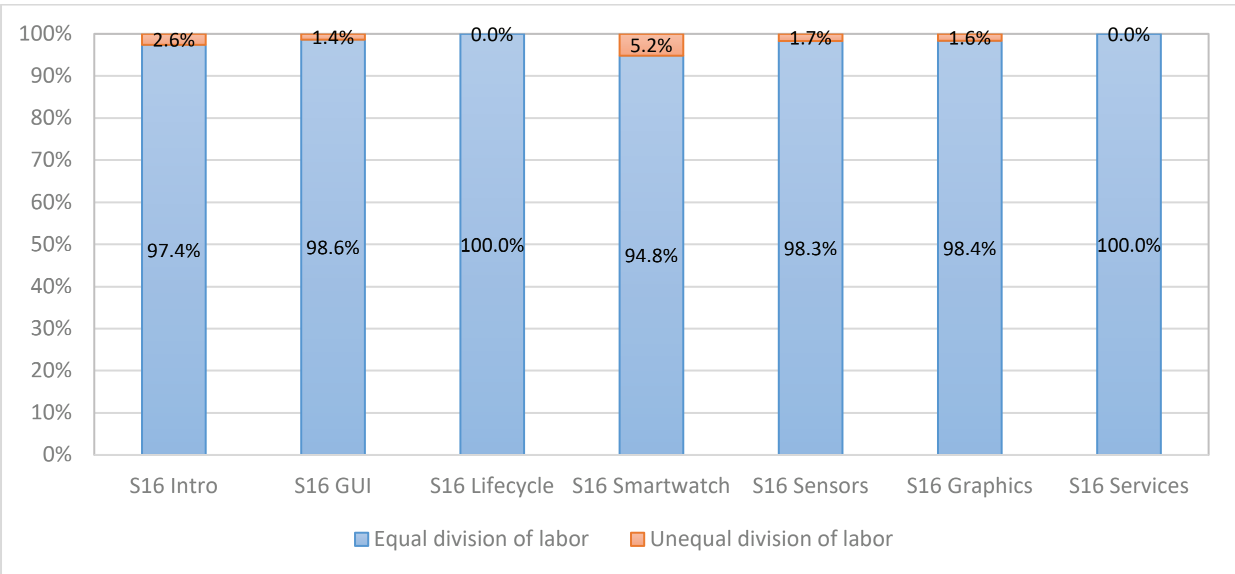

Figure 5.30 Percentage of workload division among pairs in S16

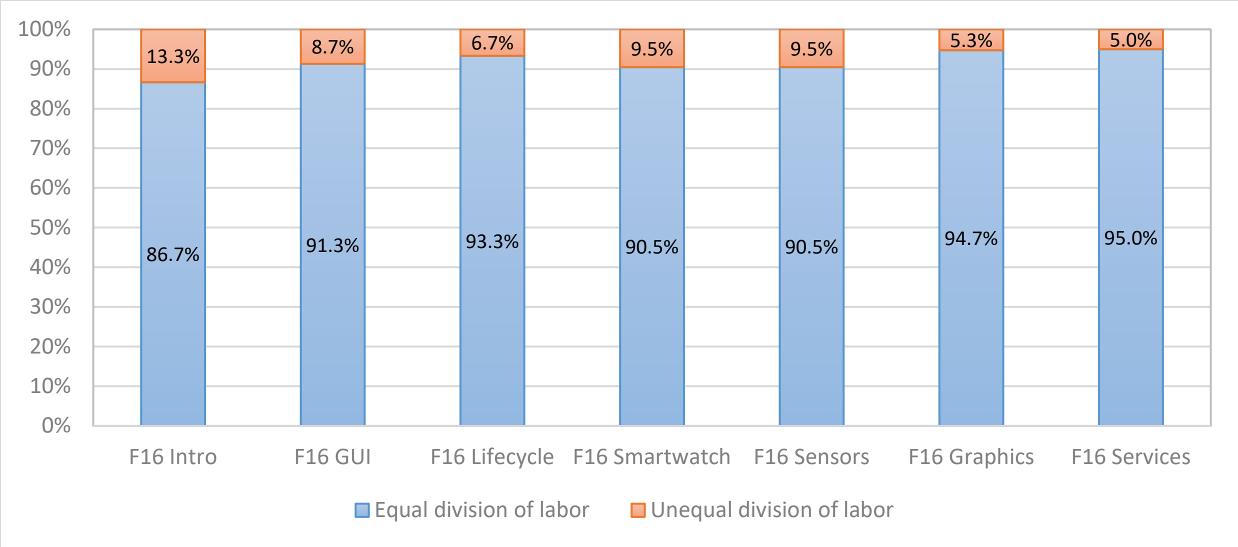

Figure 5.31 Percentage of workload division among pairs in F16

5.4.2.2 Discussion

The "Intro" session, as discussed in section 4.4, was only 20-minute session with two 10-minute rounds. Therefore, students did not have time to experience true division of workload. That explains why the intro session in the two semesters were among the lowest rates of equal division of labor.

For F16, it is important to note that the number of students who responded to this survey question with percentages that are lower than 40% or higher than 60% is 1-2 students. The open-ended questions tied to these students usually showed that one partner was busy "working on homework" or "using his/her phone". This highlights the important role of the facilitator, who should – with the help of LAs – monitor students' work and make sure that everyone is working on the assignment, not anything else. Although they are still very few cases, they should not be allowed, and extra care should be given to correct these situations once captured.

What is important for us is that students had the perception of being able to divide the workload with partners, and to be collectively responsible for their code. We believe that working in pairs through the LTP approach have helped students realize that working with another member does not mean doing all the work alone neither not working at all. Therefore, The LTP approach has helped students implement the practice of collaborative working on assignments that require coordination among multiple devices.

5.4.3 Gain from Partner

Students were asked to answer a question about whether they gain something new from their partners. A 5-point Likert scale (Likert, 1932) was used for students to provide their answers, ranging from "Strongly Disagree" to "Strongly Agree". From this question, we had two goals in mind:

1. To know if PP helped students gain more knowledge by interacting with another student instead of working alone.

2. To get students to think about whether they actually learned something new from their partners, and thus understand the value of working in pairs rather than on their own.

5.4.3.1 Findings

We chose to use diverging stacked bar charts to represent students answers to the question on whether they gained something new from their partners. Diverging stacked bar charts position the replies horizontally so positive responses are stacked to the right of a vertical baseline (which is neither agree nor disagree in our case) and negative responses are stacked to the left of this baseline (Heiberger and Robbins, 2014). The results shown in figure 5.32 and 5.33 show that in both S16 and F16, and for all the sessions, the majority of students (over 60%) either agreed or strongly agreed on gaining something new from their partners.

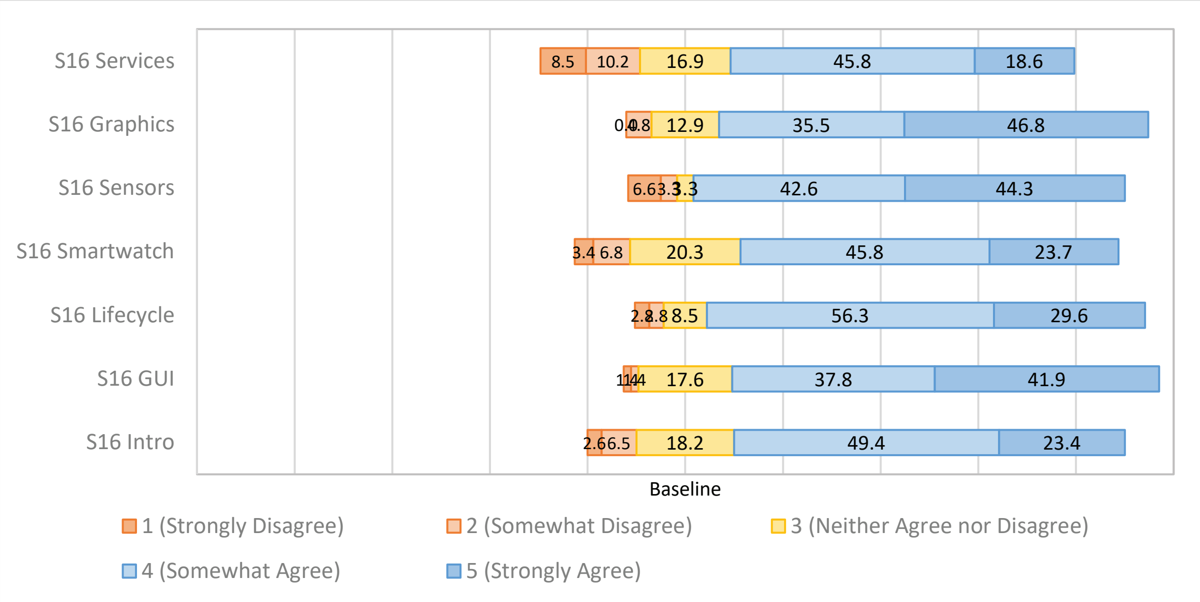

Figure 5.32 Attitude of responses to the gain from partner question in S16 (numbers are in percentage)

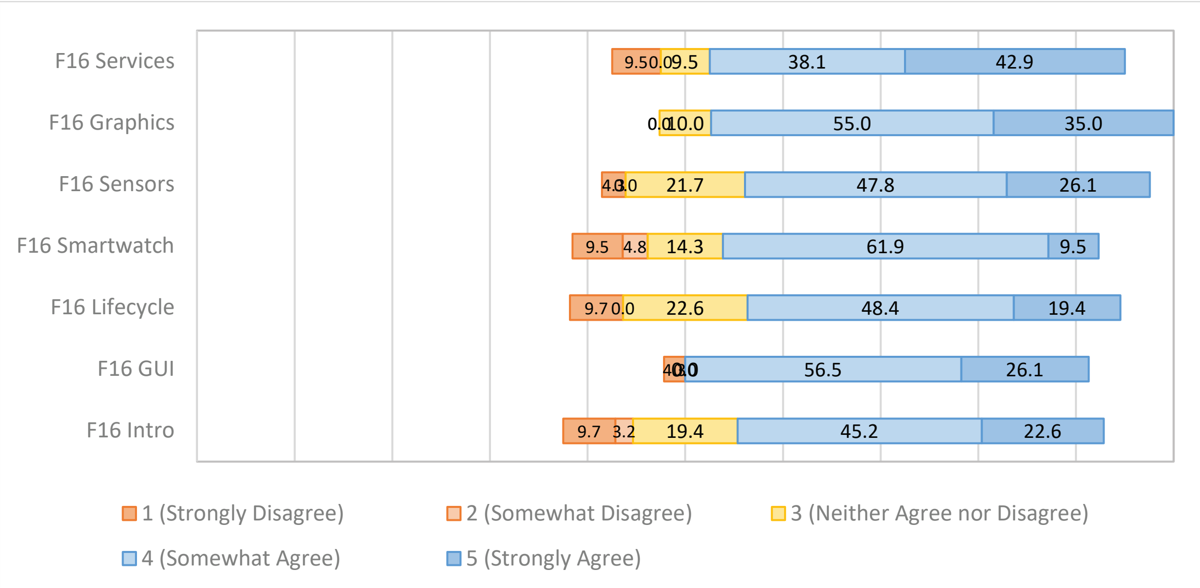

Figure 5.33 Attitude of responses to the gain from partner question in F16 (numbers are in percentage)

5.4.3.2 Discussion

It is clear from the diagrams that the percentages of the positive answers ("somewhat agree" and "strongly agree") dominate every module in the two semesters. However, the real value of the above diagrams is in showing that students were able to identify new things they were able to learn from their partners in every PP session. Together with the discussion on the division of work in section 5.4.2.2, we can conclude that PP has helped students work with new partners and learn new things from them that contributed to better comprehension of certain topics of the modules.

