Teaching Mobile Application Development through Lectures, Interactive Tutorials, and Pair Programming

Mohammed Seyam, D. Scott McCrickard, Shuo Niu, Andrey Esakia
Department of Computer Science
Virginia Tech
Blacksburg, VA
seyam@vt.edu, mccricks@cs.vt.edu, shuoniu@vt.edu, esakia@vt.edu

Woongsup Kim
Division of Computer & Information
Dongguk University
Seoul, Korea
woongsub@dongguk.edu

***Abstract*—Research suggests that different teaching styles and multiple exposures of different styles to material can aid in the learning process. While there are guidelines for identifying the best teaching style for material, new and evolving areas can present unique challenges. The emerging area of mobile software development, which combines aspects of software, hardware, and interpersonal interaction, captures many such challenges; e.g., understanding how to develop for multiple screen sizes, designing for GPS time lag, dealing with unreliable sensor data. Teaching these challenging materials seemed well suited for multiple approaches that leveraged different learning styles. This paper examines three teaching approaches employed in ten teaching modules across two semesters of a mobile software development course. The approaches included lectures, interactive tutorials, and Pair Programming. 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. Two active learning approaches used in the class were interactive tutorials and Pair Programming. Interactive tutorials presented applied development approaches, then explored their use in an individual-based hands-on demos. Pair Programming is an agile software development practice, used in both industry and education, which enforces a role-based approach to learning new programming concepts. Homeworks were used to assess learning, and surveys reflected student satisfaction. Results show areas of promise and of concern with regard to the learning styles. It seems that repetition of topics is important for mastery of the topics. Foundational theories seem well suited for lectures, while programming concepts work better in active learning situations. Additional learning took place through office hours, online question forums, and individual and group online exploration. The findings suggest specific approaches to teaching challenging and unique mobile software development topics as well as a general approach to identifying ways to distribute learning objectives across lectures, interactive tutorials, and Pair Programming sessions.**

Keywords—Mobile Application Development; Computer Science Education; Pair Programming

I. 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. With the challenges associated with developing for mobile, new approaches should be explored to ensure that students are comprehending and applying mobile development topics properly.

Pair Programming (PP) has been studied before as a pedagogical tool for programming classes, providing ways to help students collaborate on assignments. 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. These fundamentally important topics have not been studied with PP previously, but their challenging nature highlight potential benefits for pair-based learning.

This paper presents a PP-based framework that integrates PP sessions with lectures, interactive tutorials (demos), and homework assignments, to provide CS educators with a flexible and adaptable approach to teach mobile app development that would enhance students' learning. The following section presents an overview of the history of using PP in industrial and educational contexts, then it shows how mobile app development is different in nature from traditional development. The paper then proceeds by showing our previous work that led us to develop our proposed framework, which is presented in section V. A detailed explanation for the role of PP in our framework is then presented in section VI, followed by presenting an assessment framework. The paper then concludes by showing our plans to apply our framework as part of upcoming mobile development course.

To build our framework, we collected feedback from expert review sessions and also from students of Spring 2015 mobile development class. We are currently applying our proposed framework to the Spring 2016 class. Therefore, the findings provided in this paper reflect early feedback from this class, which is collected via questionnaires, assignment submissions, Teaching Assistants (TAs) sheets, and facilitator's observations. A detailed analysis and assessment of the proposed framework will be available later after the semester ends, and will be presented in a future paper for a more elaborate discussion.

II. Pair Programming for Mobile Development

Pair Programming (PP) is an Agile development practice that features two developers working on the same development task [1, 2]. 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 [3]. 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, [4] presents studies that have shown that PP enhanced understandability and maintainability of code and design [5], decreased defect rates [6] and supported knowledge transfer [7].

Since we are interested in PP for classroom settings, we focus on studies that have been performed on students, especially in higher education who are enrolled in Computer Science (CS) or Software Engineering (SE) classes. 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 [8]. Some of the foundational studies that followed are those of North Carolina State University (NCSU) [9], University of California Santa Cruz (UCSC) [10], University of Auckland [11], and the Pondicherry Engineering College [12]. These studies have shed light on PP as a pedagogical tool that can help students enhance their productivity and the quality of their work [13]. Several studies have been performed later [14], with evidence suggesting that PP could enhance enjoyment [15], increase student's confidence level [16], reduce workload [17], improve course completion rate [15], increase homework submission rate [18], improve exam performance [19], and facilitate working efficiently on programming assignments [20].