5.5 IN-CLASS OBSERVATIONS

The facilitator and LAs were observing students' work during the PP sessions. LAs were filling a sheet based on the questions from students, and the facilitator was taking notes regarding the overall session and interactions among students and LAs. This section provides more details on the observations of LAs, while the facilitator observations are used throughout the chapter to explain certain cases and provide interpretation for data findings.

5.5.1 LA Sheets

The Lab Assistant (LA) sheets were used to collect information from the LAs who were helping with answering students' questions during the PP sessions. A simple sheet is given to each of the LAs in the beginning of the session, and they were asked to fill in an entry to this sheet to work as a "log" for the questions they get from students. This was meant to help us know more about that kind of issues that students might have so that we can react by focusing on such issues in future sessions. There were nothing unexpected in the notes from LAs regarding the types of questions, so we are focusing here on the number of questions asked on every session.

5.5.1.1 Findings

Our anticipation was that students would be able to learn to work with their partners and find the solution to their problems instead of reaching out to the LAs for every problem they face. Because PP was new for students, we expected that it would take them time to feel comfortable working on their own and consulting each other, instead of calling for the LAs. Figures 5.34 and 5.35 show the number of questions answered by LAs on S16 and F16.

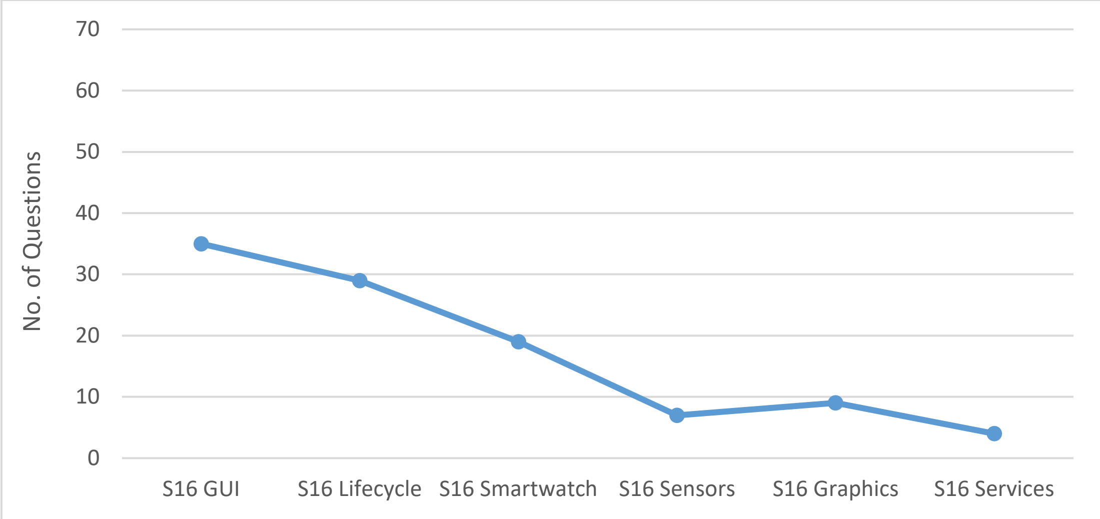

Figure 5.34 Number of questions answered by LAs in S16

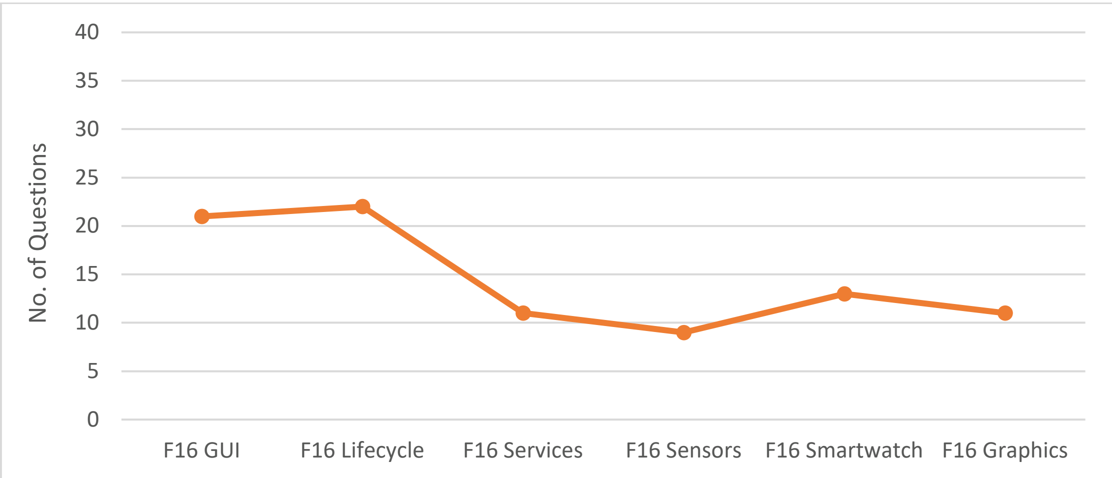

Figure 5.35 Number of questions answered by LAs in F16

5.5.1.2 Discussion

The main observations on figures 5.34 and 5.35 are:

1. The GUI and Lifecycle module are the first two modules of the two semesters, and they had the biggest number of questions asked. Since students performed well in delivering the assignments, and since most of those questions did not take more than 30 seconds to answer (based on the LA notes), these two sessions seemed to work as transitional sessions for students. Instead of being used to call for the LA's help, they learned in these sessions that they can – and should – consult each other (as a pair) to find solutions for their problems. Therefore, no matter what the following modules where, the number of questions was always lower than the first two sessions.
2. Other than the first two modules, the Smartwatch module was the one with the most questions asked. Whether it is the third module in S16 or the fifth in F16, it still got more questions that

the other modules. Section 5.7.2 provides explanation on the reason for this module to get that many questions during the sessions, as well as low satisfaction rate as shown in section 5.4.1.

5.6 END-OF-SEMESTER QUESTIONNAIRE

At the end of the two semesters, S16 and F16, students were asked to fill-in a questionnaire regarding their overall experience during the semester. The questions of the online questionnaire appears in Appendix C. The goal of this questionnaire was to collect students' feedback after working on the six modules, with all the lectures, tutorials, and PP sessions are done. We also wanted this questionnaire to be submitted after students finished working on all of their homework assignments as well as final projects. This section provides some of the interesting observations we found while analyzing students answers to the End-of-Semester questionnaire.

Although the data presented in this section belongs to the "Students Perception and Feedback" category that was discussed in section 5.4, we decided to discuss the results of the end-of-semester questionnaire in a separate category after presenting the other collected data. The reason is this questionnaire provides students overall feedback at the end of each semester, which comes after they experienced all the different stages of the LTP approach with its six modules, lectures, tutorials, and PP sessions. Therefore, with this comprehensive feedback, we wanted to make it the last collection of data we present in this chapter.

5.6.1 Number of PP Sessions

In this part, students of S16 and F16 were asked about their perception about the number of PP sessions, which was a total of seven sessions; 1 introductory and 6 full assignments. Students had to choose from too little, just right, or too much.

5.6.1.1 Findings

Figure 5.36 shows that around 38% of students in S16 thought the number of PP sessions was too much. However, in F16, this percentage dropped to around 17%.

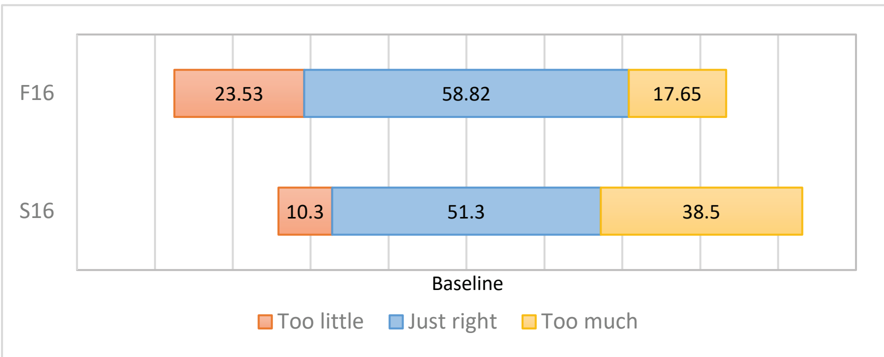

Figure 5.36 Students perception on the number of PP sessions in S16 and F16

5.6.1.2 Discussion

We believe that by better incorporation of PP within the syllabus, and with the preparation through lectures and tutorial sessions to work on the PP assignments, students were able to feel the value of PP sessions. We also noticed that in both semesters, more than half of students believed that the number of PP sessions was just right. This, again, shows that having a PP session as a core component of each module provided valuable and meaningful addition for students' perception of learning during the mobile development course.

5.6.2 Length of PP Sessions and Iterations

Since we were working in an educational institute where class times are not very flexible, we knew that the length of our sessions would be an issue. However, we tried to overcome this challenge by preparing students for PP sessions and crafting the assignments to fit within the 75-minute class time. However, it was expected that students would feel that they need more time to finalize their work.

5.6.2.1 Findings

Figure 5.37 shows that the majority of students, both in S16 and F16, believed that the length of PP sessions was too short, and that they needed more time. Besides such expected results, the figure also shows that 30% of students in S16 thought that the session time was too long, which does not reflect the fact that students usually requested more time to work on their assignments.

Figure 5.38 shows that in both S16 and F16, almost two thirds of students had no problems with the iteration length of 15 minutes, while only one third of students found it too short.

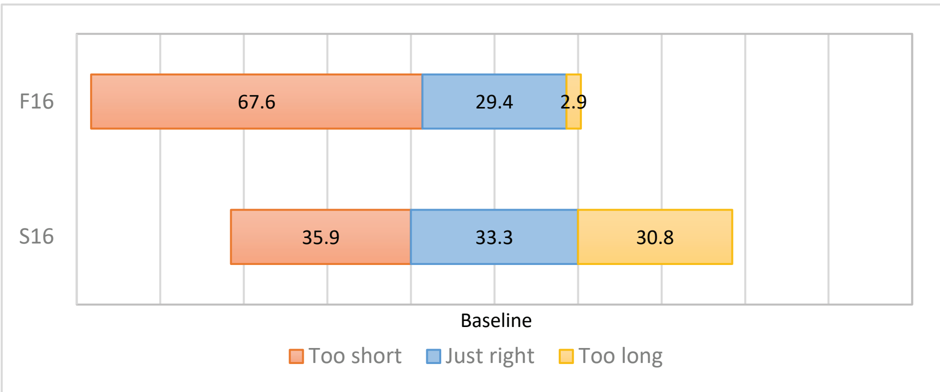

Figure 5.37 Students perception on the length of PP sessions in S16 and F16

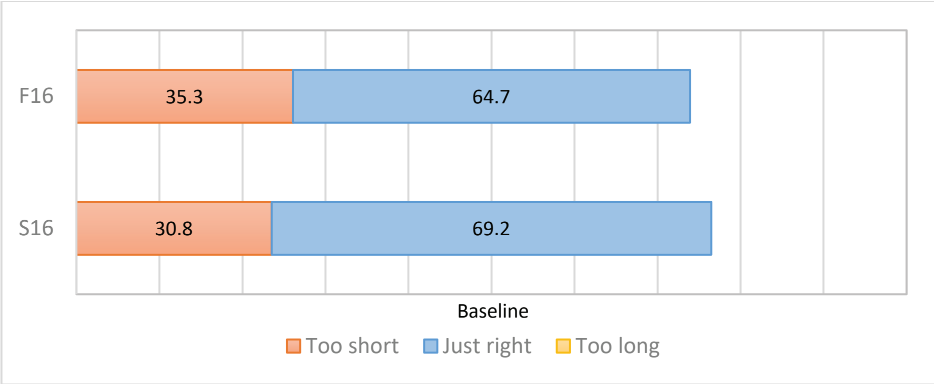

Figure 5.38 Students perception on the length of PP iterations in S16 and F16

5.6.2.2 Discussion

By reviewing the open-ended answers of those who thought the sessions were too long, we found that around 70% of them were not in favor for PP because they believed it was demanding and required them to work hard and fast. Therefore, we believe that most of the 30% who thought the sessions length was too long were actually expressing their dissatisfaction regarding PP because it required them to be "over-achievers". It is important as well to notice that this percentage is less than 3% in F16, which complies with our expectations.

As shown in Chapter 4, we wanted the session length to be 15 minutes so that every student will get to be a driver and a navigator twice each session. With this in mind, we had to make the compromise of having four 15-minute session versus fewer longer iterations. We believe that having two thirds of students doing well with the 15-minute iterations to be an indicator that our compromise worked well with the majority of students.

5.6.3 Usefulness of PP for Understanding Modules' Topics

The question about the usefulness of PP towards understanding the modules' topics aimed at assessing students' perception on how PP assignments helped them comprehend the mobile development topics presented in each of the six modules.

5.6.3.1 Findings

The results depicted in figure 5.39 show that more than 58% of S16 students and more than 67% of F16 believed that PP helped them gain better understanding of the modules' topic. Moreover, the percentage of students who think PP sessions have not helped them understand the topics was 20.5% in S16 and 17.7% in F16. The results show that the PP component of our work was essential for the majority of students to understand the topics covered in each module.

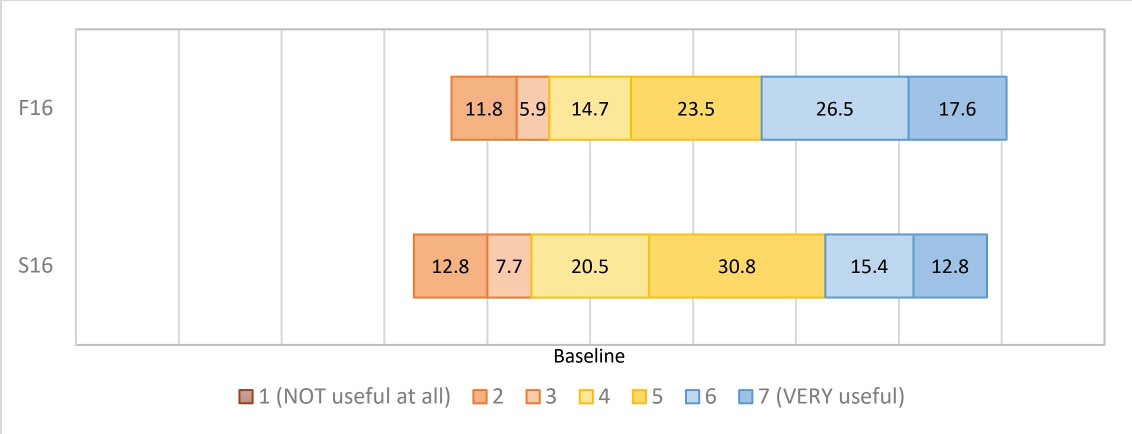

Figure 5.39 Students perception on whether PP helped them understand the modules' topics in S16 and F16

5.6.1.1 Discussion

Several studies show that PP is useful for learning programming concepts and achieves high satisfaction rates among students (Salleh et al., 2011). However, studies show that there are groups of students whose satisfaction rates are lower due to several reasons. (Balijepally, 2006) shows that "second-best" programmers in collaborative pairs were more satisfied with PP because they were able to perform at significantly higher levels than they would when performing individually. Although there were 140 students who "Strongly Agreed" that they learned more by working with a partner in (Carver et al., 2007), the study showed that there were more than 30 students "Strongly Disagreed" to the same statement. In (Chaparro et al., 2005), questionnaire data showed that, on average, only 73%of students considered pair programming was an "enjoyable task". From the previous examples, the 17%-20% rate of students who did not find PP useful in our study is in the same range (or lower than) the percentages found in other similar research studies. However, the benefit of LTP is that it provides different approaches for teaching, so that the students who did not feel the PP was useful could be able to gain from the lectures and the hands-on tutorial sessions.

To put the responses we received from students about the PP in context, we applied Chi-Square test to compare PP usefulness to lectures and tutorials in S16 and F16. The test resulted in:

- For S16: there is a significant difference: χ^2 (2, N=101) = 21.74, p=0.00009
- For F16: there is a significant difference: χ^2 (2, N=88) = 16.06, p=0.0003

Since there were significant differences in the two semesters, we applied Chi-Square test again to compare PP against lectures and tutorials, one at a time:

- S16
  - Comparing PP to lectures: there is a significant difference: χ^2 (1, N=64) = 4.43, p=0.035
  - Comparing PP to tutorials: there is a significant difference: χ^2 (1, N=68) = 7.84, p=0.005

- F16
  - Comparing PP to lectures: there is a significant difference: χ^2 (1, N=58) = 7.32, p=0.006
  - Comparing PP to tutorials: there is no significant difference: χ^2 (1, N=59) = 1.3, p=0.25

In comparison to the students' answers to the questions about how useful were lectures and tutorial sessions, the results of the statistical analysis show that PP sessions always did better than lectures and lower than tutorial sessions (for S16), as shown in figures 5.40 and 5.41.

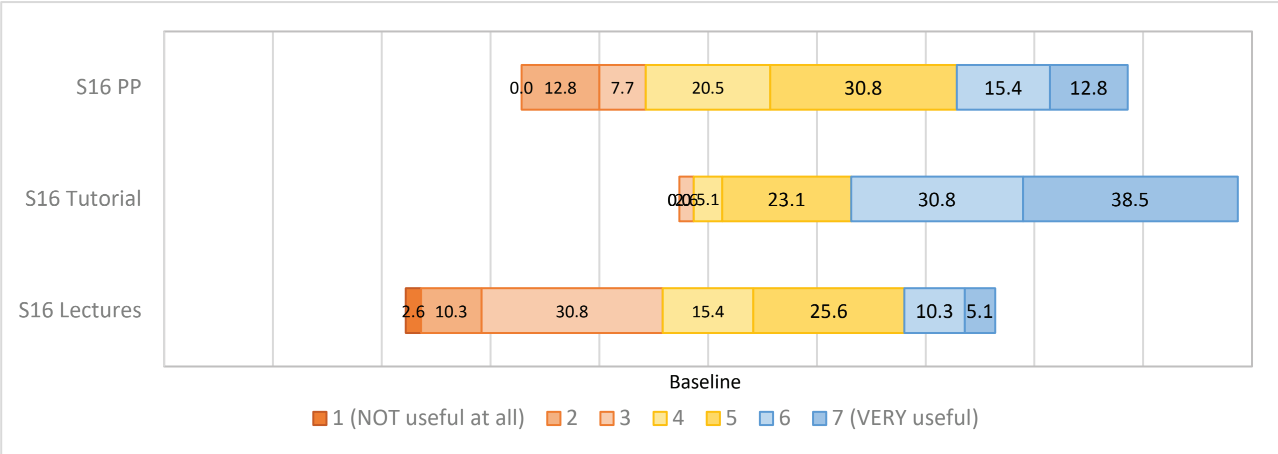

Figure 5.40 Students perception on whether Lectures, Tutorials, and PP helped them understand the modules' topics in S16