During the past decade, mobile applications became more mainstream, and several approaches and techniques have been developed to serve the goals of mobile app developers. Researchers showed how mobile development has certain issues that differentiate it from desktop/web development related to communication, mobility, portability, standards and protocols, power, storage, and user interfaces [21]. Another study proposed a pedagogy for teaching smartphone programming, which incorporated PP as a practice to encourage guided teamwork [22]. However, PP was introduced as a supporting practice 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 [23]. This research shows the need to consider several extensions while dealing with mobile development, such as wearable devices, different sensors, and multiple platforms. Different research studied using PP for teaching Human Computer Interaction (HCI) class, which considered the general HCI concepts with no specific emphasis on mobile interfaces [24].

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 and location data, accelerometers, light sensors, and cameras. PP should account for these features.

3. User interface (UI) and user experience (UX) issues: Mobile devices are very different than more traditional desktop, laptop, and web environments, with challenges and opportunities for PP.

Our 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. To understand how students would deal with those issues, we conducted expert reviews with two pairs of experienced Android developers. Then, we examined student course work during PP sessions in a mobile development class. The following two sections provide a summary of the expert reviews and the Spring 2015 in-class sessions.

III. 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 two expert review sessions with two pairs of experienced Android developers. One pair of experts was graduate students, and the others were undergraduates. We conducted two walkthrough 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 [25]. The sessions concluded by showing some pitfalls for in-class PP, as well as some recommendations that the experts think would enhance students' performance. Some of the pitfalls were:

- Lack of planning and time management
- Jumping directly into coding without working on design
- Thinking about the assignment as a "task to be graded"
- Considering on-time software delivery over product quality
- Disregarding creative ideas in favor of traditional solutions
- Giving less consideration to UI design

And to achieve better results from practicing PP with students working on mobile development in-class assignments, the expert review sessions proposed recommendations; key among them were:

- For the first PP sessions when learning the technique, students are better off not to be paired randomly
- Assign the first PP round to high-level design and session planning

- Introduce quality as a basic requirement, not as a bonus
- Emphasize the role of talking and listening
- Present the assignment as a challenging programming task, not as a regular class assignment
- Students should be asked to explicitly consider user context and usage scenarios

Based on our experience with the expert reviews, we started working with students in actual class settings. We considered the pitfalls and recommendations provided on both literature and the expert reviews, and we explored how students performed during five PP sessions.

IV. In-class PP Sessions (Spring, 2015)

On the Spring 2015 semester, we applied pair programming as part of the class activities for the Mobile Development course. A detailed discussion of our work with this class is available in [26]. Our objectives from this were to:

1. Introduce students to PP as a grounded well-known Agile practice that's been used widely in several environments
2. Get students to know about different models for software engineering
3. Encourage students to evaluate, compare, and criticize different approaches for software development
4. Assess students understanding for mobile development 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-minutes PP sessions. The five sessions covered five main topics related to mobile app development, which are:

1. Basic GUI elements
2. Graphics (Camera and Image processing)
3. Connectivity (Bluetooth and Smartwatches)
4. Sensors (GPS and Accelerometer)
5. Input/Output devices (audio, mics, and speakers)

We collected feedback from students through questionnaires and observations. Based on the feedback we got from the in-class sessions, we came to some recommendations that helped us build our framework. The main recommendations were:

1. Allow students to choose partners on the first session, then move to random pairing
2. Allow the navigator to look at the mobile device screen while the driver works on coding
3. Decide on the strategy to be used to introduce UI/UX requirements based on the amount of time available for each session
4. Plan for extra time and pre-class preparation for activities that involve using multiple devices and connections

The above recommendations worked as the basis for our proposed framework that is presented in the next section.

V. The Integrated Framework for Teaching Mobile App Development

Based on the literature review, expert review sessions, and the Spring 2015 class feedback, we came to realize that there's a need to build a framework that aims at helping CS educators who want to teach mobile development to undergraduate students. Basically, we wanted to have a framework that satisfies certain requirements:

- It has PP as a core component
- It implements PP in a way that fits the mobile development challenges and serves students' needs
- It considers integrating several teaching approaches for better understanding of mobile development topics
- It has certain assessment criteria
- It is practice-oriented so that it can be flexible and adaptable depending on different class environments

Having these guidelines in mind, we developed a framework that integrates PP sessions with lectures, tutorials, and homework assignments. Fig. 1 shows the main components of framework, and this section will describe the details of every component.