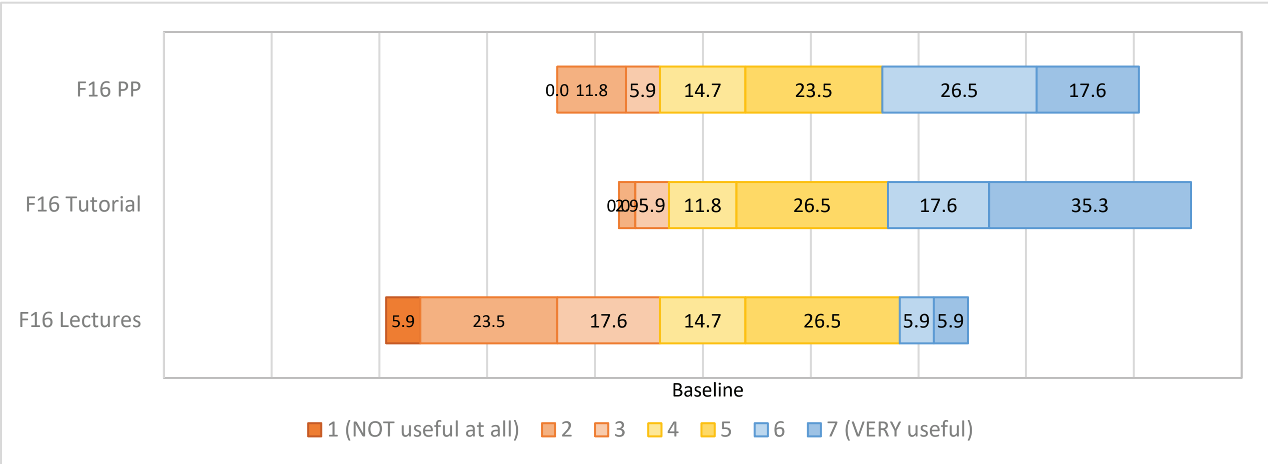

Figure 5.41 Students perception on whether Lectures, Tutorials, and PP helped them understand the modules' topics in F16

These results should not be surprising because of the different nature of the three components:

1. Lectures are more about abstract concepts, definitions, methods, and some history. Students usually do not see these important concepts as relevant to their coding experience. However, we believe that these concepts need to be covered in order to get students to understand the "why" of certain topics, before delving into the "how".

2. Tutorials are when students follow the instructor in defined programming steps. It is the transition phase from the abstracts to the actual implementation. Since they are not required to look for solutions on their own or do hard mental activities, students tend to prefer tutorial sessions because it provides an easy step-by-step guide to learning coding practices.
3. PP sessions are when students need to work within their teams to find solutions to problems that they have not experienced before. Therefore, more mental activities, challenges, and frustration of not getting the right answer on the first time are reasons for students to prefer tutorial sessions over PP. However, we believe that PP adds the component of self-learning to students, which is important in education in general, and in mobile development in particular with all of its new concepts, rapidly changing technologies and variety of devices.

5.7 CONCLUSIONS AND GENERAL OBSERVATIONS

The discussion in this chapter showed that our approach has enhanced students' understanding of the mobile development topics. Moreover, using PP for in-class assignments fostered an environment of collaboration among students that helped them building their knowledge on the six presented modules. This section provides some general observations based on the previous discussions and the facilitator's observations during several class sessions.

5.7.1 Pairing of Students

As presented in section 4.4.5, pairs were not self-selected. Students were asked to partner with each other mostly based on their location in the class. In the beginning of every PP session, students spend 3-4 minutes following the facilitator's instruction on how to find their partners. The idea behind this approach is to make it random while not to spend long time making sure of a fully random assignments. The goal was for each student to come to the class prepared to work with someone they might not know, might now have worked with before, and with no expectations on how the collaboration will work out. More details on this pairing approach is provided in section 4.4.5.

However, some students have provided feedback on the need to be able to choose their own partners, which are their friends who they feel comfortable working with. Therefore, on the last PP session of S16 (which was the Services assignment) we allowed students to choose their pairs.

Based on the numbers we had about this specific session, we found out that:

- It has a 65% delivery rate in S16 (figure 5.12)
- It has 50% of negative feedback responses in S16 (figure 5.22)
- It has the lowest number of questions asked to the LAs in S16 (figure5.34)

The Services PP assignment scored low compared to the other assignments because of some issues, and self-selected pairs was one of them. It can be argued that placing the Services assignment as the last one, which is not the best order for it among the other modules, is the main cause of students' low performance. However, the facilitator observations, together with figure 5.34 that shows that this session had the lowest number of questions asked, provides an explanation of why self-selected pairs was a major problem.

The usual time spent for student to pair in facilitator-assigned pairing is 3-4 minutes, because students follow the pairing method and they know exactly whom they are going to pair with. When they were allowed to select their partners, number of issues came up:

- Students started signaling and talking to each other to "book" a partner
- Some students responded "sure" to more than one request, which led to more discussions – among students – on how to resolve conflicts
- Some students did not get pairing requests, which led to having a small number of them uncomfortably waiting to be paired with someone
- The pairing process took around 9 minutes (compared to 3-4 minutes in earlier sessions)
- For those who paired easily with their friends, they were noticed to be having more side conversations that even noticed in earlier sessions, which caused them to waste more time.
- The small number of questions asked to LAs during this assignment, which contradicts with students' poor performance, goes back to the fact that students having side conversations and not paying enough attention to the assignment in-hand have wasted long time.
- Although this has not been studied as part of our research, we believe that those who were left until the end to pair with other un-paired students had some negative because of the awkwardness of this situation.

In summary, self-assigned pairing led to longer time spent for finding pairs, some students left behind, and wasted time in side conversations. Therefore, we decided not to follow the self-assigned pairing at all in F16, and that was one of the reasons that led the Services PP session in F16 to have better numbers than S16. Figures 5.16, 5.26, and 5.35 show that in the Services PP assignment of F16, students had better submission rate, higher percentage of positive feedback, and a number of questions to LAs that is reasonable and meaningful when compared to the other sessions of the same semester.

5.7.2 Smartwatch Module

From the findings in sections 5.3.2.3, 5.3.3.1, 5.4.1.2, and 5.5.1.2, it is clear that the Smartwatch module has been receiving the low scores in homework assignments, percentage of PP assignment completion, students' feedback, and number of questions asked to LAs. The reasons for this are:

- Dealing with Bluetooth and connectivity issues is usually very new topics for students
- Android's approach for connecting multiple devices requires lots of work on the technical programming side, and it is not a straightforward approach
- Dealing with different pieces of hardware requires lots of troubleshooting and try-and-error practices

- Manipulating multiple devices requires looking up different resources for each one (laptop, mobile phone, and smartwatch)

Based on such numbers and scores, and because this module is important in today's mobile application development world, the LTP approach tried to solve the problem of this module by:

1. Covering certain connectivity issues and coding problems as core parts of the lecture
2. Providing one extra hands-on tutorial session to cover detailed connectivity problems and examples
3. Make sure that the PP assignment covers the basic topics of this module, with a core focus on connectivity and manipulating the Smartwatch itself.

On S16, we worked on the PP assignment itself, while in F16 we worked on the lecture and the hands-on tutorial as well. Figures 5.5, 5.15, and 5.28 (repeated below) and the statistical analysis we performed in sections 5.3.2.1, 5.3.3.2, and 5.4.1.2 show that there is a significant different in the submission rate for the in-class PP assignment of the Smartwatch module. Although the results showed no significant difference in homework grades and satisfaction rate, the values we have for S16 and F16 meet our expectations regarding this module, which we know to be the most demanding to students. It is important to note that we do not have data for S14 to measure how the performance in the Smartwatch module was before applying PP in S15 and then LTP in S16 and F16.

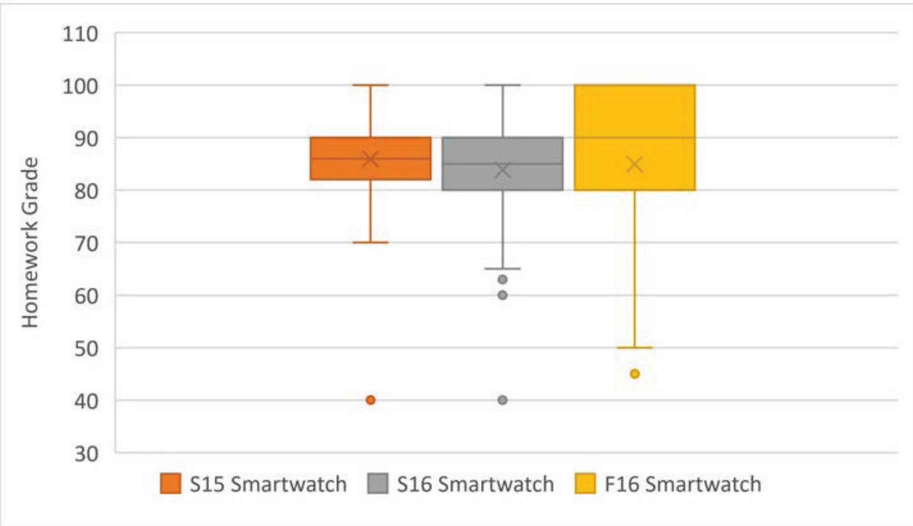

Figure 5.5 (repeated) Homework grades of the Smartwatch module in S15, S16, and F16

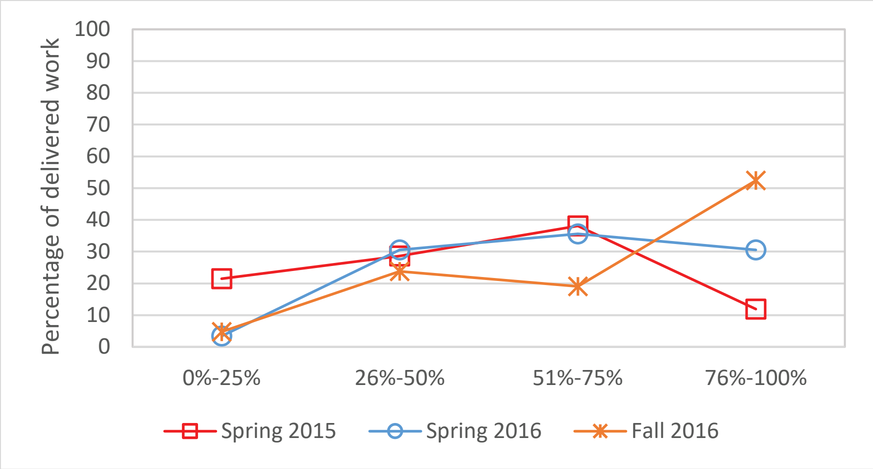

Figure 5.15 (repeated) Percentage of delivered in-class work for the Smartwatch assignment in S15, S16, and F16

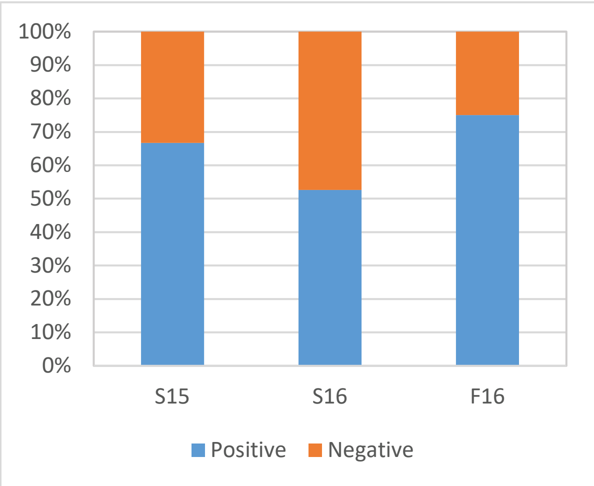

Figure 5.28 (repeated) Percentage of positive vs. negative feedback for the Smartwatch assignments in S15, S16, and F16

We believe that although the Smartwatch module is still one of the hardest to introduce, the LTP approach has helped students to work on the Smartwatch PP assignment and achieve better results than the previous semesters.

5.8 CONCLUSION

The analyses provided in this chapter provided insights from different perspectives: students' academic performance through grades, students' perceptions through questionnaires, and observations. The detailed analysis of each of those components showed that our goals of developing and applying the LTP approach have been met in S16 after applying our original version of the approach, and most of such goals have been improved in F16 after applying the modified version. Since the version presented in Chapter 4 was the finalized version, we believe that CS educators will be able to achieve the required goals of the LTP approach based on our own experience with the two mobile development classes of S16 and F16. Chapter 6 provides final thoughts on the LTP approach, as well as future work directions that would benefit from our experience with the LTP in-class practices.

Chapter 6

Conclusions and Future Work

6.1 RESEARCH SUMMARY AND CONCLUSIONS

The emerging area of mobile software development, which combines aspects of software, hardware, and interpersonal interaction, captures several challenges that are different from the traditional development environments; e.g., understanding how to develop for multiple screen sizes, using different communication channels like Bluetooth and Wi-Fi, and dealing with vast amounts of sensor data. Teaching these challenging materials seemed well suited for multiple approaches that leveraged different learning styles.

This research presents the LTP approach, which examines three teaching methods employed in mobile software development course. The three components of the LTP approach are Lectures, Tutorials, and Pair Programming. Lectures are used to introduce topics and explore underlying theories of development. Hands-on Tutorials present applied development approaches, and explore their use in an individual-based hands-on demos. Pair Programming (PP), which is an agile software development practice that has been used in both industry and education, is implemented to enforce students to apply the knowledge they have collected, and to explore new topics on their own in groups of two. Finally, homework assignments are used to assess students' learning throughout the semester.

Although the LTP approach integrates different components, PP is considered the core element of this approach due to the benefits we found in using it for handling mobile development challenges. Although PP has been studied previously as a pedagogical tool, we provide an adapted version that better fits the requirements of mobile application development. Integrating PP as a core element of mobile development classes aims at giving opportunity for students to collaborate, share experiences, and solve problems together. Therefore, this research provides detailed practices on how to apply our version of PP for in-class assignments, and how to make sure that all the other components are well integrated with the PP sessions.

The research went through different steps that helped us understand the mobile development challenges, explore how PP would be implemented, build the LTP approach, and enhance it for better results. We conducted expert reviews with two pairs of experienced Android developers. We then examined student course work during PP sessions in Spring 2015 mobile app development class. Based on our experience gained from the feedback provided during our investigation, we developed the first version of LTP approach and applied it to students of Spring 2016 mobile software development class. We made some modifications and applied the LTP approach again to students of Fall 2016 class.

We analyzed student homework and final grades over four semesters, students' answers to questionnaires, and observations from session facilitators and Lab Assistants. Results show that repetition of topics through different channels is important for mastery of the topics. Foundational theories seem well suited for lectures, while programming concepts work better in active learning situations. Our analysis also show the importance of PP in helping students recall practices, learn how to find information, communicate problems and solutions effectively, and develop better understanding of mobile development topics. The LTP approach provided in this research will help educators to develop mobile development courses by providing recommendations and practices that have been proven successful through research work and in-class experiments.

6.2 RESEARCH QUESTIONS (REVISITED)

Throughout our research, we were able to find answers to the research questions provided in section 1.4. This section revisits the research questions and provide summaries for the answers we found. Detailed answers are available in the previous chapters, which are referred to within the provided answers below.

1. How can we enhance students understanding in mobile software development classes?
   By applying the LTP approach, students were able to achieve higher final grades in the semesters it was applied in, as well as higher homework grades for several modules (as shown on sections 5.3.1 and 5.3.2).
2. Can collaborative work enhance students' performance in mobile development classes?
   Yes. Section 5.3.3 showed that collaborative work (through PP) helped increase students delivery rate for in-class assignments. the positive feedback from students regarding the PP assignment showed that it helped enhancing their performance (as presented in section 5.4.1). Moreover, sections 5.4.2 and 5.4.3 showed that students were able to divide the workload equally with their partners, as well as learning new things from their partners during the PP sessions. Therefore, incorporating PP as a main component of the LTP approach helped in enhancing students' performance.
3. How can Pair Programming be adapted to fit the requirements of mobile software development?
   Section 4.4 showed our recommendations for how to implement PP to achieve the best benefits when using it in mobile development classes. The main recommendations were:
   a. Dedicate one session for introductions
   b. Allow students to use more than one screen
   c. Consider the tutorial session and the homework while preparing for the PP assignment
   d. PP assignment presentation should include a rough draft of the required app
   e. Partners should be assigned by the facilitator
4. How to integrate Pair Programming with other teaching methods to gain the benefits of the two worlds?

The LTP approach integrates Lectures, Tutorials, and PP sessions to provide a comprehensive teaching approach, which helps students gain better understanding through a variety of pedagogical tools. The three components are not separately developed, and each of them works as a preparation step to what comes after it. Sections 4.2 and 4.3 presented how the LTP approach combines the benefits of traditional teaching methods (lectures and tutorial sessions) with the modern PP practices.

6.3 KEY LESSONS LEARNED

Throughout our work over the past years to conduct this research, and from the interactions with students, researchers, educators, and software developers, we realized the real need for new approaches to teach the rapid changing concepts and technologies in the CS filed. When it comes to mobile software development, we have learned several lessons through developing and applying our LTP approach. Below are some lessons learned through the work we have done in this research, which are tied to the research objectives we presented in section 1.4.

- Integrating multiple teaching approaches into CS syllabi is becoming necessary in today's multi-platform environments.
- Collaborative learning helps students learn the subjects while practicing listening, speaking, and communication skills. However, it is important to plan for such activities, and to mentor students during the collaboration practices to ensure that the process is implemented as planned.
- PP needs to be adapted to fit with the new mobile development environments. With multiple devices needed during programming sessions, and with the different resources available, the traditional responsibilities and expectations of drivers and navigators needed to change to cope with the new diverse software development contexts.
- Developing for mobile requires students to learn about a variety of concepts for different topics. Providing different teaching approaches ensures that student will gain the required foundational theories and concepts (through lectures), gain technical experience (through tutorials), and apply what he has learned to solve new problems with a partner (through PP). A well-planned integration is required so that the three methods complement each other.
- It is important to get students to understand the value of certain teaching practices, so that they can apply such practices knowing the benefits they would gain from them. For example: introducing Agile methods and spending time talking about its history, advantages, and current market adoption got students to understand why we do PP and how it would help them, both in the class and in their future careers.

6.4 FUTURE WORK

Exploring different methods to teach mobile software development has led us to identify several directions for our work to be extended. Below are some research opportunities that we encourage other researchers to explore in future research, as well as directions that will be on the top of our research agenda in the upcoming few years.

1. Applying the LTP approach in different contexts

As explained in Chapter 2, PP proved to be beneficial in educational contexts. In our research, we followed this trend and explored whether PP would help in mobile development classes. However, our work was limited to the mobile development classes in the CS department at Virginia Tech. As any other institution, its students share some minimum level of educational background. The authors of (Canfora et al., 2005) found confirmations that forming pairs with individuals with the same educational background emphasizes the expected benefits of collaborative work (pair design in their case). According to them, coupling a person with a scientific background and one with a non-scientific background "does not seem to improve the latter but to make worst the former". Although this finding could be very useful in universities daily practice, and although it helped students in our classes work together due to the common background, it is needed to explore how PP would help students in venues where the educational background is more diverse.

Since our work has been mainly for in-class environments, we want to use LTP in other educational contexts like community colleges, mobile development clubs, community centers, special interest groups, and workshops. These different contexts will allow us to study different factors that would affect the learning process like available time, class size, diversity of the participating groups, and the different experience levels. Moreover, and since our work has been applied in a research-focused (R1) university, we want to see LTP applied in other research as well was teaching higher education institutes, which have different resources and requirements as described in (Yuan et al., 2016).

2. Exploring how LTP can be applied in Virtual Reality courses

Virtual Reality (VR) uses the computer and a variety of accessories to create simulations of the human experience (Hedberg and Alexander, 1994). Since the VR environments has multiple devices with different and unique characteristics that differentiates them from the regular desktop computers environments, they can be considered as a variation of the mobile development environments. (Pantelidis, 2010) shows that many studies have been conducted on the applications and effectiveness of VR in education and training since the 1980s. In one extensive survey, (Youngblut, 1998) found that students enjoy developing their own virtual worlds, and the majority of the teachers said they would use VR technology "if it were affordable, available, and easy to use". However, (Chen, 2006) demonstrated that there are still issues that need further investigation including, among other issues, finding out whether its use can improve the intended performance and understanding, and investigating the VR technology impact on learners with different aptitudes. Therefore, we believe that the LTP approach can be adapted to be used in classrooms that use VR technologies, with the benefits of integrating multiple teaching methods and PP as a platform for collaborative work.