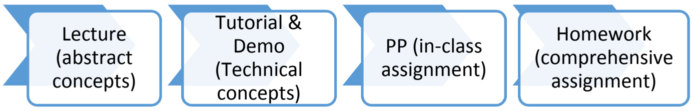
Lecture (abstract concepts)
Tutorial & Demo (Technical concepts)
PP (in-class assignment)
Homework (comprehensive assignment)

Fig. 1. A Block Diagram of the Proposed Framework

Before discussing the different components of the framework, Table I provides a summary of the different roles required for the success of the proposed framework. To ensure flexibility of our framework, 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 TA. 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 TA who is experienced with PP, so they can be the same person.

TABLE I. DIFFERENT ROLES DESCRIBED BY THE FRAMEWORK

Role	Responsibilities
Instructor	Responsible for 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 Creates questionnaires and quizzes, and collect results with help from TA
Teaching Assistant (TA)	Creates assignments together with facilitator Prepares homework assignments with facilitator Makes sure that students' machines are running and have all the required software installed Answers technical questions during sessions Evaluates students' homework Fills-in the TA sheet during the PP session
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 Fills-in the questionnaires by the end of each session Submits homework by the due date

The idea of the Table I 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). In the case of our work with the Spring 2016 class, and since we had a class of 70 students, we had one instructor, one facilitator, and three TAs.

A. *Lectures*

The lectures are taught by the main instructor, who should focus on the abstract concepts that do not require technical details. The goal of lectures is to provide the required background of abstract concepts to students before delving into technical details and the actual implementation.

B. *Tutorials and Demos*

Following the lectures, come the tutorial and demo sessions, which are usually led by a Teaching Assistant (TA) who has experience in mobile development. These sessions aim to cover the basic technical details that allow students to implement the concepts that they learned about in lectures. The TA should prepare a tutorial that explain how to turn the concepts into functioning app, as well as including some code segments for the basic topics of each module. During the session, the TA would refer to the tutorial, while demonstrating certain sections of it via live coding session with students. Students can follow the TA and are encouraged to apply what they see on their machines as the TA is working on the demo.

C. *Pair Programming Sessions*

After covering the abstract concepts of a module in the lecture, and then demonstrating implementation and technical details in the tutorial and demo session, comes the time of PP session. The goal of PP session is to ensure that students have comprehended the information provided during the class and the demo 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 time frame of a class session. The importance of using PP and the approach we recommend to apply it is provided in detail in section VI.

D. *Homeworks*

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 II provides the topics covered in lectures, interactive tutorials (demos), and PP sessions when applied to the Spring 2016 mobile app development class.

VI. THE ROLE OF PAIR PROGRAMMING IN THE FRAMEWORK FOR TEACHING MOBILE DEVELOPMENT

As shown in section II, 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 [14]. We also showed that PP seem 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 with the Spring 2015 mobile development class students, we came to realize some concerns:

- Students did not know the value of using PP until the very last 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

TABLE II. TOPICS COVERED IN LECTURES, TUTORIALS, AND PP SESSIONS

Module	Lecture (abstract concepts)	Interactive Tutorial (technical demo)	PP Session (assignment)
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
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
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
Graphics	2D Graphics Canvas Class Camera & Light Sensors Animations Speech Recognition Recording & Playing Audio Tough & Multi-touch Gestures	Real-time Drawing (Canvas) Handling Touch events Detecting Gestures SoundPool Code for Camera	Camera and Gestures
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

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

A. 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. In our Spring 2016 class, we dedicated one 75-minute session for introductions. The topics covered in this sessions were:

- 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 we finished the introduction, we asked every student to pair with the one sitting on their right, for two 10-minute PP iterations to develop a very simple mobile app (in this case, it was a simple calculator with basic operations). The objective was not to actually develop the app, but 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. We wanted to make sure that students are able to:

- 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
- Fill-in the survey right after the code submission

By the end of this session, students were familiar with the logistics of the class, as well as with the nature of developing in PP environment. Since they did not experience in mobile development, they spent the whole time exploring the development environment. Therefore, they were not expected, nor required, to submit a working app. However, working in pairs for such short time on this session helped us on all the following PP sessions to save the time of talking about logistics, and allowed us to directly work on the assignment.