3. Introducing mobile software development to K-12 students

Since the demand is very high for mobile developers, and because of the current schools tendency to introduce CS to their students, we want to investigate how the LTP approach can benefit K-12 students, and how PP can be customized for the younger kids to be able to work together.

4. Studying how PP can help with inclusion activities

During our work, we found out that students from minority and underrepresented groups get better chances to communicate and work with everyone else. Through PP and the assigned-pairs method we followed, students felt engaged and were introduced to other students that they would not have been interacting with if not for PP. This led to forming more diverse project groups, and built a more inclusive class environment. Since this was not a core area of study for our research, we want to dedicate other studies to explore how PP can affect inclusion in CS classes in general.

5. Applying LTP in iOS mobile software development class

We believe that our approach is platform-independent. However, the proposed topics within each module are highly dependent on the operating system used. Since our work has been solely with the Android version of mobile software development classes, we want to apply the LTP approach in iOS classes, where we can change the proposed topics to fit with the different platform. The results of such experience will provide evidence on how LTP can work with any platform, after only chancing the topics within each module.

6. Adapting LTP for work with online classes

We were consulted by department administrators who were developing a syllabus for an online mobile software development class. Our proposed modules, contents, and the LTP approach were very well received by them. However, it was not clear how PP was to be implemented in an online environment. Since we believe that communications and collaborative work is important for mobile development topics, we need to study other approaches to implement PP in a distributed/online environment, and whether there are other collaborative learning approaches that might better fit online courses.

7. Applying other agile practices in mobile development classrooms

Agile methods in general focus on communication and interactions among people (developers, users, designers…etc). Since we found that teaching mobile software development benefited from PP, which is a collaboration practice, we believe that implementing other agile practices can help enhance the learning process. One example can be using the information radiators, and transforming the classroom walls to design spaces where students can get to work together on the application UI design.

8. Exploring methods for more interactive lectures

Since lectures focus more on the foundational theories and abstract concepts, there are fewer opportunities to be engaged besides asking and answering questions. We believe that lectures in general, and the mobile software development topics in particular, could benefit from more innovative lecturing methods that ensure students participation and engagement during the lecture time.

9. Adapting LTP to fit with industrial environments

Most – if not all – software companies have training programs for their developers. It can be of real benefit for both trainers and trainees if they adapt an approach like LTP, which can be tailored to serve the needs of such organizations. Although LTP approach was academically developed and implemented in educational environments, we worked on it with the mentality of real developers and we borrowed practices from the industrial contexts. Therefore, applying adapted version of LTP for industrial contexts will be a promising area for future research work.

References

2001a. *Manifesto for Agile Software Development* [Online]. Available: http://agilemanifesto.org/ [Accessed 01/20/2015 2015].

2001b. *Principles behind the Agile Manifesto* [Online]. Available: http://agilemanifesto.org/principles.html [Accessed 01/20/2015 2015].

AINSWORTH, S. & TH LOIZOU, A. 2003. The effects of self-explaining when learning with text or diagrams. *Cognitive Science,* 27**,** 669-681.

BALIJEPALLY, V. 2006. *Task Complexity and Effectiveness of Pair Programming: An Experimental Study*. PhD Dissertation, THE UNIVERSITY OF TEXAS AT ARLINGTON.

BECK, K. & ANDRES, C. 2004. *Extreme Programming Explained: Embrace Change*, Addison-Wesley Professional.

BIPP, T., LEPPER, A. & SCHMEDDING, D. 2008. Pair Programming in Software Development Teams—An Empirical Study of Its Benefits. *Information and Software Technology,* 50**,** 231-240.

BRYANT, S., ROMERO, P. & DU BOULAY, B. 2008. Pair programming and the mysterious role of the navigator. *International Journal of Human-Computer Studies,* 66**,** 519-529.

CANFORA, G., CIMITILE, A., GARCIA, F., PIATTINI, M. & VISAGGIO, C. A. Confirming the influence of educational background in pair-design knowledge through experiments. Proceedings of the 2005 ACM symposium on Applied computing, 2005. ACM, 1478-1484.

CANFORA, G., CIMITILE, A., GARCIA, F., PIATTINI, M. & VISAGGIO, C. A. Performances of Pair Designing on Software Evolution: A Controlled Experiment. Proceedings of the 10th European Conference on Software Maintenance and Reengineering, 2006. 195-202.

CARVER, J. C., HENDERSON, L., HE, L., HODGES, J. & REESE, D. Increased Retention of Early Computer Science and Software Engineering Students Using Pair Programming. 20th Conference on Software Engineering Education & Training (CSEET'07), 3-5 July 2007 2007. 115-122.

CHAPARRO, E. A., YUKSEL, A., ROMERO, P. & BRYANT, S. Factors affecting the perceived effectiveness of pair programming in higher education. Proc. PPIG, 2005. 5-18.

CHEN, C. J. 2006. The design, development and evaluation of a virtual reality based learning environment. Australasian Journal of Educational Technology, 22.

CHI, M. T. H., DE LEEUW, N., CHIU, M.-H. & LAVANCHER, C. 1994. Eliciting self-explanations improves understanding. *Cognitive Science,* 18**,** 439-477.

CHOI, K. 2004. *A Discovery and Analysis of Influencing Factors of Pair Programming.* PhD Dissertation, New Jersey Institute of Technology.

COCKBURN, A. & WILLIAMS, L. The Costs and Benefits of Pair Programming. Proceedings of Second International Conference on Extreme Programming and Flexible Processes in Software Engineering, 2001a.

COCKBURN, A. & WILLIAMS, L. 2001b. *Extreme Programming Examined*, Addison-Wesley.

DEFRANCO-TOMMARELLO, J. A study of collaborative software development using groupware tools. International Conference on Information Technology: Research and Education, 2003. Proceedings. ITRE2003., 11-13 Aug. 2003 2003. 440-444.

DYBÅ, T. & DINGSØYR, T. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology,* 50**,** 833-859.

ESAKIA, A., NIU, S. & MCCRICKARD, D. S. 2015. Augmenting Undergraduate Computer Science Education With Programmable Smartwatches. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education.* Kansas City, Missouri, USA: ACM.

FLOR, N. & HUTCHINS, E. Analyzing distributed cognition in software teams: a case study of team programming during perfective maintenance. Empirical Studies of Programmers, 1992.

GOOGLE. 2017. *Android Developers* [Online]. Available: https://developer.android.com/index.html [Accessed 2/20/2017 2017].

HALVERSON, C. A. 2002. Activity theory and distributed cognition: Or what does CSCW need to DO with theories? *Computer Supported Cooperative Work (CSCW),* 11**,** 243-267.

HEDBERG, J. & ALEXANDER, S. 1994. Virtual Reality in Education: Defining Researchable Issues. Educational Media International, 31, 214-220.

HEIBERGER, R. M. & ROBBINS, N. B. 2014. Design of diverging stacked bar charts for Likert scales and other applications. *Journal of Statistical Software,* 57**,** 1-32.

HIGHSMITH, J. & COCKBURN, A. 2001. Agile software development: the business of innovation. *Computer,* 34**,** 120-127.

HOADLEY, C. M. & KILNER, P. G. 2005. Using technology to transform communities of practice into knowledge-building communities. *SIGGROUP Bull.,* 25**,** 31-40.

HOLLAN, J., HUTCHINS, E. & KIRSH, D. 2000. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.,* 7**,** 174-196.

HUTCHINS, E. 1995. *Cognition in the Wild*, MIT Press.

JENSEN, R. 2003. A pair programming experience crosstalk. *Journal of Defensive Software Engineering,* 16**,** 22-24.

JOHNSON, D. W., JOHNSON, R. T. & STANNE, M. B. 2000. Cooperative learning methods: A meta-analysis.

JOINT TASK FORCE ON COMPUTING CURRICULA, A. F. C. M. & SOCIETY, I. C. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, ACM.

JONES, P. H. & CHISALITA, C. 2005. Cognition and collaboration: analyzing distributed community practices for design. *CHI '05 Extended Abstracts on Human Factors in Computing Systems.* Portland, OR, USA: ACM.

KATIRA, N., WILLIAMS, L., WIEBE, E., MILLER, C., BALIK, S. & GEHRINGER, E. On Understanding Compatibility of Student Pair Programmers. SIGCSE Bull, 2004. 7-11.

KATRIOU, S. & TOLIAS, E. From twin training to pair programming. Proceedings of the 2nd India software engineering conference, ISEC '09, 2009 New York, USA. ACM.

KOLB, A. Y. & KOLB, D. A. 2005. Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of management learning & education,* 4**,** 193-212.

KUPPUSWAMI, S. & VIVEKANANDAN, K. 2004. The effects of pair programming on learning efficiency in short programming assignments. *Journal of Information Education,* 3**,** 251-266.

LAN, C. & RAMESH, B. An Exploratory Study on the Effects of Pair Programming,. Proceedings of Eighth International Conference on Empirical Assessment in Software Engineering Workshop - 26th International Conference on Software Engineering, 2004. 21-28.

LAVALLÉE, M., ROBILLARD, P. N. & PAUL, S. 2013. Distributed Cognition in Software Engineering: A Mapping Study. *The Fifth International Conference on Information, Process, and Knowledge Management (eKNOW 2013).* Nice, France: IARIA XPS Press.

LIKERT, R. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

LUCK, G. Subclassing xp: Breaking its rules the right way. Proceedings of the Agile Development Conference, ADC '04, 2004.

LYCETT, M., MACREDIE, R. D., PATEL, C. & PAUL, R. J. 2003. Migrating agile methods to standardized development practice. *Computer,* 36**,** 79-85.

MAYER, R. E. 2013. *Teaching and learning computer programming: Multiple research perspectives*, Routledge.

MCDOWELL, C., WERNER, L., BULLOCK, H. & FERNALD, J. The effects of pair-programming on performance in an introductory programming course. SIGCSE Bull, 2002. 38-42.

MCDOWELL, C., WERNER, L., BULLOCK, H. & FERNALD, J. The impact of pair programming on student performance, perception, and persistence. Proceedings of the 25th International Conference on Software Engineering (ICSE'03), 2003.

MCDOWELL, C., WERNER, L., BULLOCK, H. & FERNALD, J. 2006. Pair programming improves student retention, confidence, and program quality. *Communications of ACM,* 49**,** 90-95.

MELNIK, G. & MAURER, F. 2002. Perceptions of Agile Practices: A Student Survey. *In:* WELLS, D. & WILLIAMS, L. (eds.) *Extreme Programming and Agile Methods — XP/Agile Universe 2002: Second XP Universe and First Agile Universe Conference Chicago, IL, USA, August 4–7, 2002 Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg.

MENDES, E., AL-FAKHRI, L. & LUXTON-REILLY, A. Investigating pair-programming in a second-year software development and design computer science course. SIGCSE Bull, 2005. 296-300.

MENDES, E., AL-FAKHRI, L. & LUXTON-REILLY, A. A replicated experiment of pair-programming in a second-year software development and design computer science course. SIGCSE Bull, 2006. 108-112.

MUJEEB-U-REHMAN, M., XIAOHU, Y., JINXIANG, D. & GHAFOOR, M. A. Heterogeneous and Homogenous Pairs in Pair Programming: An Empirical Analysis. Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2006. 1116-1119.

MULLER, M. 2006. A Preliminary Study on the Impact of a Pair Design Phase on Pair Programming and Solo Programming. *Information and Software Technology,* 48**,** 335-344.

NAGAPPAN, N., WILLIAMS, L., FERZLI, M., WIEBE, E., YANG, K., MILLER, C. & BALIK, S. Improving the CS1 experience with pair programming. Proceedings of the 34th Technical Symposium on Computer Science Education (SIGCSE'03), 2003a.

NAGAPPAN, N., WILLIAMS, L., WIEBE, E., MILLER, C., BALIK, S., FERZLI, M. & PETLICK, J. Pair learning: With an eye toward future success. XP/Agile Universe, 2003b.

NAWROCKI, J. & WOJCIECHOWSKI, A. Experimental Evaluation of Pair Programming. Proceedings of the European Software Control and Metrics Conference, 2001. 269-276.

PANDEY, A., MIKLOS, C., PAUL, M., KAMELI, N., BOUDIGOU, F., VIJAY, V., EAPEN, A., SUTEDJO, I. & MCDERMOTT, W. Application of tightly coupled engineering team for development of test automation software – a real world experience. Proceedings of the 27th Annual International Conference on Computer Software and Applications (COMPSAC '03), 2003 Washington, DC. IEEE Computer Society.

PANTELIDIS, V. S. 2010. Reasons to use virtual reality in education and training courses and a model to determine when to use virtual reality. Themes in Science and Technology Education, 2, 59-70.

PENNINGTON, N., LEE, A. Y. & REHDER, B. 1995. Cognitive activities and levels of abstraction in procedural and object-oriented design. *Hum.-Comput. Interact.,* 10**,** 171-226.

PHAPHOOM, N., SILLITTI, A. & SUCCI, G. 2011. Pair Programming and Software Defects – An Industrial Case Study. *In:* SILLITTI, A., HAZZAN, O., BACHE, E. & ALBALADEJO, X. (eds.) *Agile Processes in Software Engineering and Extreme Programming.* Springer Berlin Heidelberg.

PHONGPAIBUL, M. & BOEHM, B. An empirical comparison between pair development and software inspection in Thailand. Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ISESE ’06, 2006 New York, USA. ACM.

PLONKA, L., SHARP, H., VAN DER LINDEN, J. & DITTRICH, Y. 2015. Knowledge transfer in pair programming: An in-depth analysis. *International Journal of Human-Computer Studies,* 73**,** 66-78.

RAHIMIAN, V. & RAMSIN, R. Designing an agile methodology for mobile software development: A hybrid method engineering approach. Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on, 3-6 June 2008 2008. 337-342.

ROGERS, Y. 1997. A brief introduction to distributed cognition. *Retrieved July,* 24**,** 1997.

SALLEH, N., MENDES, E. & GRUNDY, J. 2011. Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review. *Software Engineering, IEEE Transactions on,* 37**,** 509-525.

SANDERS, D. 2002. *Extreme Programming Perspectives*, Addison-Wesley.

SEYAM, M. & GALAL-EDEEN, G. H. 2011. Traditional versus Agile: The Tragile Framework for Information Systems development. *International Journal of Software Engineering (IJSE),* 4**,** 63-93.

SEYAM, M. & MCCRICKARD, S. Collaborating on mobile app design through pair programming: A practice-oriented approach overview and expert review. Collaboration Technologies and Systems (CTS), 2015 International Conference on, 1-5 June 2015 2015. 124-131.

SEYAM, M. & MCCRICKARD, D. S. 2016. Teaching Mobile Development with Pair Programming. *Proceedings of the 47th ACM Technical Symposium on Computing Science Education.* Memphis, Tennessee, USA: ACM.

SEYAM, M., MCCRICKARD, S., NIU, S., ESAKIA, A. & KIM, W. 2016. Teaching Mobile Software Development through Lectures, Interactive Tutorials, and Pair Programming. *Frontiers in Education 2016 (FIE'16).* Erie, PA.

SHAOCHUN, X. & RAJLICH, V. Empirical Validation of Test-Driven Pair Programming in Game Development. Proceedings of the Fifth IEEE/ACIS International Conference on Computational and Information Science - in Conjunction with First IEEE/ACIS Workshop on Component-Based Software Engineer Architecture and Reuse, 2006. 500-505.

SHARP, H. & ROBINSON, H. 2006. A Distributed Cognition Account of Mature XP Teams. *In:* ABRAHAMSSON, P., MARCHESI, M. & SUCCI, G. (eds.) *Extreme Programming and Agile Processes in Software Engineering: 7th International Conference, XP 2006, Oulu, Finland, June 17-22, 2006. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg.

SIMON, B. & HANKS, B. First Year Students' Impressions of Pair Programming in CS1. Proceedings of the Third International Computing Education Research Workshop, 2007. 73-86.

SUTHERLAND, J. 2014. *Scrum: The Art of Doing Twice the Work in Half the Time*, Crown Business.

TICHY, W. F. Agile development: evaluation and experience. Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on, 23-28 May 2004 2004. 692.

TIGREK, S. & OBADAT, M. Teaching smartphones programming using (Android Java): Pedagogy and innovation. Information Technology Based Higher Education and Training (ITHET), 2012 International Conference on, 21-23 June 2012 2012. 1-7.

VANDEGRIFT, T. Coupling pair programming and writing: learning about students' perceptions and processes. Proceedings of the 35th technical symposium on computer science education, SIGCSE 04', New York, USA. ACM.

VANHANEN, J. & KORPI, H. Experiences of using pair programming in an agile project. Proceedings of the 40th Annual Hawaii International Conference on System Sciences HICSS 07', 2007 Washington, DC. IEEE Computer Society.

VANHANEN, J. & LASSENIUS, C. Effects of pair programming at the development team level: an experiment. Proceedings of the 2005 International Symposium on Empirical Software Engineering, 2005.

VANHANEN, J. & LASSENIUS, C. Perceived effects of pair programming in an industrial context. Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007.

VANHANEN, J., LASSENIUS, C. & MANTYLA, M. Issues and tactics when adopting pair programming: A longitudinal casestudy. Proceedings of the International Conference on Software Engineering Advances, ICSEA 07', 2007. IEEE Computer Society.

WERNER, L., HANKS, B. & MCDOWELL, C. 2004. Pair-programming helps female computer science students. *Journal of Education Resources in Computing,* 4.

WILLIAMS, L. But, isn't that cheating? Proceedings of the 29th IEEE Annual Conference Frontiers in Education, 1999.

WILLIAMS, L. & KESSLER, R. 2002. *Pair Programming Illuminated*, Addison-Wesley Longman Publishing Co., Inc.

WILLIAMS, L., MCCRICKARD, S., LAYMAN, L. & HUSSEIN, K. Eleven Guidelines for Implementing Pair Programming in the Classroom. Agile, 2008. AGILE '08. Conference, 4-8 Aug. 2008 2008. 445-452.

WILLIAMS, L., WIEBE, E., YANG, K., FERZLI, M. & MILLER, C. 2002a. In support of pair programming in the introductory computer science course. *Computer Science Education,* 12**,** 197-212.

WILLIAMS, L., YANG, K., WIEBE, E., FERZLI, M. & MILLER, C. Pair programming in an introductory computer science course: Initial results and recommendations. Proceedings of the Symposium on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'02), 2002b.

WRIGHT, P. C., FIELDS, R. E. & HARRISON, M. D. 2000. Analyzing human-computer interaction as distributed cognition: the resources model. *Hum.-Comput. Interact.,* 15**,** 1-41.

YOUNGBLUT, C. 1998. Educational Uses of Virtual Reality Technology. DTIC Document.

YUAN, X., WILLIAMS, K., MCCRICKARD, S., HARDNETT, C., LINEBERRY, L. H., BRYANT, K., XU, J., ESTERLINE, A., LIU, A. & MOHANARAJAH, S. Teaching mobile computing and mobile security. Frontiers in Education Conference (FIE), 2016 IEEE, 2016. IEEE, 1-6.

Appendix (A)

IRB Approval Letter

VirginiaTech

Office of Research Compliance
Institutional Review Board
North End Center, Suite 4120, Virginia Tech
300 Turner Street NW
Blacksburg, Virginia 24061
540/231-4606 Fax 540/231-0959
email irb@vt.edu
website http://www.irb.vt.edu

MEMORANDUM

DATE: August 26, 2016

TO: Scott McCrickard, Mohammed Saad Seyam

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires January 29, 2021)

PROTOCOL TITLE: Pair Programming for Teaching Mobile App Development

IRB NUMBER: **16-746**

Effective August 26, 2016, the Virginia Tech Institution Review Board (IRB) Chair, David M Moore, approved the New Application request for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report within 5 business days to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at:

http://www.irb.vt.edu/pages/responsibilities.htm

(Please review responsibilities before the commencement of your research.)

PROTOCOL INFORMATION:

Approved As:	**Expedited, under 45 CFR 46.110 category(ies) 7**
Protocol Approval Date:	**August 26, 2016**
Protocol Expiration Date:	**August 25, 2017**
Continuing Review Due Date*:	**August 11, 2017**

*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:
Per federal regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals/work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

Invent the Future

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

An equal opportunity, affirmative action institution

Appendix (B)

PP Post-submission Questionnaire

PP1 Questionnaire

Title of Project: Pair Programming for Teaching Mobile Application Development

Investigator(s): Scott McCrickard (mccricks@cs.vt.edu (mailto:mccricks@cs.vt.edu)) – Mohammed Seyam (seyam@vt.edu (mailto:seyam@vt.edu) –

1. Purpose of this Research Project

This project studies using Pair Programming to help students working on in-class assignments for Mobile Application Development. We built a framework that integrates Pair Programming sessions with Lectures and Demo tutorials for better understanding of mobile issues.