B. Allow students to use more than once 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 Spring 2015 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 was not only about coding, but also online searching, looking up class material, and reviewing lab tutorials. Working on those tasks on the same machine was a burden for students, and 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 was the responsibility of the session leader (facilitator) and TAs to make sure that these two rules are not broken during the session. Fig. 2 shows students working during different sessions of the Spring 2016 class. Our concern was that by not following strict PP rules, students may not be working together. However, based on our observations, and the preliminary feedback that we got from students of Spring 2016 semester, we noticed that:

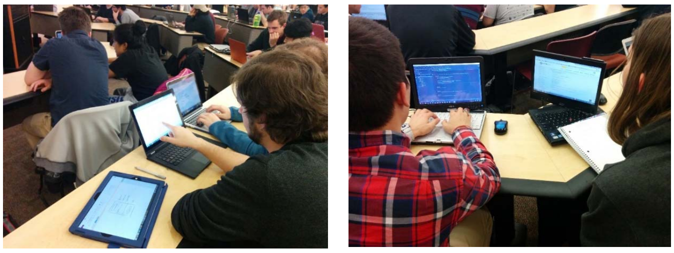

Fig. 2. Students working on multiple screens during different PP sessions

- The other laptop screen always has either Google search results, Android Studio help pages, or tutorial materials from the previous session
- Navigators used the other screen only when drivers were working on what they 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.

Based on the preliminary feedback we are getting from Spring 2016 students, the issue of the number of screens never showed up in a single comment. By reviewing the feedback of Spring 2015 students, it was clear that it was a main concern during every session, and students used to complain about being forced to use only one screen/laptop during the session. Therefore, and on a basic level, students’ concerns were addressed, but more analysis for the quality of assignments need to be performed to investigate how their performance was actually affected.

C. Consider the tutorial session and the homework while preparing 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:

1. A topic (or more) from those which have been covered in detail on the tutorial and demo session
2. 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 certain module, comes the time of the facilitator to work together with TAs 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 may not be covered on the demo session
- For the topics that are new and specific to mobile development, they should be introduced during the tutorial and demo 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 demo session, find solutions to new problems, and gain knowledge for the upcoming homework assignment.
- The homework assignment should include the topics covered during lectures, demo, and PP sessions, while being more challenging since its time is not as limited.

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

1. Collaboratively review the material provided in the tutorial and during the demo 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. This ensures that the collective experience they have would help them find better solutions and prepare for the upcoming homework.

D. *Students should be randomly paired*

This recommendation emerged after working with the Spring 2015 class. During the Spring 2016 class, we tried different ways to pair students, and the preliminary feedback shows that random pairing has benefits over fixed and/or assigned pairing. Basically, and since students will have different partners every session, this 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 personal 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.

The "pairing rule" mentioned above is the way set by facilitator every session for students to pair with. Some examples tried with the Spring 2016 class are for every student to pair with the one: sitting on the right, sitting in front, and sitting on the same position of a different row. The main idea is that no student would expect whom would be his/her partner until they start moving based on the facilitator's "rule". Another suggestion to do the random pairing is to divide students into two groups, and let every group start counting and pair those with the same numbers together. The groups, start place, and counting direction can change every session to provide different combinations every session.

From the preliminary feedback we got, 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, but as explained above, these cases are not repeated because the partners change every time.

VII. Assessment of the Proposed Framework

In order to assess our proposed framework, and to help CS educators determine how successful would their implementation of the framework be, we provide an assessment framework together with the integrated framework.

A. *Class environment and context*

Prior deciding to implement our proposed framework, an educator will need to determine if this framework is suitable to the class under consideration. We determine five main factors that would affect the implementation of our proposed framework:

1. Number of Students
2. Number of TAs
3. Number of PP sessions
4. Length of PP sessions
5. Experience levels of students and TAs

These factors and interconnected and should be considered holistically before deciding to apply our proposed framework. For example: applying the framework for 80-student class would not be a problem if there are four TAs available for PP sessions, while it will not be reasonable to have PP session with only one TA available in this case.

The ideal case for our framework, based on our two-year experience is with classes of 40-50 students, 3-4 TAs, 6-8 PP sessions, 60-90 minutes session length, with senior developers as TAs and junior-senior students. However, since every class will always have different circumstances, the adaptability of the proposed framework can handle different scenarios by providing flexible practices.