2. Procedures

Should you agree to participate, you will be asked to fill-in the questionnaire provided below. This questionnaire should take approximately 5 minutes to complete. Your answers will be anonymized and no identifying data will be collected. The results of this study may be published in research articles and papers.

3. Risks

There are no risks from participating in this study.

4. Benefits

No promise or guarantee of benefits has been made to encourage you to participate. Your participation will help the study team to develop better syllabi for future students, and will enable CS educators to use Pair Programming in a more disciplined way in their classes.

5. Extent of Anonymity and Confidentiality

Information obtained from this questionnaire might be used as part of research publications, but only in ways that make your responses anonymous. At no time will the researchers release results of the study that contain any identifying information to anyone other than individuals working on the project without your written consent. The Virginia Tech (VT) Institutional Review Board (IRB) may view the study's data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research.

6. Compensation

Your participation is completely voluntary, and no compensation is offered.

7. Freedom to Withdraw

It is important for you to know that you are free to withdraw from this study at any time without penalty. You are free not to answer any questions that you choose or respond to what is being asked of you without penalty.

8. Questions or Concerns

Should you have any questions about this study, you may contact one of the research investigators whose contact information is included at the beginning of this document. Should you have any questions or concerns about the study's conduct or your rights as a research subject, or need to report a research-related injury or event, you may contact the VT IRB Chair, Dr. David M. Moore at moored@vt.edu or (540) 231-4991.

Question 1 **1 pts**

What percentage of the required work have been done by your team towards delivering the task?

Question 2 1 pts

How would you divide the percentage of work between you and your partner?

You:

Your partner:

Question 3 1 pts

I have gained (or learned) new things from my partner during today's session.

- 1 (Strongly DISAGREE)
- 2 (Somewhat disagree)
- 3 (Neither agree not disagree)
- 4 (Somewhat agree)
- 5 (Strongly AGREE)

Question 4 1 pts

I would like to try Pair Programming for more programming assignments.

- 1 (Strongly DISAGREE)
- 2 (Somewhat disagree)
- 3 (Neither agree nor disagree)
- 4 (Somewhat agree)
- 5 (Strongly AGREE)

Question 5 1 pts

If there was one thing that you wanted to change about today's Pair Programming session, what would it be?

HTML Editor

Font Sizes Paragraph

p

Question 6

1 pts

What did you like the most about today's session?

HTML Editor

Font Sizes Paragraph

p

Question 7

1 pts

How would you describe your experience of today's Pair Programming session?

HTML Editor

Font Sizes Paragraph

p

Quiz saved at 3:13pm

Submit Quiz

Appendix (C)

End-of-semester Questionnaire

End of Semester Questionnaire

This is the final questionnaire for this semester. Just like the previous questionnaires, your answers to the provided questions would by no mean affect your grade. However, filling-in this questionnaire will count as part of your "participation" activities.

Most of the questions are multiple-choice, and the whole questionnaire shouldn't take more than 5 minutes to complete. Please provide your answers based on your own experience during the class, and feel free to use your own words in answering the open-ended questions at the end.

Title of Project: Pair Programming for Teaching Mobile Application Development

Investigator(s): Scott McCrickard (mccricks@cs.vt.edu (mailto:mccricks@cs.vt.edu)) – Mohammed Seyam (seyam@vt.edu (mailto:seyam@vt.edu) –

1. Purpose of this Research Project

This project studies using Pair Programming to help students working on in-class assignments for Mobile Application Development. We built a framework that integrates Pair Programming sessions with Lectures and Demo tutorials for better understanding of mobile issues.

2. Procedures

Should you agree to participate, you will be asked to fill-in the questionnaire provided below. This questionnaire should take approximately 5 minutes to complete. Your answers will be anonymized and no identifying data will be collected. The results of this study may be published in research articles and papers.

3. Risks

There are no risks from participating in this study.

4. Benefits

No promise or guarantee of benefits has been made to encourage you to participate. Your participation will help the study team to develop better syllabi for future students, and will enable CS educators to use Pair Programming in a more disciplined way in their classes.

5. Extent of Anonymity and Confidentiality

Information obtained from this questionnaire might be used as part of research publications, but only in ways that make your responses anonymous. At no time will the researchers release results of the study that contain any identifying information to anyone other than individuals working on the project without your written consent. The Virginia Tech (VT) Institutional Review Board (IRB) may view the study's data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research.

6. Compensation

Your participation is completely voluntary, and no compensation is offered.

7. Freedom to Withdraw

It is important for you to know that you are free to withdraw from this study at any time without penalty. You are free not to answer any questions that you choose or respond to what is being asked of you without penalty.

8. Questions or Concerns

Should you have any questions about this study, you may contact one of the research investigators whose contact information is included at the beginning of this document. Should you have any questions or concerns about the study's conduct or your rights as a research subject, or need to report a research-related injury or event, you may contact the VT IRB Chair, Dr. David M. Moore at moored@vt.edu or (540) 231-4991.

Question 1 **1 pts**

Pair Programming sessions helped you working on your **Homework assignments.**

1 (Strongly DISAGREE)

2

3

4

5

6

7 (Strongly AGREE)

Question 2 1 pts

Pair Programming sessions helped you working on the **class project**.

1 (Strongly DISAGREE)

2

3

4

5

6

7 (Strongly AGREE)

Question 3 1 pts

How useful were **Pair Programming sessions** to you for understanding the topics of each module?

1 (NOT useful at all)

2

3

4

5

6

7 (VERY useful)

Question 4 1 pts

How useful were **lectures** to you for understanding the topics of each module?

- 1 (NOT useful at all)
- 2
- 3
- 4
- 5
- 6
- 7 (VERY useful)

Question 5 1 pts

How useful were **Demo sessions (including tutorials)** to you for understanding the topics of each module?

- 1 (NOT useful at all)
- 2
- 3
- 4
- 5
- 6
- 7 (VERY useful)

Question 6 1 pts

How useful were **homework assignments** to you for understanding the topics of each module?

- 1 (NOT useful at all)
- 2

3

- 4
- 5
- 6
- 7 (VERY useful)

Question 7 1 pts

If you had the choice, what would you wish to have more of?

- Lectures
- Demo (hands-on) sessions & tutorials
- Pair Programming sessions
- Homework assignments

Question 8 1 pts

The **number of Pair Programming sessions** (7 sessions) was ....

- too little
- just right
- too much

Question 9 1 pts

The **length of each Pair Programming session** (75 minutes) was ......

- too short
- just right
- too long

Question 10 1 pts

The **length of each iteration** within the Pair Programming sessions (15-minute turns between driver and navigator) was ......

- too short
- just right
- too much

Question 11 1 pts

What did you **like the most** about Pair Programming sessions?

HTML Editor

B I U A A T_x x^2 x_2

π Font Sizes Paragraph

p

Question 12 1 pts

What do you **wish to change** about Pair Programming sessions?

HTML Editor

B I U A A T_x x^2 x_2

π Font Sizes Paragraph

p

Question 13 1 pts

How would you describe your **Pair Programming experience** during this class?

HTML Editor

Font Sizes Paragraph

p

Question 14 1 pts

How would you describe your **overall class experience**?

HTML Editor

Font Sizes Paragraph

[redacted]

p

No new data to save. Last checked at 3:16pm

Submit Quiz

Appendix (D)

Lab Assistant (LA) Sheet

PP1 (20160908) TA:________________

	Coding	IDE (Android Studio)	Mobile device issues	Irrelevant to assignment	Time spent	Notes
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						
11.						
12.						
13.						
14.						
15.						
16.						

Appendix (E)
Selected Publications and Presentations

Publications

Seyam, M., McCrickard, S., Niu, S., Esakia, A., Kim, W., (2016) "Teaching Mobile Software Development through Lectures, Interactive Tutorials, and Pair Programming", IEEE Frontiers in Education Conference (FIE'16), October 12-15, Erie, PA. Pp. 1-9.

Seyam, M., McCrickard, S. (2016) "Teaching Mobile Development with Pair Programming", Proceedings of the 47th ACM technical symposium on Computer Science Education (SIGCSE '16), March 2-6, Memphis, TN, USA. Pp. 96-101.

Seyam, M., McCrickard, S. (2015) "Collaborating on Mobile App Design through Pair Programming", Proceedings of the International Conference on Collaboration Technologies and Systems (CTS 2015), Atlanta, GA.

Seyam, M., Galal-Edeen, G., (2011), "Traditional versus Agile: The Tragile Framework for Information Systems development", The International Journal of Software Engineering (IJSE), Vol. 4, No. 1, Pp. 63-93, 2011. ISSN: 1687-6954.

Seyam, M., (2010), Agile Methodologies in Information Systems Development: How to be Agile, without losing the disciplines of being Traditional, LAP Lambert academic publishing, ISBN: 3838371925.

Galal-Edeen, G., Riad, A., **Seyam, M.**, (2007) "Agility versus discipline: Is reconciliation possible?" Computer Engineering and Systems, ICCES '07. International Conference on, pp.331-337, IEEE, 27-29 Nov. 2007

Galal-Edeen, G., Riad, A., **Seyam, M.**, (2007) "Agility Versus Discipline: Towards a Middle Ground" Computers and Industrial Engineering, The 37th International Conference on. pp.1597-1609, 20-23 Oct. 2007

Presentations and Posters

Seyam, M., (2016) "Pair Programming for Teaching Mobile Development", Poster presentation (*finalist*), the ACM Student Research Competition, the 47th ACM technical symposium on Computer Science Education (SIGCSE '16), March 2-6, Memphis, TN, USA.

Seyam, M., (2015) "Enhancing Usability through Agility", Doctoral Colloquium presentation, the International Conference on Collaboration Technologies and Systems (CTS 2015), Atlanta, GA.

Seyam, M., (2015) "User Interface Design and Agility: Practices for Integration in CS Classrooms", Poster presentation (*finalist*), the ACM Student Research Competition, the 46th ACM technical symposium on Computer Science Education (SIGCSE '15), March 4-7, Kansas City, MO, USA.

Seyam, M., (2015) "Fostering Innovation in UX Design through Agile Methods", Doctoral Consortium presentation, ACM Richard Tapia Celebration of Diversity in Computing, February 18-21, Boston MA, USA.