B. Students' performance and perception

Our proposed framework comes with different assessment steps to collect information about students' experience and performance during the different stages of the mobile development class. Table III. shows the main assessment tools.

1. In-class Interactions

   a. Facilitator's observations: At the beginning of each PP session, the facilitator is responsible for providing guidelines for 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. Besides the facilitator's intervention in such situations, it's important to record these observations to be analyzed and used to enhance the next sessions.

   b. TA's feedback sheet: TAs are provided by documents that asks them to write 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 is collected and analyzed to evaluate whether the PP assignment fits the time allowed, and whether the level is difficulty is suitable for students.

2. In-class Post-PP session

   Before students leave classroom after PP session, they are required to fill-in an online questionnaire about their PP experience. Sample questions are:

   - How would you divide the percentage of work between you and your partner?
   - Have you gained (or learned) new things from my partner during today's session? (5-point scale provided)
   - If there is one thing that you may change about today's Pair Programming session, what would it be?
   - What did you like the most about today's session?

   The objective of this questionnaire is to monitor students' perception of PP, and to watch the difference in answers among the different sessions. This will help facilitator to plan for upcoming sessions and for future planning of other classes.

3. Post-class submissions

   a. App code and UI: It's the responsibility of facilitator and TAs to evaluate students code and UI developed during the PP session. This should work as the indicator of students' performance during the programming session. The grades of these evaluations are for facilitator's evaluation, not to be returned to students.

   b. Homework: After students submit their homework, the TAs grade the assignments and provide feedback to students. This works 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 aims at getting students' feedback regarding the different framework components, and how they see the role of PP in helping them understanding specific topics throughout the semester-long course.

   b. Final grades: The final course grades will be used as a reference for comparison among different implementations of the same framework with difference classes. It would also be used to determine if there are certain patterns among different groups of students.

TABLE III. FRAMEWORK ASSESSMENT TOOLS

In-class Interactions	**In-class Post-PP session**	**Post-class Submissions**	**End-of-Semester**
- Facilitator's observations - TA's feedback sheet	- PP questionnaire	- App code and UI - Homework	- Final questionnaire - Final grades

VIII. CONCLUSION AND FUTURE WORK

This paper proposes an integrated framework for teaching mobile development to undergraduate CS students. Although the framework presents components like lectures, tutorials, demo sessions, and homework assignments, Pair Programming (PP) is considered the core element of this framework due to the benefits we see in using it for handling mobile development challenges. Therefore, the paper describes 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.

The proposed framework was developed based on our collective experience from literature review, expert review sessions, and Spring 2015 mobile development class. To provide a comprehensive and detailed evaluation for the framework, we are currently working with the Spring 2016 mobile development class, which we used some of its preliminary feedback throughout this paper. We expect to have all the required feedback by the end of semester, which will then be analyzed to assess students' performance during the semester.

Our future work will be to adapt the framework based on the feedback we are currently getting from students. Then, we will apply it to the Fall 2016 mobile development class, which will enable us to study a full implementation of the framework with all its different components. Our objective is then to encourage CS educators who may consider teaching mobile development to use our framework while developing their syllabus.

REFERENCES

[1] http://agilemanifesto.org/, accessed 01/20/2016

[2] Seyam, M., and Galal-Edeen, G.H.: 'Traditional versus Agile: The Tragile Framework for Information Systems development', International Journal of Software Engineering (IJSE), 2011, 4, (1), pp. 63-93

[3] Williams, L., and Kessler, R.: 'Pair Programming Illuminated' (Addison-Wesley Longman Publishing Co., Inc., 2002.

[4] Plonka, L., Sharp, H., van der Linden, J., and Dittrich, Y.: 'Knowledge transfer in pair programming: An in-depth analysis', International Journal of Human-Computer Studies, 2015, 73, pp. 66-78.

[5] Vanhanen, J., and Lassenius, C.: 'Perceived effects of pair programming in an industrial context', in Proceedings of the 33rd EUROMICRO Conference on Software Engineering and Advanced Applications, 2007.

[6] Phaphoom, N., Sillitti, A., and Succi, G.: 'Pair Programming and Software Defects – An Industrial Case Study', in Sillitti, A., Hazzan, O., Bache, E., and Albaladejo, X. (Eds.): 'Agile Processes in Software Engineering and Extreme Programming' (Springer Berlin Heidelberg, 2011), pp. 208-222

[7] Katriou, S., and Tolias, E.: 'From twin training to pair programming', in Proceedings of the 2nd India software engineering conference, ISEC '09, 2009.

[8] Cockburn, A., and Williams, L.: 'The Costs and Benefits of Pair Programming', in Proceedings of Second International Conference on Extreme Programming and Flexible Processes in Software Engineering, 2001.

[9] Nagappan, N., Williams, L., FERZLI, M., WIEBE, E., YANG, K., MILLER, C., and Balik, S.: 'Improving the CS1 experience with pair programming', in Proceedings of the 34th Technical Symposium on Computer Science Education (SIGCSE '03), 2003.

[10] Mcdowell, C., Werner, L., Bullock, H., and Fernald, J.: 'Pair programming improves student retention, confidence, and program quality', Communications of ACM, 2006, 49, (8), pp. 90-95.

[11] Mendes, E., Al-Fakhri, L., and Luxton-Reilly, A.: 'A replicated experiment of pair-programming in a second-year software development and design computer science course', in SIGCSE Bull, 2006, pp. 108-112.

[12] Kuppuswami, S., and Vivekanandan, K.: 'The effects of pair programming on learning efficiency in short programming assignments', Journal of Information Education, 2004, 3, (2), pp. 251-266.

[13] Simon, B., and Hanks, B.: 'First Year Students' Impressions of Pair Programming in CS1', in Proceedings of the Third International Computing Education Research Workshop, 2007, pp. 73-86.

[14] Salleh, N., Mendes, E., and Grundy, J.: 'Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review', Software Engineering, IEEE Transactions on, 2011, 37, (4), pp. 509-525.

[15] Muller, M.: 'A Preliminary Study on the Impact of a Pair Design Phase on Pair Programming and Solo Programming', Information and Software Technology, 2006, 48, (5), pp. 335-344.

[16] Bipp, T., Lepper, A., and Schmedding, D.: 'Pair Programming in Software Development Teams—An Empirical Study of Its Benefits', Information and Software Technology, 2008, 50, pp. 231-240.

[17] Canfora, G., Cimitile, A., Garcia, F., Piattini, M., and Visaggio, C.A.: 'Performances of Pair Designing on Software Evolution: A Controlled Experiment', in Proceedings of the 10th European Conference on Software Maintenance and Reengineering, 2006, pp. 195-202.

[18] Katira, N., Williams, L., Wiebe, E., Miller, C., Balik, S., and Gehringer, E.: 'On Understanding Compatibility of Student Pair Programmers', in SIGCSE Bull, 2004, pp. 7-11.

[19] Phongpaibul, M., and Boehm, B.: 'An empirical comparison between pair development and software inspection in Thailand', in Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering, ISESE 2006.

[20] Mujeeb-u-Rehman, M., Xiaohu, Y., Jinxiang, D., and Ghafoor, M.A.: 'Heterogeneous and Homogenous Pairs in Pair Programming: An Empirical Analysis', in Proceedings of the Canadian Conference on Electrical and Computer Engineering, 2006, pp. 1116-1119.

[21] Rahimian, V., and Ramsin, R.: 'Designing an agile methodology for mobile software development: A hybrid method engineering approach', in Proceedings of the Second International Conference on Research Challenges in Information Science, RCIS, 2008, pp. 337-342.

[22] Tigrek, S., and Obadat, M.: 'Teaching smartphones programming using (Android Java): Pedagogy and innovation', in Proceedings of the International Conference on Information Technology Based Higher Education and Training (ITHET), 2012, pp. 1-7.

[23] Esakia, A., Niu, S., and McCrickard, D.S.: 'Augmenting Undergraduate Computer Science Education With Programmable Smartwatches'. Proceedings of the 46th ACM Technical Symposium on Computer Science Education, Kansas City, Missouri, USA, 2015 pp. 66-71.

[24] Williams, L., McCrickard, S., Layman, L., and Hussein, K.: 'Eleven Guidelines for Implementing Pair Programming in the Classroom', in Proceedings of Agile 2008, pp. 445-452.

[25] Seyam, M., and McCrickard, S.: 'Collaborating on Mobile App Design through Pair Programming', in Proceedings of International Conference on Collaboration Technologies and Systems (CTS '15), 2015, pp.124-131.

[26] Seyam, M., and McCrickard, D.S.: 'Teaching Mobile Development with Pair Programming'. Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16), Memphis, Tennessee, USA, 2016, pp. 96-101.